

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

اللَّهُمَّ صَلِّ عَلَى مُحَمَّدٍ وَآلِ مُحَمَّدٍ وَعَجِّلْ فَرَجَهُمْ



توسعه دهنده صفحات وب با PHP و MySQL

رشته تولیدکننده و توسعه دهنده پایگاه های اینترنتی

گروه برق و رایانه

شاخه کاردانش

پایه دوازدهم دوره دوم متوسطه





وزارت آموزش و پرورش
سازمان پژوهش و برنامه‌ریزی آموزشی



نام کتاب: توسعه‌دهنده صفحات وب با PHP و MySQL - ۳۱۲۲۵۱

پدیدآورنده: سازمان پژوهش و برنامه‌ریزی آموزشی

مدیریت برنامه‌ریزی درسی و تألیف: دفتر تألیف کتاب‌های درسی فنی و حرفه‌ای و کار دانش

شناسه افزوده برنامه‌ریزی و تألیف: مجتبی انصاری پور، افشار بهمنی، محمدرضا شکرریز، حسن جعفریه، لیلا سعید و بیتا رهنمای زربیحاری (اعضای شورای برنامه‌ریزی)

محمد محمدی (مؤلف) - آزاده حق‌روستا (برنامه‌ریزی و هماهنگی تألیف)

مدیریت آماده‌سازی هنری: اداره کل نظارت بر نشر و توزیع مواد آموزشی

شناسه افزوده آماده‌سازی: جواد صفری (مدیر هنری) - مریم کیوان (طراح جلد) - سورش سعادت‌مندی و مهلا مرتضوی (صفحه‌آرا)

نشانی سازمان: تهران: خیابان ایرانشهر شمالی - ساختمان شماره ۴ آموزش و پرورش (شهید موسوی)

تلفن: ۹-۸۸۸۳۱۱۶۱، دورنگار: ۸۸۳۰۹۲۶۶، کدپستی: ۱۵۸۴۷۴۷۳۵۹

وب سایت: www.chap.sch.ir و www.irtextbook.ir

ناشر: شرکت چاپ و نشر کتاب‌های درسی ایران: تهران-کیلومتر ۱۷ جاده مخصوص کرج- خیابان ۶۱ (داروپخش)

تلفن: ۵-۴۴۹۸۵۱۶۱، دورنگار: ۴۴۹۸۵۱۶۰، صندوق پستی: ۱۳۹-۳۷۵۱۵

چاپخانه: شرکت چاپ و نشر کتاب‌های درسی ایران «سهامی خاص»

سال انتشار و نوبت چاپ: چاپ اول ۱۴۰۱

کلیه حقوق مادی و معنوی این کتاب متعلق به سازمان پژوهش و برنامه‌ریزی آموزشی وزارت آموزش و پرورش است و هرگونه استفاده از کتاب و اجزای آن به صورت چاپی و الکترونیکی و ارائه در پایگاه‌های مجازی، نمایش، اقتباس، تلخیص، تبدیل، ترجمه، عکس‌برداری، نقاشی، تهیه فیلم و تکثیر به هر شکل و نوع بدون کسب مجوز از این سازمان ممنوع است و متخلفان تحت پیگرد قانونی قرار می‌گیرند.



اگر یک ملتی نخواهد آسیب ببیند باید این ملت اولاً با هم متحد باشد، و ثانیاً در هر کاری که اشتغال دارد آن را خوب انجام بدهد. امروز کشور محتاج به کار است. باید کار کنیم تا خودکفا باشیم. بلکه ان شاءالله صادرات هم داشته باشیم. شما برادرها الان عبادت تان این است که کار بکنید. این عبادت است.
امام خمینی «قَدِّسَ سِرُّهُ»

فصل اول - نصب و راه اندازی سرویس دهنده وب

- ۱-۱- برنامه نویسی مبتنی بر وب ۲
- ۱-۲- معرفی زبان برنامه نویسی PHP ۲
- ۱-۳- نرم افزارهای مورد نیاز برای برنامه نویسی PHP ۳
- ۱-۴- آموزش نصب نرم افزارها ۵
- ۱-۵- آموزش اتصال به سرویس دهنده ۸
- ۱-۶- آموزش ایجاد اسکریپت های PHP ۱۰

فصل دوم - اصول برنامه نویسی به زبان PHP

- ۲-۱- نحوه نوشتن دستورات ۱۸
- ۲-۲- نوشتن توضیحات ۱۹
- ۲-۳- استفاده از متغیرها ۱۹
- ۲-۴- از بین بردن متغیرها ۲۱
- ۲-۵- متغیرهای رشته ای ۲۲
- ۲-۶- ثابت ها ۲۳
- ۲-۷- استفاده از عملگرها در PHP ۲۴
- ۲-۷-۱- عملگرهای محاسباتی ۲۴
- ۲-۷-۲- عملگرهای تخصیص یا انتساب ۲۶
- ۲-۷-۳- عملگرهای مقایسه ای ۲۸
- ۲-۷-۴- عملگرهای منطقی ۲۸
- ۲-۷-۵- اولویت عملگرها ۳۰

فصل سوم - ساختارهای شرطی و تکرار

- ۳-۱- ساختارهای شرطی ۳۶
- ۳-۱-۱- ساختار if ۳۶
- ۳-۱-۲- ساختار if..else ۳۷
- ۳-۱-۳- ساختار switch ۴۰
- ۳-۱-۴- عملگر : و ? ۴۲
- ۳-۲- ساختارهای تکرار ۴۲
- ۳-۲-۱- حلقه تکرار while ۴۳
- ۳-۲-۲- دستورالعمل do..while ۴۵
- ۳-۲-۳- حلقه تکرار for ۴۶
- ۳-۳- دستورات break, continue و exit ۴۹

- ۳-۳-۱- دستور break ۴۹
- ۳-۳-۲- دستور continue ۵۰
- ۳-۳-۳- دستور exit ۵۱
- ۳-۴- مثال طراحی و برنامه نویسی ۵۱

فصل چهارم - آرایه ها در PHP

- ۴-۱- تعریف آرایه در PHP ۵۸
- ۴-۲- آرایه های شمارشی ۵۸
- ۴-۳- آرایه های انجمنی ۵۹
- ۴-۴- روش نگه داری رکوردها در حافظه ۶۰
- ۴-۵- حلقه تکرار foreach ۶۱
- ۴-۶- آرایه های تودرتو ۶۳
- ۴-۷- طراحی صفحه نمایش لیست کتاب ها ۶۵

فصل پنجم - توابع در زبان PHP

- ۵-۱- انواع توابع ۷۲
- ۵-۲- روش تعریف و فراخوانی تابع ۷۳
- ۵-۳- استفاده از پارامترها در تعریف تابع ۷۴
- ۵-۴- روش مقداردهی پیش فرض پارامترها ۷۷
- ۵-۵- فراخوانی تابع به صورت پویا ۷۷
- ۵-۶- فراخوانی تابع با ارجاع ۷۸
- ۵-۷- متغیرهای محلی و سراسری ۷۸
- ۵-۸- توابع پیش ساخته در PHP ۸۰
- ۵-۸-۱- توابع کار با متغیرها ۸۰
- ۵-۸-۲- توابع کار با رشته ها ۸۱
- ۵-۸-۳- توابع محاسباتی ۸۲
- ۵-۸-۴- توابع دستکاری آرایه ها ۸۳
- ۵-۸-۵- توابع تاریخ و ساعت ۸۴
- ۵-۹- توابع ضمیمه کردن فایل ها ۸۵
- ۵-۱۰- استفاده از فایل ها در طراحی صفحات ۸۷

فصل ششم - طراحی و پردازش فرم ها

- ۶- طراحی فرم ها با HTML ۹۶

فصل نهم - کار با نرم افزار phpMyAdmin

۱۵۲	۹-۱ اجرای برنامه phpMyAdmin
۱۵۳	۹-۲ مدیریت بانک های اطلاعاتی
۱۵۴	۹-۳ ایجاد جداول
۱۵۷	۹-۳-۱ زبانه های مربوط به جدول
۱۵۸	۹-۳-۲ اصلاح ستون های جدول
۱۵۹	۹-۳-۳ درج رکورد جدید در جدول
۱۵۹	۹-۳-۴ مدیریت رکوردهای جدول
۱۶۰	۹-۳-۵ تغییر نام، انتقال و یا کپی جدول
۱۶۱	۹-۴ اجرای دستورات SQL
۱۶۱	۹-۵ عملیات روی بانک اطلاعاتی
۱۶۲	۹-۶ ورود و صدور اطلاعات

فصل دهم - تحلیل و طراحی سایت اینترنتی فروشگاه کتاب

۱۶۶	۱۰-۱ معرفی سیستم فروشگاه اینترنتی
۱۶۷	۱۰-۲ عملکرد فروشگاه اینترنتی کتاب
۱۶۸	۱۰-۳ نقش های کاربران
۱۶۹	۱۰-۴ طراحی بانک اطلاعاتی فروشگاه
۱۷۲	۱۰-۵ طراحی ساختار سایت
۱۷۵	۱۰-۶ طراحی صفحات سایت

فصل یازدهم - طراحی قالب برای صفحات فروشگاه کتاب

۱۸۰	۱۱-۱ طراحی قالب سایت
۱۸۱	۱۱-۲ طراحی قالب برای صفحات مدیریتی
۱۸۱	۱۱-۲-۱ ساختار قالب
۱۸۲	۱۱-۲-۲ طراحی فایل header
۱۸۴	۱۱-۲-۳ طراحی فایل footer
۱۸۷	۱۱-۳ طراحی قالب برای صفحات مدیریتی
۱۸۷	۱۱-۳-۱ ساختار قالب
۱۸۸	۱۱-۳-۲ طراحی header
۱۹۰	۱۱-۳-۳ طراحی footer
۱۹۱	۱۱-۳-۴ طراحی استایل شیت

۹۷	۶-۱-۱ تگ form
۹۷	۶-۱-۲ تگ input
۱۰۰	۶-۱-۳ تگ textarea
۱۰۱	۶-۱-۴ تگ های select و option
۱۰۳	۶-۲ طراحی فرم ها در محیط dreamweaver
۱۰۶	۶-۳ عملکرد کلی فرم ها
۱۰۶	۶-۴ روش های ارسال فرم
۱۰۷	۶-۵ دسترسی به اطلاعات فرم
۱۱۰	۶-۶ آرایه \$_REQUEST
۱۱۲	۶-۷ ادغام طراحی و پردازش فرم
۱۱۳	۶-۸ استفاده از آرایه ها در فرم

فصل هفتم - بارگذاری و پردازش فایل ها

۱۱۸	۷-۱ طراحی فرم برای بارگذاری فایل
۱۲۰	۷-۲ دریافت و ذخیره فایل روی سرورس دهنده
۱۲۲	۷-۳ نمایش فایل های بارگذاری شده
۱۲۳	۷-۴ تعیین محدودیت برای فایل ها
۱۲۶	۷-۵ پردازش فایل ها
۱۲۶	۷-۵-۱ بازکردن فایل
۱۲۷	۷-۵-۲ بستن فایل
۱۲۸	۷-۵-۳ نوشتن متن در فایل
۱۲۸	۷-۵-۴ بررسی انتهای فایل
۱۲۹	۷-۶ خواندن و نوشتن رکوردها در فایل

فصل هشتم - آشنایی با بانک اطلاعاتی MySQL

۱۳۶	۸-۱ سیستم های مدیریت بانک اطلاعاتی
۱۳۷	۸-۲ طراحی بانک اطلاعاتی
۱۴۱	۸-۳ معرفی دستورات مهم SQL
۱۴۱	۸-۳-۱ دستور ایجاد اجزای بانک اطلاعاتی
۱۴۳	۸-۳-۲ دستور خواندن از بانک اطلاعاتی
۱۴۵	۸-۳-۳ دستورات به روزرسانی داده ها
۱۴۷	۸-۴ معرفی MySQL
۱۴۸	۸-۵ مدیریت بانک های اطلاعاتی MySQL

فصل دوازدهم - اتصال به بانک اطلاعاتی MySQL و نمایش

اطلاعات

- ۱۹۶-۱۲-۱ دسترسی به بانک اطلاعاتی MySQL.....
- ۱۹۷-۱۲-۲ دستورات اتصال به بانک اطلاعاتی.....
- ۱۹۸-۱۲-۳ مراحل لازم برای کار با بانک اطلاعاتی.....
- ۲۰۰-۱۲-۴ توابع کار با بانک اطلاعاتی.....
- ۲۰۱-۱۲-۵ نمایش اطلاعات جدول.....
- ۲۰۵-۱۲-۶ نمایش کتاب‌ها در فروشگاه اینترنتی.....
- ۲۰۵-۱۲-۶-۱ طراحی صفحه نمایش لیست کتاب‌ها.....
- ۲۰۷-۱۲-۶-۲ طراحی صفحه نمایش جزئیات کتاب.....
- ۲۰۹-۱۲-۷ نمایش لیست موضوعات.....

فصل سیزدهم - افزودن رکورد به بانک اطلاعاتی

- ۲۱۴-۱۳-۱ درج رکورد به بانک اطلاعاتی.....
- ۲۱۷-۱۳-۲ دریافت اطلاعات رکورد جدید از کاربر.....
- ۲۱۹-۱۳-۳ استفاده از انواع ورودی‌ها در فرم.....
- ۲۲۵-۱۳-۴ طراحی فرم افزودن کتاب با استفاده از قالب.....

فصل چهاردهم - حذف و ویرایش رکوردهای بانک اطلاعاتی

- ۲۳۴-۱۴-۱ حذف رکوردهای بانک اطلاعاتی.....
- ۲۳۶-۱۴-۲ ویرایش رکوردهای بانک اطلاعاتی.....
- ۲۳۹-۱۴-۳ تغییر مسیر درخواست‌ها.....
- ۲۴۰-۱۴-۴ تکمیل مدیریت کتاب‌ها در فروشگاه اینترنتی.....
- ۲۴۰-۱۴-۴-۱ طراحی صفحه نمایش لیست کتاب‌ها.....
- ۲۴۲-۱۴-۴-۲ طراحی صفحه ویرایش کتاب.....
- ۲۴۶-۱۴-۴-۳ ویرایش گروهی رکوردها.....

فصل پانزدهم - جست‌وجو و محدود کردن نمایش رکوردها

- ۲۵۴-۱۵-۱ بازیابی و نمایش رکوردها بر اساس کلید خارجی.....
- ۲۵۵-۱۵-۲ جست‌وجوی رکوردها.....
- ۲۵۹-۱۵-۳ محدود کردن تعداد رکوردها.....
- ۲۶۱-۱۵-۴ صفحه‌بندی نمایش رکوردها.....

فصل شانزدهم - کار با Sessionها

- ۲۶۸-۱۶-۱ تعریف Session.....
- ۲۶۹-۱۶-۲ دستورات PHP برای کار با Session.....
- ۲۶۹-۱۶-۲-۱ شروع به کار Session.....
- ۲۶۹-۱۶-۲-۲ نوشتن داده‌ها در Session.....
- ۲۷۰-۱۶-۲-۳ خواندن داده‌ها از Session.....
- ۲۷۰-۱۶-۲-۴ از بین بردن Session.....
- ۲۷۱-۱۶-۳ کاربرد Session در نمایش پیغام بازخورد.....
- ۲۷۳-۱۶-۴ پیاده‌سازی سبد خرید فروشگاه.....

فصل هفدهم - احراز هویت و کنترل دسترسی کاربران

- ۲۸۲-۱۷-۱ کنترل دسترسی کاربران.....
- ۲۸۳-۱۷-۲ احراز هویت کاربران.....
- ۲۸۶-۱۷-۳ پیاده‌سازی احراز هویت برای فروشگاه کتاب.....
- ۲۸۶-۱۷-۳-۱ ایجاد جدول کاربران فروشگاه.....
- ۲۸۷-۱۷-۳-۲ پیاده‌سازی صفحه ورود کاربر.....
- ۲۸۹-۱۷-۳-۳ کنترل دسترسی به صفحات سایت.....
- ۲۹۲-۱۷-۳-۴ خروج کاربر از سایت.....
- ۲۹۳-۱۷-۴ پیاده‌سازی صفحه تغییر کلمه عبور.....
- ۲۹۵-۱۷-۵ پیاده‌سازی امکانات شخصی کاربران.....
- ۲۹۶-۱۷-۶ صفحه پیگیری سفارشات قبلی.....

منابع ۳۰۳

خداوند بزرگ را سپاسگزارم که به من فرصتی عطا کرد تا بتوانم خدمتی در جهت کمک به فرزندان این مرز و بوم ارائه نموده و قدمی در جهت پیشرفت ایران سربلند بردارم.

کتاب حاضر برای آموزش طراحی نرم‌افزارهای کاربردی مبتنی بر وب با استفاده از زبان برنامه‌نویسی PHP تألیف شده است. PHP یک زبان برنامه‌نویسی سمت سرور می‌دهنده بوده و رایگان و قابل حمل و کد منبع باز است و دارای محبوبیت بالایی می‌باشد.

محتوای آموزشی ارائه شده در این کتاب با هدف آموزش اصول کلی و ضروری به هنرجویان و افزایش هر چه بیشتر مهارت آنها برای طراحی و پیاده‌سازی نرم‌افزارهای کاربردی مبتنی بر وب تدوین شده و همواره سعی شده است تا پیوستگی کلی و هدفمند بین مطالب و مثال‌های برنامه‌نویسی وجود داشته باشد.

شیوه آموزش در این کتاب به صورت پروژه محور بوده و بر اساس سایت یک فروشگاه اینترنتی کتاب تدوین شده است به گونه‌ای که در هر فصل کتاب بعد از ارائه مفاهیم کلی و مثال‌های طراحی و برنامه‌نویسی مربوطه، صفحاتی از فروشگاه کتاب طراحی و نمایش داده شده است.

در فصل اول روش نصب نرم‌افزارها و آماده‌سازی محیط برای شروع برنامه‌نویسی PHP و روش اجرای صفحات PHP آموزش داده می‌شود و در فصل‌های دوم تا هفتم اصول برنامه‌نویسی به زبان PHP از جمله اجزای تشکیل‌دهنده برنامه‌ها، ساختارهای شرطی و تکرار، استفاده از آرایه‌ها، طراحی فرم‌ها و پردازش آنها به همراه بارگذاری فایل‌ها آموزش داده شده است.

در فصل هشتم و نهم روش تحلیل داده‌ها در یک محیط عملیاتی و طراحی بانک اطلاعاتی آموزش داده شده و دستورات مهم زبان SQL و روش ایجاد و مدیریت بانک اطلاعاتی MySQL با کمک نرم‌افزار phpMyAdmin آموزش داده شده است.

در فصل دهم، یک سیستم فروشگاه اینترنتی کتاب به صورت خلاصه مورد بررسی و تحلیل قرار گرفته، امکانات اصلی آن، نقش‌های کاربری و عملکرد هر نقش تعیین شده و بانک اطلاعاتی آن طراحی شده است. همچنین ساختار سایت فروشگاه و طرح کلی هر کدام از صفحات آن تعیین شده است و در فصل یازدهم با استفاده از CSS و bootstrap دو قالب برای صفحات عمومی و صفحات مدیریتی سایت فروشگاه اینترنتی طراحی شده و نحوه استفاده از آنها تشریح شده است.

در فصل‌های دوازدهم تا پانزدهم، عملیات مختلف روی بانک اطلاعاتی آموزش داده شده ضمن اینکه در انتهای هر فصل صفحات کامل برای ساخت بخش‌های مختلف فروشگاه طراحی و کدنویسی شده است و در فصل شانزدهم و هفدهم sessionها معرفی شده و کنترل دسترسی به بخش‌های مختلف فروشگاه پیاده‌سازی شده است.

در پایان امیدوارم این اثر بتواند برای استفاده‌کنندگان مفید واقع شود و از زحمات همه کسانی که در مراحل تألیف، بررسی و ویرایش محتوای کتاب همکاری داشتند، قدردانی می‌نمایم.



فصل اول

نصب و راه اندازی سرویس دهنده وب



هدف‌های رفتاری

- انتظار می‌رود هنرجو در پایان این فصل بتواند:
- وب سرور را نصب و راه‌اندازی کند.
- فایل تنظیمات وب سرور را باز کرده و تنظیمات را تغییر دهد.
- با استفاده از مرورگر به وب سرور متصل شود.
- اسکریپت PHP را ایجاد کرده و در پوشه وب سرور ذخیره کند.
- اسکریپت‌های PHP را اجرا کند.

۱-۱- برنامه‌نویسی مبتنی بر وب

برنامه‌های کاربردی تحت وب، دسته بزرگی از برنامه‌ها هستند که در بستر وب قرار گرفته و اجرا می‌شوند. کاربران می‌توانند از هر جای دنیا بدون نیاز به نصب آنها روی سیستم تنها با اتصال به اینترنت و اجرای یک مرورگر وب ساده، به این برنامه‌ها وارد شده و از آنها استفاده کنند.

این راحتی استفاده و دسترس‌پذیری باعث شده است تا سازمان‌ها برای انجام امور خود، به ویژه زمانی که دارای شعبه‌های مختلف هستند و یا با انبوه کاربران پراکنده در سطح کشور یا جهان سروکار دارند، از برنامه‌های کاربردی تحت وب استفاده کنند.

با کمک نرم‌افزارهای تحت وب، شرکت‌های تجاری بزرگ می‌توانند با مشتریان خود در سرتاسر جهان ارتباط داشته، آخرین محصولات خود را به‌صورت سریع در سطح جهانی عرضه کرده و خدمات پشتیبانی را ارائه دهند. با استفاده از این نرم‌افزارها، مرزها برداشته شده و فعالیت شرکت‌ها در سطح جهانی گسترش می‌یابد. طراحی و برنامه‌نویسی مبتنی بر وب از اهمیت و جایگاه بالایی در بازار توسعه نرم‌افزار برخوردار بوده و برنامه‌نویسان وب، اغلب دارای بازار کار خوب و درآمدهای بالا هستند.

برای ایجاد برنامه‌های کاربردی تحت وب باید به آخرین ابزارها، زبان‌ها و فناوری‌ها و تغییرات آنها دسترسی داشته و به استفاده از آنها تسلط داشته باشید. بنابراین نیاز است تا به‌صورت مداوم در حال یادگیری بوده و اطلاعات و مهارت‌های خود را در این زمینه افزایش دهید.

یک برنامه کاربردی تحت وب مجموعه‌ای از صفحات وب است که برای طراحی آن از زبان‌ها و ابزارهایی مثل HTML، PHP، CSS و bootstrap استفاده می‌گردد. زبان PHP برای برنامه‌نویسی و ایجاد صفحات وب پویا کاربرد دارد که در این درس برنامه‌نویسی آن را فرا می‌گیرید. با کمک این زبان خواهید توانست برنامه‌های کاربردی تحت وب کامل را طراحی و پیاده‌سازی و اجرا کنید.

۱-۲- معرفی زبان برنامه‌نویسی PHP

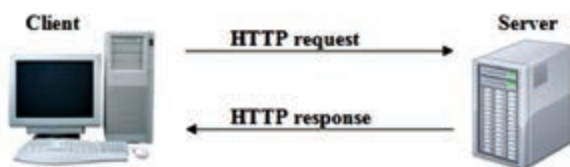
PHP از عبارت PHP:HyperText Preprocessor گرفته شده است که به معنی پیش پردازشگر ابرمتن می‌باشد. یک زبان برنامه‌نویسی مبتنی بر وب است و دستورات آن در صفحه وب نوشته شده و باعث پویایی صفحه می‌شود. این دستورات می‌تواند شامل دستورات محاسباتی، ساختارهای شرطی، حلقه‌های تکرار باشد که در زمان درخواست صفحه، اجرا شده و باعث پویایی محتوای آن می‌شود. در ادامه به مهم‌ترین ویژگی‌های زبان برنامه‌نویسی PHP اشاره می‌شود:

PHP یک زبان رایج برای برنامه‌نویسی وب است که از آخرین مفاهیم مورد استفاده در برنامه‌نویسی پشتیبانی کرده و دارای امکانات برنامه‌نویسی و کتابخانه‌های فوق‌العاده برای نوشتن برنامه در زمان کوتاه است. PHP یک زبان با کد منبع باز است و برنامه‌های نوشته شده با آن به صورت سورس کد عرضه می‌شوند و استفاده کننده به کدهای برنامه دسترسی دارد. به علاوه یک زبان رایگان است یعنی برای تهیه مفسر PHP هزینه‌ای پرداخت نمی‌شود.

یکی از مزایای زبان PHP قابل حمل بودن آن است به این علت که مفسرهای مختلفی از PHP برای نصب و اجرا در محیط‌های مختلف وجود دارد. در نتیجه نرم‌افزارهای نوشته شده به زبان PHP می‌توانند بدون هیچ‌گونه تغییری روی اکثر سخت‌افزارها و سیستم عامل‌ها (مثل ویندوز و لینوکس) اجرا شوند. قابل حمل بودن PHP این امکان را فراهم می‌کند تا برنامه‌ای که در محیط ویندوز پیاده‌سازی می‌شود بعداً بتواند روی سرورهایی که از سیستم عامل لینوکس استفاده می‌کنند، نصب و اجرا شود. این ویژگی به شدت مورد علاقه مؤسسات دولتی است چون وابستگی کشور به سیستم عامل ویندوز را کمتر می‌کند. بر اساس آمارها در برنامه‌نویسی بیش از ۷۰ درصد سایت‌های اینترنتی از زبان PHP استفاده شده است. برای ساخت و پشتیبانی این سایت‌ها نیاز به برنامه‌نویسان PHP است. بنابراین با یادگیری زبان PHP دارای یک مهارت مفید و کاربردی شده و در آینده فرصت‌های شغلی زیادی پیش روی شما خواهد بود. جامعه بزرگی از برنامه‌نویسان در سرتاسر جهان از زبان PHP استفاده کرده و از آن حمایت می‌کنند، این جوامع در تولید نسخه‌های جدید PHP و اضافه کردن امکانات و تولید کتابخانه‌های مفید برای آن همکاری می‌کنند. منابع آموزشی متنی و تصویری زیادی برای آموزش زبان PHP وجود داشته و افراد زیادی در این زمینه مهارت دارند و برای هر سؤال یا مشکل با طرح آن در انجمن‌های گفت‌وگویی اینترنتی می‌توان پاسخ مناسبی برای آن پیدا کرد.

۳-۱-۱- نرم‌افزارهای مورد نیاز برای برنامه‌نویسی PHP

همان‌طور که می‌دانید در وب برای درخواست صفحات و دریافت پاسخ از پروتکل HTTP استفاده می‌شود. در این پروتکل یک سرویس‌دهنده وب و چند سرویس‌گیرنده وب وجود دارد و همیشه در ابتدا سرویس‌گیرنده درخواست صفحه را به سرویس‌دهنده ارسال کرده و منتظر پاسخ می‌ماند تا سرویس‌دهنده پاسخ را آماده کرده و برگشت دهد. در شکل ۱-۱ روش کار پروتکل HTTP را مشاهده می‌کنید.



شکل ۱-۱- نحوه کار پروتکل HTTP

برنامه‌نویسی به زبان PHP از نوع سمت سرور می‌باشد. بنابراین صفحات وبی که دارای دستورات PHP باشند، باید روی سرور سرورس‌دهنده وب قرار گرفته، از طرف سرورس‌گیرنده درخواست شده و توسط سرورس‌دهنده اجرا شده و خروجی آنها به سرورس‌گیرنده برگشت داده شود.

یک سیستم که برای نگه‌داری و اجرای صفحات PHP استفاده می‌شود باید دارای نرم‌افزار سرورس‌دهنده وب باشد و نیز برای تفسیر دستورات PHP نیاز به مفسر این زبان است و اگر در برنامه از بانک‌اطلاعاتی استفاده شود (که معمولاً هم استفاده می‌شود) نیاز به سرورس‌دهنده داده خواهد بود. از طرفی برای درخواست برنامه‌ها نیاز به سرورس‌گیرنده وب می‌باشد.

به‌طور خلاصه برای اجرای برنامه‌های PHP، به سرورس‌دهنده وب، مفسر PHP و سرورس‌دهنده داده نیاز است که همه این امکانات باید روی یک سیستم نصب شده و در حال اجرا باشند تا آن سیستم بتواند به‌عنوان سرورس‌دهنده عمل کرده و برنامه‌های PHP را اجرا کند.

در ادامه به معرفی هر کدام از این امکانات پرداخته می‌شود:

سرورس‌دهنده وب نرم‌افزاری است که می‌تواند روی یک کامپیوتر سرور یا حتی کامپیوتر معمولی نصب و اجرا شود. نرم‌افزار Apache یکی از رایج‌ترین سرورس‌دهنده‌های وب است که روی انواع سیستم‌ها قابل نصب و اجرا می‌باشد.

سرورس‌گیرنده وب برنامه‌ای است که روی یک سیستم اجرا شده و درخواست‌های خود را به سرورس‌دهنده فرستاده و پاسخ دریافت شده را استفاده کرده یا نمایش می‌دهد. مرورگرهای وب یک نمونه از سرورس‌گیرنده‌های وب هستند.

درخواست‌های سرورس‌گیرنده از سرورس‌دهنده همیشه به‌صورت URL هستند که شامل نام پروتکل، نام دامنه یا آدرس سرورس‌دهنده، مسیر و نام و پسوند فایل درخواستی می‌باشند. در شکل ۱-۲ ساختار یک URL را مشاهده می‌کنید:

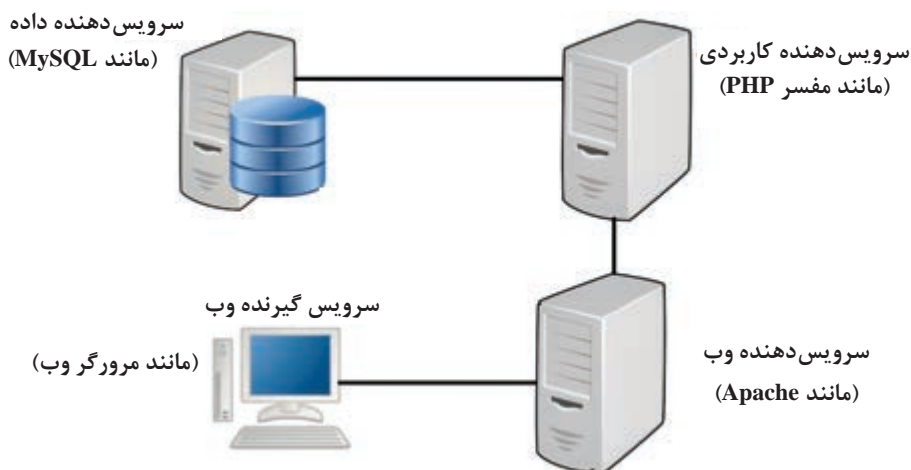


شکل ۱-۲- قالب URL

سرورس‌دهنده کاربردی مجموعه‌ای از امکانات نرم‌افزاری است که برای تفسیر و اجرای دستورات نوشته شده به یک زبان برنامه‌نویسی مورد نیاز است. به عنوان مثال برای اجرای دستورات PHP نیاز به مفسر زبان PHP می‌باشد.

سرورس‌دهنده داده که به آن سیستم مدیریت بانک‌اطلاعاتی هم می‌گویند برای ایجاد بانک‌اطلاعاتی و دسترسی به آن مورد نیاز است. MySQL یکی از نرم‌افزارهایی است که به این منظور استفاده می‌شود و در فصل‌های بعدی با آن آشنا می‌شوید.

در شکل ۱-۳ ارتباط بین این قسمت‌ها نشان داده شده است. همه این برنامه‌ها می‌تواند روی یک سیستم یا هر کدام از آنها روی یک سیستم جداگانه نصب شده باشد و در هر صورت می‌توانند با هم ارتباط برقرار کرده و وظیفه خود را به درستی انجام دهند.



شکل ۱-۳- امکانات نرم‌افزاری سرویس‌دهنده

۴-۱- آموزش نصب نرم‌افزارها

گفته شد برای تبدیل یک سیستم به سرویس‌دهنده‌ای که بتواند برنامه‌های PHP را اجرا کند، نیاز به نصب چند برنامه است. برای راحتی کار، همه این برنامه‌ها در بسته‌هایی قرار داده شده‌اند که در صورت نصب هر کدام از این بسته‌ها، نرم‌افزارهای سرویس‌دهنده وب apache، مفسر زبان PHP و سرویس‌دهنده داده MySQL به همراه امکانات جانبی دیگر روی سیستم، نصب خواهند شد. این بسته‌ها عبارت‌اند از:

EasyPHP ■

WAMP Server ■

XAMP ■

همه این بسته‌ها به صورت رایگان عرضه شده و به صورت کامل و قانونی قابل دریافت و نصب هستند. نصب یکی از آنها کافی است تا سیستم برای برنامه‌نویسی PHP آماده شود. برای دریافت هر کدام از این بسته‌ها می‌توانید نام آن را در اینترنت جست‌وجو کنید.

از آنجا که Wamp Server برای نصب و استفاده راحت‌تر است، در اینجا روش دانلود، نصب و تنظیم آن را آموزش می‌دهیم.

برای دریافت جدیدترین نسخه WampServer به سایت مرجع آن با آدرس زیر وارد شوید:

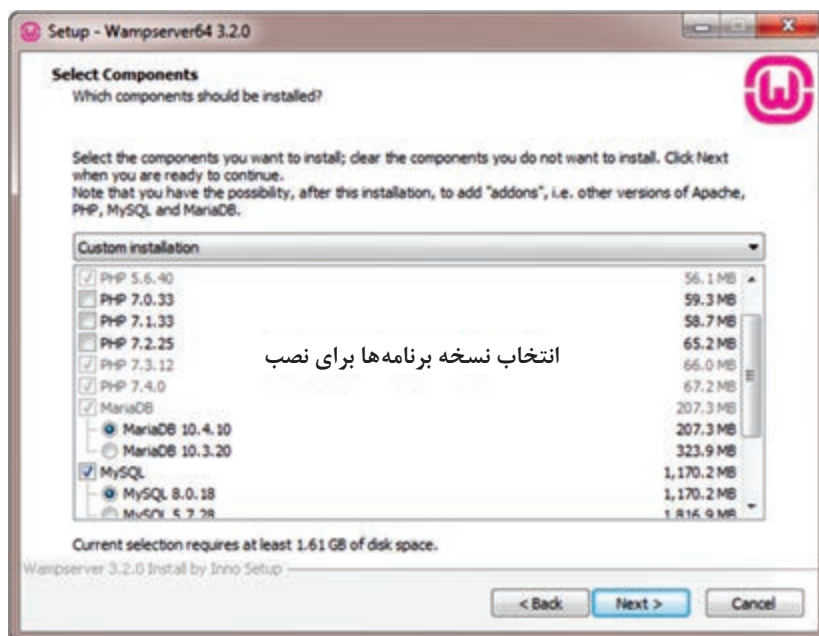
<https://www.wampserver.com>

سپس با توجه به نسخه ویندوز سیستم خودتان، نسخه ۳۲ بیتی یا ۶۴ بیتی را دانلود کنید. جدیدترین نسخه در زمان نگارش کتاب ۳,۲,۰ بوده است.



شکل ۴-۱- Wamp Server

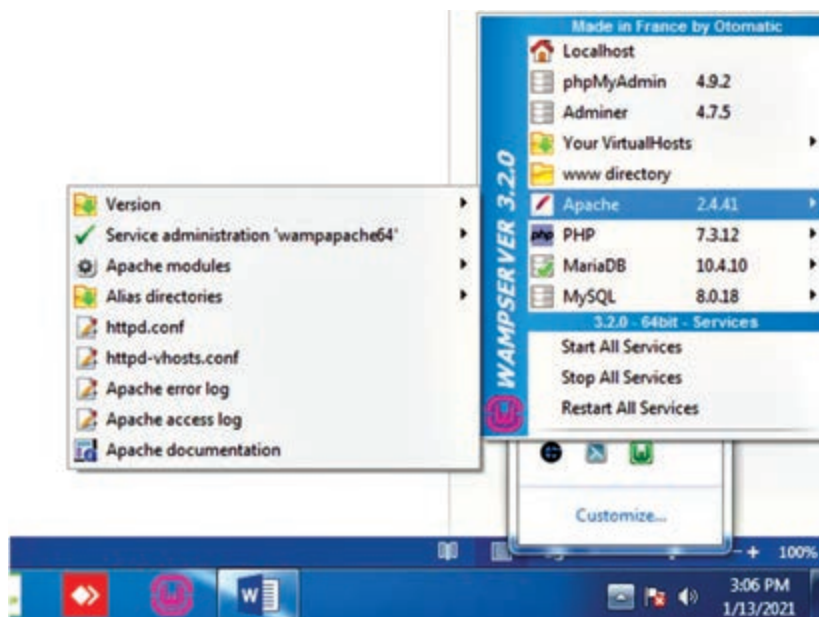
پس از دانلود فایل قابل نصب Wamp Server، آن را اجرا کرده و مراحل آن را ادامه دهید. در طول نصب برنامه پوشه محل نصب نرم افزار سؤال می شود که می تواند به عنوان مثال c:\wamp64 باشد. همچنین در یکی از مراحل مطابق شکل ۵-۱ می توانید نسخه برنامه ها را برای نصب، انتخاب کنید:



شکل ۵-۱- تعیین امکانات برای نصب در Wamp Server

این نسخه از Wamp Server، به طور پیش فرض PHP نسخه ۷,۳ و ۷,۴ و برای MySQL نسخه ۸,۰ را نصب می کند. می توانید نسخه های دیگر را هم در صورت نیاز انتخاب کنید. در پایان روی Next کلیک کنید تا نصب انجام شود. هر نسخه جدید PHP دارای امکانات جدیدتر است و ممکن است برخی از امکانات نسخه قبلی را نداشته باشد. برنامه هایی که با یک نسخه از PHP نوشته شده اند ممکن است با نسخه های جدید PHP سازگاری نداشته باشند و برای اجرای آنها نیاز به نسخه های قبلی باشد.

پس از نصب Wamp Server، می‌توانید آن را اجرا کنید تا آیکون W مانند آن در گوشه سمت راست نوار وظیفه (ناحیه هشدار) ظاهر شود. با نصب WAMP در اصل شما مجموعه‌ای از امکانات را نصب کرده‌اید و چنانچه روی این آیکون کلیک کنید در منوی باز شده مطابق شکل ۱-۶ گزینه‌هایی برای ورود و مدیریت این امکانات را خواهید دید:



شکل ۱-۶ پنجره تنظیمات WAMP

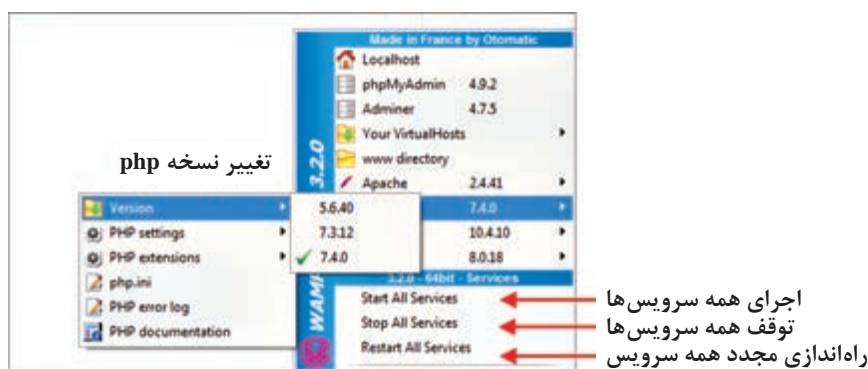
اصلی‌ترین امکاناتی که به نصب Wamp Server روی سیستم نصب می‌شود عبارت‌اند از:

■ **Apache:** برنامه سرویس‌دهنده وب

■ **PHP:** مفسر زبان PHP که برای اجرای دستورات PHP استفاده خواهد شد.

■ **MySQL:** برنامه مدیریت بانک اطلاعاتی

برای هر کدام از این برنامه‌ها، یک زیرمنو با تعدادی گزینه خواهید دید که با کمک آنها می‌توانید کارهای مختلفی انجام دهید. به عنوان مثال برای تغییر نسخه PHP در حال اجرا می‌توانید از زیر منوی PHP و سپس زیر منوی داخلی Version مطابق شکل ۱-۷ استفاده کنید.



شکل ۱-۷ تغییر نسخه PHP

همان‌گونه که در شکل ۷-۱ دیده می‌شود، نسخه فعلی PHP دارای شماره ۷,۴,۰ می‌باشد، برای انتخاب نسخه دیگر می‌توانید روی نسخه مورد نظر کلیک کنید و توجه داشته باشید که برای اعمال این تنظیمات باید با اجرای گزینه Restart All Services سرویس‌ها را دوباره راه اندازی کنید.

رنگ آیکون W نشان‌دهنده وضعیت سلامت و اجرای برنامه‌ها می‌باشد:

■ رنگ سبز: همه برنامه‌ها به درستی اجرا شده و آماده استفاده هستند.

■ رنگ نارنجی: برخی از برنامه‌ها با مشکل مواجه هستند.

■ رنگ قرمز: همه برنامه‌ها با مشکل مواجه هستند.

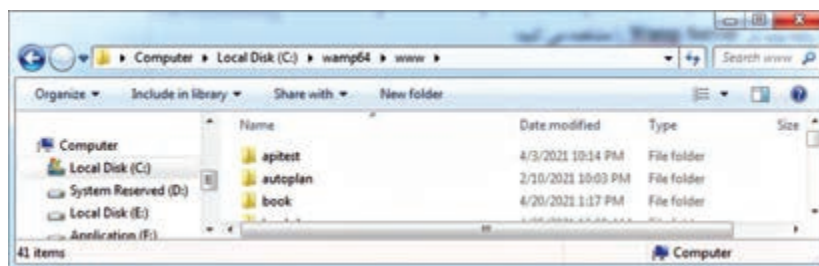
در صورتی که رنگ W سبز نباشد باید با تغییر دادن تنظیمات یا نصب مجدد، اشکالات را برطرف کنید. یکی از اشکالاتی که ممکن است هنگام نصب Wamp Server پیش آید، اشغال بودن پورت ۸۰ است که به‌طور پیش‌فرض، توسط Apache استفاده می‌شود. برای رفع این مشکل، می‌توان پورت مورد استفاده Apache را به عددی دیگر تنظیم کرد که برای این منظور، فایل httpd.conf (که در شکل ۶-۱ دیده می‌شود) را باز کرده و عدد ۸۰ را جست‌وجو کنید و آن را به عددی دیگر مثل ۸۱ تغییر داده و فایل را ذخیره نمایید سپس همه سرویس‌ها را دوباره راه اندازی کنید.

هر برنامه اجرا شده روی رایانه از جمله برنامه سرویس‌دهنده وب (Apache) از شماره پورت مشخصی استفاده می‌کند. این شماره، مشابه کدپستی برنامه است و برنامه‌های دیگر هنگام ارتباط، پیام‌ها را به این شماره ارسال می‌کنند. سرویس‌دهنده‌های وب، به‌طور پیش‌فرض از پورت شماره ۸۰ استفاده می‌کند. حال اگر قبلاً برنامه‌ای دیگر این پورت را استفاده کرده باشد، سرویس‌دهنده وب Apache با مشکل مواجه می‌شود و رنگ آیکون W هم قرمز خواهد بود.

۵-۱- آموزش اتصال به سرویس‌دهنده

هر کدام از بسته‌های ذکر شده، پس از نصب، قابلیت نگه‌داری صفحات و سایت‌های وب و پاسخگویی به درخواست‌های سرویس‌گیرندگان را دارند. هر کدام از آنها دارای یک فهرست مخصوص هستند که فهرست ریشه مجازی وب نامیده شده و فایل‌های صفحات وب، برنامه‌ها، تصاویر و غیره باید در آن فهرست ذخیره شوند. تا از طریق وب قابل دسترس باشند.

نام و مسیر فهرست ریشه وب بستگی به بسته نصب شده دارد. به‌عنوان مثال در صورت نصب Wamp Server، نام فهرست ریشه وب، www و در صورت نصب XAMP، نام این فهرست htdocs خواهد بود. در شکل ۸-۱، فهرست ریشه وب در Wamp Server را مشاهده می‌کنید:



شکل ۸-۱- فهرست ریشه وب

برای باز کردن فهرست ریشه مجازی وب (www) همچنین می‌توانید از گزینه `www directory` که در منوی مدیریتی `wamp server` وجود دارد و در شکل ۱-۶ دیده می‌شود، استفاده کنید.

وقتی سرویس‌گیرنده درخواست صفحه‌ای را به سرویس‌دهنده می‌فرستد، سرویس‌دهنده به این فهرست مراجعه کرده و فایل آن را جست‌وجو می‌کند، اگر وجود داشته، برای درخواست دریافت شده، پاسخ ارسال می‌شود. در غیر این صورت خطای ۴۰۴ برگشت داده می‌شود که به معنی پیدا نشدن صفحه درخواست شده می‌باشد. صفحات `PHP` باید در داخل فهرست ریشه وب یا زیرپوشه‌های آن قرار بگیرند تا بتوانند توسط مرورگر درخواست شده و اجرا شوند. برای اجرای اسکریپت‌های `PHP` توسط مرورگر باید ابتدا مرورگر درخواست صفحه را به سرویس‌دهنده بفرستد و سرویس‌دهنده در پاسخ، آنها را اجرا کند و نمی‌توان با باز کردن معمولی این فایل‌ها، در مرورگر، اجرای آنها را دید.

برای اتصال به سرویس‌دهنده وب از طریق پروتکل `HTTP` باید مرورگر وب را باز کرده و در نوار آدرس آن یکی از آدرس‌های زیر را وارد کرده و درخواست کنید:
نوشتن آی پی ۱,۰,۰,۱۲۷ که همیشه مشخص‌کننده رایانه محلی است.

`http://127.0.0.1`

(به جای شماره ۱ می‌توان اعداد ۱ تا ۲۵۵ را نیز استفاده کرد.)

نوشتن عبارت `localhost` که همیشه مشخص‌کننده رایانه محلی است.

`http://localhost`

نوشتن نام یا `IP` رایانه مورد نظر

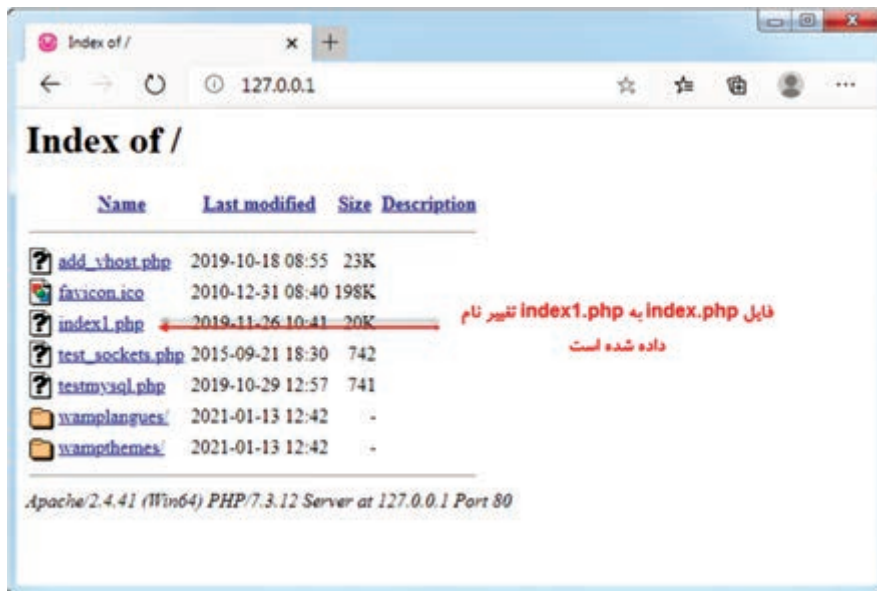
نام رایانه `http://`

رایانه `http:// IP`

همچنین برای اتصال به سرویس‌دهنده وب می‌توانید از گزینه `localhost` در منوی مدیریتی `wamp server` که در شکل ۱-۶ نشان داده شد، استفاده کنید.

سرویس‌دهنده پس از دریافت هر کدام از درخواست‌های فوق، فهرست ریشه وب را برای فایلی با نام `index.php` بررسی می‌کند، اگر این فایل وجود داشته، آن را اجرا کرده و خروجی آن را بر می‌گرداند. در غیر این صورت معمولاً لیست فایل‌ها و زیر فهرست‌های آن را به عنوان پاسخ برگشت می‌دهد.

برای راحتی کار پیشنهاد می‌شود: فایل `index.php` را از فهرست ریشه وب حذف کرده یا آن را به اسمی دیگر تغییر نام دهید تا بتوانید با اتصال به سرویس‌دهنده، لیست فایل‌ها و پوشه‌ها را مشابه شکل ۱-۹ ببینید:



شکل ۹-۱- اتصال به سرویس دهنده وب

اگر سرویس دهنده وب از شماره پورت دیگری غیر از ۸۰ استفاده کند باید بعد از آدرس های ذکر شده در بالا علامت : (دو نقطه روی هم) و سپس شماره پورت جدید را بنویسیم. به عنوان مثال اگر شماره پورت سرویس دهنده را به ۸۱ تغییر داده اید، برای اتصال به سرویس دهنده، آدرس `http://127.0.0.1:81` یا `http://localhost:81` را استفاده کنید.

نکته



۶-۱- آموزش ایجاد اسکریپت های PHP

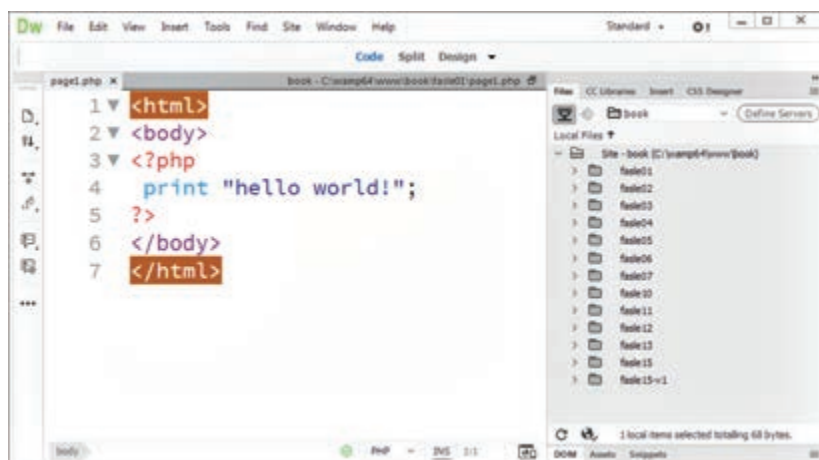
زبان PHP یک زبان تعبیه شده در HTML است و دستورات PHP می توانند در لابه لای تگ های HTML نوشته شوند. فایل های برنامه نویسی PHP دارای پسوند php هستند و محتوای آنها مشابه صفحات وب از تگ های HTML تشکیل شده است با این تفاوت که در جاهایی از آنها دستورات PHP هم استفاده شده است.

برای طراحی و برنامه نویسی صفحات PHP می توان از ویرایشگرهای متنی ساده مثل Notepad یا Notepad++ یا هر محیط طراحی و برنامه نویسی پیشرفته تر مثل Dreamweaver، PHP Desinger، Web Storm، PHP Storm استفاده کرد. در این کتاب از نرم افزار DreamWeaver برای طراحی و برنامه نویسی استفاده شده است.

مثال ۱: صفحه ای برای نمایش پیغام «Hello World!» طراحی کنید.

اولین برنامه PHP را به صورت زیر با نام `page1.php` ایجاد کرده و در زیرپوشه `fasle\` از زیرپوشه `www` ذخیره کنید.

همان‌طور که قبلاً هم گفته شد، فایل‌های PHP که به آنها اسکریپت PHP هم گفته می‌شود برای اینکه بتوانند اجرا شوند باید در فهرست ریشه وب قرار گیرند (یعنی فهرست www در پوشه نصب Wamp Server). کد مربوطه در شکل ۱-۱۰ نشان داده شده است:

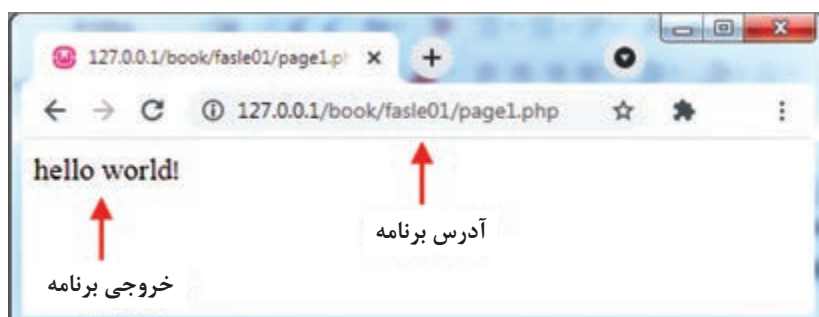


شکل ۱-۱۰ برنامه نمایش hello world

با نگاه به برنامه فوق در شکل ۱-۱۰ متوجه می‌شوید که دستورات PHP در داخل علامت‌های `<? php ?>` نوشته شده و برای نمایش اطلاعات از دستور `print` استفاده شده است. (به جای دستور `print` می‌توان از دستور `echo` هم استفاده کرد)

توصیه می‌شود فایل‌های مربوط به هر وب سایت را در زیرپوشه مشخصی قرار دهید. (به عنوان مثال در زمان نگارش این کتاب، در پوشه `www`، یک زیرپوشه به نام `book` ایجاد کرده و مثال‌های برنامه‌نویسی هر فصل را در یک زیرپوشه جداگانه به نام `fasle1`، `fasle2` و اسامی دیگر فصل‌ها ذخیره شده‌اند). برای اجرای برنامه فوق:

- ۱ مطمئن شوید که برنامه `Wamp Server` در حال اجرا بوده و آیکون آن به رنگ سبز است.
 - ۲ مرورگر را اجرا کرده و در نوار آدرس آن، `127.0.0.1` را نوشته و کلید `Enter` را فشار دهید.
 - ۳ با مشاهده لیست فهرست‌ها و فایل‌های داخل `www`، روی زیرپوشه مربوط به سایت خود (`fasle1`) کلیک کرده و سپس با کلیک روی برنامه `page1.php`، صفحه را اجرا کنید.
- نتیجه اجرای این برنامه مشابه شکل ۱-۱۱ نشان داده می‌شود:



شکل ۱-۱۱ اجرای برنامه hello world

یک بار دیگر هم یادآوری می‌شود: به نحوه آدرس دهی برنامه در شکل فوق دقت کنید. برای اجرای برنامه‌های PHP باید حتماً با اتصال به سرویس دهنده وب، آنها را درخواست و اجرا کنید.

مثال ۲: برنامه‌ای برای نمایش مشخصات PHP بنویسید.

حال که با روش برنامه‌نویسی و اجرای برنامه‌های PHP آشنا شدید در دومین برنامه تابع مفیدی از PHP را اجرا می‌کنید. تابع `phpinfo()` برای نمایش مشخصات و تنظیمات PHP استفاده می‌شود.

برای اجرای این تابع می‌توان یک فایل PHP جدید به نام `page2.php` با دستورات زیر ایجاد کرد:

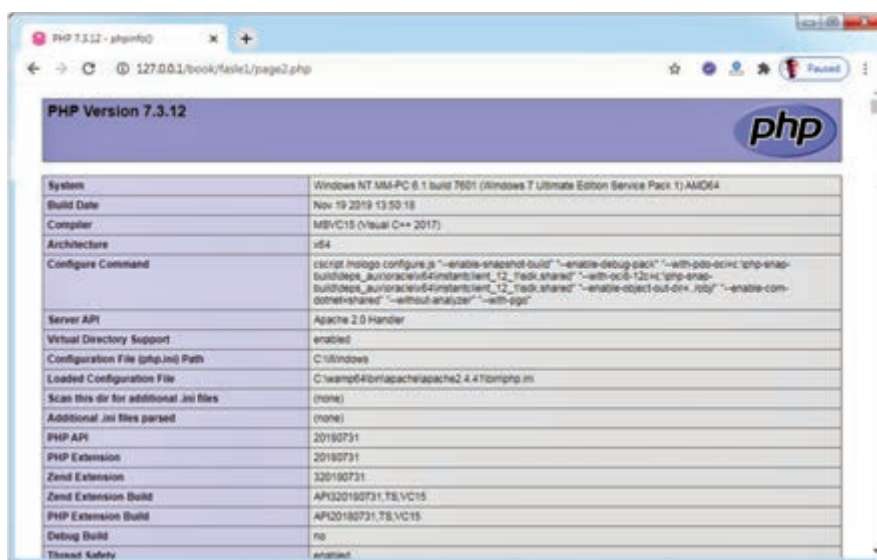
```
<?php
echo phpinfo();
?>
```

با اجرای این برنامه، مشخصات سیستم، برنامه‌های نصب شده با Wamp Server و تنظیمات آنها از جمله موارد زیر نمایش داده می‌شود:

جدول ۱-۱

ردیف	نام ویژگی	توضیح
۱	PHP Version	نسخه نصب شده و مورد استفاده PHP
۲	System	مشخصات سیستم
۳	Apache Version	نسخه نصب شده وب سرور آپاچی
۴	HTTP_HOST	آدرس وب سرور

خروجی حاصل از اجرای `page2.php` را در شکل ۱-۱۲ مشاهده می‌کنید:



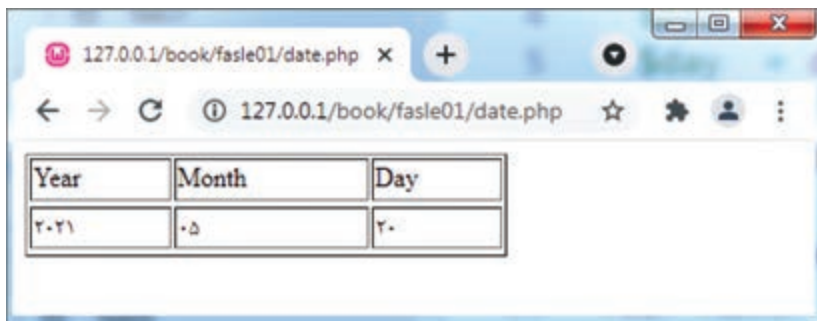
شکل ۱-۱۲- خروجی تابع `PHPinfo()`

مثال ۳: صفحه‌ای برای دریافت تاریخ جاری و نمایش آن در یک جدول طراحی کنید. در این مثال باید با کمک تابع `date()` از زبان PHP، شماره سال، ماه و روز را از سیستم گرفته و در سلول‌های یک جدول HTML نمایش دهید.

صفحه <code>date.php</code>
<pre> <html> <body> <?php \$year= date("Y"); \$mon = date("m"); \$day = date("d"); ?> <table border="1" width="300"> <tr> <td> Year </td> <td> Month </td> <td> Day </td> <tr> <tr> <td><?php print \$year; ?></td> <td><?php print \$mon; ?></td> <td><?php print \$day; ?></td> <tr> </table> </body> </html> </pre>

در صفحه فوق ابتدا با کمک تابع `date()` شماره سال، ماه و روز از سیستم دریافت شده و به ترتیب در متغیرهای `$year`، `$mon` و `$day` ذخیره شده است (در فصل‌های بعد با نحوه تعریف متغیرها و عملکرد کامل تابع `date()` آشنا خواهید شد). سپس با کمک تگ‌های HTML یک جدول طراحی شده و در سطر دوم آن هر کدام از متغیرها در یک سلول نمایش داده شده است. همان‌طور که می‌بینید در جاهای مختلف صفحه از دستورات PHP استفاده شده و هر بخش از دستورات PHP داخل علامت‌های `<?php?>` قرار گرفته‌اند. به هر کدام از این بخش‌ها یک بلاک دستور PHP گفته شده و می‌توانند شامل یک یا چند دستور PHP باشند.

نتیجه درخواست و اجرای صفحه date.php به صورت زیر می باشد:



شکل ۱۳-۱ نمایش تاریخ جاری

خودآزمایی

- ۱ چهار نمونه از ویژگی های زبان برنامه نویسی را بنویسید.
- ۲ ویژگی قابل حمل بودن PHP به چه معنا است و چه مزیتی دارد؟
- ۳ نحوه کار پروتکل HTTP را با رسم شکل شرح دهید.
- ۴ کاربر URL چیست و از چه قسمت هایی تشکیل شده است؟
- ۵ عبارت «برنامه نویسی PHP سمت سرور است» به چه معنا است؟ توضیح دهید.
- ۶ برای ایجاد نرم افزارهای کاربردی با PHP به چه برنامه هایی روی سیستم نیاز است؟
- ۷ سه بسته برای نصب PHP را نام ببرید.
- ۸ با نصب بسته WAMP سرور، چه برنامه هایی روی سیستم نصب می شود؟
- ۹ فهرست ریشه مجازی وب چیست؟ توضیح دهید.

پرسش های چهار گزینه ای

- ۱ عبارت PHP سرنام کدام عبارت است؟
 - الف) Performed Home Page
 - ب) Personal Home Page
 - ج) PHP: Hypertext Preprocessor
 - د) Personal Hypertext Page
- ۲ کدام گزینه از ویژگی های زبان PHP نمی باشد؟
 - الف) قابل حمل بودن
 - ب) منبع باز (open source) بودن
 - ج) رایج بودن
 - د) کاربر پسند بودن
- ۳ در پروتکل HTTP کدام سمت شروع کننده ارتباط است؟
 - الف) سرویس دهنده
 - ب) سرویس گیرنده
 - ج) سرویس دهنده وب
 - د) سیستم عامل
- ۴ کدام یک از موارد زیر برای ساخت نرم افزارهای کاربردی به زبان PHP ضروری نمی باشد؟
 - الف) سرویس دهنده وب
 - ب) مفسر PHP
 - ج) سرویس دهنده داده
 - د) مرورگر وب

- ۵ کدام برنامه از قسمت‌های نصب شده هنگام نصب Wamp Server نمی‌باشد؟
 الف) SQL Server (ب) MySQL (ج) PHP (د) Apache
- ۶ کدام گزینه، تعریف کامل‌تری از برنامه Apache است؟
 الف) نرم‌افزار برنامه‌نویسی PHP (ب) نرم‌افزار مدیریت فایل
 ج) نرم‌افزار سرویس‌دهنده وب (د) نرم‌افزار پیکربندی سرور
- ۷ کدام رنگ از آیکون Wamp Server، نشان‌دهنده وجود مشکل در برخی از برنامه‌های آن است؟
 الف) سبز (ب) آبی (ج) قرمز (د) نارنجی
- ۸ برنامه‌های سرویس‌دهنده وب، به‌طورپیش‌فرض از چه شماره پورتی استفاده می‌کنند؟
 الف) ۲۱ (ب) ۲۸ (ج) ۸۰ (د) ۸۱
- ۹ کدام یک از اطلاعات زیر توسط تابع phpinfo، نمایش داده نمی‌شود؟
 الف) مشخصات و تنظیمات PHP (ب) لیست برنامه‌های نصب شده
 ج) مشخصات و تنظیمات Apache (د) مشخصات سیستم عامل
- ۱۰ برای اتصال به سرویس‌دهنده وب محلی از کدام آدرس استفاده نمی‌شود؟
 الف) 127.0.0.0 (ب) localhost (ج) 127.0.0.1 (د) 127.0.0.2
- ۱۱ فهرست با نام www در پوشه نصب Wamp Server، چه کاربردی دارد؟
 الف) نگه‌داری برنامه‌های PHP (ب) نگه‌داری صفحات وب
 ج) نگه‌داری فایل‌های تصاویر سایت (د) همه موارد
- ۱۲ قطعه کدهای PHP در داخل کدام علامت‌ها نوشته می‌شوند؟
 الف) <? ?> (ب) <? php ?> (ج) <! —> (د) /* */

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
جواب	ج	د	ب	د	الف	ج	د	ج	ب	الف	د	ب

دستور کار عملی

- با استفاده از اینترنت در مورد امکانات و آخرین نسخه بسته‌های نصب PHP که در این فصل ذکر شده‌اند، تحقیق کنید.
- بسته نصبی Wamp Server را روی سیستم نصب کنید.
- یک صفحه PHP ایجاد کرده و دستوراتی در آن بنویسید که پیغام Hello World را به صورت تیترا درجه ۱ نمایش دهد:

Hello World



فصل دوم

اصول برنامه‌نویسی به زبان PHP



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- متغیرها را تعریف کرده و در برنامه استفاده کند.
- ثابت‌ها را تعریف کرده و استفاده کند.
- انواع داده‌ها در PHP را شرح دهد.
- انواع عملگرها و روش کار آنها را توضیح دهد.
- عملگرها را در برنامه به کار ببرد.

هر زبان برنامه‌نویسی از جمله PHP دارای مجموعه‌ای از نشانه‌ها، کلمات، اصول، قواعد و امکانات است که برای نوشتن برنامه‌ها باید با آنها آشنا باشید. خوشبختانه زبان PHP و اکثر زبان‌های برنامه‌نویسی از نظر اصول و قواعد برنامه‌نویسی مشابه هم هستند و در صورت آشنایی با یکی از آنها، با مطالعه‌ای اندک می‌توانید با زبان‌های دیگر هم برنامه‌نویسی کنید.

در این فصل و فصل‌های بعدی با اصول اولیه برنامه‌نویسی با زبان PHP برای شروع برنامه‌نویسی با زبان PHP مثل نحوه تعریف و نامگذاری متغیرها، استفاده از عملگرها، ایجاد بلاک‌های برنامه‌نویسی، ساختارهای شرطی و حلقه‌های تکرار، کار با آرایه‌ها و توابع، دریافت داده‌ها آشنا می‌شوید.

۱-۲- نحوه نوشتن دستورات

در فصل قبل استفاده از چند مثال برنامه‌نویسی نشان داده شد که زبان PHP از HTML جدا نیست و برنامه‌های PHP را می‌توان همان صفحات HTML دانست با این تفاوت که دارای پسوند php بوده و در آنها دستوراتی به زبان PHP نوشته شده است. یک قطعه کد (یا بلوک برنامه‌نویسی) PHP، همیشه با `<?php` شروع شده و با `>` پایان می‌یابد و می‌تواند در هر جای صفحه قرار گیرد.

```
<?php
```

```
PHP دستورات به زبان
```

```
?>
```

برای نمایش خروجی در PHP از دو دستور `echo` و `print` استفاده می‌شود که خروجی می‌تواند داخل پرانتز () یا بدون پرانتز نوشته شود. در مثال زیر از دستور `echo` برای نمایش متن `Hello World` و `Hello Friends!` به صورت درشت، استفاده شده است.

```
<html>
```

```
<body>
```

```
<h1><?php echo("Hello World!"); ?></h1>
```

```
<h1><?php echo "Hello Friends!"; ?></h1>
```

```
</body>
```

```
</html>
```

نتیجه حاصل از اجرای یک برنامه PHP، شامل تگ‌های HTML صفحه به علاوه خروجی دستورات داخل بلاک‌های PHP است. بنابراین اسکریپت فوق، خروجی HTML زیر را تولید می‌کند:

```
<html>
<body>
<h1>Hello World!</h1>
<h1>Hello Friends!</h1>
</body>
</html>
```

و زمانی که مرورگر این صفحه را درخواست کند، قطعه HTML فوق را دریافت کرده و صفحه زیر را به کاربر نمایش خواهد داد:

Hello World!

Hello Friends!

یادآوری: فایل PHP باید حتماً دارای پسوند php باشد. اگر پسوند آن html باشد، دستورات آن اجرا نخواهد شد.

۲-۲- نوشتن توضیحات

در PHP از // برای ایجاد یک خط توضیح و از /* و */ برای ساختن چند سطر توضیحات استفاده می‌شود:

```
<html>
<body>
<?php
//This is a comment
/*
This is a
comment block
*/
?>
</body>
</html>
```

۲-۳- استفاده از متغیرها

هر متغیر مکانی از حافظه است که برای ذخیره کردن مقادیری مختلف مثل رشته‌ها و اعداد استفاده می‌شود. در زبان PHP، نام متغیرها با علامت \$ شروع می‌شود و در ادامه می‌تواند شامل یک حرف لاتین (z-a)، یا خط زیر (_) و سپس شامل حروف، ارقام یا خط زیر (_) باشد. مثل:

\$num , \$_name , \$a123 , \$first_name

همچنین در زبان PHP با مقداردهی یک متغیر، متغیر ایجاد شده و به آن حافظه داده می‌شود:

```
$var_name = value;
```

مثال: در دستورات زیر یک متغیر برای نگهداری رشته و یک متغیر برای نگهداری اعداد تعریف شده است:

```
<?php  
$txt="Hello World!";  
$x=16;  
?>
```

هر دستور PHP با علامت سمیکالن (;) خاتمه پیدا می‌کند. سمیکالن جداکننده است و برای تشخیص دستورات از یکدیگر استفاده می‌شود.

توجه داشته باشید که زبان PHP نسبت به کوچک یا بزرگ بودن حروف حساس است و هر دستور یا متغیر باید دقیقاً با حروف تعیین شده نوشته شود. به عنوان مثال اگر متغیری با نام \$a تعریف شود در قسمت دیگر کد، نمی‌توان آن متغیر را با نام \$A استفاده کرد.

PHP یک زبان برنامه‌نویسی با نوع داده ضعیف است یعنی در زمان ایجاد متغیر، لازم نیست نوع داده آنها به صورت صریح تعریف شود در حالی که در زبان‌های برنامه‌نویسی با تعیین نوع داده قوی باید نوع داده متغیر به صورت صریح مشخص شود.

در PHP، در صورت مقداردهی یک متغیر، آن متغیر ایجاد شده، به آن حافظه تخصیص داده شده و با توجه به مقدار تعیین شده نوع داده آن تعیین می‌گردد و نیز بعداً در صورت انتساب مقدار جدید با نوع داده دیگر به آن، نوع داده متغیر عوض می‌شود. با این حال متغیرها در زبان PHP می‌توانند دارای انواع زیر باشند:

جدول ۱-۲

نوع داده	کاربرد	مثال
integer	اعداد صحیح	123
double	اعداد اعشاری	34.60
string	رشته‌ها	"hello"
boolean	منطقی (درست تا نادرست)	true یا false

گفته شد برای تعریف متغیر به کمک زبان PHP نیازی به تعیین نوع داده‌ای که توسط متغیر نگهداری می‌شود، نیست. همچنین در PHP نوع داده متغیر ثابت نیست و ممکن است با نسبت دادن مقادیر جدید در دستورات بعدی برنامه، نوع داده آن عوض شود. به دستورات زیر توجه کنید:

```
<?php  
$a = 100;  
$a = 16.75 ;  
$a = "hello" ;  
$a = true;  
?>
```

در قطعه کد قبل مقادیر ثابت زیر به متغیر \$a نسبت داده شده‌اند:

100: ثابت عددی صحیح

16.75: ثابت اعشاری

"hello": ثابت رشته‌ای (رشته‌های ثابت بین علامت‌های " یا " نوشته می‌شوند)

true یا false: ثابت منطقی

در دستور اول، متغیر \$a با عدد صحیح ۱۰۰ مقداردهی شده پس نوع داده آن integer تعیین شده است. سپس در دستور دوم، عدد اعشاری ۱۶٫۷۵ به آن نسبت داده شده در نتیجه نوع داده آن به double تغییر کرده است.

در دستور سوم، متغیر \$a با رشته "hello" مقداردهی شده و نوع داده آن رشته‌ای خواهد بود و در ادامه در دستور چهارم، مقدار منطقی true به آن نسبت داده شده و نوع داده آن به boolean تغییر کرده است.

۴-۲- از بین بردن متغیرها

در زبان PHP در هر جای برنامه امکان از بین بردن متغیر و آزاد کردن حافظه آن وجود دارد. برای از بین بردن یک متغیر از تابع unset به صورت زیر استفاده می‌شود:

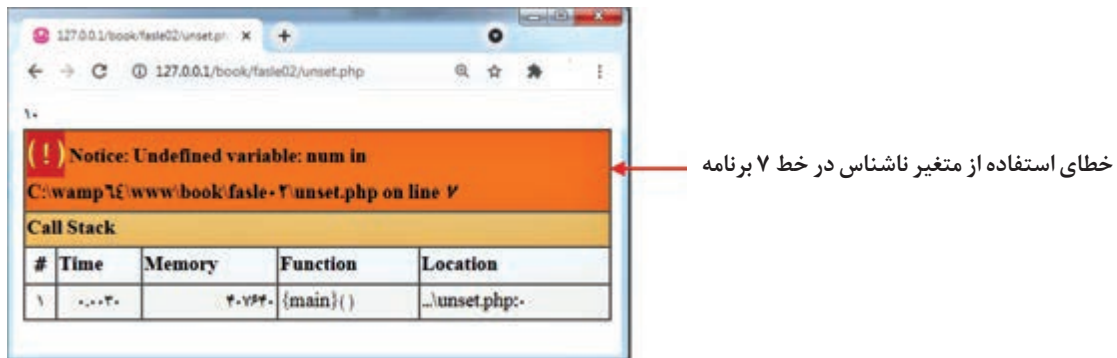
unset (نام متغیر);

در صفحه unset.php، متغیر از حافظه پاک شده است:

صفحه unset.php

```
<html>
<body>
<?php
$num = 10;
echo $num;
unset($num); // از بین بردن متغیر
echo $num;
?>
</body>
</html>
```

در کد فوق، ابتدا متغیر \$num ایجاد می‌شود، سپس مقدار آن نمایش داده می‌شود. در ادامه متغیر از حافظه حذف شده و دستور نمایش بعدی، چون متغیر \$num وجود ندارد، مطابق شکل ۲-۱، پیغام خطا نمایش داده می‌شود.



شکل ۲-۱- نمایش پیغام خطای متغیر ناشناس

در زمان بازکردن صفحات در صورتی که برنامه دارای خطای گرامری باشد، پیغام خطا به شکل فوق نمایش داده می‌شود که علت خطا، نام صفحه و شماره خط را اعلام می‌کند. با دقت به این پیغام‌های خطا، به مرور زمان مهارت لازم برای برطرف کردن هر خطایی از برنامه را بدست می‌آورید.

۲-۵- متغیرهای رشته‌ای

متغیرهای رشته‌ای برای ذخیره کردن مقادیری که شامل کاراکترها باشند استفاده می‌شوند. در زبان PHP رشته‌های ثابت بین علامت‌های نقل قول تکی (") یا دوتایی (") قرار می‌گیرند:

```
<?php
$name = 'ali' ;
$name = "ali" ;
?>
```

بیشتر اوقات مهم نیست از کدام علامت برای تعیین رشته‌های ثابت استفاده شود ولی اگر از نقل قول دوتایی (") استفاده گردد، PHP به صورت هوشمندانه متغیرهای داخل رشته را شناسایی کرده و مقدار آنها را در نظر می‌گیرد.

مثال ۱: در قطعه کد زیر، مقدار متغیر \$m چاپ شده است.

کد برنامه	خروجی
<pre><?php \$m = 30; echo " sum = \$m "; ?></pre>	sum = 30

مثال ۲: قطعه برنامه زیر متغیر رشته‌ای \$txt را نشان می‌دهد که حاوی رشته "Hello World" است.

خروجی	کد برنامه
Hello World	<pre><?php \$txt="Hello World"; echo \$txt; ?></pre>

تنها یک عملگر رشته‌ای در PHP وجود دارد و آن هم عملگر الحاق رشته‌ها یعنی نقطه (.) است که دو رشته را به هم می‌چسباند.

برای الحاق دو رشته با یکدیگر از عملگر الحاق به صورت زیر استفاده کنید:

خروجی	کد
Hello World! What a nice day!	<pre><?php \$txt1="Hello World!"; \$txt2="What a nice day!"; echo \$txt1. "
". \$txt2; ?></pre>

در کد فوق، دو بار از عملگر الحاق استفاده شده و سه رشته به هم چسبیده‌اند و برای اینکه هر جمله در یک سطر جداگانه نوشته شود بین آنها تگ
 قرار داده شده است.

۶-۲- ثابت‌ها

شناسه‌های ثابت‌ها در PHP مشابه متغیرها، خانه‌هایی از حافظه برای نگهداری داده‌ها هستند ولی روش تعریف و مقداردهی و کاربرد آنها متفاوت است.

یک ثابت، در اصل نام یا شناسهٔ مربوط یک مقدار ساده است که معمولاً در ابتدای برنامه تعیین شده و در جاهای مختلف برنامه به جای نوشتن مقدار ثابت، این شناسه استفاده می‌شود. نام یک شناسه ثابت بدون علامت \$ نوشته شده و مقدار آن در طول اجرای برنامه، ثابت بوده و تغییر نمی‌کند. برای تعریف ثابت به صورت زیر عمل می‌شود:

define ("نام ثابت", مقدار);

به عنوان مثال برای تعریف شناسهٔ ثابت MAX با مقدار ۱۰۰ دستور زیر را استفاده می‌شود:

```
<?php
define("MAX",100);
?>
```

بعد از اینکه یک ثابت تعریف شد، در هر جای برنامه می توان از آن استفاده کرد. به عنوان مثال در دستورات زیر ابتدا شناسه ثابت MAX تعریف شده و سپس متغیر \$sum مساوی MAX به علاوه ۵ قرار داده شده است:

```
<?php
define("MAX",100 );
$sum = MAX + 5 ;
?>
```

مزیت استفاده از شناسه های ثابت این است که با نسبت دادن مقدار جدید به شناسه ثابت، در همه جای برنامه، این مقدار جدید مورد استفاده قرار خواهد گرفت. به عنوان مثال فرض کنید در یک سایت فایل های تصویری در پوشه خاصی قرار گرفته اند و در جاهای مختلف صفحه برای آدرس دهی تصاویر، نام این پوشه استفاده می شود. در این صورت می توان در ابتدا به صورت زیر یک ثابت برای نگهداری نام پوشه تعریف کرده و در جاهای مختلف صفحه این ثابت را استفاده کرد تا بعداً در صورت تغییر نام پوشه فقط مقدار این ثابت را تغییر داده و نیاز به انجام تغییرات زیاد در جاهای مختلف برنامه نباشد.

```
<?php
define("IMAGE_DIR" , "images");
?>
```

۷-۲- استفاده از عملگرها در PHP

عملگرها بین متغیرها یا ثابتها قرار گرفته و روی آنها عمل مشخصی انجام می دهند. به عنوان مثال برای جمع یا تفریق دو عدد از عملگر + یا - استفاده می شود. در هر زبان برنامه نویسی از جمله PHP انواع مختلف عملگرها وجود دارند. در ادامه به معرفی انواع عملگرها در زبان PHP می پردازیم:

۱-۷-۲- عملگرهای محاسباتی

عملگرهای محاسباتی برای انجام اعمال محاسباتی استفاده می شوند، این عملگرها در جدول ۲-۲ نشان داده شده اند:

جدول ۲-۲

نتیجه	مثال	شرح	عملگر
y=4	x=2 ; y=x+2;	جمع	+
y=3	x=2 ; y=5-x;	تفریق	-
y=20	x=4 ; y=x*5;	ضرب	*
y=3 z=2.5	y=15/5; z=5.0/2;	تقسیم	/
x=1 y=2 z=0	x=5%2; y=10%8; z=10%2;	باقیمانده	%

عملگر	شرح	مثال	نتیجه
++	افزایش	x=5 ; x++; y=5; ++y;	x=6 y=6
--	کاهش	x=5; x -- ; y=5 ; -- y;	x=4 y=4

عملگرهای ++ و -- می‌توانند قبل یا بعد از یک متغیر قرار گیرند. اگر این عملگرها قبل از متغیر قرار گیرند، دارای بالاترین اولویت بوده و اول اجرا شده و مقدار جدید متغیر در عبارت استفاده می‌شود. ولی اگر بعد از متغیر قرار داشته باشند، دارای کمترین اولویت هستند، به این معنی که بعد از محاسبه کامل عبارت، مقدار آن تغییر می‌کند.

مثال ۱: در کد زیر، ابتدا \$x در عبارت محاسباتی استفاده شده و در پایان به آن یک واحد اضافه می‌شود:

صفحه plus1.php	خروجی
<pre><html> <body> <?php \$x = 5; \$y = \$x++ * 2 ; echo "X: \$x , Y: \$y "; ?></body></html></pre>	X: 6 , Y: 10

مثال ۲: در کد زیر، ابتدا یک واحد به \$x اضافه شده و سپس در عبارت محاسباتی استفاده می‌شود:

صفحه plus2.php	خروجی
<pre><html> <body> <?php \$x = 5; \$y = ++\$x * 2 ; echo "X: \$x , Y: \$y "; ?> </body> </html></pre>	X: 6 , Y: 12

۲-۷-۲ عملگرهای تخصیص یا انتساب

عملگرهای انتساب برای نسبت دادن یک مقدار یا نتیجه یک عبارت به یک متغیر استفاده می‌شوند. بعضی از این عملگرها عمل مشخصی را روی یک متغیر انجام داده و حاصل را در همان متغیر ذخیره می‌کنند:

جدول ۲-۳

عبارت معادل	مثال	عملگر
$x=y$	$x=y$	=
$x=x+y$	$x+=y$	+=
$x=x-y$	$x-=y$	-=
$x=x*y$	$x*=y$	*=
$x=x/y$	$x/=y$	/=
$x=x.y$	$x.=y$.=
$x=x'/y$	$x'/=y$	'/=

مثال ۳: برنامه‌ای بنویسید که با داشتن قیمت کتاب، تعداد کتاب و درصد تخفیف، مبلغ، جمع مبلغ، مبلغ تخفیف و مبلغ قبل پرداخت را نمایش دهد:

برنامه pay.php	خروجی
<pre> <html> <body> <?php \$price = 23000; // قیمت کتاب \$num = 20 ; // تعداد \$percent = 10 ; // درصد تخفیف \$total = \$num * \$price ; // جمع مبلغ \$discount=\$total * \$percent / 100;// کل تخفیف \$payable = \$total - \$discount ; // قابل پرداخت echo "TOTAL : \$total
 "; echo "DISCOUNT: \$discount
 "; echo "PAYABLE : \$payable"; ?> </body> </html> </pre>	<pre> TOTAL : 460000 DISCOUNT: 46000 PAYABLE : 414000 </pre>

توجه داشته باشید که عملگر % فقط برای متغیرها و داده‌های صحیح کاربرد داشته و عملگر / اگر بین دو مقدار صحیح قرارگیرد، خارج قسمت تقسیم را محاسبه می‌کند و اگر یک طرف آن اعشاری باشد، تقسیم اعشاری انجام خواهد داد.

با تغییراتی در صفحه فوق می‌توان خروجی را به صورت جدول زیر نمایش داد:

TOTAL	DISCOUNT	PAYABLE
460000	46000	414000

برای این منظور تگ‌های HTML را برای نمایش جدول و ساخت سطر و سلول‌های آن چاپ کرده و نتایج برنامه را در داخل سلول‌های جدول چاپ کنید:

```

برنامه pay2.php
<html>
<body>
<?php
$price = 23000; // قیمت کتاب
$num = 20 ; // تعداد
$percent = 10 ; // درصد تخفیف
$total = $num * $price ; // جمع مبلغ
$discount=$total * $percent / 100; // کل تخفیف
$payable = $total - $discount ; // قابل پرداخت
?>
<table border="1" >
<tr>
<td> TOTAL </td>
<td>DISCOUNT</td>
<td>PAYABLE</td>
</tr>
<tr>
<td><?php echo $total ?></td>
<td><?php echo $discount ?></td>
<td><?php echo $payable ?></td>
</tr>
</table>
</body>
</html>

```

در صفحه فوق برای طراحی کل صفحه و همچنین جدول از تگ‌های HTML استفاده شده و فقط در داخل هر کدام از سلول‌های جدول با استفاده از PHP مقدار متغیرها نمایش داده شده است. همان‌گونه که مشاهده می‌کنید، چند بلاک برنامه‌نویسی PHP وجود دارد که بعد از اجرای هر بلاک، خروجی آن در بین تگ‌های HTML قرار می‌گیرد.

۲-۷-۳- عملگرهای مقایسه‌ای

عملگرهای مقایسه‌ای یا رابطه‌ای برای مقایسه دو مقدار به کار می‌روند. زمانی که لازم باشد تا مقدار یک متغیر با یک متغیر دیگر یا یک مقدار ثابت مقایسه شود، از این عملگر استفاده می‌شود. اگر نتیجه مقایسه درست باشد، نتیجه true و در غیر این صورت نتیجه false به دست خواهد آمد. در جدول ۲-۴ عملگرهای مقایسه‌ای را مشاهده می‌کنید:

جدول ۲-۴

عملگر	کاربرد	مثال
==	مساوی بودن مقدار	نتیجه $5==8$ مساوی false است
===	مساوی بودن مقدار و نوع داده	نتیجه $5===5$ مساوی false است
!=	نامساوی	نتیجه $5!=8$ مساوی true است
!==	نامساوی مقدار یا نوع داده	نتیجه $5!==5$ مساوی true است
<>	نامساوی	نتیجه $5<8$ مساوی true است
>	بزرگ‌تر	نتیجه $5>8$ مساوی false است
<	کوچک‌تر	نتیجه $5<8$ مساوی true است
>=	بزرگ‌تر یا مساوی	نتیجه $5>=8$ مساوی false است
<=	کوچک‌تر یا مساوی	نتیجه $5<=8$ مساوی true است

عملگرهای == و != هم مقدار و هم نوع داده دو طرف را مورد بررسی قرار می‌دهند. در صورتی با هم مساوی هستند که هم مقدار و هم نوع داده دو طرف مساوی باشد و در صورتی نامساوی هستند که مقدار یا نوع داده دو طرف یکسان نباشد.

۲-۷-۴- عملگرهای منطقی

عملگرهای منطقی زمانی استفاده می‌شوند که لازم شود دو مقایسه را با هم ترکیب کنیم، به‌عنوان مثال بررسی اینکه دو شرط هم‌زمان با هم درست است یا بررسی اینکه حداقل یکی از دو شرط داده شده درست است. در جدول ۲-۵ عملگرهای منطقی را مشاهده می‌کنید:

جدول ۲-۵

مثال	کاربرد	عملگر
x=6; y=3; (x < 10 && y > 1) → true	و	and
x=6; y=3; (x==5 y==5) → false	یا	or
x=6; y=3; (x < 10 && y > 1) → true	و	&&
x=6; y=3; (x==5 y==5) → false	یا	
x=6; y=3; !(x==y) → true	نقیض	!

همه عملگرهای منطقی به جز نقیض (!)، بین دو گزاره قرار می‌گیرند. که هر کدام از این گزاره‌ها می‌توانند درست (true) یا نادرست (false) باشند.

عملگرهای and و && معادل هم هستند و اگر هر دو گزاره درست باشد، نتیجه درست و در غیر این صورت نتیجه نادرست (false) را تولید می‌کنند.

عملگرهای or و || نیز معادل هم بوده و اگر هر دو گزاره نادرست باشد، نتیجه نادرست (false) و در غیر این صورت نتیجه درست (true) را تولید می‌کنند.

عملگر ! قبل از یک گزاره قرار گرفته و اگر گزاره درست بود، نتیجه نادرست (false) و اگر گزاره نادرست بود، نتیجه درست (true) را تولید می‌کند.

در جداول زیر عملکرد هر کدام از عملگرهای منطقی در حالت‌های مختلف نشان داده شده است:

A	B	A and B A && B
false	false	false
false	true	false
true	false	false
true	true	true

A	B	A or B A B
false	false	False
false	true	True
true	false	True
true	true	True

A	! A
False	true
True	false

یکی از کاربردهای عملگرهای منطقی، بررسی محدوده مقدار یک متغیر است. به عنوان مثال جهت بررسی اینکه که متغیر \$x\$ بین دو مقدار ۱۰ و ۲۰ باشد از دستور مقابل استفاده می‌کنیم ($x \geq 10 \ \&\& \ x \leq 20$) عملگرهای مقایسه‌ای و منطقی در ساختارهای شرطی کاربرد دارند که در فصل بعد به آن پرداخته خواهد شد و در آینده، مثال‌های بیشتری از کاربرد این عملگرها خواهید دید.

۵-۷-۲- اولویت عملگرها

عملگرها دارای اولویت‌های مشخص بوده و هنگام اجرای برنامه، به ترتیب اولویت اجرا می‌شوند. بنابراین برای استفاده صحیح از عملگرها، باید اولویت آنها را در نظر داشته باشید. همچنین ترتیب اجرای آنها در یک عبارت می‌تواند از چپ به راست یا از راست به چپ باشد. در جدول ۲-۶ اولویت عملگرها نشان داده شده است: (به عملگرهای !، ++ و -- که تنها روی یک گزاره عمل می‌کنند عملگرهای یکانی می‌گویند)

جدول ۲-۶

اولویت	عملگرها	جهت اجرا
عملگرهای یکانی	! ++ -	راست به چپ
ضرب و تقسیم	% / *	چپ به راست
جمع و تفریق	+	چپ به راست
مقایسه‌ای	< <= > >=	چپ به راست
تساوی	== != === !==	چپ به راست
And منطقی	&& and	چپ به راست
Or منطقی	or	چپ به راست
انتساب	= += -= *= /= %=	راست به چپ

جهت اجرای عملگرها به این معنی است که اگر در یک عبارت چند عملگر دارای اولویت یکسان باشند، ترتیب اجرای آنها از چپ به راست یا از راست به چپ می‌باشد. برای همه عملگرها با پرانتزگذاری عبارت، می‌توان ترتیب اجرای آنها را به دلخواه تغییر داد. به عنوان مثال، در عبارت زیر علی‌رغم اینکه * اولویت بالاتری نسبت به + دارد، ابتدا عبارت داخل پرانتز محاسبه شده و حاصل آن در ۲ ضرب می‌شود.

$$y = (x + 5) * 2 ;$$

در این فصل با مفاهیم مقدماتی برنامه‌نویسی PHP مثل تعریف متغیرها، ثابت‌ها و عبارت‌های مختلف آشنا شدید. در فصل بعد با مفاهیم و ساختارهای پیشرفته‌تر زبان PHP آشنا خواهید شد.

خودآزمایی

- ۱ در زبان PHP، تعریف متغیر چگونه است؟
- ۲ عبارت «زبان PHP دارای نوع داده ضعیف است» به چه معنی است؟ توضیح دهید.
- ۳ انواع داده‌های اصلی در PHP را نام برده و با ذکر مثال متغیرهایی از هر نوع داده ایجاد کنید؟
- ۴ عملگرهای محاسباتی، مقایسه‌ای و منطقی در PHP و کاربرد آنها را بنویسید؟
- ۵ منظور از اولویت عملگرها چیست؟ توضیح دهید.
- ۶ با در نظر گرفتن اولویت عملگرها، نتیجه عبارت زیر را به دست آورید.

`$x= 6;`

`$y = ++$x * ($x / 2) * ($x + 3)`

۷ عملگر نقطه در PHP چه کاربردی دارد؟ با ذکر مثال توضیح دهید.

۸ شناسه‌های ثابت چه کاربردی دارند؟ روش تعریف آنها چگونه است؟

پرسش‌های چهار گزینه‌ای

- ۱ کدام یک از اسامی زیر برای متغیرها نادرست است؟
الف) `$for` (ب) `$num` (ج) `sum` (د) `$_sum`
- ۲ نوع داده `double` برای نگه‌داری چه مقادیری استفاده می‌شود؟
الف) اعداد بزرگ (ب) اعداد اعشاری (ج) اعداد صحیح (د) همه نوع اعداد
- ۳ در PHP توضیحات چند سطر بین کدام علامت‌ها قرار می‌گیرند؟
الف) `//` (ب) `<!-->` (ج) `/* */` (د) همه موارد
- ۴ کدام یک از عملگرهای محاسباتی فقط برای اعداد صحیح کاربرد دارد؟
الف) `/` (ب) `%` (ج) `.*` (د) `+`
- ۵ عملگر نقطه (.) در PHP چه کاربردی دارد؟
الف) جداکردن دستورات (ب) ضرب دو عدد (ج) الحاق دو رشته (د) کاربردی ندارد
- ۶ عملگرهای منطقی چه کاربردی دارند؟
الف) بررسی درست یا نادرست بودن (ب) ترکیب دو شرط (ج) نقیض کردن یک شرط (د) موارد ب و ج
- ۷ در PHP برای `and` کردن دو گزاره از کدام عملگر استفاده می‌شود؟
الف) `&&` (ب) `and` (ج) `||` (د) موارد الف و ب
- ۸ خروجی دستور زیر کدام است؟
`$a=4 ;`
`echo "$a + 5 ";`

الف) 9 (ب) 4+5 (ج) \$a+5 (د) نمایش خطا

۹ خروجی عبارت زیر چیست؟

```
$a=4;
```

```
$b=8;
```

```
echo !($a>5 && $b <10)
```

الف) true ب) false ج) 1 د) 0

۱۰ کدام عملگر زیر دارای بالاترین اولویت نسبت به دیگران است؟

الف) * ب) && ج) ++ قبل از متغیر د) ==

۱۱ حاصل عبارت زیر کدام است؟

```
$x= 5;
```

```
$y = ++$x + 2 * $x ;
```

الف) 15 ب) 18 ج) 21 د) 48

۱۲ کدام گزینه در مورد انواع داده‌ها در PHP صحیح می‌باشد؟

- الف) نوع داده متغیر، بستگی به داده ذخیره شده در آن دارد.
- ب) هنگام تعریف متغیر، باید نوع داده آن به صورت صریح تعیین شود.
- ج) نوع داده یک متغیر در طول برنامه ثابت است.
- د) متغیرها می‌توانند بدون نوع داده باشند.

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
پاسخ	الف	ب	ج	ب	ج	ب	د	ب	الف	ج	ب	الف

دستور کار عملی

۱ صفحه‌ای طراحی کنید که با فرض قرار داشتن تعداد ثانیه‌ها در متغیر \$n بتواند تعداد ساعت، دقیقه و ثانیه را محاسبه کرده و در جدول زیر نمایش دهد. (راهنمایی: هر ساعت دارای ۳۶۰۰ ثانیه و هر دقیقه دارای ۶۰ ثانیه است و با فرض اینکه \$n=8000 باشد خروجی زیر را نمایش دهد)

$$8000 = 2 * 3600 + 13 * 60 + 20$$

hours	2
minutes	13
seconds	20

۲ ابتدا عبارت‌های زیر را روی کاغذ نوشته و نتایج آنها را به دست آورید. سپس صفحه‌ای طراحی کنید که حاصل عبارت‌های زیر را محاسبه کرده و در جدول نمایش دهد و نتایج روی کاغذ را با نتایج روی صفحه نمایش مقایسه کنید.

$x=5;$

$y=7;$

$A = ++x + 4 * x ;$

$B = x / (y - 4) ;$

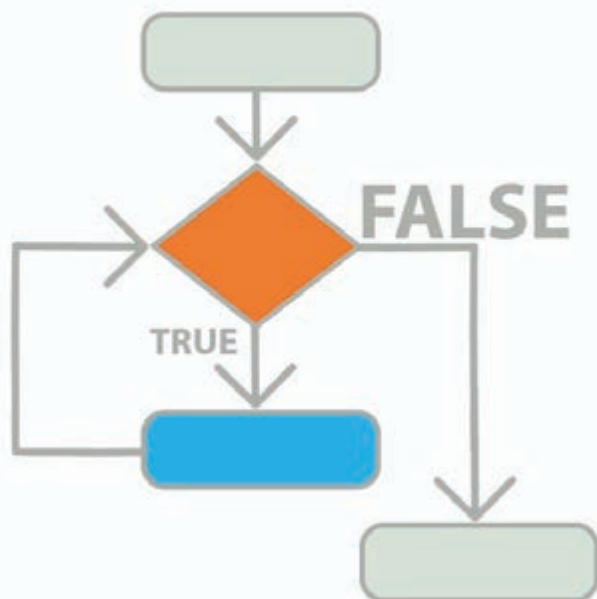
$C = (1 + x) \% (y++ - 4) ;$

x	Y	A	B	C
?	?	?	?	?



فصل سوم

ساختارهای شرطی و تکرار



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- کاربرد ساختارهای شرطی را توضیح دهد.
 - ساختارهای شرطی را در برنامه استفاده کند.
 - کاربرد ساختارهای مختلف تکرار را توضیح دهد.
 - ساختارهای مختلف تکرار را در برنامه استفاده کند.
 - عملکرد دستورات `break`، `continue` و `exit` را توضیح دهد.
 - دستورات `break`، `continue` و `exit` را در برنامه استفاده کند.
- در فصل قبل اصول اولیه برنامه‌نویسی به زبان PHP بیان شد، در این فصل به معرفی ساختارهای شرطی و تکرار می‌پردازیم. با کمک ساختارهای شرطی می‌توان برای اجرای یک بخش از برنامه یا نمایش یک قطعه از طراحی صفحه، شرط مورد نظر را تعیین کرد تا در صورت برقرار بودن شرط آن بخش، اجرا یا نمایش داده شود. همچنین با کمک حلقه‌های تکرار می‌توان بخشی از دستورات یا طراحی را چند بار تکرار نمود.

۱-۳- ساختارهای شرطی

گاهی لازم می‌شود دستوراتی از برنامه در شرایط خاصی اجرا شود. به عبارت دیگر در صورت برقرار بودن شرط مدنظر، یک مجموعه از دستورات و در غیر این صورت دستورات دیگری اجرا شود. در این مواقع از ساختارهای شرطی استفاده می‌شود.

از ساختارهای شرطی برای انجام عملیات مختلف با توجه به شرایط مختلف استفاده می‌شوند. در زبان PHP برای ایجاد ساختارهای شرطی از دستورات زیر استفاده می‌شود:

■ دستور `if`

■ دستور `if..else`

■ دستور `switch`

در ادامه به شرح و نحوه کارکرد هر کدام از این ساختارها می‌پردازیم:

۱-۱-۳ ساختار `if`

ساختار `if` برای بررسی یک شرط و اجرای دستورات در صورت برقرار بودن آن استفاده می‌شود. شکل کلی ساختار `if` به صورت زیر است:

(شرط) `if`

{

دستوراتی که در صورت درست بودن شرط اجرا می‌شوند

}

مثال ۱: ماکزیمم (بزرگترین) عدد بین دو عدد `$a` و `$b` را نمایش دهید.

برنامه max1.php

```
<?php
$max = $a ;
if ( $max < $b )
{
    $max = $b;
}
echo $max;
?>
```

در مثال فوق ابتدا متغیر \$max مساوی عدد \$a قرار می‌گیرد، سپس بررسی می‌شود اگر \$b از \$max بزرگ‌تر بود، \$b به \$max نسبت داده شده و در پایان \$max نمایش داده می‌شود. در مواقعی که فقط یک دستور در بدنه ساختار if داشته باشیم، نیازی به گذاشتن {} نیست. برنامه فوق را می‌توان مجدداً به صورت زیر نوشت:

برنامه max2.php

```
<?php
$max = $a ;
if ( $max < $b )
    $max = $b;
echo $max;
?>
```

۲-۱-۳ ساختار if..else

ساختار if..else در مواقعی استفاده می‌شود که قصد داریم یک شرط بررسی شود، اگر درست بود یک مجموعه دستور و اگر نادرست بود مجموعه دیگری از دستورات اجرا شوند. شکل کلی ساختار if..else به صورت زیر است:

```
if ( شرط )
{
    دستوراتی که در صورت درست بودن شرط اجرا می‌شوند
} else {
    دستوراتی که در صورت نادرست بودن شرط اجرا می‌شوند
}
```

مثال ۲: در قطعه برنامه زیر ماکزیمم دو عدد محاسبه می‌شود:

برنامه max3.php
<pre><?php if (\$a > \$b) { \$max = \$a; }else{ \$max = \$b ; } echo \$max;</pre>

در این مثال هم چون در بخش‌های if و else فقط یک دستور داریم نیازی به گذاشتن {} نیست. مثال ۳: در برنامه زیر ساعت جاری از سیستم گرفته شده و بررسی می‌شود. اگر از ۲۰ کوچک‌تر بود پیغام «روز بخیر» و در غیر این صورت پیغام «شب بخیر» را نمایش می‌دهد.

برنامه show_time.php	خروجی برنامه
<pre><?php \$t = date("H"); if (\$t < "20") { echo "روز بخیر"; } else { echo "شب بخیر"; } ?></pre>	روز بخیر

در برنامه فوق از تابع date() برای دریافت ساعت جاری سیستم استفاده شده است. با کمک این تابع می‌توان بخش‌های مختلف تاریخ و زمان شامل سال، ماه، روز، ساعت، دقیقه و ثانیه را از سیستم دریافت کرد. در فصل ۵ با این تابع آشنا خواهید شد.

مثال ۴: در کد زیر بررسی می‌شود، اگر جمع مبلغ یک سفارش خرید از یکصد هزار تومان بیشتر باشد، مبلغ ۱۰ هزار تومان در غیر این صورت مبلغ ۵ هزار تومان تخفیف در نظر گرفته می‌شود:

برنامه discount.php	خروجی
<pre><?php \$total=14000; if (\$total < 100000) \$disc = 5000; else \$disc = 10000; echo "Discount: \$disc "; ?></pre>	Discount: 5000

مثال ۵: در برنامه زیر متغیرهای \$user و \$pass که حاوی نام کاربری و کلمه عبور کاربر هستند با عبارت های admin و ۱۲۳ مقایسه شده و در صورتی که با آنها مساوی بود پیغام خوش آمدگویی با رنگ سبز و در غیر این صورت پیغام خطای با رنگ قرمز نشان می دهد:

برنامه checkuser.php	خروجی برنامه
<pre><!doctype html> <html> <body> <?php \$user="admin"; \$pass="123"; if(\$user=="admin" && \$pass=="123"){ ?> <div style="color:green"> کاربر عزیز خوش آمدید </div> <?php } else { ?> <div style="color:red"> نام کاربری یا کلمه عبور درست نیست </div> <?php } ?> </body> </html></pre>	کاربر عزیز خوش آمدید

در برنامه فوق، ساختار if..else به صورت سه قطعه کد PHP جداگانه نوشته شده که برای هر کدام از قطعات از علامت های <? php?> استفاده شده است.

۳-۱-۳ ساختار switch

ساختار switch با هدف مقایسه یک متغیر یا نتیجه یک عبارت با چند مقدار ثابت و اجرای دستورات معین در صورت مساوی بودن با هر کدام از آنها استفاده می‌شود. هر کدام از این ثابت‌ها می‌توانند از نوع عددی صحیح یا رشته‌ای باشند. شکل کلی ساختار switch به صورت زیر است:

switch (متغیر یا عبارت)

{

case مقدار ۱:

دستوراتی که در صورت مساوی بودن متغیر یا نتیجه عبارت با مقدار ۱، باید اجرا شوند

break;

case مقدار ۲ :

دستوراتی که در صورت مساوی بودن متغیر یا نتیجه عبارت با مقدار ۲، باید اجرا شوند

break;

...

default:

دستوراتی که در صورت مساوی نبودن متغیر یا نتیجه عبارت با هیچ کدام از مقادیر، باید اجرا شوند

}

با توجه به شکل کلی ساختار switch، متغیر یا عبارت مورد نظر در مقابل کلمه switch نوشته شده و هر کدام از مقادیر ثابت در مقابل یک کلمه case قرار می‌گیرند. هنگام اجرا متغیر یا نتیجه عبارت به ترتیب با مقادیر ثابت مقایسه می‌شوند، با هر کدام که مساوی بود، دستورات مقابل آن اجرا شده و با رسیدن به دستور break به اولین دستور بعد از ساختار switch پرش می‌کند.

مثال ۶: در قطعه برنامه زیر، رقم موجود در متغیر \$digit به صورت حروف نمایش داده می‌شود:

برنامه d2a.php	خروجی برنامه
<pre><?php \$digit= 2; switch(\$digit){ case 0: \$alpha = 'zero'; break; case 1: \$alpha = 'one'; break; case 2: \$alpha = 'two'; break; // برای رقم‌های ۳ تا ۸ به همین صورت نوشته می‌شود // case 9: \$alpha = 'nine'; break; default : \$alpha = 'error'; } echo \$alpha; ?></pre>	two

در برنامه فوق، رقم \$digit به ترتیب با کاراکترهای 0، 1، 2، ... و 9 مقایسه شده با هر کدام که مساوی بود متغیر \$alpha با نام آن رقم مقداردهی شده و سپس دستور break اجرا می‌شود. دستور break باعث می‌شود، کنترل اجرای برنامه به اولین دستور بعد از ساختار switch منتقل شود. بعد از هر قسمت باید حتماً دستور break نوشته شود، در غیر این صورت اجرا ادامه پیدا کرده و دستورات بخش‌های بعدی را هم به اشتباه اجرا خواهد کرد.

اگر \$digit با هیچ کدام از رقم‌ها مساوی نباشد، دستورات بخش default اجرا می‌شود، بنابراین بهتر است دستوراتی را در بخش default بنویسیم که می‌بایست در صورت برقرار نبودن هیچ کدام از شرایط اجرا شوند، در غیر این صورت و در صورتی که بخش default نوشته نشده باشد، هیچ یک از دستورات داخل برچسب‌های switch اجرا نمی‌شود و کنترل برنامه به اولین دستور بعد از ساختار switch منتقل می‌شود.

در مثال فوق اگر \$digit با هیچ کدام از رقم‌ها مساوی نباشد، مقدار error نمایش داده خواهد شد. **مثال ۷:** فرض کنید متغیر \$day حاوی نام اختصاری روز هفته است، در برنامه زیر نام کامل روز به صورت انگلیسی و فارسی به دست آمده و نمایش داده شده است:

برنامه dayname.php	خروجی برنامه
<pre> <?php \$day= 'Sun'; \$fday=""; \$eday=""; switch(\$day){ case 'Sat':\$eday='Saturday'; \$fday ='شنبه'; break; case 'Sun':\$eday ='Sunday'; \$fday = 'یکشنبه'; break; // برای دیگر روزها هم نوشته شود case 'Fri':\$eday ='Sunday'; \$fday = 'جمعه'; break; } echo "English: \$eday
". "Farsi:\$fday " ; ?> </pre>	<p>English: Sunday Farsi: یکشنبه</p>

در برنامه فوق، در صورت مساوی بودن متغیر \$day با هر کدام از ثابت‌های رشته‌ای دو دستور مقابل آن اجرا شده و سپس دستور break اجرا می‌شود تا کنترل به دستور echo منتقل شود. در دستور echo عبارت‌های ثابت و متغیرها و تگ‌های HTML در داخل علامت‌های " " قرار گرفته‌اند که عبارت‌های ثابت، تگ br و مقادیر متغیرها نمایش داده شده و خروجی مورد نظر ما به دست می‌آید.

۴-۱-۳- عملگر : و ؟

از علامت‌های ؟ و : برای ایجاد ساختارهای شرطی کوچک استفاده می‌شود. به عنوان مثال اگر برای مقداردهی یک متغیر یا نمایش یک خروجی نیاز به بررسی شرط خاصی باشد از این علامت‌ها استفاده می‌شود. شکل کلی استفاده از این عملگر به صورت زیر است:

؛ عبارت دوم : عبارت اول ؟ (شرط مورد نظر)
که ابتدا شرط بررسی شده و در صورت درست بودن شرط، عبارت اول و در غیر این صورت عبارت دوم استفاده خواهد شد.
مثال ۸: اگر متغیر \$gender جنسیت (۱- دختر ۲- پسر) را داشته باشد، برای نمایش نام جنسیت می‌توان دستورات زیر را استفاده کرد:

```
<?php
$gender = 1;
echo ( $gender == 1 ) ? "پسر" : "دختر" ;
?>
```

مثال ۹: در دستورات زیر با توجه به شماره ماه تعداد روزهای آن محاسبه شده است: (۶ ماه اول ۳۱ روزه و ۶ ماه دوم، ۳۰ روزه فرض شده‌اند)

```
<?php
$mon = 8 ;
$days = ( $mon <= 6 ) ? 31 : 30 ;
echo $days;
?>
```

نتیجه اجرای دستورات فوق، نمایش عدد ۳۰ می‌باشد.

۲-۳- ساختارهای تکرار

اغلب وقتی کدنویسی می‌کنیم، مایل هستیم قسمت‌هایی از برنامه بیشتر از یک بار اجرا شوند در این مواقع به جای اضافه کردن چندین کپی از دستورات به صورت پشت سرهم، از حلقه‌های تکرار برای انجام این کار استفاده می‌کنیم. حلقه‌های تکرار یک دسته از دستورات را به تعداد دفعات معین یا تا زمان برقرار بودن شرط مورد نظر، تکرار می‌کنند. در PHP چندین دستورالعمل برای تکرار داریم:

- **while**: برای تکرار یک مجموعه از دستورات تا زمان برقرار بودن یک شرط استفاده می‌شود.
 - **do...while**: برای اجرای مجموعه دستورات برای یکبار و سپس تکرار آن در صورت برقرار بودن یک شرط استفاده می‌شود.
 - **for**: معمولاً برای تکرار یک مجموعه دستورات به تعداد دفعات معین استفاده می‌شود.
 - **foreach**: برای تکرار یک مجموعه دستورات به ازای تمام عناصر آرایه استفاده می‌شود.
- در ادامه این ساختارهای تکرار را بررسی می‌کنیم:

۱-۲-۳- حلقه تکرار while

حلقه تکرار while تا زمان درست بودن یک شرط، دستورات را تکرار می‌کند و شکل کلی آن به صورت زیر است:

while (شرط)

```
{
مجموعه دستوراتی که باید تکرار شوند
}
```

مثال ۱: برنامه زیر حلقه تکراری را نشان می‌دهد که با $i=1$ شروع شده و تا زمانی که i کوچک‌تر یا مساوی ۵ باشد تکرار می‌کند. در هر بار تکرار به i یک واحد اضافه می‌شود.

کد برنامه number1.php	خروجی
<pre><html> <body> <?php \$i=1; while(\$i<=5) { echo "number ".\$i."
"; \$i++; } ?> </body> </html></pre>	<pre>number 1 number 2 number 3 number 4 number 5</pre>

مثال ۲: یک جدول برای چاپ سال‌های ۱۳۹۰ تا ۱۳۹۹ و آمار دانش‌آموزان هر سال طراحی و برنامه‌نویسی کنید. (آمار دانش‌آموزان برای همه سال‌ها را ۵۰۰ در نظر بگیرید)

حل: برای طراحی این صفحه از Dreamweaver استفاده می‌کنیم. ابتدا یک صفحه جدید به نام years1.php ایجاد کرده و داخل آن یک جدول با دو سطر و دو ستون درج می‌نماییم:

```
صفحه years.php
<html>
<body>
<table width="200" border="1">
<!-- سطر عنوان جدول که ثابت است -- ! >
<tr>
<td>Year</td><td>Students</td>
</tr>
<!-- سطر بدنه جدول که به‌ازای هر سال تکرار می‌شود -- ! >
<tr>
<td>1390</td><td>500</td>
</tr>
</table>
</body>
</html>
```

سطر دوم جدول باید برای هر کدام از سال‌های ۱۳۹۰ تا ۱۳۹۵ تکرار شود. بنابراین سطر دوم را داخل حلقه تکرار قرار می‌دهیم. از طرفی برای اینکه شماره سال و آمار دانش‌آموزان در داخل سلول‌های جدول نمایش داده شود در داخل تگ‌های td دستوراتی برای چاپ سال و آمار دانش‌آموزان اضافه می‌کنیم. برای این منظور برنامه years1.php به‌صورت زیر پیاده‌سازی می‌شود:

```
برنامه کامل شده years.php
<html>
<body>
<table width="200" border="1">
<tbody>
<tr>
<td>Year</td><td>Students</td>
</tr>
<?php
$y = 1390 ;
```

```

$n = 500 ;
while ( $y <=1395 ) { ?>
<tr>
<td><?= $y ?></td><td><?= $n ?></td>
</tr>
<?php
$y++;
} ?>
</tbody>
</table>

</body>
</html>

```

برای نمایش متغیرها از علامت <? نام متغیر = ؟> استفاده شده است. در شکل ۳-۱ خروجی برنامه years1.php را مشاهده می کنید:

Year	Students
۱۳۹۰	۵۰۰
۱۳۹۱	۵۰۰
۱۳۹۲	۵۰۰
۱۳۹۳	۵۰۰
۱۳۹۴	۵۰۰
۱۳۹۵	۵۰۰

شکل ۳-۱- خروجی حلقه تکرار

۳-۲-۲ دستورالعمل do... while

دستورالعمل do...while مجموعه دستورات داخل حلقه را یک بار اجرا کرده و سپس شرطی را بررسی می کند. اگر شرط درست بود، به ابتدای حلقه برگشته و دستورات را دوباره اجرا می کند و این تکرار را تا زمان برقرار بودن شرط ادامه می دهد. ساختار do..while حداقل یک بار دستورات داخل حلقه را اجرا می کند. شکل کلی این حلقه تکرار به صورت زیر است:

```

do{
مجموعه دستوراتی که باید اجرا شوند
} while (شرط);

```

مثال ۳: در کد زیر یک حلقه تکرار تعریف شده که با $i=1$ شروع می‌کند. سپس i را یک واحد افزایش می‌دهد و خروجی را چاپ می‌کند سپس شرط را بررسی می‌کند و تا زمانی که i کوچک‌تر یا مساوی ۵ باشد حلقه ادامه پیدا می‌کند.

کد برنامه numbers2.php	خروجی
<pre><?php \$i=1; do { echo "number ". \$i."
"; \$i++; } while (\$i <= 5); ?></pre>	<pre>number 1 number 2 number 3 number 4 number 5</pre>

۳-۲-۳ حلقه تکرار for

حلقه تکرار for معمولاً زمانی استفاده می‌شود که تعداد دفعات تکرار دستورات را بدانیم و دارای بخش‌هایی برای مقداردهی اولیه متغیرها، بررسی شرط مورد نظر و دستورات افزایش یا کاهش متغیرها می‌باشد. شکل کلی حلقه for به صورت زیر است:

```
for (init; condition; increment)
{
// دستوراتی که باید تکرار شوند اینجا نوشته می‌شوند
}
```

توضیح پارامترهای حلقه for

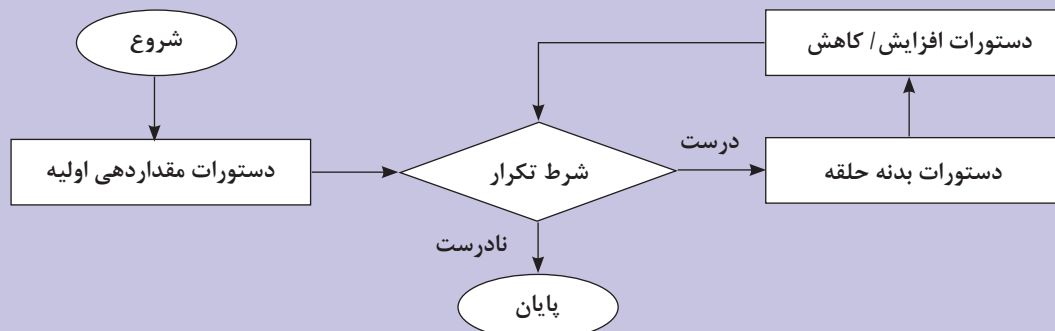
init: اغلب برای مقدار اولیه دادن به شمارنده استفاده می‌شود و می‌تواند شامل هر تعداد دستور باشد که در شروع اجرای حلقه اجرا خواهند شد.

condition: برای هر تکرار حلقه بررسی می‌شود. اگر حاصل آن TRUE باشد حلقه ادامه پیدا می‌کند. اگر مقدار آن نادرست باشد حلقه پایان می‌یابد.

increment: اغلب برای افزایش شمارنده استفاده می‌شود. (ولی می‌تواند شامل هر مجموعه دستوری باشد که در انتهای حلقه تکرار می‌شود)



هر کدام از پارامترهای فوق می تواند خالی باشد یا می تواند بیش از یک دستور باشد (جدا شده با ویرگول). در شکل ۲-۳، نحوه اجرای حلقه تکرار for نشان داده شده است:



شکل ۲-۳- روش اجرای حلقه for

مثال ۴: برنامه numbers3.php از حلقه for برای نمایش اعداد ۱ تا ۵ نوشته شده است که با $i=1$ شروع شده و تا زمانی که i کوچک تر یا مساوی ۵ باشد، دستور echo تکرار می شود:

برنامه numbers3.php	خروجی
<pre><?php for (\$i=1; \$i<=5; \$i++) { echo "number ".\$i."
"; } ?></pre>	<pre>number 1 number 2 number 3 number 4 number 5</pre>

مثال ۵: برای اجرای برنامه چاپ سال ها و آمار دانش آموزان با استفاده از حلقه تکرار for می توان برنامه years2.php را به صورت زیر پیاده سازی کرد:

برنامه years2.php
<pre><html> <body> <table width="200" border="1"> <tbody> <tr> <td>Year</td><td>Students</td></pre>

```

</tr>
<?php
$n = 500 ;
for( $y =1390 ; $y <= 1399 ; $y++ )
echo "<tr>
<td> $y </td><td> $n </td>
</tr>";

?>
</tbody>
</table>
</body>
</html>

```

در برنامه فوق، یک دستور echo در بدنه حلقه for قرار گرفته که یک رشته ثابت شامل تگ‌های HTML و متغیرها را نمایش می‌دهد. چون رشته ثابت داخل علامت‌های " " قرار گرفته‌اند پس به صورت هوشمندانه، متغیرها شناسایی شده و مقدار آنها نمایش داده می‌شود.

در مثال فوق به جای اینکه تگ‌های HTML را با دستور echo نمایش دهیم، بهتر است از چند بلوک برنامه‌نویسی PHP استفاده کرده و اجازه دهیم تگ‌های HTML خارج از بلوک PHP قرار گیرند:

قرار دادن تگ‌های HTML خارج از بلوک‌های PHP

```

<?php
$n = 500 ;
for( $y =1390 ; $y <= 1399 ; $y++ ) { ?>
<tr>
<td><?= $y ?></td><td><?= $n ?></td>
</tr>
<?php } ?>

```


مثال ۶: برای نمایش اعداد ۱ تا ۱۰ و مجذور آنها می‌توان صفحه `majzoor.php` را به صورت زیر پیاده‌سازی کرد:

روش اول	روش دوم
<pre><html> <body> <table width="300" border="1"> <?php for (\$i=1 ; \$i <=10 ; \$i++) echo "<tr><td>". \$i. "</td><td>".(\$i*\$i). "</td></tr>" ; ?> </table> </body> </html></pre>	<pre><html> <body> <table width="300" border="1"> <?php for (\$i=1 ; \$i<=10;\$i++) { ?> <tr> <td> <?php echo \$i; ?> </td> <td> <?php echo \$i*\$i; ?> </td> </tr> <?php } ?> </table> </body> </html></pre>

همان‌گونه که در برنامه فوق مشاهده می‌کنید، برای حل این مثال دو روش وجود دارد، یک روش نوشتن دستورات خروجی به زبان PHP برای چاپ تگ‌های مربوط به جدول بوده و روش دوم نوشتن تگ‌های جدول به صورت HTML و نوشتن حداقل دستورات PHP است. معمولاً از روش دوم برای طراحی چنین صفحاتی استفاده می‌شود.

۳-۳-۲ دستورات break، continue و exit

دستورات `break` و `continue` در داخل حلقه‌های تکرار نوشته شده و کاربردهای زیر را دارند:

۳-۳-۱ دستور break

دستور `break` در هر کدام از حلقه‌های تکرار از جمله `while`، `do..while` و `for` یا ساختار `switch` استفاده شده و باعث قطع کردن تکرار حلقه می‌شود. به این صورت که اگر نیاز باشد در داخل یک حلقه تکرار شرط خاصی بررسی شده و در صورت برقرار بودن شرط سریعاً اجرای حلقه قطع شود از دستور `break` استفاده می‌گردد. همچنین در بخش قبل دیدید که برای قطع اجرای دستورات در ساختار `switch` از دستور `break` استفاده شد.

مثال ۱: در برنامه break.php، هر زمان مقدار \$num از عدد ۷ بیشتر شد، حلقه تکرار قطع می‌شود و کنترل به دستور بعد از حلقه تکرار منتقل می‌گردد.

برنامه break.php	خروجی
<pre><?php \$num=1; while (\$num<=10) { if (\$num >= 7) break ; echo "number \$num
"; \$num++; } ?></pre>	<pre>number 1 number 2 number 3 number 4 number 5 number 6</pre>

در کد فوق علی‌رغم این که شرط حلقه تکرار برای رسیدن \$num به ۱۰ نوشته شده ولی در عمل در صورت رسیدن \$num به ۷ حلقه قطع می‌شود و بنابراین اعداد ۱ تا ۶ نمایش داده خواهند شد.

۲-۳-۳ دستور continue

از دستور continue برای صرف‌نظر از ادامه اجرای دستورات داخل حلقه و پرش به مرحله بعد آن استفاده می‌شود. اگر نیاز باشد در شرایط خاصی از اجرای دستورات بعدی حلقه، صرف‌نظر شده و کنترل به ابتدای حلقه برگشت کند از دستور continue استفاده می‌شود.

مثال ۲: در برنامه continue.php تنها اعداد فرد بین ۱ تا ۱۰ نمایش داده شوند:

برنامه continue.php	خروجی
<pre><?php for (\$i=1 ; \$i<=10 ; \$i++) { if (\$i % 2 ==0) continue; echo "number \$i
 "; } ?></pre>	<pre>number 1 number 3 number 5 number 7 number 9</pre>

در برنامه فوق، شمارنده \$i از ۱ تا ۱۰ تغییر می‌کند و هر بار بررسی می‌شود اگر \$i بر ۲ بخش‌پذیر باشد، از اجرای ادامه حلقه (دستور echo) صرف‌نظر شده و مرحله بعد اجرا می‌شود.

۳-۳-۳ دستور exit

از این دستور برای قطع اجرا و خروج از برنامه استفاده می‌شود. معمولاً در اغلب موارد دستورات برنامه از ابتدا تا انتها اجرا می‌شوند مگر در مواقعی که خطا رخ داده یا شرایط مورد نظر برای اجرای ادامه برنامه فراهم نباشد که می‌توان با اجرای دستور exit اجرای برنامه را قطع کرد.

مثال ۳: در برنامه exit.php، ساعت جاری از سیستم گرفته شده و بررسی می‌شود اگر کوچکتر از ۶ بود، بعد از نمایش یک پیغام، اجرای برنامه قطع می‌شود در غیر این صورت اجرای برنامه ادامه پیدا می‌کند:

برنامه exit.php

```
<?php
$hour = date("H");
if ( $hour < 6 ) {
echo "Sleep now!";
exit;
}
echo "Good morning!";
echo "How are you?";
?>
```

۴-۳-۴ مثال طراحی و برنامه‌نویسی

در صفحه زیر با کمک تگ‌های div و img و تنظیمات CSS، یک جعبه به صورت زیر طراحی شده است:

```
<div class="box">

<strong>کتاب<?=$k?></strong>
</div>
```

سپس با کمک یک حلقه تکرار for، ۶ نمونه از جعبه فوق در صفحه تکرار شده است. توجه کنید که برای نمایش صحیح این برنامه باید فایل book.png در زیرپوشه images موجود باشد.

صفحه book-index.php

```
<head>
<style>
.box{
width:70px;
height:100px;
```

```

border: 1px solid #444;
float: right;
margin: 5px;
padding: 5px;
text-align: center;
}
.box img {
width: 70px;
height: 70px;
}

</style>
</head>
<body>
<?php
for($k=1 ; $k<=6 ; $k++) { ?>
<div class="box">

<strong>کتاب<?=$k?></strong>
</div>
<?php } ?>
</body>
</html>

```

نتیجه اجرای صفحه book_index.php در شکل ۳-۳ نشان داده شده است:



شکل ۳-۳- نمایش لیست کتاب‌ها

ساختارهای شرطی و تکرار، کاربرد زیادی در برنامه‌نویسی داشته و برای حل مسائل محاسباتی و پیچیده استفاده می‌شوند. در این فصل برای رعایت اختصار و یادگیری راحت‌تر مطالب، تنها به ارائه مثال‌های ساده از این ساختارها پرداخته شده و از بیان جزئیات بیشتر و ارائه مثال‌های محاسباتی و پیچیده‌تر پرهیز گردید. در فصل‌های بعد برحسب نیاز از این ساختارها استفاده خواهد شد.

خودآزمایی

- ۱ ساختارهای شرطی چه کاربردی در برنامه‌ها دارند؟
- ۲ ساختارهای شرطی در PHP را نام ببرید.
- ۳ دستور break در ساختار switch چه کاربردی دارد؟
- ۴ ساختارهای تکرار PHP را نام ببرید.
- ۵ ساختارهای تکرار while و do..while چه تفاوتی با هم دارند؟
- ۶ شکل کلی دستور for را نوشته و قسمت‌های آن را توضیح دهید.
- ۷ با استفاده از حلقه تکرار while، قطعه کدی برای چاپ اعداد فرد دو رقمی بنویسید.
- ۸ با استفاده از حلقه تکرار for برنامه‌ای برای نمایش یک جدول دو ستونی که در ستون اول آن اعداد ۱ تا ۱۰ و ستون دوم آن شامل اعداد زوج ۲ تا ۲۰ باشد بنویسید.
- ۹ دستور break و continue چه کاربردی دارند؟
- ۱۰ کاربرد دستور exit را بنویسید.

سوالات چهار گزینه‌ای

- ۱ کدام یک از ساختارهای شرطی برای مقایسه یک متغیر با چند مقدار ثابت و انجام دستورات مختلف در هر حالت مناسب‌تر است؟
الف) switch ب) if..else ج) if د) موارد ب و ج
- ۲ برای صرف‌نظر از ادامه اجرای یک حلقه و شروع مرحله بعد آن از کدام دستور استفاده می‌شود؟
الف) next ب) break ج) continue د) cancel
- ۳ دستور break چه کاربردی در ساختار تکرار for دارد؟
الف) قطع اجرای حلقه تکرار ب) قطع اجرای مرحله جاری
ج) پرش از روی دستورات بعدی د) همه موارد

۴ نمایش خروجی قطعه کد زیر در مرورگر به چه صورت است؟

```
$K=1;
while ( $K < 6 )
{
  echo "<br> $K ";
  $K++;
}
```

الف) نمایش اعداد 1 تا 5 در یک سطر
ب) نمایش 1 تا 5 در سطرهای زیر هم
ج) نمایش $\$K
$ در سطرهای مختلف
د) نمایش پیغام خطا

۵ در کدام یک از حلقه‌های تکرار، دستورات داخل حلقه حداقل یک بار اجرا می‌شوند؟

الف) for (ب) while (ج) foreach (د) do..while

۶ خروجی قطعه کد زیر کدام است؟

```
$x=0 ;
for( $i=1 ; $i<5 ; $i = $i+3)
  $x++;
echo $x;
```

الف) 1 (ب) 2 (ج) 3 (د) 4

۷ خروجی قطعه کد زیر کدام است؟

```
$x=5;
$sum=0;
while ( $x <=8 ){
  $x++;
  if ( $x>7) continue;
  $sum = $sum + $x;
}
echo $sum;
```

الف) 0 (ب) 6 (ج) 13 (د) 21

۸ خروجی قطعه کد زیر چیست؟

```
$x=100;
if ( $x %2 ==0 )
  echo "<tr><td>$x+1</td></tr>";
```

الف) $\langle tr \rangle \langle td \rangle \$x+1 \langle /td \rangle \langle tr \rangle$
ب) $\langle tr \rangle \langle td \rangle 101 \langle /td \rangle \langle tr \rangle$
ج) $\langle tr \rangle \langle td \rangle 100+1 \langle /td \rangle \langle tr \rangle$
د) خروجی ندارد

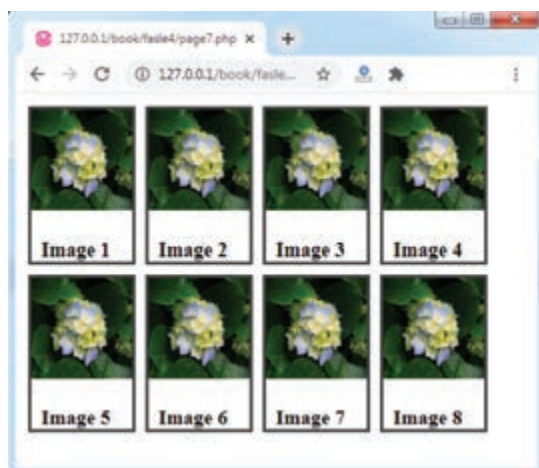
پاسخنامه

۸	۷	۶	۵	۴	۳	۲	۱	سؤال
ج	ج	ب	د	ب	الف	ج	الف	پاسخ

دستور کار عملی

X	X+۵
۱	۶
۲	۷
۳	۸
۴	۹
۵	۱۰

۱ با کمک تگ‌های html، کلاس‌های stylesheet و حلقه‌های تکرار، صفحه مشابه زیر طراحی کنید که سطرها را با استفاده از ساختار تکرار نمایش دهد و با استفاده از ساختارهای شرطی برای سطرهای زوج و فرد، رنگ‌های پس‌زمینه متفاوت استفاده نماید.



۲ با کمک تگ‌های div و کلاس‌های stylesheet، یک جعبه طراحی کرده و سپس آن را داخل یک حلقه تکرار به طول ۸ قرار دهید تا خروجی مقابل را تولید کند.



فصل چهارم

آرایه‌ها در زبان PHP

	name	price	color
0	■	■	■
1	■	■	
2	■	■	■

هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- انواع آرایه‌ها را توضیح دهد.
- آرایه‌ها را در برنامه تعریف و مقداردهی کند.
- برای نگه‌داری لیست‌های مختلف از آرایه استفاده کند.
- آرایه‌ها را پیمایش کند.
- عملکرد توابع مهم روی آرایه‌ها را توضیح دهد.
- توابع آرایه‌ای را در برنامه استفاده کند.

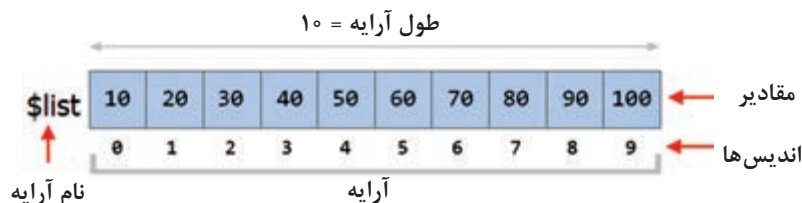
در دو فصل قبل با اصول اولیه برنامه‌نویسی و ایجاد ساختارهای شرطی و حلقه‌های تکرار در زبان PHP آشنا شدید. در این فصل با آرایه‌ها آشنا می‌شوید که برای نگه‌داری مجموعه‌ای از داده‌ها در حافظه استفاده شده و کاربرد زیادی در برنامه‌نویسی دارند.

۴-۱- تعریف آرایه‌ها در PHP

به تعدادی از خانه‌های حافظه که دارای نام یکسان ولی اندیس‌های مختلف هستند، آرایه گفته می‌شود. هر کدام از خانه‌های آرایه دارای اندیس یکتا بوده و می‌توانند یک یا چند مقدار داده را نگه‌داری کنند. در زبان PHP برای ایجاد آرایه نیاز به تعریف متغیر یا تعیین نوع داده و طول آن نیست بلکه کافی است خانه‌های آن مقداردهی شود و هر خانه‌ای از آرایه که مقداردهی شود، بلافاصله ایجاد می‌شود. همچنین طول آرایه مشخص و ثابت نیست و می‌توان در هر زمان خانه‌های جدیدی برای آن مقداردهی و ایجاد کرد. نوع داده هر کدام از خانه‌های آرایه با توجه به مقدار قرارگرفته در آن تعیین شده و در طول اجرای برنامه با مقداردهی‌های بعدی ممکن است تغییر کند همچنین لزومی ندارد که همه خانه‌های آرایه دارای نوع داده یکسان داشته باشند. به‌طور کلی دو نوع آرایه در زبان PHP وجود دارد که در ادامه به ارائه روش تعریف و پردازش آنها پرداخته می‌شود.

۴-۲- آرایه‌های شمارشی

آرایه‌های شمارشی از اندیس‌های عددی استفاده می‌کنند. این اندیس‌ها از صفر شروع شده و قابل شمارش هستند (به‌صورت ۰، ۱، ۲، ...). در شکل ۴-۱ یک آرایه شمارشی دارای ۱۰ خانه نشان داده شده است:



شکل ۴-۱- آرایه شمارشی به طول ۱۰

مثال: نگهداری نمرات دانش‌آموزان در یک آرایه

برای ایجاد آرایه می‌توان به چند روش عمل کرد. در روش اول می‌توان هر کدام از خانه‌های آرایه را به صورت زیر مقداردهی کرد:

```
$mark1[0] = 16.5 ;
```

```
$mark1[1] = 18.75 ;
```

```
$mark1[2] = 16.25 ;
```

```
$mark1[3] = 19.5 ;
```

در روش دوم خانه‌های آرایه بدون نوشتن اندیس‌ها مقداردهی می‌شود که اندیس‌ها به صورت خودکار از صفر شروع شده و افزایش پیدا می‌کند:

```
$mark1[] = 16.5 ;
```

```
$mark1[] = 18.75 ;
```

```
$mark1[] = 16.25 ;
```

```
$mark1[] = 19.5 ;
```

در روش سوم می‌توان از تابع `array` استفاده کرده و مقادیر اولیه برای آرایه را به ترتیب تعیین نمود:

```
$mark1=array(16.5 , 18.75 , 16.25 , 19.5 );
```

با اجرای دستور فوق، یک آرایه به طول ۴ و اندیس‌های ۰ تا ۳ ایجاد شده و مقادیر تعیین شده در آن ذخیره می‌شوند. به جای استفاده از تابع `array` می‌توان از علامت `[]` هم استفاده کرد:

```
$mark1=[ 16.5 , 18.75 , 16.25 , 19.5 ];
```

مهم نیست از کدام روش برای ایجاد آرایه استفاده شود، در هر صورت یک آرایه دارای ۴ خانه با اندیس‌های ۰ تا ۳ ایجاد خواهد شد.

برای پیمایش آرایه به عنوان مثال نمایش همه عناصر آرایه می‌تواند هر کدام از عناصر را با یک دستور جداگانه نمایش داد:

```
echo $mark1[0];
```

```
echo $mark1[1];
```

```
echo $mark1[2];
```

```
echo $mark1[3];
```

که البته بهتر است برای پیمایش آرایه از حلقه تکرار استفاده شود:

```
for ( $i =0 ; $i<4 ; $i++ )
```

```
    echo $mark1[ $i ];
```

۳-۴- آرایه‌های انجمنی

آرایه‌های انجمنی نوع دیگری از آرایه‌ها در PHP هستند که برای خانه‌های آنها به جای اندیس‌های عددی از کلیدهای متنی استفاده می‌شود. این روش اندیس‌گذاری کمک می‌کند تا با داشتن کلید یک خانه مستقیماً به محتوای آن خانه آرایه در دسترس باشد.

به عنوان مثال برای ذخیره کردن نمرات دانش آموزان در یک آرایه انجمنی می توان اسامی دانش آموزان را به عنوان کلید و نمرات را به عنوان مقادیر استفاده کرد:

```
$mark2['ali']= 16.5 ;
```

```
$mark2['hamid']= 18.75 ;
```

```
$mark2['amin']= 16.25 ;
```

```
$mark2['hadi']= 19.5 ;
```

همچنین به جای مقادیردهی تک تک خانه ها می توان از تابع array یا گروه ها [] به شکل زیر استفاده کرد:

```
$mark2 = array ('ali' =>16.5 , 'hamid' =>18.75 , 'amin' =>16.25 , 'hadi' => 19.5 );
```

با استفاده از [] هم دستور زیر برای تعریف آرایه استفاده می شود:

```
$mark2 = ['ali' =>16.5 , 'hamid' =>18.75 , 'amin' =>16.25 , 'hadi' =>19.5 ];
```

در این دستورات برای هر خانه از آرایه ابتدا کلید آن نوشته شده، سپس عملگر => و بعد از آن مقدار مربوط به خانه قرار می گیرد.

مزیت استفاده از آرایه های انجمنی این است که با داشتن کلید یک خانه به سادگی و بدون نیاز به بررسی خانه های آرایه، مقدار آن به دست می آید. به عنوان مثال در آرایه \$marks برای به دست آوردن نمره دانش آموزی با نام ali کافی است نوشته شود:

```
$m = $mark2['ali'];
```

برای پیمایش آرایه های انجمنی و دسترسی به همه عناصر آنها از حلقه تکرار foreach استفاده می شود که در ادامه این فصل، به آن پرداخته شده است.

۴-۴- روش نگهداری رکوردها در حافظه

یک رکورد مجموعه ای از چند فیلد اطلاعاتی مربوط به یک موضوع است. به عنوان مثال رکورد کتاب دارای فیلدهای نام، مؤلف، سال انتشار و قیمت است.

در زبان PHP برای نگهداری رکوردها از آرایه های انجمنی استفاده می شود به این صورت که نام فیلدها به عنوان کلیدهای آرایه و مقدار آنها به عنوان مقادیر خانه های آرایه استفاده می شود.

به عنوان مثال برای نگهداری رکورد کتاب PHP که در جدول ۴-۱ مشاهده می شود، از آرایه انجمنی به شکل زیر استفاده می گردد.

جدول ۴-۱

ویژگی	مقدار
bid	1001
bname	PHP
author	Mohammad
price	500

صفحه book.php

```
<?php
$book=array('bid' =>1001 ,
'byname' =>'PHP',
'author' =>'Mohammad',
'price' =>500 );

var_dump($book);
?>
```

خروجی کد فوق کلیدها، مقادیر و نوع داده برای همه عناصر آرایه می‌باشد:

```
C:\wamp64\www\book\fasle04\book.php:9:
```

```
array(size=4)
'bid' =>int1001
'byname' =>string'PHP'(length=3)
'author' =>string'Mohammad'(length=8)
'price' =>int500
```

تابع `var_dump()` نوع داده و محتوای متغیرها و عناصر آرایه را نمایش می‌دهد. این تابع بیشتر توسط برنامه‌نویس جهت مشاهده محتوای آرایه و رفع اشکال برنامه استفاده می‌شود.

۵-۴- حلقه تکرار foreach

برای پیمایش آرایه‌های شمارشی می‌توان از هرکدام از حلقه‌های تکرار `while`، `do..while` و `for` استفاده کرد و کافی است یک متغیر برای شمارش اندیس‌های آرایه استفاده شده و مقدار آن از صفر تا آخرین اندیس آرایه تغییر کند درحالی که این حلقه‌ها برای آرایه‌های انجمنی با توجه به این که دارای اندیس‌های قابل شمارش نیستند، کاربرد ندارند.

حلقه تکرار `foreach` برای پیمایش و گرفتن خروجی از هر دوی آرایه‌های شمارشی و انجمنی کاربرد دارد. شکل کلی این حلقه تکرار به صورت زیر است:

```
(متغیر مقدار=<متغیر کلید as نام آرایه) foreach
{
// دستورات بدنه حلقه
}
```

در این حلقه، عناصر آرایه یکی یکی پیمایش شده، برای هر عنصر کلید آن در متغیر کلید و مقدار آن در متغیر مقدار قرار گرفته و دستورات بدنه حلقه اجرا می‌شود. این دستورات می‌توانند از متغیر کلید و متغیر مقدار استفاده کنند.

حلقه تکرار foreach برای گرفتن خروجی از آرایه استفاده شده و حتی اگر متغیرهای کلید و مقدار در بدنه حلقه تغییر هم کند، این تغییرات روی آرایه اعمال نخواهد شد.

مثال ۱: برنامه show-mark1.php برای چاپ نمرات دانش‌آموزان از آرایه \$mark2 نوشته شده است:

برنامه show-mark1.php	خروجی
<pre><?php \$mark2 = array ('ali' =>16.5 , 'hamid' =>18.75 , 'amin' =>16.25 , 'hadi' => 19.5); foreach(\$mark2 as \$v) echo " \$v
" ; ?></pre>	<pre>16.5 18.75 16.25 19.5</pre>

مثال ۲: برنامه show-mark2.php برای چاپ اسامی و نمرات دانش‌آموزان از آرایه \$mark2 نوشته شده است:

برنامه show-mark2.php	خروجی
<pre><?php \$mark2 = array ('ali' =>16.5 , 'hamid' =>18.75 , 'amin' =>16.25 , 'hadi' =>19.5); foreach(\$mark2 as \$k => \$v) echo " \$k : \$v
" ; ?></pre>	<pre>ali:16.5 hamid:18.75 amin:16.25 hadi:19.5</pre>

در برنامه فوق برای هر خانه از آرایه کلید آن (نام دانش‌آموز) در متغیر \$k و مقدار آن (نمره) در متغیر \$v قرار می‌گیرد و هر دو نمایش داده می‌شوند. نام متغیرهای \$k و \$v به صورت دلخواه تعیین شده است.

۴-۶- آرایه‌های تودرتو

در زبان PHP هر کدام از خانه‌های آرایه می‌توانند یک مقدار ساده یا یک آرایه را نگه‌داری کنند. اگر هر کدام از خانه‌های آرایه حاوی یک آرایه باشند یک آرایه دو بعدی خواهیم داشت. به علاوه هر کدام از خانه‌های آرایه‌های داخلی هم می‌توانند مقدار ساده یا آرایه داشته باشند و آرایه‌های سه بعدی یا بیشتر به وجود خواهند آمد. مثال: در آرایه زیر مشخصات ۳ تا از کتاب‌ها ذخیره شده است به گونه‌ای که هر سطر از آرایه مشخصات یک کتاب را دارا بوده و در هر سطر آرایه‌ای با سه خانه قرار دارد که در هر خانه آن یکی از فیلدهای کتاب نگه‌داری می‌شود. به عبارت دیگر هر سطر حاوی یک آرایه انجمنی است و کلیدهای آن اسامی فیلدهای کتاب است.

```
$books=array(array('name' =>'PHP', 'author' =>'Mina' ) ,
              array('name' =>'Physics','author' =>'Ali'),
              array('name' =>'Math','author'=>'Sara' ) );
```

می‌توان آرایه books را یک آرایه دو بعدی دارای ۳ سطر با اندیس‌های ۰ تا ۲ و سه ستون دارای اندیس‌های name و author به شکل ۴-۲ در نظر گرفت:

	name	author	
\$books[0]	PHP	Mina	← سطر ۰
\$books[1]	Physics	Ali	← سطر ۱
\$books[2]	Math	Sara	← سطر ۲

↑ ↑
ستون name ستون author

شکل ۴-۲- آرایه دو بعدی

برای پیمایش آرایه‌های دو بعدی می‌توان از حلقه‌های تکرار تودرتو استفاده کرد. در صفحه book_list.php، مشخصات کتاب‌ها در یک جدول HTML نمایش داده شده است:

صفحه book-list1.php

```
<html>
<body>
<?php
$books=array(array('name' =>'PHP', 'author' =>'Mina' ) ,
                array('name' =>'Physics','author' =>'Ali'),
                array('name' =>'Math','author'=>'Sara' ) );
?>
<table width="600" border="1">
```

```

<tbody>
<tr>
<td>Name</td>
<td>Author</td>
</tr>
<?php foreach($books as $row ){ ?>
<tr>
<?php foreach($row as $v ) { ?>
<td><?=$v ?></td>
<?php } ?>
</tr>
<?php } ?>
</tbody>
</table>

</body>
</html>

```

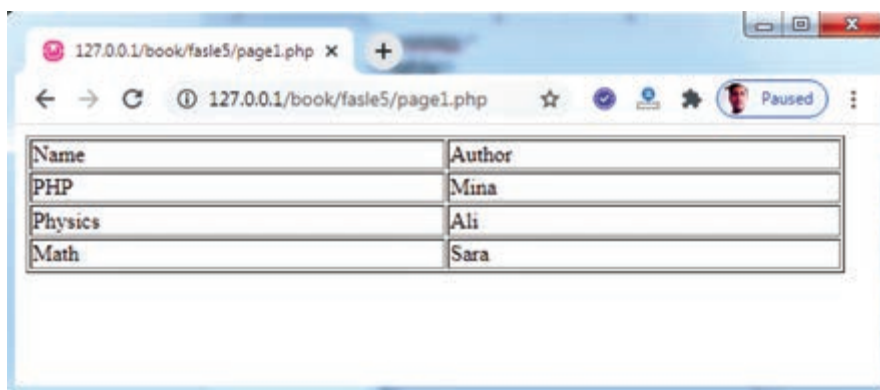
در کد فوق می بینید که سطرهای آرایه با کمک یک حلقه تکرار foreach پیمایش شده و هرکدام از سطرهای آرایه \$books در متغیر \$row قرار گرفته اند. \$row نیز یک آرایه انجمنی بوده و برای پیمایش آن از حلقه تکرار foreach دیگری استفاده شده است. در این مثال چون تعداد خانه های آرایه \$row فقط دو تا (نام و مؤلف) می باشد می توان به جای نوشتن حلقه تکرار foreach داخلی با دو دستور جداگانه نام و مؤلف کتاب را به صورت زیر نمایش داد:

```

<?php foreach($books as $row ){ ?>
<tr>
<td><?=$row['name'] ?></td>
<td><?=$row['author'] ?></td>
</tr>
<?php } ?>

```


خروجی برنامه فوق در شکل ۳-۴ نشان داده شده است:



Name	Author
PHP	Mina
Physics	Ali
Math	Sara

شکل ۳-۴- نمایش آرایه دو بعدی

۷-۴- طراحی صفحه نمایش لیست کتاب‌ها

در این قسمت با کمک تگ‌های HTML و تنظیمات CSS و استفاده از آرایه‌ها و حلقه‌های تکرار یک صفحه وب کامل طراحی شده است. این صفحه مشابه شکل زیر دارای نوار پیمایش شامل لینک‌های موضوعات در بالای صفحه بوده و لیست کتاب‌ها را در بدنه صفحه نمایش می‌دهد. آرایه‌های مورد استفاده به صورت زیر تعریف می‌شوند: آرایه موضوعات:

```
<?php
$subs = ['1'=>'داستانی',
        '2'=>'سیاسی',
        '3'=>'تاریخی',
        '4'=>'جغرافیایی'];
?>
```

آرایه کتاب‌ها:

```
$books=[
    ['bname'=>'شاهزاده و گدا', 'cover'=>'b1.jpg'],
    ['bname'=>'نجات دلفین کوچولو', 'cover'=>'b2.jpg'],
    ['bname'=>'آینده کهن', 'cover'=>'b3.jpg'],
    ['bname'=>'نقشه نابودی ماه', 'cover'=>'b4.jpg'],
    ['bname'=>'پاریس من و پدرم', 'cover'=>'b5.jpg'];
];
```

کلید cover برای نگه‌داری نام فایل تصویر جلد کتاب استفاده شده و برای نمایش کامل صفحه باید تصاویری با این اسامی در زیرپوشه images سایت موجود باشند.

همچنین برای نمایش موضوعات در ستون کناری از یک لیست غیرترتیبی (ul) استفاده شده و برای نمایش کتاب‌ها در بدنهٔ صفحه از تگ div و تنظیمات استایل شیت استفاده شده است. این تنظیمات در یک فایل CSS خارجی به صورت زیر نوشته شده است:

فایل style.css

```
body{ direction: rtl;}
.sub{
background-color:#eee;
display: block;
width:70px;
height: 20px;
padding: 5px;
margin: 3px;
float: right;
}
.content{
background-color: #eee;
}
.content .box{
width:100px;
height:150px;
background-color:#fff;
float:right;
margin:5px;
}
.content .box img{
width:100px;
height:100px;
}
```

در نهایت صفحه `book_index.php` به صورت زیر طراحی می‌شود. ابتدا در یک حلقه تکرار به ازای هر کدام از عناصر آرایه موضوعات، یک تگ `a` و به ازای هر کدام از عناصر آرایه کتاب‌ها، یک تگ `div` نمایش داده می‌شود:

```

صفحه book-index.php
<html>
<head>
<link href="style1.css" rel="stylesheet" type="text/css"/>
</head>
<body>
<?php
$subs=['1'=>'سیاسی',
'2'=>'داستانی',
'3'=>'تاریخی',
'4'=>'جغرافیایی'];
$books=[
['bname' =>'آخرین دقایق', 'cover'=>'a1.jpg' ],
['bname' =>'آینده کهن', 'cover'=>'a2.jpg' ],
['bname' =>'شهر زمرد', 'cover'=>'a3.jpg' ],
['bname' =>'سی شعر خیام', 'cover'=>'a4.jpg' ],
['bname' =>'نجات دلفین', 'cover'=>'a5.jpg' ]];
?>
<?php foreach($subs as $sid=>$sname){ ?>
<a class="sub" href=""><?=$sname?></a>
<?php } ?>
<br><br>
<h3>لیست کتاب‌ها</h3>
<div class="content">
<?php
foreach($books as $book) { ?>
<div class="box">

<strong><?=$book['bname']?></strong>
</div>
<?php } ?>
</div>

</body>
</html>

```

اجرای صفحه `book_index.php` در شکل ۴-۴ نشان داده شده است:



شکل ۴-۴- نمایش لیست موضوعات و کتاب ها

در این فصل با مفهوم آرایه در زبان PHP و نحوه تعریف، مقداردهی و پیمایش آرایه‌های شمارشی، انجمنی و تودرتو آشنا شدید. همچنین با استفاده از آرایه‌ها توانستید اطلاعات یک رکورد یا لیستی از رکوردها را در حافظه نگه‌داری کرده و آنها را به شکل دلخواه در صفحه وب نمایش دهید. در فصل‌های آتی مثال‌های بیشتری از کاربرد آرایه‌ها را خواهید دید.

خودآزمایی

- ۱ تفاوت بین آرایه‌های شمارشی و انجمنی چیست؟ توضیح دهید.
- ۲ روش تعریف و مقداردهی آرایه انجمنی را با ذکر مثال توضیح دهید.
- ۳ یک آرایه انجمنی برای نگه‌داری دمای هوای ۷ روز هفته تعریف کرده و مقداردهی کنید. (راهنمایی: نام روزها را به عنوان کلید و دمای هوا را به عنوان مقدار استفاده کنید)
- ۴ با ذکر مثال توضیح دهید چگونه برای نگه‌داری رکوردها از آرایه انجمنی استفاده می‌شود؟
- ۵ فرض کنید آرایه شمارشی `$list` دارای اسامی ۱۰ کتاب است، برنامه‌ای بنویسید که اسامی را در یک جدول با دو ستون نمایش دهد که ستون اول ردیف و ستون دوم نام کتاب باشد.
- ۶ کاربرد آرایه `foreach` چیست؟ شکل کلی آن را نوشته و توضیح دهید.
- ۷ با کمک حلقه تکرار `foreach` از روی آرایه زیر یک لیست انتخابی چاپ کنید.

```
$states=array('۱'=>'تهران','۲'=>'فارس','۳'=>'کرمان');
```

- ۸ مشخصات ۴ دانش‌آموز دلخواه شامل نام، نام خانوادگی و معدل را در یک آرایه دو بعدی ذخیره کنید.
- ۹ برنامه‌ای برای نمایش مشخصات دانش‌آموزان از آرایه سؤال ۸ در یک جدول HTML بنویسید.

سؤالات چهارگزینه‌ای

۱ کدام گزینه از ویژگی‌های آرایه در PHP می‌باشد؟

- (الف) تعدادی از خانه‌های پشت سر هم در حافظه
 (ب) تعدادی از خانه‌های حافظه با نام یکسان و اندیس‌های مختلف
 (ج) تعدادی خانه حافظه با نوع داده یکسان
 (د) همه موارد

۲ کدام گزینه برای مقداردهی آرایه درست نمی‌باشد؟

- (الف) `$a=array(4,5)` (ب) `$a=[4,5]` (ج) `$a={4,5}` (د) `$a[0]=4 ; $a[1]=5`

۳ کدام گزینه برای آرایه‌های انجمنی نادرست است؟

- (الف) عناصر آرایه می‌توانند از کلیدهای یکسان استفاده کنند.
 (ب) آرایه‌های با سرعت بالا برای دسترسی براساس کلید هستند.
 (ج) آرایه‌هایی با اندیس‌های متنی هستند.
 (د) با داشتن کلید یک خانه، مقدار به راحتی به دست می‌آید.

۴ کدام علامت در زبان PHP بین کلید و مقدار آرایه انجمنی قرار می‌گیرد؟

- (الف) `= >` (ب) `- >` (ج) `=` (د) `<<`

۵ خروجی دستورات زیر چیست؟

```
$list[] = 56;
$list[] = 87;
echo $list[1];
```

- (الف) خالی (ب) 56 (ج) 87 (د) نمایش خطا

۶ برای نگهداری مشخصات یک مشتری شامل نام، آدرس و تلفن بهتر است از کدام آرایه استفاده شود؟

- (الف) شمارشی (ب) انجمنی (ج) دوبعدی (د) همه موارد

۷ در دستورات زیر، برای نمایش نام کتاب، به جای علامت ؟ کدام گزینه قرار می‌گیرد؟

```
$book=array( array('name'=>'PHP'));
```

```
echo ? ;
```

- (الف) `$book['name']` (ب) `$book['name'][0]` (ج) `$book[0]['name']` (د) `$book[0]`

۸ برای نمایش کلیدها و مقادیر آرایه در دستورات زیر به جای علامت ؟ دقیقاً باید کدام گزینه قرار گیرد؟

```
$codes=array('98'=>'ایران', '61'=>'استرالیا', '43'=>'اتریش');
```

```
foreach($codes as ? )
```

```
echo "$code : $name ";
```

- (الف) `$code` (ب) `$name` (ج) `$code:$name` (د) `$code=>$name`

۹ هنگام اشکال‌یابی برنامه، از کدام دستور برای نمایش سریع کلیدها و مقادیر آرایه استفاده می‌شود؟

- (الف) `dump` (ب) `print` (ج) `var_dump` (د) `arr_print`

۹	۸	۷	۶	۵	۴	۳	۲	۱	سؤال
ج	د	ج	ب	ج	الف	الف	ب	ب	پاسخ

دستور کار عملی

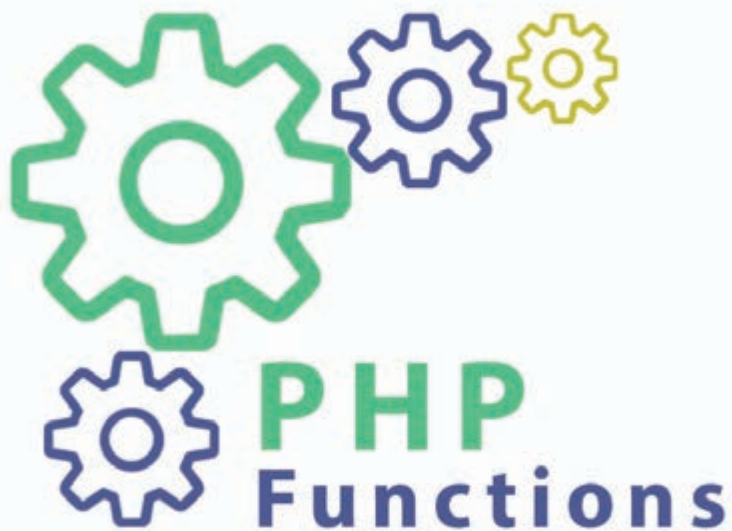
- ۱ یک آرایه شمارشی برای نگه‌داری اسامی ۱۰ دانش‌آموز تعریف و مقداردهی کنید. سپس با استفاده از حلقه‌های تکرار `while`، `do..while`، `for` و `foreach` برنامه‌هایی برای نمایش لیست اسامی بنویسید.
- ۲ یک آرایه انجمنی برای نگه‌داری روزهای هفته و تاریخ هر روز تعریف و مقداردهی کنید به گونه‌ای که نام روزهای هفته به عنوان کلید و تاریخ به عنوان مقدار استفاده شود. سپس با کمک تگ‌های `HTML` و حلقه تکرار لیست روزها و تاریخ‌ها را در جدول زیر نمایش دهید:

روز هفته	تاریخ
شنبه	۱۴۰۰/۰۲/۲۳
...	...

- ۳ یک آرایه دو بعدی برای نگه‌داری مشخصات ۵ کشور شامل نام کشور، جمعیت و نام قاره و نام تصویر پرچم آن تعریف و مقداردهی کنید. سپس یک صفحه وب شامل یک جدول برای نمایش لیست کشورها طراحی و پیاده‌سازی کنید که دارای ستون‌های پرچم، نام کشور، جمعیت و نام قاره باشد. (ابتدا تصاویر مربوط به پرچم کشورها را آماده و در یک زیر پوشه از سایت داشته باشید)

فصل پنجم

توابع در زبان PHP



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

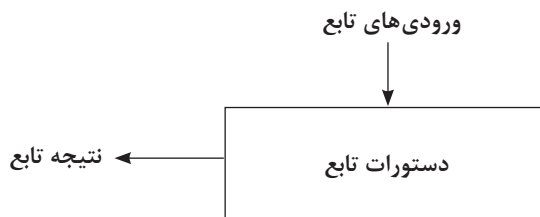
- تابع جدید در برنامه تعریف کند.
- توابع را به صورت درست فراخوانی کند.
- توابع پیش ساخته PHP را در برنامه استفاده کند.
- چند تابع پرکاربرد در PHP را نام ببرد.
- توابع رشته‌ای را در برنامه استفاده کند.

هر زبان برنامه‌نویسی از جمله PHP دارای مجموعه‌ای از توابع است که برنامه‌نویس باید با آنها آشنایی داشته و در مواقع لازم از آنها استفاده نماید. یک تابع مجموعه‌ای از دستورات از قبل نوشته شده تحت یک نام است که با فراخوانی نام آن، دستورات داخل آن اجرا می‌شوند همچنین اکثر زبان‌های برنامه‌نویسی امکان تعریف و پیاده‌سازی توابع جدیدی را در برنامه فراهم می‌کنند. در این فصل با روش تعریف و پیاده‌سازی توابع و نیز مهم‌ترین توابع پیش ساخته در زبان PHP آشنا می‌شوید.

۵-۱- انواع توابع

تابع دارای تعدادی دستور است که با یک نام مشخص تعریف شده‌اند. اگر در جایی از برنامه به آن دستورات نیاز باشد، به جای نوشتن آنها کافی است نام تابع نوشته شود تا دستورات داخل آن اجرا شوند که به این کار فراخوانی تابع می‌گویند.

هر تابع می‌تواند تعدادی ورودی را پذیرفته، روی آنها دستوراتی اجرا کرده و حاصل را نمایش داده یا به عنوان نتیجه برگرداند. شکل ۵-۱ نمای کلی از تابع را نشان می‌دهد:



شکل ۵-۱- شکل کلی تابع

در داخل یک تابع می‌توان از همه دستورات از جمله دستورات محاسباتی، ساختارهای شرطی و حلقه‌های تکرار استفاده کرد.

در زبان PHP، توابع دارای انواع زیر هستند:

توابع پیش ساخته: توابعی هستند که از قبل در زبان PHP تعریف شده‌اند و می‌توانند در برنامه فراخوانی شوند.
توابع کتابخانه‌ای: توابعی در زمینه‌های مختلف هستند که از قبل در فایل‌های کتابخانه‌ای نوشته شده و آماده می‌باشند. با ضمیمه نمودن این فایل‌ها و فراخوانی این توابع می‌توان از آنها استفاده نمود.

توابع برنامه‌نویس: توابعی هستند که توسط خود برنامه‌نویس ایجاد می‌شوند. هر برنامه‌نویس می‌تواند توابع مورد نیاز خود را در برنامه تعریف و پیاده‌سازی کرده و سپس آنها را فراخوانی نماید. در این فصل ابتدا روش تعریف و پیاده‌سازی تابع را با چند مثال ساده نشان داده و در ادامه توابع پیش ساخته مفید و پرکاربرد زبان PHP را معرفی خواهیم کرد.

۲-۵- روش تعریف و فراخوانی تابع

همان‌طور که پیش‌تر گفته شد، برنامه‌نویس می‌تواند برحسب نیاز، دستوراتی را به‌عنوان یک تابع، تعریف و پیاده‌سازی کند و آن را در ادامه برنامه فراخوانی نماید. در زبان PHP، در هر جای برنامه می‌توان تابع را تعریف و پیاده‌سازی کرد. برای تعریف تابع در PHP از شکل کلی زیر استفاده می‌شود:

(پارامترهای ورودی تابع) نام تابع function

```
{
دستورات تابع
return نتیجه تابع ;
}
```

نام تابع یک کلمه است که با یک حرف لاتین یا خط زیر (_) شروع شده و در ادامه می‌تواند شامل حروف و ارقام باشد.

پارامترهای ورودی مقادیر مورد نیاز را به‌داخل تابع منتقل می‌کنند تا دستورات روی آنها اجرا شده و نتیجه مورد نظر به‌دست آید. پارامترهای ورودی، اختیاری هستند و ممکن است تابعی هیچ ورودی نداشته باشد. نتیجه حاصل می‌تواند نمایش داده شده یا توسط دستور return برگردانده شود. مثال: تابع زیر برای نمایش عبارت Hello به‌صورت درشت استفاده می‌شود:

برنامه show_hello.php	خروجی
<pre><html> <body> <?php function showHello() { echo "<h1> Hello </h1>"; } // فراخوانی تابع showHello(); ?> </body> </html></pre>	Hello

تابع فوق، هیچ پارامتر ورودی ندارد و نتیجه را نمایش پس نیازی به دستور return ندارد. برای فراخوانی این تابع در هر جای برنامه کافی است دستور زیر نوشته شود:

```
<?php  
showHello();  
?>
```

نتیجه هر بار فراخوانی تابع فوق به صورت زیر می باشد:

```
<h1>Hello </h1>
```

که توسط مرورگر به صورت زیر نمایش داده خواهد شد:

Hello

دقت کنید که حتی اگر تابع پارامتر ورودی نداشته باشد باز هم در زمان تعریف و یا فراخوانی تابع باید برای آن پرانتز باز و بسته () نوشته شود.

۳-۵- استفاده از پارامترها در تعریف تابع

با هر بار فراخوانی تابع نوشته شده در مثال ۱، پیغام ثابت Hello نمایش داده می شود. برای تعریف تابعی که بتواند برای نمایش پیغام های مختلف استفاده شود، نیاز به استفاده از پارامتر است تا پیغام مورد نظر را از طریق پارامتر به آن منتقل کنیم. با استفاده از پارامترها می توان مقادیر مورد نظر را به داخل تابع انتقال داده و تابع می تواند کارایی و کاربر بیشتری داشته باشد.

مثال ۱: برای بهتر کردن عملکرد تابع در مثال ۱ و این که بتوان از آن برای چاپ پیغام های مختلف استفاده کرد، می توان برای آن پارامتر ورودی در نظر گرفت:

برنامه show-msg.php	خروجی
<pre><html> <body> <?php function showMsg(\$msg) { echo "<h1> \$msg </h1>"; } // فراخوانی تابع showMsg("Hello"); showMsg("Good Bye"); ?> </body> </html></pre>	<pre>Hello Good Bye</pre>

تابع فوق در هر جای برنامه می تواند برای نمایش هر پیغام دلخواهی استفاده شود:

```
<?php
showMsg("Hello");
showMsg("Good Bye");
?>
```

در زمان فراخوانی تابع فوق باید برای پارامترهای آن مقدار تعیین شود. به این مقادیر آرگومان های تابع می گویند. تعداد و ترتیب آرگومان ها باید به صورت دقیق با تعداد و ترتیب پارامترها هم خوانی داشته باشد. با دو بار فراخوانی تابع فوق خروجی زیر تولید خواهد شد:

```
<h1>Hello</h1>
<h1>Good Bye</h1>
```

که این خروجی توسط مرورگر به صورت متن های عنوان زیر نمایش داده می شود:

Hello

Good Bye

تابع تعریف شده در مثال ۲ می تواند برای نمایش هر پیغام دلخواه استفاده شود. بنابراین نسبت به تابع مثال ۱، مفیدتر بوده و کاربرد بیشتری دارد. پیشنهاد می شود برای هر تابع پارامترهای ورودی مناسب تعریف کنید تا بتوانید در موارد مختلف و با آرگومان های ورودی متفاوت از آنها استفاده نمایید. **مثال ۲:** در تابع زیر، پارامترهای سال، ماه و روز وارد شده و در کنار هم قرار گرفته و تاریخ با فرمت مورد نظر برگردانده می شود:

برنامه format-date.php	خروجی
<pre><?php function formatDate(\$y , \$m , \$d) { \$date = " \$y/\$m/\$d "; return \$date; } // فراخوانی تابع echo formatDate(1399,11,22); \$today = formatDate(1399,11,22); echo "
".\$today; ?></pre>	<pre>1399/11/22 1399/11/22</pre>

تابع فوق دارای سه پارامتر ورودی است و نتیجه با دستور return برگشت داده می شود. دستورات صفحه بعد برای فراخوانی این تابع نوشته شده است:

```

<?php
echo formatDate ( 1399 , 11 , 22 );
//...
$today = formatDate ( 1399 , 11 , 22 );
?>

```

در هنگام فراخوانی تابع، آرگومان‌ها باید به ترتیب سال، ماه و روز تعیین شود همچنین اگر قرار است از تابع نتیجه‌ای برگشت داده شود، در هنگام فراخوانی، این نتیجه باید نمایش داده شده یا در یک متغیر ذخیره شود. در مثال فوق در فراخوانی اول، نتیجه فراخوانی (رشته ۱۳۹۹/۱۱/۲۲) نمایش داده می‌شود و در فراخوانی دوم، نتیجه فراخوانی (رشته ۱۳۹۹/۱۱/۲۲) در متغیر \$today ذخیره می‌گردد.

مثال ۳: تابع زیر دارای یک پارامتر برای ورود شماره روز هفته است و شامل دستوراتی است که براساس پارامتر ورودی، نام روز هفته را پیدا کرده و برمی‌گرداند:

برنامه week-day.php	خروجی
<pre> <?php //تعریف تابع function weekDay(\$day) { switch(\$day) { case 1: \$name='شنبه' ; break; case 2: \$name='یکشنبه' ; break; case 3: \$name='دوشنبه' ; break; case 4: \$name='سه شنبه' ; break; case 5: \$name='چهارشنبه' ; break; case 6: \$name='پنجشنبه' ; break; case 7: \$name='جمعه' ; break; default:\$name='نامعتبر' ; } return \$name; } //فراخوانی تابع echo weekDay (2); echo "
"; echo weekDay (4); ?> </pre>	<pre> یکشنبه سه شنبه </pre>

۴-۵- روش مقداردهی پیش فرض پارامترها

معمولاً برای فراخوانی تابع تعیین آرگومان به‌ازای همه پارامترهای آن اجباری است ولی اگر هنگام تعریف تابع، برای پارامترهای آن مقدار پیش فرض تعیین گردد، نوشتن آرگومان برای آن پارامترها اختیاری می‌شود. بنابراین هرکدام از پارامترهای تابع که دارای مقدار پیش فرض هستند، در زمان فراخوانی می‌توانند مقداردهی نشوند. مثال: در تابع زیر پارامترهای \$d و \$m دارای مقدار پیش فرض هستند پس مقداردهی آنها اختیاری می‌باشد:

برنامه format-date2.php	خروجی
<pre><?php function formatDate2(\$y,\$m=2,\$d=1) { \$date = "\$y/\$m/\$d"; return \$date; } echo formatDate2(1399 , 8 ,21); echo "
"; echo formatDate2(1399 , 8); echo "
"; echo formatDate2(1399); ?></pre>	<pre>1399/8/21 1399/8/1 1399/2/1</pre>

در تابع formatDate2، پارامترهای \$m و \$d دارای مقدار پیش فرض بوده و در نتیجه هنگام فراخوانی تابع می‌توانند بدون مقدار بمانند که در این صورت مقدار پیش فرض برای آنها استفاده خواهد شد.

۵-۵- فراخوانی تابع به صورت پویا

یکی از امکانات جالب زبان PHP امکان فراخوانی پویای تابع است به این صورت که می‌توان نام تابع را در یک متغیر دلخواه ذخیره کرده و برای فراخوانی از این متغیر استفاده نمود. به‌عنوان مثال برای فراخوانی formatDate2 می‌توان به صورت زیر عمل کرد:

```
<?php
$name = "formatDate2";
echo $name( 1399 , 4 , 14 );
?>
```

۶-۵- فراخوانی تابع با ارجاع

معمولاً هنگام فراخوانی تابع، یک کپی از آرگومان‌ها در پارامترها قرار گرفته و تغییراتی که در داخل تابع روی پارامترها داده شود، روی آرگومان‌ها تأثیری ندارد ولی اگر هنگام تعریف تابع قبل از نام پارامتر علامت & قرار داده شود، این پارامتر ارجاعی شده و هنگام فراخوانی به جای استفاده از کپی آرگومان، مستقیماً از همان حافظه آرگومان استفاده کرده و تغییرات روی این پارامتر روی آرگومان هم انجام خواهد شد.

مثال: در تابع زیر پارامتر \$x به صورت معمولی و پارامتر \$y به صورت ارجاعی تعریف شده است:

برنامه ref-param.php	خروجی
<pre><?php function add(\$x , &\$y) { \$x = \$x + 2 ; \$y = \$y + 3 ; } // فراخوانی \$a = 10 ; \$b = 10 ; add (\$a , \$b); echo "A= \$a , B=\$b "; ?></pre>	A= 10 , B=13

در برنامه فوق، تغییرات پارامتر \$x روی آرگومان \$a تأثیری ندارد ولی تغییرات پارامتر ارجاعی \$y، به طور مستقیم روی آرگومان \$b انجام شده و مقدار آن را تغییر داده است.

۷-۵- متغیرهای محلی و سراسری

در زبان PHP، برای ایجاد یک متغیر به آن مقدار داده می‌شود و اگر برنامه دارای توابع هم باشد، این متغیرها می‌توانند اولین بار در خارج یا داخل توابع مقداردهی شوند بنابراین متغیرها می‌توانند در خارج از توابع یا در داخل یکی از توابع ایجاد شوند.

بر این اساس دو نوع متغیر وجود دارد:

متغیرهای سراسری: متغیرهای تعریف شده در خارج از همه توابع (در بدنه اصلی برنامه) که تا پایان اجرای برنامه در حافظه باقی می‌مانند.

متغیرهای محلی تابع: متغیرهای تعریف شده در داخل یک تابع که هنگام فراخوانی تابع ایجاد شده و با پایان یافتن تابع از بین می‌روند.

مثال ۱: در تابع زیر، متغیر x به صورت سراسری و متغیر y به صورت محلی تعریف شده است:

متغیرهای سراسری و محلی
<pre><?php \$x = 0; function add() { \$y = 5 ; \$x = \$x + \$y ; // با خطا مواجه می شود } add (5); ?></pre>

در تابع فوق، متغیر y به صورت محلی و متغیر x به صورت سراسری تعریف شده‌اند. در حالت معمول متغیرهای سراسری در داخل توابع شناخته نمی‌شوند بنابراین دستور $x = x + y$ با خطای «استفاده از متغیر ناشناس» مواجه می‌شود.

متغیرهای مقداردهی شده در داخل یک تابع، متغیر محلی برای همان تابع هستند حتی اگر هم‌نام با متغیرهای سراسری باشند. برای اینکه در داخل یک تابع متغیرهای سراسری شناخته شده و قابل استفاده گردد باید در داخل تابع با دستور `global` نام آنها معرفی شود.

مثال ۲: در برنامه زیر، متغیر x به صورت سراسری تعریف شده و در داخل تابع با دستور `global` معرفی شده و سپس ۵ واحد به مقدار آن اضافه شده است:

برنامه <code>global-vars.php</code>	خروجی
<pre><?php \$x = 23; function addFive() { global \$x; \$x = \$x + 5 ; } echo "Before: \$x "; addFive(); echo "
After: \$x "; ?></pre>	<p>Before: 23 After: 28</p>

۸-۵- توابع پیش ساخته در PHP

توابع پیش ساخته به توابعی گفته می‌شود که از قبل در زبان برنامه‌نویسی تعریف شده‌اند. یکی از ویژگی‌های خوب PHP وجود توابع پیش‌ساخته بسیار زیاد آن است. در PHP بیش از ۱۰۰۰ تابع پیش‌ساخته وجود دارد که می‌تواند برای انجام عملیات مختلف روی اعداد، رشته‌ها، آرایه‌ها، تاریخ و... استفاده شوند. وقتی با زبان PHP برنامه‌نویسی می‌کنید تقریباً برای هر کاری که فکر کنید، تابع مناسبی وجود داشته و بجای نوشتن دستورات طولانی می‌توانید به راحتی تابع مناسب را فراخوانی کنید.

مراجع مختلفی در اینترنت وجود دارد که لیست کامل توابع کتابخانه‌ای PHP را با توضیحات کافی و مثال‌هایی برای نمایش نحوه فراخوانی آنها ارائه کرده‌اند. از جمله این مراجع می‌توان به سایت سازمانی PHP به آدرس www.php.net اشاره کرد.

معرفی لیست کامل توابع پیش ساخته PHP از حوصله این کتاب خارج است و در این فصل چند نمونه از توابع مفید و کاربردی، مورد بررسی قرار گرفته است. همچنین در طی فصل‌های بعد برحسب نیاز، توابع دیگری معرفی شده و مورد استفاده قرار خواهند گرفت.

توابعی که در این فصل بررسی می‌شوند در دسته زیر قرار دارند:

- توابع کار با متغیرها
- توابع رشته‌ای
- توابع تاریخ و زمان
- توابع آرایه
- توابع ضمیمه کردن

۸-۵-۱- توابع کار با متغیرها

در فصل دوم با متغیرها آشنا شدید. در زبان PHP، توابع مختلفی برای کار با متغیرها وجود دارد که مهم‌ترین آنها در جدول زیر نشان داده شده‌اند:

نام تابع	کاربرد	مثال	توضیح مثال
gettype	برگرداندن نوع داده متغیر	<code>\$x= 34.5 ; \$type=gettype(\$x)</code>	رشته double در متغیر \$type قرار می‌گیرد.
settype	تعیین نوع داده متغیر	<code>settype (\$x , "integer");</code>	نوع داده متغیر به integer تغییر می‌کند.
isset	بررسی وجود متغیر	<code>if (isset(\$x)) \$x++; else \$x=1;</code>	اگر متغیر \$x وجود داشت، یک واحد به آن اضافه کرده در غیر این صورت ۱ را به آن نسبت می‌دهد.
empty	بررسی خالی بودن متغیر	<code>if (empty(\$name)) echo "نام وارد نشده است"</code>	اگر \$name خالی بود، پیام مناسب نشان می‌دهد.
unset	از بین بردن متغیر	<code>unset(\$num);</code>	متغیر \$num را از حافظه حذف می‌کند.
var_dump	نمایش متغیر	<code>var_dump(\$sum);</code>	مقدار و نوع داده متغیر \$sum را نشان می‌دهد.
intval	برگرداندن ارزش عددی متغیر	<code>\$str="123"; \$num = intval(\$str);</code>	مقدار عددی رشته \$str را در متغیر \$num قرار می‌دهد.

۲-۸-۵- توابع کار با رشته‌ها

توابع پیش ساخته زیادی در PHP وجود دارد. تقریباً هر پردازشی روی رشته‌ها را می‌توان با فراخوانی این توابع انجام داد. در جدول زیر لیستی از پرکاربردترین توابع رشته‌ای آمده است:

نام تابع	کاربرد	مثال	توضیح مثال
strlen	محاسبه طول رشته	<pre>\$str="ali reza"; \$len=strlen(\$str);</pre>	طول رشته یعنی عدد ۸ در متغیر \$len قرار می‌گیرد.
strpos	جست‌وجو در رشته	<pre>\$str="php is better"; \$p=strpos(\$str, "is");</pre>	موقعیت عبارت is در \$str یعنی عدد ۴ را برمی‌گرداند.
substr	برگرداندن بخشی از رشته	<pre>\$str=substr("php is better" , 4 , 2);</pre>	بخشی از رشته با شروع از موقعیت ۴ به طول ۲ کاراکتر را برمی‌گرداند. پس \$str مساوی is می‌شود.
trim	برش زدن فاصله‌ها	<pre>\$name=trim(" php ");</pre>	عبارت "php" بدون فاصله‌های دو طرف را برمی‌گرداند.
nl2br	تبدیل کاراکترهای سطر جدید به تگ br	<pre>\$text1="my name"; \$text2=nl2br(\$text1);</pre>	رشته "my name" در متغیر text2 قرار می‌گیرد.
addslashes	اضافه کردن علامت \ قبل از کاراکترهای ویژه	<pre>\$str = addslashes('What does "php" mean?'); echo(\$str);</pre>	خروجی به صورت زیر خواهد بود: What does \"php\" mean?
strtolower	تبدیل همه حروف به صورت کوچک	<pre>\$name=strtolower("HtMI")</pre>	\$name مساوی html می‌شود.
strtoupper	تبدیل همه حروف به صورت بزرگ	<pre>\$name=strtoupper("HtMI")</pre>	\$name مساوی HTML می‌شود.
explode	شکستن رشته براساس یک جداکننده و تولید آرایه	<pre>\$str="add.blade.php"; \$arr=explode('.', \$str);</pre>	هر کدام از کلمات \$str در خانه‌های آرایه \$arr قرار می‌گیرند. در نتیجه، \$arr یک آرایه با عناصر add، blade و php می‌شود.
implode	الحاق عناصر آرایه و تولید رشته	<pre>\$arr=['A', 'B', 'C']; \$str=implode('-', \$arr);</pre>	رشته \$str شامل رشته زیر می‌شود: 'A-B-C'

۳-۸-۵- توابع محاسباتی

اعمال محاسباتی مختلفی مثل محاسبه ماکزیمم، مینیمم و جذر، در ریاضیات وجود دارد. در زبان PHP برای انجام هرکدام از این اعمال، تابع مخصوصی در نظر گرفته شده است که مهم‌ترین آنها در جدول زیر نشان داده شده‌اند:

نام تابع	کاربرد	مثال	توضیح مثال
abs	محاسبه قدرمطلق عدد	<code>\$x = -50 ; \$y=abs(\$x);</code>	قدرمطلق عدد 50- یعنی 50 در متغیر \$y قرار می‌گیرد.
max	محاسبه بزرگ‌ترین عدد	<code>\$m = max(3,6,8,2); \$arr=[3,6,8,2]; \$m = max(\$arr);</code>	در هر دو بار بزرگ‌ترین عدد، یعنی 8 در متغیر \$m قرار می‌گیرد.
min	محاسبه کوچک‌ترین عدد	<code>\$m = min(3,6,8,2); \$arr=[3,6,8,2]; \$m = min(\$arr);</code>	در هر دو بار کوچک‌ترین عدد، یعنی 2 در متغیر \$m قرار می‌گیرد.
pow	محاسبه توان	<code>\$p =pow(2,10);</code>	1024 در متغیر \$p قرار می‌گیرد.
rand	تولید عدد تصادفی	<code>echo(rand()."
"); echo(rand()."
"); echo(rand(10,100));</code>	خروجی می‌تواند سه عدد تصادفی مشابه زیر باشد: 512549293 1132363175 79 در دستور فراخوانی سوم، محدوده مشخص شده است.
round	گرد کردن عدد اعشاری	<code>echo(round(5.465,2)."
"); echo(round(5.463,2));</code>	خروجی به صورت زیر است: 5.47 5.46 اگر رقم سوم بزرگ‌تر یا مساوی 5 بود به سمت بالا و در غیر این صورت به سمت پایین گرد می‌شود.
ceil	محاسبه جزء صحیح عدد اعشاری	<code>echo (ceil(5.6));</code>	خروجی به صورت زیر است: 5
sqrt	محاسبه جذر عدد	<code>echo sqrt(16);</code>	خروجی آن جذر 16 یعنی عدد 4 می‌باشد.

۴-۸-۵- توابع دستکاری آرایه‌ها

در زبان php توابع از پیش تعریف شده زیادی برای مدیریت آرایه‌ها در نظر گرفته شده است که هر کدام دارای دستورالعمل خاص و هدف مشخصی هستند.

در ادامه به معرفی برخی از این توابع که ساده‌تر و رایج‌تر هستند، پرداخته شده است:

تابع array(): برای تعریف آرایه و مقداردهی اولیه آن استفاده می‌شود که در فصل قبل از آن استفاده شد.

تابع print_r(): برای چاپ عناصر آرایه استفاده می‌شود.

تابع count(): محاسبه طول آرایه (تعداد خانه‌های آرایه)

مثال ۱: در برنامه زیر یک آرایه تعریف شده و سپس با کمک یک حلقه تکرار، عناصر آن چاپ شده است:

برنامه print-arr.php	خروجی
<code><?php</code>	2
<code>\$arr =array(2 , 56, 43 , 78 , 9);</code>	56
<code>for (\$i=0 ; \$i<count(\$arr) ; \$i++)</code>	43
<code> echo \$arr[\$i]."
";</code>	78
<code>?></code>	9

در کد فوق، با کمک تابع count، طول آرایه به دست آمده و در نوشتن شرط تکرار حلقه، از آن استفاده شده است.

زبان PHP دارای چندین تابع برای مرتب‌سازی آرایه‌های شمارشی و انجمنی می‌باشد:

تابع sort(): برای مرتب‌سازی آرایه‌های شمارشی به صورت صعودی استفاده می‌شود.

تابع rsort(): برای مرتب‌سازی آرایه‌های شمارشی به صورت نزولی استفاده می‌شود.

تابع asort(): برای مرتب‌سازی آرایه‌های انجمنی به صورت صعودی استفاده می‌شود.

تابع arsort(): برای مرتب‌سازی آرایه‌های انجمنی به صورت نزولی استفاده می‌شود.

تابع ksort(): برای مرتب‌سازی آرایه‌های انجمنی براساس کلیدها و به صورت صعودی استفاده می‌شود.

تابع krsort(): برای مرتب‌سازی آرایه‌های انجمنی براساس کلیدها و به صورت نزولی استفاده می‌شود.

مثال ۲: در برنامه زیر ابتدا عناصر آرایه براساس کلیدها به صورت نزولی مرتب شده و سپس عناصر آن نمایش

داده می‌شود:

برنامه print-arr2.php	خروجی
<code><?php</code>	Array ([sara] => 18 [javad] => 19 [ali] =>
<code>\$arr=array('ali'=>17,</code>	17)
<code> 'sara'=>18,</code>	
<code> 'javad'=>19);</code>	
<code>krsort(\$arr);</code>	
<code>print_r(\$arr);</code>	
<code>?></code>	

تابع array_keys() برای قراردادن کلیدهای آرایه انجمنی در یک آرایه جداگانه استفاده می‌شود.

مثال ۳: در برنامه زیر کلیدهای آرایه جدا شده و لیست کلیدها نمایش داده می‌شوند:

برنامه print-arr3.php	خروجی
<pre><?php \$arr =array('ali'=>17 , 'sara'=>18, 'javad'=>19); \$names = array_keys(\$arr); print_r(\$names); ?></pre>	<pre>Array ([0] => ali [1] => sara [2] => javad)</pre>

۵-۸-۵- توابع تاریخ و ساعت

در زبان PHP توابع پیش ساخته مختلفی برای دریافت تاریخ و ساعت از سیستم و عملیات دیگر روی آنها وجود دارد. در ادامه مهم‌ترین توابع تاریخ و ساعت معرفی شده‌اند.
تابع `date()` برای دریافت تاریخ از سیستم یا از یک timestamp استفاده می‌شود و شکل کلی آن به صورت زیر است:

```
date ("فرمت" , [timestamp] );
```

timestamp عددی صحیح بوده و تعداد کل ثانیه‌ها از اول ژانویه ۱۹۷۰ تاکنون می‌باشد. timestamp اختیاری است و می‌تواند وارد نشود.

برای تعیین فرمت تاریخ یا زمان درخواستی از یک مجموعه کاراکترهای ویژه به صورت زیر استفاده می‌شود:

Y: سال به صورت چهار رقمی

y: سال به صورت دو رقمی

M: نام ماه

m: شماره ماه

D: نام روز هفته

d: شماره روز

H: ساعت در فرمت ۲۴ ساعته

h: ساعت در فرمت ۱۲ ساعته

i: دقیقه

s: ثانیه

مثال ۴: برنامه زیر تاریخ و ساعت به چند فرمت مختلف نمایش داده می‌شود:

برنامه show-date.php	خروجی
<pre><?php echo date("Y/m/d")."
"; echo date("Y-m-d H:i:s")."
"; echo date("D , d M")."
"; ?></pre>	<pre>2021/04/07 2021-04-07 13:35:47 Wed , 07 Apr</pre>

تابع `time()`: زمان جاری یعنی تعداد ثانیه‌های گذشته از اول ژانویه ۱۹۷۰ را برمی‌گرداند.
مثال ۵: در برنامه زیر زمان جاری از سیستم گرفته و سپس نمایش داده شده است:

برنامه show-time.php	خروجی
<pre><?php \$t = time(); echo \$t; ?></pre>	<pre>1617803105</pre>

کد فوق زمان جاری را از سیستم دریافت می‌کند و در صورت اجرای این کد در زمان‌های مختلف، اعداد مختلفی نمایش می‌دهد.

۹-۵- توابع ضمیمه کردن فایل‌ها

یکی از امکانات خوب PHP این است که می‌توانید بخش‌هایی از کدنویسی‌ها یا طراحی‌ها را در فایل‌های جداگانه ذخیره کرده و در صفحات سایت آنها را ضمیمه کنید.

برای ضمیمه کردن فایل‌ها از توابع زیر استفاده می‌شود:

تابع `include()`: برای ضمیمه کردن یک فایل

تابع `include_once()`: برای ضمیمه کردن یک فایل به صورت یک بار

تابع `require()`: برای ضمیمه کردن یک فایل ضروری

تابع `require_once()`: برای ضمیمه کردن یک فایل ضروری به صورت یک بار

هر چهار تابع برای ضمیمه کردن استفاده می‌شوند ولی با هم تفاوت‌های مهمی دارند. اگر توابع `include()` و `include_once()` به دلایلی نتوانند فایل را ضمیمه کنند، پیغام هشدار نشان داده شده و ادامه برنامه اجرا خواهد شد ولی اگر توابع `require()` یا `require_once()` نتوانند فایل را ضمیمه کنند، خطای غیرقابل چشم‌پوشی نشان داده شده و اجرای برنامه متوقف می‌شود.

از توابع ضمیمه کردن فایل می‌توان برای استفاده از فایل‌های کتابخانه‌ای شامل توابع مفید یا فایل‌های تنظیمات در صفحات مختلف سایت استفاده کرد که اولاً با یک بار نوشتن توابع می‌توان آنها را در صفحات مختلف استفاده کرد و ثانیاً با تغییر توابع یا تنظیمات، همه صفحات از توابع یا تنظیمات جدید بهره‌مند شوند. مثال: در برنامه زیر دو تابع مفید برای محاسبه محیط و مساحت مستطیل در فایل جداگانه‌ای به نام unitlity.php قرار گرفته است.

فایل unitlity.php
<pre> <?php function mohit(\$stool , \$sarz) { \$m=2 *(\$stool + \$sarz); return \$m; } function masahat(\$stool , \$sarz) { \$m=\$stool*\$sarz; return \$m; } ?> </pre>

بعد از ایجاد فایل unitlity.php می‌توان در صفحات دیگر به جای پیاده‌سازی دوباره این توابع فایل فوق را ضمیمه کرد:

فایل rect.php	خروجی
<pre> <html> <body> <?php include('utility.php'); echo "Mohit :".mohit(13 ,4)."
"; echo "Masahat:".masahat(13,4); ?> </body> </html> </pre>	<pre> Mohit :34 Masahat:52 </pre>

وقتی در جایی از برنامه، دستور ضمیمه کردن فایل نوشته شود، در زمان اجرا این دستور از برنامه حذف شده و به جای آن یک نسخه از محتوای فایل ضمیمه شده قرار می‌گیرد. حال اگر یک فایل یکسان با تابع `include()` یا `require()` در چند جای برنامه ضمیمه شود، در زمان اجرا چند نسخه از فایل در برنامه قرار خواهد گرفت. ولی اگر بجای آنها از توابع `include_once()` یا `require_once()` استفاده شود، تنها در اولین نوبت محتوای فایل درج شده و از دفعات بعدی صرف‌نظر خواهد شد.

۱۰-۵- استفاده از فایل‌ها در طراحی صفحات

فرض کنید در حال طراحی یک وب سایت با چندین صفحه وب هستید که همه آنها دارای ظاهر یکسان بوده و دارای سرصفحه، پاصفحه و ستون کناری مشابه هستند. برای طراحی این صفحات از چه روشی استفاده می‌کنید؟ آیا پس از طراحی یکی از صفحات، دیگر صفحات را از روی آن کپی کرده و به شکل مورد نظر تغییر می‌دهید؟ یا اینکه با استفاده از توابع ضمیمه کردن فایل راه‌حل بهتری پیدا می‌کنید؟

در چنین مواقعی می‌توان طراحی قسمت‌های مشترک مثل سرصفحه، پاصفحه و ستون کناری را در فایل‌های جداگانه قرار داده و آنها را در هر صفحه سایت ضمیمه کرد.

مثال ۱: اگر همه صفحات سایت از یک منو در ستون کناری خود استفاده کنند. می‌توانید یک بار این منو را طراحی کرده، در فایل جداگانه‌ای قرار داده و در هر صفحه این فایل را ضمیمه کنید.

صفحه menu.php

```
<ul>
<li><a href="home.php">پیشخوان</a></li>
<li><a href="subs.php">موضوعات</a></li>
<li><a href="books.php">کتاب‌ها</a></li>
<li><a href="custs.php">مشتریان</a></li>
<li><a href="orders.php">سفارشات</a></li>
</ul>
```

حال در همه صفحات به عنوان مثال در صفحهٔ پیشخوان طراحی زیر را انجام دهید:

صفحه home.php

```
<html>
<head>
<style>
    body{
        direction: rtl;
    }
    .sidebar{
        width: 20%;
        float: right;
        height: 100%;
        border-left: 1px solid #999;
    }
    .content{
        float: right;
        height: 100%;
        width: 79%;
    }
</style>
</head>
<body>
<div class="sidebar" >
    <?php include('menu.php'); ?>
</div>
<div class="content">
<h1>
پیشخوان
</h1>
<hr>
<h4>
    کاربر عزیز خوش آمدید
</h4>
</div>
</body>
</html>
```


در شکل ۵-۲ اجرای صفحه home.php را مشاهده می‌کنید:



شکل ۵-۲- اجرای صفحه home.php

در صفحه home.php، از دو تگ div استفاده شده که با کمک تنظیمات CSS در کنار هم قرار گرفته‌اند. در div اول که ستون کناری صفحه را تشکیل می‌دهد، فایل menu.php ضمیمه شده و در div دوم که بدنه اصلی صفحه را تشکیل می‌دهد، عنوان صفحه و پیغام خوش‌آمدگویی نشان داده شده است. برای طراحی دیگر صفحات سایت هم می‌توان به همین صورت عمل کرده و در ستون کناری آنها فایل menu.php را ضمیمه کرد.

مثال ۲: اگر به طراحی صفحات سایت‌های اینترنتی دقت کنید، متوجه خواهید شد که همه صفحات از نظر طراحی ظاهری مشابه هستند. معمولاً همه صفحات یک سایت از یک بخش سرصفحه، پاصفحه و گاهی یک یا چند ستون کناری به صورت مشترک استفاده می‌کنند. به عنوان مثال همه صفحات وب سایت وزارت آموزش و پرورش دارای طراحی مشابه شکل ۵-۳ می‌باشند:



شکل ۵-۳- طراحی سایت وزارت آموزش و پرورش

در فصل های بعد این کتاب هم از یک قالب مشترک برای همه صفحات یک فروشگاه اینترنتی به شکل ۵-۳ استفاده شده است:



شکل ۵-۴- قالب سایت فروشگاه کتاب

برای طراحی قالب فوق ابتدا یک فایل Stylesheet به صورت زیر طراحی کنید:

فایل style1.css

```
body{
direction: rtl;
margin: 0px;
background-color:#082d2a;
}
.header{
height: 100px;
background-color:#fff;
border-bottom: 1px solid #082d2a;
}
.content{
padding: 20px;
min-height: 100px;
background-color:#fff;
display: block;
}
.footer{
text-align: center;
padding: 10px;
color:#fff;
font-weight: bold;}
```

سپس فایلی با نام دلخواه (مثلاً temp.php) به صورت شکل ۵-۵ طراحی کنید که از فایل stylesheet فوق استفاده کرده و دارای سه عدد تگ div برای ایجاد هدر، بدنه و فوتر باشد:

```

1 <html>
2 <head>
3 <title>کتابخانه کتاب</title>
4 <link href="style1.css" type="text/css"
5   rel="stylesheet"/>
6 </head>
7 <body>
8 <div class="header">
9   
10 </div>
11 <div class="content">
12
13
14 </div>
15 <div class="footer">
16   طراحی شده توسط منرجویان
17 </div>
18 </body>
19 </html>

```

شکل ۵-۵- طراحی فایل قالب

در شکل فوق، قسمت‌هایی که داخل مستطیل نشان داده شده‌اند، بخش‌های مشترک هستند و در اکثر صفحات استفاده می‌شوند.

بخش اول را در فایل جداگانه به نام header.php (یا هر نام دلخواهی) و بخش دوم را در فایل جداگانه footer.php قرار دهید.

حال برای طراحی هر کدام از صفحات سایت این دو فایل را در شروع و پایان صفحه ضمیمه کنید. به عنوان مثال برای صفحه پیشخوان می‌توانید از این دو فایل استفاده نمایید:

صفحه home.php
<pre> <?php include('header.php'); ?> <h1>پیشخوان</h1><hr> <p>کاربر عزیز خوش آمدید</p> <- ادامه طراحی صفحه -!> <?php include('footer.php'); ?> </pre>

در این فصل با روش تعریف و پیاده‌سازی و فراخوانی توابع و نیز با مجموعه‌ای توابع پیش ساخته مفید PHP آشنا شدید. پیاده‌سازی توابع مناسب باعث سازماندهی بهتر کدهای برنامه و پرهیز از نوشتن کدهای تکراری و خسته‌کننده در برنامه می‌شود. و استفاده از توابع پیش ساخته و کتابخانه‌ای لازم بوده و باعث سرعت بخشیدن به توسعه و تکمیل برنامه می‌گردد.

خودآزمایی

- ۱ انواع توابع در زبان PHP را نام ببرید.
- ۲ استفاده از پارامتر برای تابع چه مزیتی دارد؟
- ۳ برای تعریف پارامتر با مقدار پیش فرض به چه صورت عمل می‌شود؟
- ۴ فرق بین متغیرهای محلی و سراسری چیست؟
- ۵ توابع isset و unset چه کاربردی دارند؟
- ۶ هر کدام از توابع sort, asort, ksort و rsort چه کاربردی دارند؟
- ۷ تفاوت بین توابع include و require چیست؟
- ۸ برای نمایش تاریخ به صورت ۲۰۲۱/۰۴/۲۵ تابع date باید با چه پارامتری فراخوانی شود؟
- ۹ کاربرد توابع print_r و var_dump چیست؟
- ۱۰ برای قرار دادن کلیدهای یک آرایه انجمنی در آرایه جداگانه از چه تابعی استفاده می‌شود؟ با ذکر مثال نشان دهید.

سوالات چهار گزینه‌ای

- ۱ کدام گزینه در مورد توابع پیش ساخته درست است؟
(الف) از قبل در خود زبان PHP تعریف شده‌اند.
(ب) زبان PHP دارای توابع پیش ساخته خیلی زیادی است.
(ج) برای استفاده از توابع پیش ساخته کافی است فراخوانی شوند.
(د) همه موارد
- ۲ کدام گزینه در مورد توابع درست است؟
(الف) نام تابع با علامت \$ شروع می‌شود.
(ب) نام تابع می‌تواند شامل حروف، ارقام و خط زیر باشد.
(ج) همه توابع باید نتیجه را برگشت دهند.
(د) همه توابع باید پارامتر ورودی داشته باشند.
- ۳ چگونه برای تابع پارامتر ورودی اختیاری تعریف می‌کنیم؟
(الف) برای پارامتر مقدار پیش فرض تعیین می‌کنیم.
(ب) بعد از نام پارامتر علامت ? قرار می‌دهیم.
(ج) پارامتر اختیاری را در ابتدا می‌نویسیم.
(د) همه پارامترها باید الزامی باشند.
- ۴ برای تابع به صورت `func1($a, $b=0, $c=10)` کدام فراخوانی نادرست است؟
(الف) `func1()` (ب) `func1(5)` (ج) `func1(5,7)` (د) `func1(1,2,3)`
- ۵ کدام گزینه در مورد متغیرهای محلی و سراسری نادرست است؟
(الف) متغیرهای سراسری خارج از توابع تعریف می‌شوند.
(ب) متغیرهای محلی در داخل بدنه تابع تعریف می‌شوند.
(ج) متغیرهای سراسری طول عمر بیشتری نسبت به متغیرهای محلی دارند.
(د) متغیرهای سراسری به‌طورپیش فرض در داخل توابع قابل استفاده هستند.

۶ از کلمه global برای چه منظوری استفاده می‌شود؟

- الف) تعریف متغیرهای سراسری
 ب) معرفی متغیرهای سراسری در داخل تابع
 ج) تبدیل متغیرهای محلی به سراسری
 د) همه موارد صحیح است.

۷ خروجی دستور زیر کدام است؟

```
echo strpos('it is a book' , 'book')
```

- الف) 4 (الف) 8 (ب) 9 (ج) پیغام خطا (د)

۸ خروجی دستور زیر چیست؟

```
$x= 'entertainment';
```

```
echo substr( $x , strlen($x)-5 , 3 );
```

- الف) ert (الف) inm (ب) nme (ج) men (د)

۹ خروجی قطعه کد زیر چیست؟

```
$arr = array( 2 , 56, 43 , 78 , 9 );
```

```
echo $arr[ count( $arr ) - 2 ];
```

- الف) 56 (الف) 43 (ب) 78 (ج) 9 (د)

۱۰ برای ضمیمه کردن یک فایل، به گونه‌ای که فقط یک نسخه از آن در برنامه قرار گیرد و در صورتی که پیدا نشود، اجرای برنامه قطع شود، از کدام تابع استفاده می‌کنیم؟

- الف) include (الف) require (ب) include_once (ج) require_once (د)

۱۱ کدام تابع برای شکستن یک رشته و قرار دادن هر بخش آن در یک خانه از آرایه استفاده می‌شود؟

- الف) break (الف) to_array (ب) explode (ج) implode (د)

۱۲ کدام تابع برای مرتب کردن عناصر آرایه انجمنی براساس کلیدها و به صورت صعودی استفاده می‌شود؟

- الف) sort (الف) ksort (ب) asort (ج) krsort (د)

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
پاسخ	د	ب	الف	الف	د	ب	ب	ج	ج	د	ج	ب

دستور کار عملی

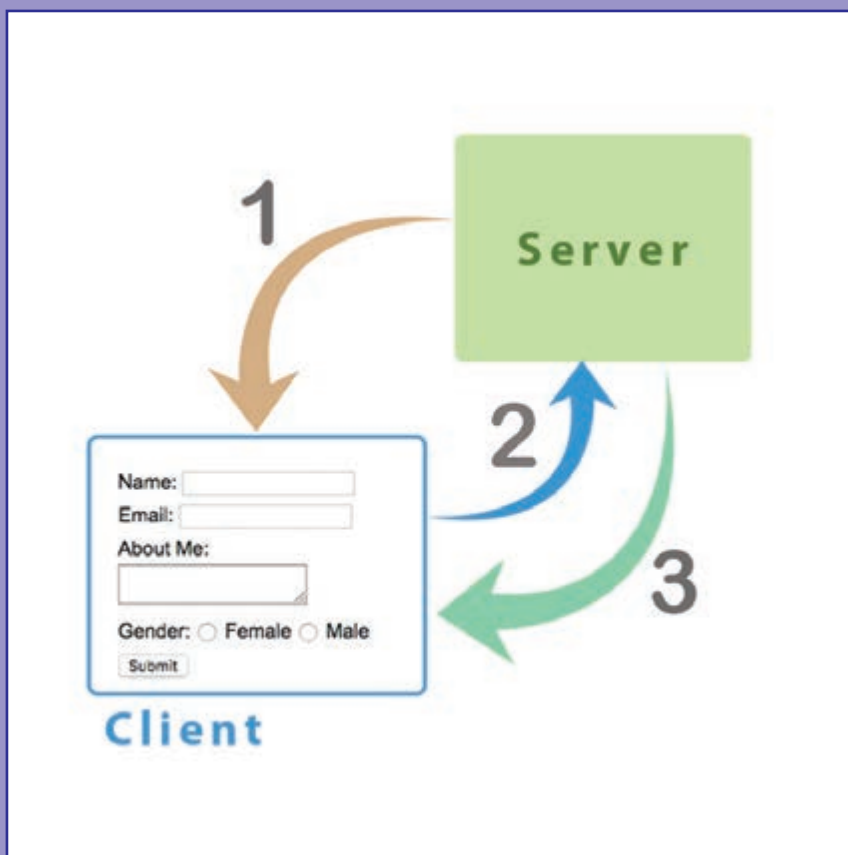
- ۱ یک صفحه وب PHP دارای یک تابع برای محاسبه ماکزیمم بین دو عدد پیاده‌سازی کنید سپس این تابع را برای مقادیر دلخواه فراخوانی کنید.
- ۲ یک صفحه php دارای یک تابع پیاده‌سازی کنید که شماره ماه را به‌عنوان پارامتر دریافت کرده و نام ماه را برگرداند. سپس این تابع را برای شماره‌های ۳ و ۹ فراخوانی کنید.
- ۳ با استفاده از تگ‌های div و stylesheet صفحه الگوی زیر را برای یک سایت طراحی کنید. سپس بخش‌هایی از آن را که باید در همه صفحات سایت یکسان باشد، در فایل‌های جداگانه (فایل‌های header.php و footer.php) قرار داده و در طراحی یک صفحه از سایت از آنها استفاده کنید.

نام سایت

بدنه صفحه
طراحی شده توسط من

فصل ششم

طراحی و پردازش فرم‌ها



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- اجزای تشکیل‌دهنده فرم‌های ورود داده را شرح دهد.
- فرم‌ها با ورودی‌های مختلف را طراحی کند.
- انواع کنترل‌های ورودی را در فرم استفاده کند.
- با استفاده از امکانات dreamweaver، فرم‌ها را طراحی کند.
- شیوه‌های ارسال فرم و ویژگی‌های آنها را شرح دهد.
- داده‌های ارسالی فرم را در php استفاده کند.

حتماً تاکنون فرم‌های وب را در سایت‌های اینترنتی مشاهده کرده‌اید که برای دریافت اطلاعات از کاربر و ارسال آن به سرور استفاده می‌شوند. این فرم‌ها دارای انواع مختلف ورودی‌ها مثل جعبه دریافت متن، لیست‌های انتخابی و دکمه‌های انتخابی هستند که همگی با استفاده از تگ‌های HTML ساخته می‌شوند. در این فصل با تگ‌های HTML برای ساخت فرم‌ها و نمایش انواع ورودی‌ها در آنها آشنا می‌شوید. همچنین فرامی‌گیرید که چگونه با استفاده از زبان PHP اطلاعات وارد شده در فرم را به‌دست آورده و پردازش کنید.

۱-۶- طراحی فرم‌ها با HTML

هر نرم‌افزاری باید بتواند داده‌ها را از کاربر دریافت کرده و مورد استفاده قرار دهد. نرم‌افزارهای تحت وب نوشته شده به زبان php هم باید این امکان را داشته باشند. این نوع نرم‌افزارها برای دریافت داده‌های کاربر از فرم‌های وب استفاده می‌کنند.

هنگام استفاده از فرم‌ها، ابتدا سرویس‌گیرنده فرم را از سرویس‌دهنده درخواست کرده و به کاربر نمایش می‌دهد. کاربر فرم را پر کرده و روی دکمه ارسال کلیک می‌کند تا سرویس‌گیرنده داده‌های فرم را به برنامه‌ای روی سرویس‌دهنده ارسال کند.

فرم‌های وب مشابه دیگر صفحات وب با استفاده از تگ‌های HTML ساخته می‌شوند. برای ساخت یک فرم از تگ‌های خاصی استفاده می‌شود که در جدول زیر نشان داده شده است:

تگ	کاربرد
form	ساخت فرم
input	ساخت شکل‌های مختلف ورودی
select	ساخت لیست‌های انتخابی
option	تعریف آیتم برای لیست‌ها
textarea	تعریف جعبه متن چند سطری

۱-۱-۶- تگ form

از تگ form برای ایجاد فرم استفاده می‌شود. در مثال زیر روش استفاده از این تگ نشان داده شده است:

```
<form action="add.php" method="post" enctype="multipart/form-data">
اجزای فرم اینجا قرار می‌گیرند
</form>
```

تگ فرم سه خاصیت خیلی مهم دارد:

خاصیت action آدرس و نام صفحه‌ای را تعیین می‌کند که اطلاعات این فرم به آن ارسال خواهد شد.
خاصیت method روش ارسال اطلاعات فرم را تعیین می‌کند. برای ارسال داده‌ها یکی از دو روش زیر استفاده می‌شود:

get: در این روش می‌توان مجموعه کوچکی از داده‌ها را به ادامه url اضافه شده و به سرورس‌دهنده ارسال کرد.
post: در این روش می‌توان داده‌های طولانی و حتی فایل‌ها را به صورت مخفی به سرورس‌دهنده ارسال کرد. معمولاً برای ارسال فرم‌ها از روش post استفاده می‌شود تا بتوان اطلاعات بیشتری را با امنیت بالاتر ارسال نمود.
خاصیت enctype نحوه کدگذاری اطلاعات در زمان ارسال فرم را تعیین می‌کند. تنها در صورتی که فرم با روش post ارسال شود کاربرد داشته و اگر فرم دارای فایل‌هایی برای بارگذاری باشد باید مقدار آن را مساوی multipart/form_data قرار داده شود.

۲-۱-۶- تگ input

از تگ input برای ایجاد کنترل‌های و روی با شکل‌های مختلف مثل جعبه متن ساده، جعبه متن کلمه عبور، دریافت تاریخ، دکمه رادیویی استفاده می‌شود.

شکل کلی استفاده از input به صورت زیر است:

```
<input name="نام ورودی" type="نوع ورودی" value="مقدار اولیه" />
```

جدول ۱-۶

کاربرد	نوع
جعبه متن یک سطری	text
جعبه متن دریافت کلمه عبور	password
دکمه رادیویی	radio
جعبه انتخاب	checkbox
انتخاب و ارسال فایل	file
نگهداری اطلاعات مخفی	hidden
دکمه فشاری	button
دکمه برای ارسال فرم به سرورس‌دهنده	submit
دکمه بازنویسی اطلاعات فرم	reset
جعبه متن دریافت عدد	number
جعبه متن انتخاب تاریخ	date

خاصیت name: مشخص‌کننده نام ورودی است، هر ورودی باید دارای یک نام یکتا باشد که بعد از ارسال فرم برای به دست آوردن مقدار ورودی از آن استفاده خواهد شد. نام می‌تواند یک کلمه یا یک عبارت باشد و توصیه می‌شود فقط از حروف و ارقام انگلیسی برای نام‌گذاری استفاده شود.

خاصیت type شکل ورودی را تعیین می‌کند. با مقداردهی آن می‌توان انواع ورودی‌ها مثل جعبه متن، دکمه ارسال را ایجاد کرد. در جدول روبه‌رو مقادیر مختلف که می‌توان برای خاصیت type استفاده کرد، نشان داده شده است.

خاصیت **value**: مقدار اولیه را تعیین می کند، برای نمایش ورودی با مقدار پیش فرض از این خاصیت استفاده می شود.

خواص دیگر تگ **input** عبارتند از:

خاصیت **size**: اندازه عرض یک فیلد ورودی به نسبت تعداد کاراکتر را تعیین می کند و برای ورودی های **text**، **password**، **submit** و **reset** کاربرد دارد.

خاصیت **value**: برای تعیین مقدار اولیه یک ورودی و یا تعیین عنوان دکمه ها استفاده می شود.

خاصیت **readonly**: برای فقط خواندنی کردن محتوای ورودی استفاده می شود. در صورت نوشتن این خاصیت کاربر اجازه تغییر آن را ندارد.

تگ زیر یک جعبه متن فقط خواندنی با نام **bid** و مقدار اولیه ۱۰۰ نمایش می دهد:

```
<input type="text" name="bid" readonly value="100" />
```

خاصیت **checked**: برای انتخاب دکمه های رادیویی و جعبه انتخاب به طور پیش فرض استفاده می شود.

در مثال زیر دو دکمه رادیویی برای انتخاب جنسیت نمایش داده شده است که جنسیت پسر به صورت پیش فرض انتخاب شده است:

Gender :

```
<input type="radio" name="gender" value="1" checked /> پسر
```

```
<input type="radio" name="gender" value="2" /> دختر
```

خاصیت **disabled**: برای غیرفعال کردن یک ورودی استفاده می شود. هنگام ارسال فرم به سرور دهنده از ارسال ورودی های غیرفعال صرف نظر می شود.

خاصیت **maxlength**: حداکثر تعداد کاراکترهای قابل دریافت در یک جعبه متن را تعیین می کند. به عنوان مثال در ورودی زیر کاربر می تواند حداکثر تا ۱۱ رقم برای شماره همراه وارد کند:

```
<input type="number" name="mobile" maxlength="11" />
```

در مثال های زیر روش ساخت انواع کنترل های ورودی نشان داده شده است:

صفحه stud-add1.php

```
<html>
<body dir="rtl">
<form action="stud-add2.php" method="post"
enctype="multipart/form-data">
<h2>افزودن دانش آموز</h2>
<hr>
نام<br>
<input type="text" name="name" /><br>
کلمه عبور<br>
<input type="password" name="pass" /><br>
```

```
<br>جنسیت  
<input type="radio" name="gen" value="1" checked />پسر<br>  
<input type="radio" name="gen" value="2" />دختر<br>  
  
<br>علاقمندی‌ها  
<input type="checkbox" name="mu" />موسیقی<br>  
<input type="checkbox" name="sp" checked />ورزش<br>  
  
<br>تصویر  
<input type="file" name="pic" /><br><br>  
  
<input type="submit" value="ارسال فرم" />  
<input type="reset" value="پاک کردن فرم" />  
  
</form>  
</body>  
</html>
```

با درخواست صفحه فوق در مرورگر، فرمی مشابه شکل ۶-۱ را مشاهده خواهیم کرد:



شکل ۶-۱- فرم افزودن دانش آموز

اگر با دقت به تگ‌ها و خصوصیات آنها در فرم قبل توجه کنید متوجه می‌شوید که:

- برای ساخت انواع مختلف ورودی‌ها از تگ `input` با `type`های مختلف استفاده می‌شود.
- از تگ `
` برای شکستن خط استفاده شده و در نتیجه ورودی‌ها زیر هم نمایش داده شده‌اند.
- جهت ساخت یک گروه دکمه‌های رادیویی برای انتخاب جنسیت از دو تگ `input` با نوع `radio` استفاده شده که دارای `name` یکسان ولی `value`های متفاوت هستند. در واقع گذاشتن `name` یکسان باعث می‌شود که هر بار تنها یکی از دکمه‌های رادیویی بتوانند انتخاب شوند.
- برای نمایش جعبه‌های انتخاب علاقمندی‌ها از چند تگ `input` با نوع `checkbox` و اسامی مختلف استفاده شده است.
- برای اینکه دکمه رادیویی جنسیت پسر یا جعبه انتخابی ورزش در علاقه‌مندی‌ها از قبل انتخاب شده باشد، از خاصیت `checked` استفاده شده است.
- برای انتخاب تصویر از تگ `input` با نوع `file` استفاده شده که هنگام نمایش فرم یک دکمه نمایش داده شده و اگر کاربر روی آن کلیک کند، کادر `open` نمایش داده شده و می‌توان فایل را انتخاب کرد.
- برای ساخت دکمه ارسال از تگ `input` با نوع `submit` استفاده شده و عنوان دکمه در خاصیت `value` نوشته شده است. اگر کاربر روی این دکمه کلیک کند، فرم به صفحه‌ای که در خاصیت `action` فرم تعیین شده فرستاده می‌شود.
- برای ساخت دکمه پاک کردن از تگ `input` با نوع `reset` استفاده شده و عنوان دکمه در خاصیت `value` نوشته شده است و اگر کاربر روی این دکمه کلیک کند، محتوای فرم پاک خواهد شد.

در ادامه تگ‌های دیگری از `HTML` را معرفی می‌کنیم که در طراحی فرم استفاده می‌شوند.

۳-۱-۶- تگ `textarea`

از این تگ برای نمایش یک جعبه متن بزرگ چند سطری استفاده می‌شود که بتوان متن‌های طولانی چند سطری مثل آدرس، متن نظر و متن خبر را در آن وارد کرد. تگ `textarea` به صورت زیر استفاده می‌شود:

```
<textarea name="نام" rows="تعداد سطرها" cols="تعداد ستون‌ها">
```

```
</textarea>
```

خاصیت `name`: نام ورودی را تعیین می‌کند. خاصیت `rows` تعداد سطرهای نمایش داده شده و به عبارت دیگر ارتفاع جعبه متن را تعیین می‌کند. خاصیت `cols` تعداد ستون‌ها یا تعداد کارکترهای یک سطر و در واقع عرض جعبه متن را تعیین می‌کند.

`textarea` دارای تگ شروع و پایان است و متن اولیه یا پیش‌فرض در بین تگ شروع و پایان نوشته می‌شود. هنگام وارد کردن متن با زدن اینتر، سطرهای جدیدی به آن اضافه شده و در صورت نیاز نوار پیمایش عمودی برای آن ظاهر خواهد شد.

مثال: ایجاد یک جعبه متن برای دریافت متن خبر با ارتفاع ۱۰ سطر و عرض ۵۰ کاراکتر به شکل زیر می‌باشد:

```
<textarea name="matn" rows="10" cols="50"></textarea>
```

۴-۱-۶- تگ‌های select و option

از این دو تگ برای نمایش لیست‌های انتخابی مثل لیست استان‌ها، رشته‌ها و دسته‌بندی‌ها استفاده می‌شود. در مثال زیر موضوعات کتاب‌ها، به صورت لیست کشویی نمایش داده شده و کاربر می‌تواند هر کدام از آنها را انتخاب کند:

```
<select name="subject">
<option value="1">داستانی</option>
<option value="2">سیاسی</option>
<option value="3">تاریخی</option>
</select>
```

ابتدا تگ select باز شده و نام آن تعیین شده است. در بدنه آن برای هر گزینه از لیست، یک تگ option باز شده مقدار (value) آن تعیین شده و بسته می‌شود و در بدنه option عنوان گزینه قرار می‌گیرد. در لیست فوق، برای هر موضوع، یک تگ option نوشته شده، کد موضوع برای مقدار و نام موضوع برای عنوان آن استفاده شده است.

لیست فوق در مرورگر، به شکل ۶-۲ نمایش داده می‌شود:



شکل ۶-۲- نمایش لیست کشویی

از طرفی لیست فوق به صورت فشرده نمایش داده می‌شود و فقط امکان انتخاب یک گزینه وجود دارد. اگر قصد نمایش لیستی دارید که ارتفاع بیشتری داشته و اجازه دهد دو یا چند گزینه هم زمان انتخاب شود، باید خاصیت size را استفاده کرده و به آن مقدار ۲ یا ۳ یا بیشتر بدهیم. هر چه size بزرگ‌تر داده شود، ارتفاع لیست بزرگ‌تر خواهد شد.

```
<select name="subject " size="2">
<option value="1">داستانی</option>
<option value="2">سیاسی</option>
<option value="3">تاریخی</option>
</select>
```



لیست فوق در مرورگر به شکل مقابل نمایش داده می‌شود:

برای هر کدام از تگ‌های option می‌توان از خاصیت selected استفاده کرد تا آن گزینه به‌طور پیش‌فرض انتخاب شده باشد. در مثال زیر موضوع سیاسی به‌طور پیش‌فرض انتخاب شده است:

```
<select name="subject">
<option value="0"> --- </option>
<option value="1">داستانی</option>
<option value="2" selected >سیاسی</option>
<option value="3">تاریخی</option>
</select>
```

مثال: نمایش عناصر یک آرایه به‌صورت لیست کشویی

فرض کنید لیستی از موضوعات در یک آرایه انجمنی ذخیره شده که کد موضوع به‌عنوان کلیدها و نام موضوع به‌عنوان مقدار آن باشد. با استفاده از تگ SELECT و تکرار تگ OPTION می‌توان لیست انتخابی شامل موضوعات را نمایش داد:

صفحه subjects.php

```
<html>
<body>
<?php
$subs=['1' => 'تاریخی',
       '2' => 'سیاسی',
       '3' => 'داستانی'];
?>
<select name="subject" >
<option value="0"> – </option>
<?php foreach($subjects as $sid=>$sname ) { ?>
<option value="<?=$sid?>">
<?=$sname?>
</option>
<?php } ?>
</select>
</body>
</html>
```

در صفحه فوق از یک حلقه تکرار foreach استفاده شده که عناصر آرایه را پیمایش می‌کند. برای هر عنصر، کلید در متغیر \$sid و مقدار در متغیر \$sname قرار گرفته و این دو برای مقدار و عنوان تگ OPTION نمایش داده می‌شوند.

توجه کنید که در زبان php برای نمایش یک مقدار یا نتیجه یک عبارت به جای نوشتن:

```
<?php echo عبارت >
```

می‌توان از :

```
<? عبارت =? >
```

استفاده کرد.

در مثال‌های فوق با روش کلی طراحی فرم و کاربرد تگ‌های html برای اضافه کردن انواع ورودی‌ها به فرم آشنا شدید. باید بدانید که برای داشتن یک فرم با طراحی زیباتر باید از تگ‌های دیگر html نیز استفاده شود. به‌عنوان مثال با استفاده از تگ‌های جدول می‌توان اجزای فرم را در یک جدول قرار داد تا نظم و ظاهر بهتری پیدا کند.

همچنین می‌توان با کمک stylesheetها، ویژگی‌های ظاهری مثل رنگ، اندازه متن، ترازبندی را به دلخواه تنظیم کرد یا با کمک کلاس‌های bootstrap می‌توان فرم‌هایی کاملاً حرفه‌ای طراحی نمود.

۲-۶- طراحی فرم‌ها در محیط dreamweaver

نرم‌افزار dreamweaver به‌شما کمک می‌کند تا به‌راحتی و بدون نوشتن تگ‌ها فرم‌های مختلف با انواع ورودی‌ها را طراحی کنید. برای این کار از گزینه‌های زیر منوی Form از منوی Insert استفاده می‌کنید. این زیرمنو دارای گزینه‌هایی برای اضافه کردن هر کدام از انواع ورودی‌های فرم مثل جعبه متن، دکمه رادیویی و لیست انتخابی می‌باشد.

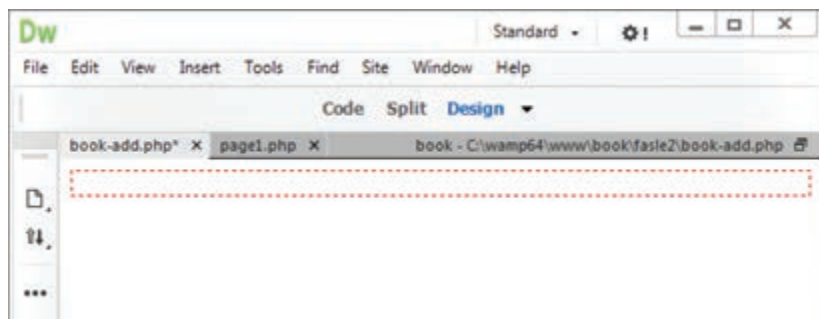
برای آموزش روش طراحی فرم در dreamweaver، یک فرم برای دریافت مشخصات کتاب، شامل نام کتاب، نوع انتشار، نام مؤلف، موضوع کتاب، توضیحات و تصویر روی جلد طراحی شده است.

جدول ۲-۶- ویژگی‌های کتاب

نام ویژگی	نوع ورودی
نام	جعبه متن
مؤلف	جعبه متن
نوع انتشار	دکمه‌های رادیویی (چاپی، فایل)
موضوع	لیست انتخابی از موضوعات (داستانی، روانشناسی، سیاسی و...)
توضیحات	جعبه متن چند سطری
تصویر روی جلد	جعبه متن انتخاب فایل

۱ ابتدا صفحه را در نمای طراحی (Design) قرار داده و در جایی از صفحه که باید فرم قرار گیرد، کلیک کنید تا مکان نما به آنجا منتقل شود.

۲ گزینه Insert|Form|Form (یعنی گزینه Form از زیرمنوی Form از منوی Insert) را اجرا کنید تا کادر خط چین قرمز رنگی روی صفحه ظاهر شود. اجزای فرم باید در داخل این کادر قرار گیرند. (شکل ۶-۳)



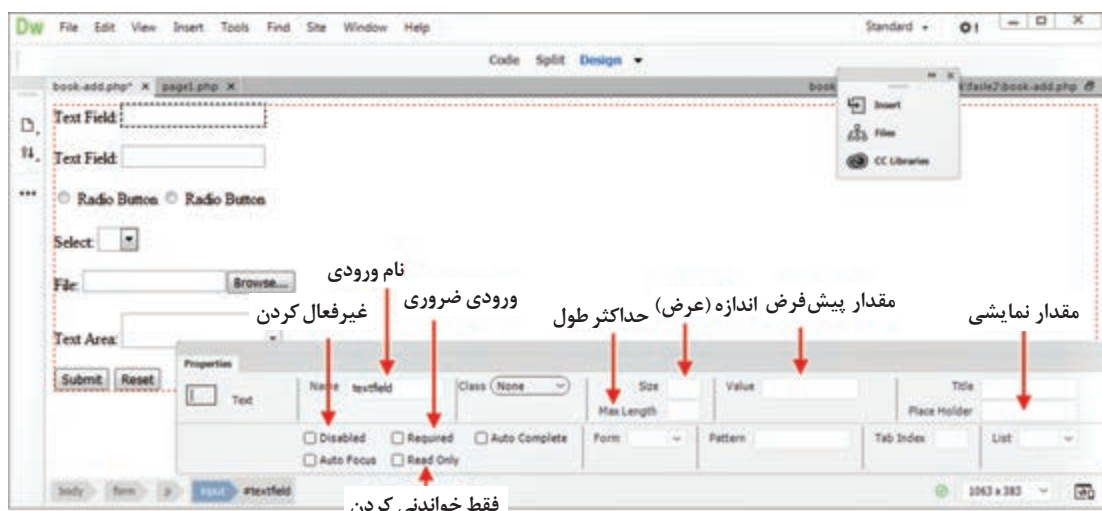
شکل ۶-۳- طراحی فرم در dreamweaver

۳ مکان نما را داخل کادر خط چین قرار داده و برای اضافه کردن هر کدام از ورودی‌های فرم، یکی از گزینه‌های زیرمنوی Insert|Form که در شکل ۶-۴ نشان داده شده را اجرا کنید.



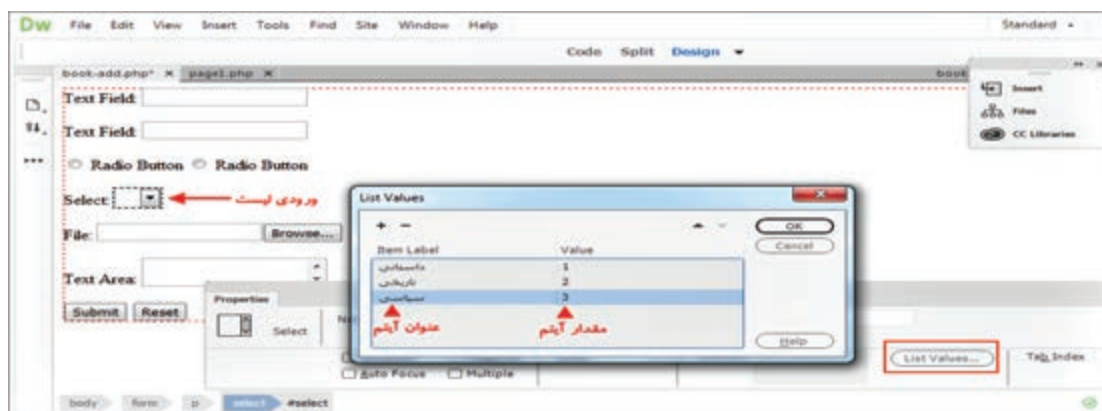
شکل ۶-۴- گزینه‌های زیرمنوی form

- ۴ با توجه به مشخصات کتاب و نوع ورودی‌های موردنظر، عناصر شکل ۵-۶ را به فرم اضافه کنید. با اضافه کردن هر عنصر، تگ‌های مربوط به آن به کد صفحه اضافه خواهند شد.
- ۵ هر ورودی یک مجموعه تنظیمات اصلی مثل نام و اندازه دارد که باید انجام شود. تنظیمات را می‌توان در نمای code و با مقداردهی خاصیت‌های تگ که در بخش‌های قبل یاد گرفتید و یا با کمک پنجره تنظیمات (properties) که در شکل ۵-۶ مشاهده می‌کنید، انجام دهید.
- ۶ برای نمایش پنجره properties می‌توانید گزینه‌ای با این نام در منوی Window اجرا کنید.



شکل ۵-۶- تنظیمات ورودی‌ها

- ۷ با انتخاب خود فرم (کادر قرمز رنگ)، و با استفاده از پنجره properties، باید ویژگی‌های action، method و enctype آن را مقداردهی کنید.
- ۸ برای اضافه کردن آیتم‌ها به ورودی لیست انتخابی (لیست انتخاب موضوعات کتاب) لیست را انتخاب کرده و در پنجره properties روی دکمه list values کلیک کنید و در کادر باز شده آیتم‌ها را مطابق شکل ۶-۶ یکی یکی اضافه نمایید.



شکل ۶-۶- تعیین آیتم‌های لیست

با استفاده از گزینه‌های زیرمنوی Insert|Form در dreamweaver می‌توانید هر عنصری به فرم اضافه کرده و با استفاده از پنجره properties می‌توانید همه تنظیمات را روی عناصر فرم انجام داد.

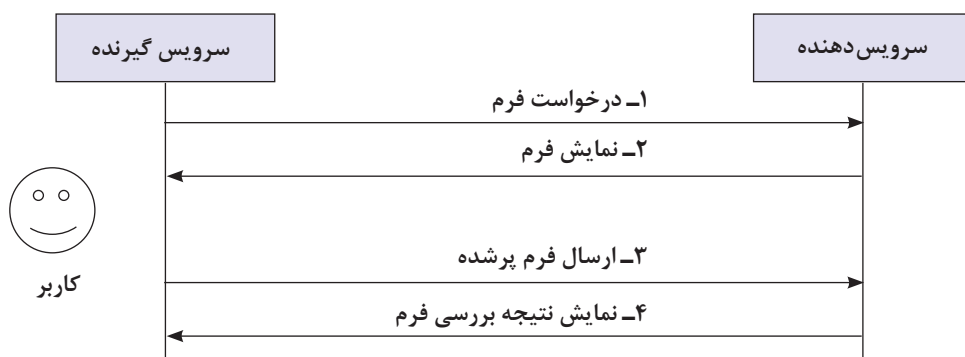
۳-۶- عملکرد کلی فرم‌ها

در برنامه‌های تحت وب برای دریافت داده‌ها از طریق فرم می‌توان از دو صفحه وب استفاده کرد:

۱ در صفحه اول تگ‌های html برای طراحی فرم قرار گیرد.

۲ در صفحه دوم دستوراتی به زبان php برای دریافت داده‌های فرم نوشته شود.

ابتدا سرویس گیرنده (مرورگر) با یک درخواست از سرویس دهنده، فرم را درخواست می‌کند و آن را به کاربر نمایش می‌دهد، کاربر فرم را پر کرده و روی دکمه ارسال کلیک می‌کند، سرویس گیرنده داده‌های فرم را طی یک درخواست جدید به صفحه دوم روی سرویس دهنده که آدرس آن در action فرم تعیین شده است ارسال می‌کند. سرویس دهنده داده‌های فرم را دریافت کرده و مورد بررسی قرار می‌دهد و نتیجه را به سرویس گیرنده برمی‌گرداند. در شکل ۶-۷ این مراحل نشان داده شده است:



شکل ۶-۷- مراحل درخواست و ارسال داده‌های فرم

کاربر ابتدا صفحه فرم را درخواست می‌کند و پس از پرکردن فرم، اطلاعات را به صفحه دوم ارسال می‌کند.

۴-۶- روش‌های ارسال فرم

در قسمت قبلی با طراحی فرم آشنا شدید. در این قسمت فرم ارسال شده را دریافت کرده و پردازش می‌کنید. برای این کار دستوراتی با زبان php می‌نویسید.

همان‌طور که پیش‌تر گفته شد، خاصیت action از تگ form، آدرس مقصد و صفحه‌ای که اطلاعات فرم به آن ارسال می‌شود را تعیین می‌کند. وقتی کاربر روی دکمه ارسال فرم کلیک می‌کند، مرورگر این صفحه را از سرویس دهنده درخواست کرده و داده‌های فرم را به همراه درخواست، به آن ارسال می‌کند.

خاصیت method از تگ form، روش ارسال فرم را تعیین می‌کند و روش ارسال فرم می‌تواند get یا post باشد.

مثال: صفحه login.html برای نمایش یک فرم ورود کاربر طراحی شده است:

صفحه login.html
<pre><form action="login.php" method="get"> نام کاربری: <input name="user" type="text" />
 کلمه عبور: <input name="pass" type="password" />
 <input type="submit" value="ورود"/> </form></pre>

فرم فوق دارای دو ورودی با نام‌های user و pass است. فرض کنید برای اولی مقدار admin و برای دومی مقدار ۱۲۳ وارد شده است. چون روش ارسال فرم، get تعیین شده است، پس هنگام ارسال فرم داده‌های فرم به ادامه url اضافه شده و به صفحه login.php ارسال می‌شوند و شکل url درخواستی به صورت زیر خواهد بود:

<http://127.0.0.1/login.php?user=admin&pass=123>

همان گونه که مشاهده می‌کنید، بعد از نام صفحه علامت سؤال (?) قرار گرفته و در ادامه برای هر ورودی، نام آن مساوی مقدار آن شده و بین ورودی‌ها علامت & نوشته شده است. این url در نوار آدرس مرورگر نمایش داده می‌شود و کاربر به راحتی می‌تواند داده‌های فرم را مشاهده کند. بنابراین نمی‌توان از روش get برای ارسال اطلاعات مهم مثل کلمه عبور استفاده کرد. همچنین از آنجا که طول url محدودیت دارد و نمی‌تواند بیشتر از حدود ۲۰۰۰ کاراکتر باشد، با روش get نمی‌توان حجم بالایی از داده‌ها را ارسال کرد.

اگر از روش post برای ارسال داده‌های فرم استفاده شود، داده‌ها با هر حجمی که دارند به صورت مخفی ارسال می‌شوند. به عنوان مثال اگر در صفحه login.html، خاصیت method مساوی post قرار گیرد، هنگام ارسال فرم، در نوار آدرس مرورگر، url زیر را مشاهده می‌کنید:

<http://127.0.0.1/login.php>

بنابراین برای ارسال فرم دارای اطلاعات مهم و پر حجم از روش post استفاده می‌کنیم.

۵-۶- دسترسی به اطلاعات فرم

در قسمت قبل توضیح داده شد که اطلاعات فرم به کمک خاصیت action به صفحه مقصد ارسال می‌شود. این صفحه می‌تواند با استفاده از آرایه‌های خاصی به داده‌های فرم دسترسی داشته باشد. اگر فرم با روش get ارسال شود، اطلاعات آن در آرایه \$_GET و اگر با روش post ارسال شود، در آرایه \$_POST قرار می‌گیرند.

دوباره فرم ورود کاربر (login.html) را در نظر بگیرید که ورودی‌های آن دارای اسامی user و pass بوده و به صفحه login.php ارسال می‌شوند. در صفحه login.php، اگر فرم با روش get ارسال شده باشد، آرایه انجمنی \$_GET به صورت زیر در دسترس می‌باشد:

\$_GET	
user	admin
pass	123

در این صورت برای به دست آوردن اطلاعات، می‌توان دستورات زیر را استفاده کرد:

```
<?php
$user = $_GET["user"];
$pass = $_GET["pass"];
?>
```

اگر فرم فوق با روش post ارسال شده باشد، آرایه انجمنی \$_POST به صورت زیر در دسترس می‌باشد:

\$_POST	
user	admin
pass	123

که در این صورت برای به دست آوردن اطلاعات، دستورات زیر را نوشته می‌شود:

```
<?php
$user = $_POST["user"];
$pass = $_POST["pass"];
?>
```

کلمات user و pass همان اسامی ورودی‌های فرم ارسال شده هستند که در اینجا به عنوان کلیدهای آرایه‌های انجمنی \$_GET و \$_POST استفاده می‌شوند. این کلیدها باید از نظر کوچکی و بزرگی حروف دقیقاً به همان شکلی استفاده شوند که در فرم نوشته شده‌اند. همچنین اسامی آرایه‌های \$_GET و \$_POST باید همیشه با حروف بزرگ نوشته شود.

مثال: صفحاتی برای دریافت طول و عرض مستطیل و محاسبه مساحت آن پیاده‌سازی کنید. برای این کار دو صفحه ایجاد کنید:

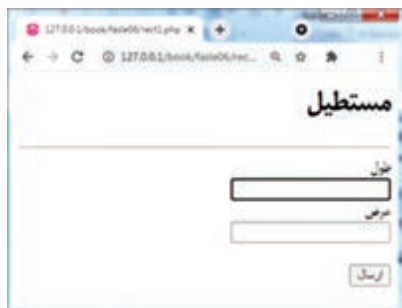
۱ صفحه اول شامل طراحی فرم (rect_form.php)

۲ صفحه دوم شامل دستوراتی برای دریافت اطلاعات فرم (rect_action.php)

ابتدا صفحه وب rect_form.php را به صورت زیر طراحی کنید:

صفحه rect_form.php

```
<html>
<body>
<form action="rect_action.php" method="get" dir="rtl">
<h1>مستطیل</h1>
<hr>
طول<br>
<input type="text" name="tool" /><br>
عرض<br>
<input type="text" name="arz" /><br><br>
<input type="submit" value="ارسال" />
</form>
</body>
</html>
```



شکل ۸-۶- اجرای rect_form.php

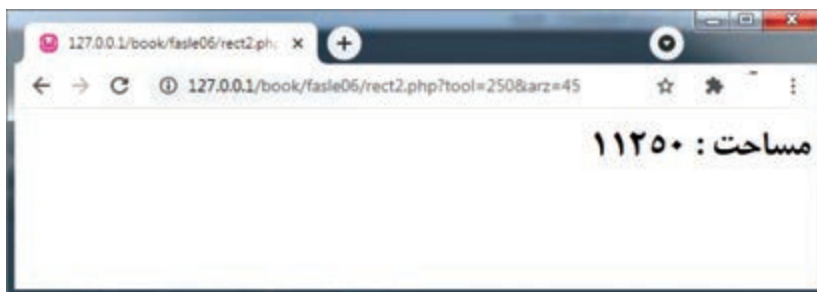
در این فرم action با مقدار rect_action.php و method با get مقداردهی شده است. بنابراین هنگام کلیک روی دکمه ارسال فایل rect_action.php درخواست شده و داده‌های فرم با روش get به آن ارسال می‌شوند. در شکل ۸-۶ نمایش صفحه rect_form.php نمایش داده شده است:

سپس فایل rect_action.php را به صورت زیر پیاده‌سازی کنید که شامل دستورات php برای دریافت اطلاعات فرم شامل طول و محاسبه مساحت مستطیل و نمایش نتیجه است:

صفحه rect_action.php

```
<html>
<body dir="rtl">
<?php
$tool = $_GET['tool'];
$arz = $_GET['arz'];
$mas = $tool * $arz ;
?>
<h2>مساحت : <?php print $mas; ?></h2>
</body>
</html>
```

در این فایل ابتدا داده‌های دریافتی از آرایه `$_GET` در متغیرهای `$tool` و `$arz` ذخیره شده و با ضرب این دو مساحت مستطیل به دست آمده و در متغیر `$mas` ذخیره می‌شود. سپس مساحت در داخل تگ `h2` نمایش داده می‌شود. خروجی صفحه `rect_action.php` را در صورتی که در فرم، اعداد `۲۲۵` و `۵۰` برای طول و عرض وارد شده باشند، در شکل ۶-۹ مشاهده می‌کنید.



شکل ۶-۹- اجرای `rect_action.php`

توجه داشته باشید که باید ابتدا صفحه `rect_form.php` را درخواست کرده و پس از کلیک روی دکمه ارسال فرم به صورت خودکار صفحه `rect_action.php` اجرا شود و داده‌ها به آن ارسال شوند. چنانچه این ترتیب را رعایت نکرده و مستقیماً `rect_action.php` را درخواست کنید، در سطرهای ۴ و ۵ آن با خطای "undefined index" مواجه شده و صفحه اجرا نخواهد شد.

در مثال فوق فرم با روش `get` ارسال شد، اگر همین فرم با روش `post` هم ارسال شود، با تغییرات جزئی در `rect_action.php` می‌توان داده‌های طول و عرض را به دست آورد. (حتماً حدس زده‌اید که در این صورت باید به جای آرایه `$_GET` از آرایه `$_POST` استفاده شود.)

۶-۶- آرایه `$_REQUEST`

آرایه `$_REQUEST` می‌تواند به جای آرایه‌های `$_GET` و `$_POST` استفاده شود یعنی به جای آرایه‌های `$_GET` و یا `$_POST` می‌توان در `rect_action.php` دستورات زیر را بنویسید:

```
$tool = $_REQUEST['tool'];
$arz = $_REQUEST['arz'];
```

در این صورت با هر شیوه‌ای که فرم ارسال شود، صفحه بدون خطا اجرا شده و داده‌ها را به دست می‌آورد. هرچند به ندرت از این آرایه استفاده می‌شود. همیشه ساده‌ترین راه حل بهترین راه حل نیست! توصیه می‌شود به عنوان تحقیق، دلیل عدم استفاده از این آرایه را پیدا کنید.

مثال: در صفحه زیر یک فرم برای دریافت روز هفته نمایش داده می‌شود. کاربر شماره روز را وارد کرده و فرم را ارسال می‌کند، این شماره در صفحه دوم استفاده شده، نام روز هفته به دست آمده و به کاربر نمایش داده می‌شود:

day-form.php

```
<html>
<body>
<form action="day-action.php" method="get" >
Enter num : <br>
<input type="text" name="num" />
<input type="submit" value="show"/>
</form>
</body>
</html>
```

day-action.php

```
<html>
<body>
<?php
$num = $_REQUEST['num'];
$name="";
switch( $num ) {
case 1: $name ='شنبه' ; break;
case 2: $name ='یکشنبه' ; break;
case 3: $name ='دوشنبه' ; break;
//..
case 7:$name='جمعه';break;
}
?>
<h1> name of day : <?=$name ?></h1>
</body>
</html>
```

۶-۷- ادغام طراحی و پردازش فرم

در هر کدام از مثال‌های فوق دو صفحه جداگانه برای نمایش فرم و پردازش اطلاعات فرم استفاده شد. دلیل استفاده از دوم صفحه مجزا، توضیح بهتر مراحل درخواست فرم و ارسال داده‌های آن بود. می‌توان به جای استفاده از دو صفحه تنها از یک صفحه استفاده کرد که شامل تگ‌های طراحی فرم و نیز دستورات پردازش فرم می‌باشد.

در صفحه زیر صفحه rect.php شامل تگ‌های نمایش فرم و نیز دستورات پردازش آن است:

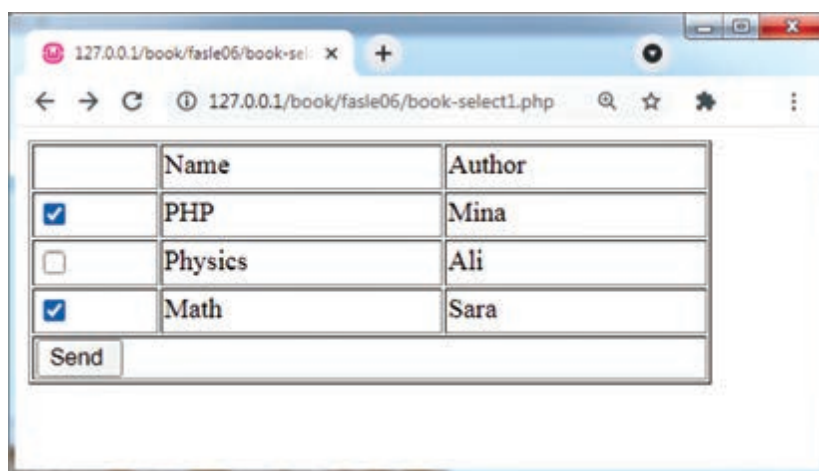
صفحه rect.php

```
<html>
<body dir="rtl">
<?php
if(isset($_POST['tool'])) {
    $tool = $_POST['tool'];
    $arz = $_POST['arz'];
    $mas = $tool * $arz ;
    echo "<h2>مساحت : $mas </h2>" ;
} else { ?>
<form action="rect.php" method="POST" dir="rtl">
<h1>مستطیل</h1>
<hr>
طول<br>
<input type="text" name="tool" /><br>
عرض<br>
<input type="text" name="arz" /><br><br>
<input type="submit" value="ارسال" />
</form>
<?php } ?>
</body>
</html>
```

در کد فوق ابتدا با استفاده از تابع `isset()` بررسی شده که آیا در آرایه `$_POST` کلیدی به نام `tool` وجود دارد یا خیر؟ اگر وجود داشت به معنی ارسال فرم می‌باشد پس دستورات مربوط به پردازش فرم اجرا می‌شود. در غیر این صورت بخش دوم صفحه اجرا شده و تگ‌های مربوط به طراحی فرم نمایش داده می‌شود.

۸-۶- استفاده از آرایه‌ها در فرم

بعضی اوقات لازم است تا لیستی طولانی از مقادیر را از طریق فرم دریافت کنید به‌عنوان مثال در یک صفحه وب که لیستی از رکوردها را نمایش می‌دهد، ممکن است نیاز به انتخاب یک یا چند رکورد جهت انجام عملیات باشد. در این صورت از یک فرم استفاده می‌شود که دارای چندین جعبه انتخاب هرکدام برای یکی از رکوردها بوده و کاربر بعد از انتخاب یک یا چند جعبه، فرم را ارسال می‌نماید. در این مواقع برای نام‌گذاری جعبه‌های انتخاب از نام آرایه استفاده می‌شود و برای پردازش این فرم هم از عملیات آرایه‌ای کمک گرفته می‌شود. مثال: در صفحه `book-select1.php` لیست کتاب‌ها نشان داده شده و کاربر می‌تواند با انتخاب هر تعداد از آنها در نهایت روی دکمه "Send" کلیک کرده و فرم را ارسال کند.



شکل ۱۰-۶- نمایش لیست جعبه انتخاب

صفحه `book-select1.php` به‌صورت زیر پیاده‌سازی شده است:

```

صفحه book_select.php
<html>
<body>
<?php
$books=array(array( 'name' =>'php','author' =>'Mina' ) ,
    array( 'name' =>'Physics','author' =>'Ali' ) ,
    array( 'name' =>'Math' , 'author' =>'Sara' ) );

?>
<form action="book_search2.php" method="post">
<table width="600" border="1">
<tr>
<td></td>

```

```

<td>Name</td>
<td>Author</td>
</tr>
<?php foreach($books as $book ){ ?>
<tr>
<td>
<input type="checkbox" name="books[]" value="<?=$book['name']?>" /></td>
<td><?=$book['name'] ?></td>
<td><?=$book['author'] ?></td>
</tr>
<?php } ?>
<tr>
<td colspan="3">
<input type="submit" value="Send" >
</td>
</tr>
</table>
</form>
</body>
</html>

```

در کد فوق، یک سطر برای جدول طراحی شده و داخل حلقه تکرار قرار گرفته تا به ازای هر کتاب تکرار شود. در هر سطر یک تگ `input` قرار داده شده که نام آن آرایه `books[]` و مقدار آن نام کتاب تعیین شده است. این فرم با روش `post` به صفحه `book-select2.php` ارسال می‌شود و آرایه‌ای با نام `$books` حاوی اسامی کتاب‌ها را به آن می‌فرستد. در زیر پیاده‌سازی صفحه `book-select2.php` را می‌بینید:

صفحه `book-select2.php`

```

<?php
$books = $_POST['books'];
var_dump($books)
?>

```

در کد فوق ابتدا خانه با کلید `books` از آرایه `$_POST`، به متغیر `$books` نسبت داده شده است. `$books` یک آرایه می‌باشد و با تابع `var_dump()` تمام خانه‌های آن چاپ شده است. خروجی صفحه `book-select2.php` به صورت زیر می‌باشد:

C:\wamp64\www\book\fasle5\page4.php:3:

```
array (size=2)
```

```
0 =>string'php'(length=3)
```

```
1 =>string'Physics'(length=7)
```

مشاهده می‌کنید که چون دو کتاب انتخاب شده بود، آرایه \$books دارای دو خانه است که نام کتاب‌ها را دارا هستند. با پیمایش این آرایه می‌توان هر استفاده‌ای از مقادیر آن داشت. در مثال فوق آرایه‌ای از جعبه‌های انتخاب ایجاد شده است. شما می‌توانید با همین روش آرایه‌ای از جعبه‌های متنی را ایجاد کرده، تا مقادیر وارد شده در جعبه‌های متنی به عنوان مقادیر آرایه ارسال شوند. در این فصل با اصول کلی دریافت داده‌های فرم‌های ساده، استفاده از آنها در محاسبات و نمایش نتایج آشنا شدید. برای استفاده از داده‌های فرم‌های پیشرفته‌تر نیاز به مهارت‌های بیشتر صفحه‌نویسی php دارید که در فصل‌های بعد، مثال‌های کامل‌تری از فرم‌ها را خواهید دید و مهارت‌های کامل‌تری کسب خواهید کرد.

خودآزمایی

- ۱ تگ‌های html برای طراحی فرم‌های وب را نام ببرید.
- ۲ خاصیت type در تگ input چه کاربردی دارد؟ شرح دهید.
- ۳ کاربرد تگ‌های select و option را با ذکر مثال نمایش دهید.
- ۴ خاصیت‌های rows و cols مربوط به کدام تگ html هستند و کاربرد آنها چیست؟
- ۵ فرایند درخواست فرم و ارسال داده‌های آن را با رسم شکل توضیح دهید؟
- ۶ دو روش ارسال داده‌های فرم را نام برده و با هم مقایسه کنید.
- ۷ برای دستیابی به داده‌های فرم در صفحه php به چه شکل عمل می‌شود؟ با مثال نمایش دهید.
- ۸ کاربرد آرایه \$_REQUEST چیست؟ توضیح دهید.

پرسش‌های چهار گزینه‌ای

- ۱ برای نمایش دکمه‌های رادیویی از کدام تگ استفاده می‌شود؟
الف) radio (الف) ب) radionbutton (ب) ج) select (ج) د) input (د)
- ۲ کدام مقدار برای خاصیت type در تگ input برای نمایش جعبه انتخابی استفاده می‌شود؟
الف) box (الف) ب) checkbox (ب) ج) select (ج) د) checked (د)
- ۳ خاصیت action از تگ form برای تعیین چه چیزی به کار می‌رود؟
الف) روش ارسال فرم به صفحه دیگر (الف) ب) نوع عمل درخواستی روی فرم (ب)
ج) آدرس و نام صفحه دریافت‌کننده فرم (ج) د) فعال نگه داشتن فرم در زمان اجرا (د)
- ۴ خاصیت method از تگ form چه کاربردی دارد؟
الف) تعیین روش ارسال فرم (الف) ب) تعیین متد پردازش فرم (ب)
ج) تعیین صفحه پردازشگر فرم (ج) د) کاربردی ندارد (د)
- ۵ تگ option چه کاربردی دارد؟
الف) نمایش لیست گزینه‌ها (الف) ب) تعریف آیتم در لیست انتخابی (ب)
ج) انتخاب پیش فرض آیتم‌ها (ج) د) نمایش دکمه انتخابی (د)

۶ اندیس‌های آرایه \$_GET چگونه تعیین می‌شوند؟

الف) با تعریف متغیرها

ب) همان اسامی ورودی‌های فرم هستند

ج) به صورت دلخواه استفاده می‌شوند

د) کلمات از پیش تعریف شده هستند

۷ کدام گزینه از تفاوت‌های روش‌های ارسال با get و post نمی‌باشد؟

الف) روش تعیین برنامه فرم پرداز

ب) حجم داده‌هایی که فرستاده می‌شوند

ج) امنیت داده‌های ارسالی

د) روش به دست آوردن داده‌ها در برنامه

۸ برای دریافت داده‌هایی که با post ارسال شده‌اند از کدام آرایه استفاده می‌شود؟

الف) \$_GET (ب) \$_POST (ج) \$_REQUEST (د) موارد ب و ج

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸
پاسخ	د	ب	ج	الف	ب	ب	الف	د

دستور کار عملی

۱ یک فرم وب برای دریافت مشخصات یک کتاب شامل عنوان، سال انتشار، موضوع، چکیده و تصویر روی جلد طراحی کنید طوری که موضوع کتاب از لیست موضوعات ۱- ریاضیات ۲- علوم تجربی ۳- علوم انسانی انتخاب شود و امکان وارد کردن چند سطر متن برای چکیده وجود داشته باشد.

۲ یک فرم وب دارای سه جعبه متن برای دریافت حقوق ثابت، تعداد ساعت اضافه کاری و حق الزحمه به ازای هر ساعت طراحی کرده و سپس صفحه‌ای بنویسید که با دریافت داده‌ها از این فرم، حقوق کل را محاسبه کرده و نمایش دهد.

(راهنمایی: حقوق کل مساوی حقوق ثابت + تعداد ساعت * حق الزحمه ساعتی است)

فصل هفتم

بارگذاری و پردازش فایل‌ها



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- فرم مناسب برای بارگذاری فایل‌ها را طراحی کند.
- فایل‌های بارگذاری شده را روی سرور ذخیره کند.
- محدودیت‌های اندازه و نوع فایل را هنگام بارگذاری اعمال کند.
- فایل تصویری بارگذاری شده را در صفحه نمایش دهد.
- محتوای فایل‌های متنی را خوانده و نمایش دهد.
- اطلاعات رکوردها را با فرمت مناسب در فایل متنی بنویسد.
- اطلاعات رکوردها را از فایل متنی خوانده و نمایش دهد.

در فصل قبل فرم‌های وب طراحی شده و مورد پردازش قرار گرفتند. در این فصل فرم‌هایی طراحی می‌شوند که قابلیت بارگذاری فایل‌ها را داشته و با استفاده از دستورات php فایل‌های بارگذاری شده در پوشه دلخواه روی سرور می‌ذخیره می‌شوند. منظور از بارگذاری فایل، انتقال فایل از درایوهای سیستم کاربر به سرور می‌دهنده و ذخیره کردن در درایوهای سرور می‌دهنده است.

همچنین در این فصل روش بازکردن فایل‌های متنی، ذخیره کردن اطلاعات در این فایل‌ها و خواندن اطلاعات از فایل‌ها را فرا می‌گیرید. با کمک این روش‌ها خواهید توانست لیست رکوردهای اطلاعاتی را با قالب مناسب در فایل ذخیره کرده و یا با خواندن این رکوردها از فایل آنها را در صفحه وب نمایش دهید.

۱-۷- طراحی فرم برای بارگذاری فایل

در این قسمت فرم‌هایی طراحی می‌شوند که به کاربر اجازه انتخاب فایل‌ها و ارسال آنها به سرور را می‌دهند. برای طراحی فرم بارگذاری فایل از تگ‌های `form` و `input` با تنظیمات مخصوص استفاده می‌شود. برای اینکه یک فرم بتواند فایل‌ها را ارسال کند به‌طور مشخص باید دو خاصیت آن به صورت زیر مقداردهی شود: خاصیت `method`: این خاصیت روش ارسال داده‌ها را تعیین می‌کند و برای ارسال فایل باید آن را با `post` مقداردهی کرد. چون با استفاده از این روش، امکان ارسال حجم بالای داده‌ها و فایل‌ها به صورت کدگذاری شده وجود دارد.

خاصیت `enctype`: این خاصیت روش کدگذاری محتوای فرم را تعیین می‌کند و برای بارگذاری فایل‌ها باید مقدار آن `multipart/form-data` تعیین شود.

مقداردهی این دو خاصیت ضروری است و اگر هر کدام از آنها مقداردهی نشود، بارگذاری فایل با مشکل مواجه می‌گردد.

برای هر کدام از ورودی‌های فایل باید یک جعبه انتخاب فایل با کمک تگ `input` از نوع `file` به فرم اضافه شود:

```
<input type="file" name="نام ورودی" >
```

مثال: در صفحه book1.php یک فرم برای تعیین مشخصات یک کتاب به همراه تصویر جلد کتاب و یک فایل شامل محتوای کتاب طراحی شده است:

```

صفحه book1.php
<html>
<body dir="rtl">
<h2>افزودن کتاب</h2>
<hr>
<form action="book2.php" method="post"
enctype="multipart/form-data">
نام کتاب:
<input type="text" name="name" />
تصویر جلد:
<input type="file" name="cover" />
متن کتاب:
<input type="file" name="content" />
<br>
<input type="submit" value="ارسال" />
<input type="reset" value="بازنویسی" />
</form>
</body>
</html>

```



فرم فوق مشابه شکل ۷-۱ در مرورگر نمایش داده شده و دکمه "browse" یا "Choose File" (بسته به نوع مرورگر) برای انتخاب دیده می‌شود:

شکل ۷-۱- فرم برای بارگذاری فایل

۷-۲- دریافت و ذخیره فایل روی سرویس دهنده

بعد از طراحی فرم باید در صفحه فرم پرداز که اطلاعات فرم را دریافت می‌کند، به کمک دستورات php اطلاعات و فایل‌های ارسال شده را شناسایی و دریافت کرده، سپس فایل‌های بارگذاری شده در پوشه موردنظر روی سرویس دهنده ذخیره شوند.

در فصل قبل با نحوه دریافت اطلاعات از فرم آشنا شدید و مشاهده کردید که اطلاعات ارسال شده با روش post در آرایه‌ای به نام \$_POST قرار داشت که در متغیرهای دلخواه گذاشته و استفاده شد. حال باید بدانید که برای نگه‌داری مشخصات فایل‌های ارسال شده از آرایه دیگری به نام \$_FILES استفاده می‌شود. این آرایه به صورت دو بعدی و دارای سطر و ستون بوده و مشخصات هر فایل در یکی از سطرهای آن قرار می‌گیرد و هر ستون آن حاوی یکی از ویژگی‌های فایل مطابق جدول زیر می‌باشد:

جدول ۷-۱- ستون‌های آرایه \$_FILES

نام ستون	کاربرد
name	نام فایل آپلود شده
size	اندازه فایل آپلود شده بر حسب بایت
type	نوع فایل آپلود شده
tmp_name	نام کپی موقتی فایل که بر روی سرویس دهنده ذخیره شده است
error	کد خطای تولید شده در حین آپلود فایل

با کمک آرایه \$_FILES می‌توان مشخصات کامل فایل را به دست آورد. مثال ۱: فرض کنید فایل با نام pic از طریق فرم ارسال شده باشد، برای به دست آوردن مشخصات آن در صفحه می‌توانید از کد زیر استفاده کنید:

به دست آوردن مشخصات فایل
<pre><?php \$name = \$_FILES['cover']['name'] ; \$type = \$_FILES['cover']['type'] ; \$size = \$_FILES['cover']['size'] ; \$tmp_name = \$_FILES['cover']['tmp_name'] ; ?></pre>

معمولاً فایل‌ها دارای اندازه و حجم بالایی هستند و نمی‌توانند به صورت یکجا به سرویس دهنده منتقل شوند و لازم است تا مرورگر وب هر فایل را به قسمت‌های کوچک تقسیم کرده و هر کدام را به صورت یک بسته به سرویس دهنده ارسال کند.

سرویس‌دهنده نیز برای دریافت فایل در پوشه tmp خود یک فایل موقت ایجاد کرده و قسمت‌های دریافتی فایل را به آن اضافه می‌کند. ستون tmp_name از آرایه \$_FILES آدرس و نام این فایل موقت را نگه‌داری می‌کند. پس از اینکه همه اطلاعات فرم و فایل‌ها دریافت شد و درخواست به صورت کامل پایان یافت، فایل‌های موقت از پوشه tmp به شکل خودکار حذف خواهند شد. دلیل این کار، جلوگیری از پر شدن فضای دیسک با فایل‌های موقت است.

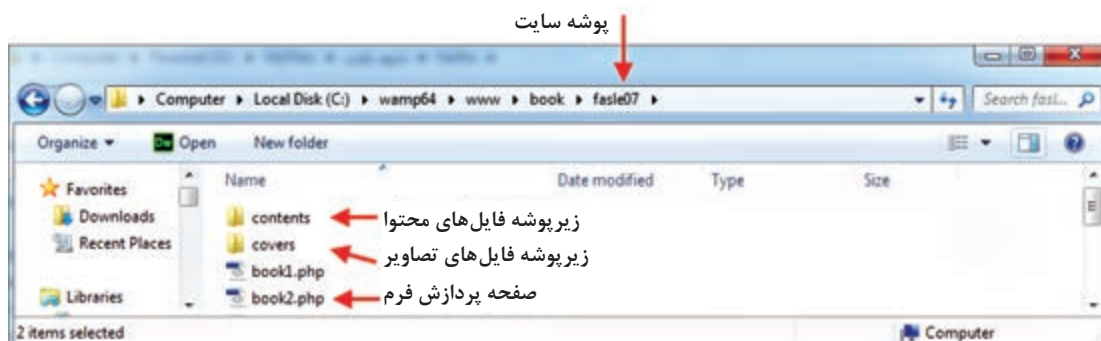
بنابراین در برنامه فرم پردازش باید علاوه بر دریافت اطلاعات فرم، فایل‌های ارسالی نیز شناسایی شده و این فایل‌ها از پوشه tmp به پوشه دائمی در داخل سایت منتقل گردند. برای انتقال فایل از پوشه موقت به پوشه دائمی از تابع مخصوصی به نام move_uploaded_file استفاده می‌شود که شکل کلی این تابع به صورت زیر است:

move_uploaded_file(مسیر و نام اصلی فایل , مسیر و نام موقت فایل);

مثال ۲: در بخش قبل فرم دریافت مشخصات کتاب شامل نام، تصویر جلد و فایل pdf محتوای کتاب را طراحی کردید که اطلاعات این فرم با توجه به خاصیت action آن به صفحه book2.php ارسال می‌شود، در این قسمت صفحه book2.php را پیاده‌سازی کنید.

قبل از پیاده‌سازی این صفحه باید محل ذخیره‌سازی فایل‌های دریافتی از فرم را مشخص کنید. به عنوان مثال می‌توانید در همان پوشه‌ای که صفحات وب قرار دارند، دو زیرپوشه با اسامی دلخواه برای نگه‌داری جداگانه فایل‌های تصویر جلد و محتوای کتاب ایجاد کنید تا از قرار گرفتن فایل‌های داده‌ای در کنار صفحات سایت و شلوغ شدن پوشه سایت جلوگیری شود.

در شکل ۲-۷، ساختار پوشه سایت و زیرپوشه‌های آن را مشاهده می‌کنید:



شکل ۲-۷- ساختار پوشه سایت

با توجه به اینکه در پیاده‌سازی صفحه book2.php، از زیرپوشه contents برای نگه‌داری فایل‌های متن کتاب و از زیرپوشه covers برای نگه‌داری فایل‌های تصویری جلد کتاب‌ها استفاده شده است، باید این پوشه‌ها قبل از اجرای صفحه ایجاد شده باشند در غیر این صورت اجرای صفحه با مشکل مواجه می‌شود.

صفحه book2.php برای دریافت اطلاعات فرم و ذخیره کردن فایل‌های دریافتی به صورت دائمی پیاده‌سازی شده است:

```
صفحه book2.php
<?php
//دست آوردن نام کتاب
$bname= $_POST['bname'] ;

//دست آوردن نام اصلی و نام موقت فایل تصویر جلد
$cv_fname= $_FILES['cover']['name'];
$cv_tname= $_FILES['cover']['tmp_name'];
move_uploaded_file($cv_tname , "covers/$cv_fname");

//دست آوردن نام اصلی و نام موقت فایل متن کتاب
$cn_fname= $_FILES['content']['name'];
$cn_tname= $_FILES['content']['tmp_name'];
move_uploaded_file($cn_tname , "contents/$cn_fname");

?>
```

در صفحه فوق برای هر فایل ابتدا نام اصلی و نام موقت فایل در متغیرها قرار گرفته و سپس تابع `move_uploaded_file` برای انتقال آن از مسیر و با نام موقت به پوشه دائمی و با نام اصلی استفاده شده است.

۳-۷- نمایش فایل‌های بارگذاری شده

بعد از بارگذاری فایل‌ها و ذخیره آنها در پوشه‌های مشخص روی سرویس‌دهنده، در صورتی که نام و آدرس آنها را بدانید، می‌توانید آنها را در داخل صفحه نمایش دهید یا می‌توانید با گذاشتن یک لینک به فایل‌های موجود، امکان دانلود کردن آنها را فراهم کنید.

مثال: در صفحه book3.php بعد از ذخیره کردن فایل‌ها با اضافه کردن تگ‌های HTML که در شبه کد زیر آورده شده است، تصویر جلد کتاب یا لینک دانلود فایل محتوای کتاب نمایش داده شده است.

```
صفحه book3.php
<?php
$bname= $_POST['bname'] ;
$cv_fname= $_FILES['cover']['name'];
$cv_tname= $_FILES['cover']['tmp_name'];
move_uploaded_file($cv_tname , "covers/$cv_fname");
```

```

$cn_fname= $_FILES['content']['name'];
$cn_tname= $_FILES['content']['tmp_name'];
move_uploaded_file($cn_tname , "contents/$cn_fname");
?>
<html>
<body dir="rtl">
<h1>نام کتاب: <?=$bname?></h1>
<h3>
<a href="contents/<?=$cn_fname?>">دانلود متن کتاب</a>
</h3>

</body>
</html>

```

۷-۴- تعیین محدودیت برای فایل‌ها

هنگام بارگذاری فایل‌ها توسط کاربران باید بررسی‌های لازم انجام شود تا فرمت فایل‌های ارسالی مناسب و مورد نیاز برنامه باشد. به‌عنوان مثال اگر قرار است کاربر تصویر جلد کتاب با حداکثر اندازه معین را بارگذاری کند، باید بررسی شود که فایل ارسال شده دارای پسوند تصویری بوده و اندازه آن از عدد موردنظر بیشتر نباشد. فرم‌های ورود اطلاعات یکی از جاهایی هستند که هرکجا با وارد کردن داده‌های غیر معمول سعی می‌کنند با گمراه کردن برنامه و ارسال اطلاعات مخرب عملکرد سایت را مختل کنند. مخصوصاً زمانی که فرم دارای ورودی فایل باشد، افراد می‌توانند با ارسال فایل‌های خطرناک و سپس اجرای آنها موفق به حذف یا تغییر اطلاعات سایت شوند. بنابراین کنترل نوع و اندازه فایل دارای اهمیت بالایی است و باید با دقت و حساسیت زیاد انجام شود.

در ادامه با تکنیک‌هایی برای ایجاد این کنترل‌ها در صفحه آشنا می‌شوید.

ساده‌ترین کار برای محدود کردن نوع فایل‌های ارسالی استفاده از خاصیت `accept` در تگ ورودی فایل از فرم می‌باشد که توسط نسخه‌های جدید مرورگرها پشتیبانی می‌شود. به‌عنوان مثال اگر این خاصیت با `image/*` مقداردهی شود، کاربر تنها امکان انتخاب فایل‌های تصویری را برای بارگذاری خواهد داشت:

```
<input type="file" name="cover" accept="image/*" />
```

مقادیر قابل استفاده برای خاصیت `accept` و توصیف هرکدام از آنها در جدول ۷-۲ نشان داده شده‌اند:

جدول ۷-۲

مقدار	توصیف
image/*	فقط انتخاب فایل‌های تصویری
video/*	فقط انتخاب فایل‌های ویدیویی
audio/*	فقط انتخاب فایل‌های صوتی
media type	فقط انتخاب فایل‌های با media type تعیین شده

media type یا mime type نوع داده فایل را نشان می‌دهد. برای آگاهی از mime type فایل‌ها می‌توانید جست‌وجوی ساده در اینترنت انجام دهید. به عنوان مثال فایل‌های با پسوند png دارای mime type مساوی image/png هستند و خاصیت accept به صورت زیر نوشته می‌شود:

```
<input type="file" name="cover" accept="image/png" />
```

همچنین می‌توان برای خاصیت accept ترکیبی از چند عبارت را استفاده کرد. در تگ زیر اجازه انتخاب فایل‌های صوتی و ویدئویی داده می‌شود:

```
<input type="file" name="cover" accept="audio/*,video/*" />
```

با توجه به اینکه خاصیت accept تنها در مرورگرهای جدید پشتیبانی می‌شود نمی‌توان به آن اکتفا کرد بلکه باید با استفاده از دستورات php در زمان دریافت فایل‌ها بررسی‌های لازم را انجام داد. با فرض اینکه نام ورودی فایل بارگذاری شده cover باشد، برای به دست آوردن اندازه فایل بنویسید:

```
$size = $_FILES['cover']['size'];
```

و برای به دست آوردن نوع فایل بنویسید:

```
$size = $_FILES['cover']['type'];
```

در دستور فوق به عنوان مثال برای فایل‌های تصویری با پسوند jpg عبارت image/jpeg و یا برای فایل متنی با پسوند txt عبارت text/plain در متغیر \$type قرار گرفته است. همچنین با پردازش رشته‌ای روی نام فایل می‌توان پسوند فایل را به دست آورده و خیلی راحت تر با پسوند‌های مورد نظر مقایسه کرد. دستورات زیر پسوند فایل را جدا کرده و در متغیر \$ext قرار می‌دهد:

```
$fname=$_FILES['cover']['name'];
```

```
$arr = explode('.', $fname);
```

```
$ext = end ($arr);
```

تابع explode برای تقسیم یک رشته به چند قسمت و قرار دادن آنها در یک آرایه استفاده می‌شود و تابع end آخرین خانه آرایه داده شده را برمی‌گرداند. بنابراین در دستورات فوق ابتدا نام آرایه براساس نقطه به چند کلمه شکسته شده و در آرایه \$arr نوشته می‌شوند و در ادامه آخرین خانه \$arr که پسوند فایل است در متغیر \$ext قرار می‌گیرد.

مثال: برای ایجاد محدودیت در فرم افزودن کتاب به صورتی که کاربر برای تصویر جلد کتاب تنها اجازه بارگذاری فایل‌های تصویری با پسوند png یا jpg با اندازه حداکثر ۱۰۰ کیلوبایت داشته باشد، می‌توان دستوراتی مطابق صفحه book4.php استفاده کرد:

صفحه book4.php

```

<?php
$name= $_POST['bname'] ;

$cv_fname= $_FILES['cover']['name'];
$cv_tname= $_FILES['cover']['tmp_name'];
$cv_fsize= $_FILES['cover']['size'];

$cn_fname= $_FILES['content']['name'];
$cn_tname= $_FILES['content']['tmp_name'];
$cn_fsize= $_FILES['content']['size'];

$errors=array();
// به دست آوردن پسوند تصویر روی جلد
$arr = explode('.', $cv_fname );
$cv_ext = end($arr);
// به دست آوردن پسوند فایل محتوا
$arr = explode('.', $cn_fname );
$cn_ext = end($arr);
// بررسی درستی پسوند و اندازه فایل‌ها
if ( $cv_ext!='jpg' && $cv_ext!='png')
    $errors[]='فرمت تصویر جلد قابل قبول نیست';
if($cv_fsize > 100*1024 )
    $errors[]='اندازه تصویر جلد بیش از حد مجاز است';

if ( $cn_ext!='pdf' )
    $errors[]='فرمت فایل محتوا قابل قبول نیست';
if( $cn_fsize > 500*1024 )
    $errors[]='اندازه فایل محتوا بیش از حد مجاز است';

if (count($errors)==0) {
    move_uploaded_file($cv_tname , "covers/$cv_fname");
    move_uploaded_file($cn_tname,"contents/$cn_fname");
} else {
    echo "<ul>";
    foreach($errors as $err)

        echo "<li> $err </li> ";
    echo "</ul>";
}
?>

```

در صفحه قبل ابتدا مشخصات فایل‌ها به دست آمده و در متغیرهای دلخواه ذخیره شده سپس پسوند فایل‌ها جدا شده است. در ادامه یک آرایه با نام دلخواه \$errors تعریف شده که برای نگه‌داری پیغام‌های خطا استفاده می‌شود. در شروع کار این آرایه خالی است و در صورتی که پسوند یا اندازه هر کدام از فایل‌ها نادرست بود، پیغام مناسب به آن اضافه می‌شود.

در پایان طول آرایه \$errors بررسی می‌شود اگر مساوی صفر بوده به معنی این است که هیچ مشکلی وجود نداشته پس فایل‌ها به پوشه‌های دائمی منتقل می‌شوند در غیر این صورت در یک لیست علامت‌دار همه خطاها نمایش داده می‌شود.

استفاده از آرایه‌ای مثل \$errors و اضافه کردن پیغام‌های خطا به آن می‌تواند به عنوان یک روش جالب در بررسی درستی داده‌های هر فرمی مورد استفاده قرار گیرد.

۵-۷- پردازش فایل‌ها

پردازش فایل‌ها اهمیت زیادی در برنامه‌های کاربردی دارد. منظور از پردازش یا دستکاری فایل‌ها بازکردن فایل، خواندن از فایل و یا نوشتن در فایل است. همچنین در زبان php توابع زیادی برای کار با فایل‌ها وجود دارد که با کمک این توابع می‌توانید مدیریت فایل‌ها مثل ایجاد، حذف، کپی، انتقال یا تغییر نام فایل‌ها را انجام داده و یا اطلاعات را در فایل نوشته و از فایل خواند.

در این قسمت عملیات مهم شامل بازکردن فایل، نوشتن در فایل متنی و خواندن از فایل متنی بیان شده و در انتها دو مثال بسیار کاربردی برای ذخیره لیستی از رکوردها در فایل و یا خواندن لیست رکوردها از فایل پیاده‌سازی شده است.

۱-۵-۷ بازکردن فایل

پیش از نوشتن دستورات پردازش فایل نیاز هست تا فایل مورد نظر به کمک تابع fopen() باز شده و به صورت یک متغیر که به آن دستگیره فایل می‌گویند، در دسترس قرار گیرد. تابع fopen() دارای دو پارامتر است که اولین پارامتر نام فایلی است که باید باز شود و دومین پارامتر تعیین می‌کند که فایل در چه حالتی و با چه نوع سطح دسترسی باز شود.

قطعه کد زیر برای بازکردن فایل books.txt موجود در فهرست جاری به صورت خواندنی نوشته شده است. اگر تابع fopen() قادر به بازکردن فایل تعیین شده نباشد، صفر (false) را برمی‌گرداند. در صفحه file_open.php، اگر تابع fopen() نتوانست فایل تعیین شده را باز کند، یک پیغام نمایش داده و اجرای برنامه قطع می‌شود.

صفحه file-open.php

```
<html>
<body>
<?php
$file=fopen("books.txt","r") or die("فایل نمی‌تواند باز شود");
?>
</body>
</html>
```

پارامتر دوم تابع `fopen()` نحوه باز شدن فایل را تعیین می‌کند. این پارامتر می‌تواند یکی از مقادیر جدول ۷-۳ با عملکرد مشخص باشد :

حالت	عملکرد
r	فقط خواندنی: در این حالت فایل از ابتدای آن و به صورت فقط خواندنی باز می‌شود و در این حالت امکان تغییر فایل برای کاربر وجود ندارد.
r+	خواندنی/نوشتنی: در این حالت فایل از ابتدای آن و به صورت فقط خواندنی و قابل ویرایش باز می‌شود و در این حالت امکان تغییر فایل برای کاربر وجود دارد.
w	فقط نوشتنی: فایل باز شده و تمام محتویات قبلی فایل پاک می‌شود، اگر فایل از قبل وجود نداشته باشد ایجاد می‌گردد.
w+	خواندنی/نوشتنی: فایل باز شده و تمام محتویات قبلی فایل پاک می‌شود، اگر فایل از قبل وجود نداشته باشد، ایجاد می‌گردد. همچنین امکان خواندن از فایل وجود دارد.
a	اضافه شدنی: فایل باز شده و اطلاعات جدید به انتهای فایل اضافه می‌شود. اگر فایل از قبل وجود نداشته باشد ایجاد می‌گردد.
a+	خواندنی/اضافه شدنی: محتوای فایل می‌تواند خوانده شود و اطلاعات جدید می‌تواند به انتهای آن اضافه گردد.
x	فقط نوشتنی: اگر فایل از قبل موجود باشد مقدار <code>false</code> برمی‌گرداند در غیر این صورت فایل جدیدی ایجاد می‌کند.
x+	خواندنی/نوشتنی: اگر فایل از قبل موجود باشد مقدار <code>false</code> برمی‌گرداند. در غیر این صورت فایل جدیدی ایجاد می‌کند.

۷-۵-۲ بستن فایل

تابع `fclose()` برای بستن یک فایل باز استفاده می‌شود. بعد از استفاده از فایل لازم است که فایل باز شده به کمک این تابع بسته شود تا آخرین تغییرات به آن منتقل شده و از لیست فایل‌های باز سیستم عامل حذف شود.

بستن فایل
<pre><?php \$fp = fopen("test.txt","r"); // دستورات پردازش فایل fclose(\$fp); ?></pre>

۷-۵-۳- نوشتن متن در فایل

برای نوشتن اطلاعات متنی در فایل از تابع `fwrite` استفاده می‌شود. در صفحه `file_write_text.php` فایل `newfile.txt` به صورت فقط نوشتنی باز شده و چند سطر متن در آن نوشته می‌شود:

صفحه `file-write-text.php`

```
<?php
$fp = fopen("myfile.txt", "w");
$txt = "Hello \r\n";
fwrite($fp, $txt);
$txt = "How Are You \r\n ";
fwrite($fp, $txt);
fclose($fp);
?>
```

۷-۵-۴- بررسی انتهای فایل

وقتی یک فایل برای خواندن باز می‌شود، اشاره‌گر فایل به ابتدای فایل اشاره می‌کند، با هر بار خواندن، اشاره‌گر به تعداد بایت‌های خوانده شده به سمت انتهای فایل انتقال می‌یابد و بالاخره بعد از خواندن همه اطلاعات به انتهای فایل می‌رسد.

تابع `feof()` بررسی می‌کند که اشاره‌گر به انتهای فایل رسیده است یا خیر و معمولاً در حلقه‌های تکرار استفاده می‌شود تا با بررسی نتیجه این تابع بعد از هر بار خواندن از فایل بتوانند رسیدن به انتهای فایل را تشخیص دهند:

```
while ( ! feof( $file ) ) {
    دستوراتی برای خواندن داده‌ها از فایل و پردازش آنها//
}
```

مثال: خواندن فایل به صورت خط به خط

تابع `fgets()` برای خواندن تنها یک خط از فایل استفاده می‌شود. بعد از هر بار فراخوانی این تابع، اشاره‌گر فایل به خط بعدی منتقل می‌گردد.

صفحه `file_read_text.php`، یک فایل را به صورت خط به خط می‌خواند تا زمانی که به انتهای فایل برسد:

صفحه `file-read-text.php`

```
<?php
$fp = fopen("myfile.txt", "r");
while(!feof($fp))
{
    echo fgets($fp). "<br />";
}
fclose($fp);
?>
```


در صفحه قبل، ابتدا فایل myfile.txt که در پوشه جاری وجود دارد به صورت خواندنی باز می‌شود. سپس تا زمانی که اشاره‌گر آن به انتهای فایل نرسیده، هر بار یک سطر از فایل خوانده شده و در صفحه نمایش داده می‌شود.

۶-۷- خواندن و نوشتن رکوردها در فایل

گاهی نیاز است که لیستی از رکوردها را در فایل ذخیره کرده یا لیستی از رکوردها از فایل خوانده شود. به عنوان مثال فرض کنید کاربر از طریق یک فرم اطلاعاتی، فایلی که حاوی چندین رکورد است را بارگذاری می‌کند. این فایل ممکن است از برنامه‌ای مثل excel گرفته شده باشد. نرم‌افزار باید بتواند رکوردهای فایل را خوانده و نمایش داده و یا در بانک اطلاعاتی ذخیره کند. گاهی اوقات هم نرم‌افزار باید یک فایل متنی حاوی چند رکورد را ایجاد کند.

توجه داشته باشید که اکثر برنامه‌های کاربردی قابلیت صدور اطلاعات رکوردها به صورت فایل‌های متنی و یا دریافت فایل‌های متنی و پردازش آن و همچنین استخراج رکوردها را دارند.

برای ذخیره کردن چند رکورد در فایل باید از یک قالب مناسب استفاده شود تا هنگام خواندن از فایل، رکوردها از هم تفکیک شده و فیلدهای هر کدام استخراج گردد. معمولاً فیلدهای رکورد را با استفاده از یک جداکننده مثل ویرگول (,) و یا سمیکالن (;) و یا هر علامت دیگری کنار هم قرار داده و آن را به صورت یک سطر در فایل متنی می‌نویسند. این فایل دارای قالب CSU می‌باشد.

مثال ۱: نوشتن آرایه‌ای از رکوردها در فایل متنی

فرض کنید لیستی از دانش‌آموزان به صورت جدول زیر موجود باشد:

Name	family	average
Ali	amini	17.5
sara	ahmadi	18
mitra	akbari	19.25

می‌توان این اطلاعات را با استفاده از جدا کننده ویرگول به صورت زیر در فایل ذخیره کرد:

```
name,family,average
ali,amini,17.5
sara,ahmadi,18
mitra,akbari,19.25
```

در صفحه زیر اطلاعات دانش‌آموزان که در آرایه \$studs قرار دارد، در فایل متنی allstuds.txt ذخیره شده است:

صفحه file-write-records.php

```
<?php
$studs=array(
array('name'=>'ali','family'=>'amini','avg'=>17.5),
array('name'=>'sara','family'=>'ahmadi','avg'=>18),
array('name'=>'mitra','family'=>'akbari','avg'=>19.25)
);
$fp = fopen('allstuds.txt','w');
$line = "name,family,average\r\n";
fwrite($fp, $line);
foreach($studs as $row )
{
    $line=implode(',',$row);
    $line.="\r\n";
    fwrite($fp, $line);
}
fclose($fp);
?>
```

در صفحه فوق برای تعیین انتهای خط از کاراکترهای ویژه `\r\n` استفاده شده و از تابع `implode` برای به هم چسباندن خانه‌های آرایه استفاده گردیده است. در ورودی این تابع ابتدا کاراکتر جداکننده و سپس نام آرایه نوشته می‌شود.

مثال ۲: خواندن رکوردها از فایل متنی

در مثال ۱ لیستی از رکوردهای دانش‌آموزان در فایل متنی `allstuds.txt` ذخیره شد. حال برای خواندن همین فایل و نمایش مشخصات دانش‌آموزان در یک جدول HTML می‌توان از کد زیر استفاده کرد:

صفحه file-read-records.php

```
<html>
<body>
<table width="600" align="center" border="1">
<?php
$fp = fopen("allstuds.txt","r");
while( ! feof($fp )){
    $line= fgets($fp);
    $row = explode(',',$line);
```

```

?>
<tr>
<td><?=$row[0] ?></td>
<td><?=$row[1] ?></td>
<td><?=$row[2] ?></td>
</tr>
<?php } ?>
</table>
</body>
</html>

```

تابع `explode` همان‌گونه که پیش‌تر کاربرد آن توضیح داده شده برای تقسیم سطر خوانده شده از فایل به چند بخش براساس جداکننده ویرگول استفاده می‌شود.

در این صفحه با استفاده از دستورات `php` فایل `allstuds.txt` برای خواندن باز شده است. سپس در یک حلقه تکرار تا زمانی که اشاره‌گر به انتهای فایل نرسیده باشد، هر بار یک سطر از فایل را خوانده و در متغیر `$line` قرار می‌دهد.

سطر خوانده شده شامل اطلاعات یک رکورد است که فیلدهای آن با علامت کاما (,) از هم جدا شده اند پس با استفاده تابع `explode` فیلدهای آن از هم جدا شده و در آرایه `$row` قرار می‌گیرند و در ادامه هر کدام از عناصر آرایه `$row` در یکی از سلول‌های جدول نمایش داده می‌شوند. نتیجه اجرای صفحه فوق را در شکل ۷-۳ مشاهده می‌کنید:



The screenshot shows a web browser window with the URL `127.0.0.1/book/fasle07/file-read-records.php`. The browser displays a table with the following data:

name	family	average
ali	amini	۱۷.۵
sara	ahmadi	۱۸
mitra	akbari	۱۹.۲۵

شکل ۷-۳- نمایش لیست رکوردهای خوانده شده از فایل متنی

در این فصل با ارسال فرم‌های حاوی فایل و دریافت و ذخیره‌سازی فایل‌ها در پوشه سایت آشنا شدید، توانستید با استفاده از توابع پردازش فایل، فایل‌ها را برای خواندن یا نوشتن باز کرده و اطلاعات متنی را در آنها نوشته یا از آنها بخوانید. همچنین با دو نمونه برنامه مفید برای صدور رکوردهای داده‌ای به فرمت `CSV` و خواندن فایل‌های `CSV` و تشخیص رکوردها و نمایش آنها آشنا شدید.

خودآزمایی

- ۱ کدام خاصیت‌ها از تگ فرم برای بارگذاری فایل‌ها باید مقداردهی شوند؟ و چه مقادیری می‌گیرند؟
- ۲ با استفاده از تگ‌های `html`، یک صفحه دارای فرم مناسب برای دریافت نام و تصویر دانش‌آموز طراحی کنید.
- ۳ یک صفحه برای دریافت اطلاعات فرم شامل نام و تصویر دانش‌آموز و نمایش مجدد نام و تصویر طراحی و کدنویسی کنید.
- ۴ فرض کنید فایلی از طریق ورودی با نام `attach` از فرم دریافت شده است. دستوراتی بنویسید که پسوند فایل را جدا کرده و با کلمات `pdf` و `doc` مقایسه کند و اگر غیر از این دو بود پیغام خطا نمایش دهد.
- ۵ یک دستور برای باز کردن فایل با نام `test.txt` به صورت اضافه شدن و امکان خواندن از آن بنویسید.
- ۶ قالب `csv` برای فایل‌ها چه کاربردی دارد؟ توضیح دهید.

سوالات چهارگزینه‌ای

- ۱ منظور از بارگذاری فایل چیست؟
الف) باز کردن فایل در مرورگر
ب) ارسال فایل به سرور
ج) دریافت فایل از سرور
د) همه موارد
- ۲ مقدار `multipart/form_data` برای کدام خاصیت فرم استفاده می‌شود؟
الف) `action` (ب) `method` (ج) `type` (د) `enctype`
ب) خاصیت `method` در فرم‌های بارگذاری فایل باید چه مقداری داشته باشد؟
الف) `get` (ب) `file` (ج) `post` (د) `upload`
- ۳ چه اطلاعاتی در آرایه `$_FILES` قرار داده می‌شود؟
الف) اطلاعات فرم ارسالی
ب) مشخصات فایل‌های ارسالی
ج) داده‌های فایل‌های ارسالی
د) همه موارد
- ۴ کدام دستور زیر برای به دست آوردن آدرس و نام موقت فایل ارسال شده از طریق ورودی `p` درست است؟
الف) `$p=$_FILES['tmp_name']` (ب) `$t=$_FILES['p']['tmp_name']`
ج) `$name=$p['tmp_name']` (د) `$name=$_POST['p']['tmp_name']`
- ۵ با یک بار اجرای تابع `move_uploaded_file` می‌توان
الف) یک فایل را از محل موقت به محل دائمی منتقل کرد.
ب) همه فایل‌ها را از محل موقت به محل دائمی منتقل کرد.
ج) بررسی کرد فایل‌ها آپلود شده‌اند یا خیر.
د) فایل‌ها را از یک پوشه به پوشه دیگر منتقل کرد.

۷ بعد از اجرای دستورات زیر محتوای متغیر \$x چه خواهد بود؟

```
$fname="how.are.you";
$arr = explode('.', $fname);
$x = end($arr);
```

الف (how) ب (are) ج (you) د (how are you)

۸ در دستور زیر، کدام مقدار برای متغیر \$mode، اگر فایل از قبل موجود باشد، محتوای آن را پاک می‌کند؟

```
$fp=fopen("aaa.txt" , $mode )
```

الف (r) ب (r+) ج (w) د (a)

۹ تابع fgets برای چه کاری استفاده می‌شود؟

الف (خواندن کل متن فایل) ب (خواندن یک سطر از فایل)
ج (خواندن یک کلمه از فایل) د (خواندن یک کاراکتر از فایل)

۱۰ برای بررسی اینکه ببینیم اشاره‌گر فایل به انتهای آن رسیده یا خیر از کدام تابع استفاده می‌شود؟

الف (end) ب (check) ج (eof) د (feof)

۱۱ تابع explode برای چه منظوری به کار می‌رود؟

الف (جداکردن بخشی از متن) ب (جدا کردن چند خانه از آرایه)
ج (تقسیم رشته به چند قسمت) د (به هم چسباندن عناصر آرایه)

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱
پاسخ	ب	د	ج	ب	د	الف	ج	ج	ب	د	ج

دستور کار عملی

۱ یک صفحه شامل فرم افزودن دانش‌آموز شامل نام، نام خانوادگی و تصویر شخصی طراحی کنید. سپس

صفحه دوم را برای دریافت این مشخصات از فرم و نمایش آنها در صفحه طراحی و کدنویسی نمایید.

۲ در پردازش فرم مشخصات دانش‌آموز در سؤال قبل، دستوراتی اضافه کنید تا با بررسی پسوند فایل ارسال شده اجازه آپلود فایل‌های با پسوند غیر از jpg، png و gif را ندهد.

۳ یک آرایه دو بعدی دارای مشخصات سه کتاب شامل نام، مؤلف و قیمت تعریف کنید. سپس دستوراتی برای ذخیره کردن اطلاعات کتاب‌ها در یک فایل با فرمت CSV بنویسید.

۴ صفحه‌ای برای خواندن مشخصات کتاب‌ها از فایل CSV (ایجاد شده در سؤال قبل) و نمایش آنها در یک جدول بنویسید.

۵ صفحه‌ای طراحی کنید که دارای یک فرم برای بارگذاری فایل با قالب txt باشد. سپس دستوراتی برای خواندن محتوای فایل بارگذاری شده، استخراج مشخصات و نمایش آنها در یک جدول بنویسید.



فصل هشتم

آشنایی با بانک اطلاعاتی MySQL



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- بانک اطلاعاتی و اجزای آن را تعریف کند.
 - برای یک محیط عملیاتی، موجودیت‌ها، ویژگی‌ها و روابط آنها را شناسایی کند.
 - برای یک سیستم، بانک اطلاعاتی مناسب طراحی کند.
 - ویژگی‌های بانک اطلاعاتی MySQL را بیان کند.
 - دستورات اصلی زبان SQL و ساختار کلی آنها را بنویسد.
 - برای درخواست موردنظر از بانک اطلاعاتی، دستور SQL مناسب را بنویسد.
- در فصل‌های قبلی با مفاهیم پایه برنامه‌نویسی مبتنی بر وب به زبان PHP آشنا شدید. می‌دانید که PHP یک زبان برنامه‌نویسی کاربردی مبتنی بر وب است و برنامه‌های کاربردی تحت وب اغلب با حجم زیادی از اطلاعات مثل اخبار، محتوای آموزشی، محصولات و مشتریان سروکار داشته و برای نگهداری آنها از بانک‌های اطلاعاتی استفاده می‌کنند.
- در این فصل به آموزش مفاهیم ضروری برای طراحی و پیاده‌سازی بانک اطلاعاتی و دستورات مهم زبان SQL برای کار بانک اطلاعاتی پرداخته شده است.

۸-۱- سیستم‌های مدیریت بانک اطلاعاتی

در هر محیط عملیاتی حجم زیادی اطلاعات وجود دارد که باید به صورت دائمی نگهداری شده و روی آنها کار شود. به عنوان مثال در یک محیط آموزشی مثل مدرسه اطلاعات معلمان، دانش‌آموزان، درس‌ها و نمرات باید نگهداری شود که برای آنها از بانک اطلاعاتی استفاده می‌شود.

به طور کلی بانک اطلاعاتی یا پایگاه داده مجموعه‌ای از رکوردهای ذخیره شده دارای ساختار مناسب در فایل‌ها است که امکان اجرای اعمال زیر را فراهم کند:

۱ ساخت، اصلاح و حذف فایل‌ها

۲ اضافه، حذف، ویرایش و بازیابی رکوردها

انواع مختلفی از بانک‌های اطلاعاتی وجود دارد که رایج‌ترین آنها بانک‌های اطلاعاتی رابطه‌ای هستند. یک بانک اطلاعاتی رابطه‌ای داده‌ها را به صورت جدول‌های مرتبط به هم نمایش می‌دهد به گونه‌ای که هر رکورد به صورت یک سطر از جدول نمایش داده شده و فیلدهای رکورد خانه‌ها یا ستون‌های آن سطر هستند.

نرم‌افزارهای مختلفی وجود دارند که امکان ایجاد بانک‌های اطلاعاتی رابطه‌ای، ایجاد جداول و اجزای دیگر بانک اطلاعاتی و مدیریت سطرها را فراهم می‌کنند. به این نرم‌افزارها سیستم‌های مدیریت بانک اطلاعاتی گفته می‌شود و می‌توانند نقش سرویس‌دهنده داده را در سیستم‌های تحت وب داشته باشند. نمونه‌هایی از این نرم‌افزارها عبارت‌اند از:

■ **Access**: یکی از نرم‌افزارهای بسته آفیس متعلق به شرکت مایکروسافت است.

■ **SQL Server**: محصول شرکت مایکروسافت است و دارای قدرت و امکانات بیشتری نسبت به Access می‌باشد.

■ **MySQL**: نرم‌افزاری open source یا متن باز و رایگان است که دارای نسخه‌های مختلف برای نصب در محیط‌های مختلف از جمله ویندوز و لینوکس می‌باشد.

زبان PHP قابلیت ارتباط با انواع مختلف سیستم‌های مدیریت بانک‌های اطلاعاتی را دارد، با این حال استفاده از بانک اطلاعاتی MySQL در PHP خیلی معمول است زیرا هر دو رایگان هستند و می‌توانند در سیستم‌های مختلف از جمله لینوکس اجرا شوند. در این درس بانک اطلاعاتی MySQL مورد استفاده قرار می‌گیرد. هر کدام از سیستم‌های بانک اطلاعاتی می‌توانند درخواست‌های مربوط به داده‌ها را از کاربر یا از برنامه‌های دیگر دریافت کرده، روی داده‌ها اجرا کرده و نتیجه را برگردانند. این درخواست‌ها را با زبان مشخصی دریافت می‌کنند که هر سه نرم‌افزار مذکور از زبان SQL برای دریافت دستورات استفاده می‌کنند.

۸-۲- طراحی بانک اطلاعاتی

یکی از مهم‌ترین کارهایی که باید قبل از شروع طراحی و برنامه‌نویسی هر سیستم نرم‌افزاری انجام شود، شناسایی، تحلیل و بررسی داده‌ها و طراحی جداول بانک اطلاعاتی برای آن است. طراحی بانک اطلاعاتی برای یک نرم‌افزار مشابه بتن‌ریزی برای بناکردن یک ساختمان بزرگ است. همان‌طور که یک ساختمان مستحکم باید بتن‌ریزی مستحکم و اصولی داشته باشد، یک نرم‌افزار قوی هم باید طراحی بانک اطلاعاتی استاندارد و قوی داشته و نیازهای بعدی سازمان و توسعه نرم‌افزار در آن پیش‌بینی شود.

برای طراحی بانک اطلاعاتی مربوط به یک نرم‌افزار باید با حضور در محیط سازمان و مشاهده روش انجام کارها و مصاحبه با کارکنان با تجربه سازمان، اطلاعاتی که باید در بانک اطلاعات نگه‌داری شوند مثل اطلاعات افراد، نقش‌ها، مکان‌ها، دستگاه‌ها، عملیات، وقایع، گزارشات و دیگر اطلاعات مورد نیاز را شناسایی و تجزیه و تحلیل کرد.

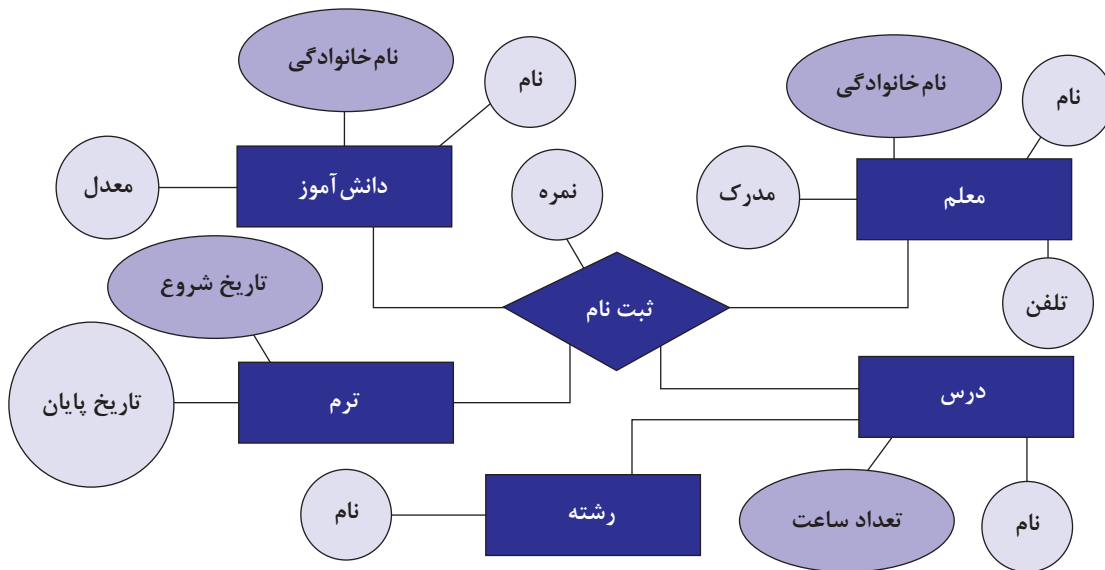
در مرحله تحلیل داده‌ها باید موجودیت‌ها مورد شناسایی و بررسی قرار گیرند، هر چیزی که باید در مورد آن داده‌هایی نگه‌داری شود را اصطلاحاً موجودیت می‌گویند و هر موجودیت دارای یک مجموعه ویژگی است که باید تعیین شود و نیز ارتباط بین موجودیت‌ها شناسایی شده و در طراحی بانک اطلاعاتی در نظر گرفته شود. به عنوان مثال در محیط آموزشی موجودیت‌ها می‌توانند معلم، دانش‌آموز، درس و ترم باشد.

- موجودیت معلم دارای ویژگی‌های کد پرسنلی، نام، نام خانوادگی، مدرک تحصیلی و تلفن است.
- موجودیت دانش‌آموز دارای ویژگی‌های نام، نام خانوادگی و معدل است.
- موجودیت درس دارای ویژگی‌های نام درس و تعداد ساعت است.
- موجودیت ترم دارای ویژگی‌های عنوان، تاریخ شروع و تاریخ پایان است.

از طرفی موجودیت‌ها با هم رابطه‌های زیر را دارند:

- درس توسط معلم تدریس می‌شود.
- درس‌ها توسط دانش‌آموزان اخذ می‌شوند.
- هر درس در یک ترم ارائه می‌شود.

برای داشتن تصویر خواناتر و فهم بهتر داده‌ها، می‌توان از نمودار ER مشابه شکل ۸-۱ برای نمایش موجودیت‌ها و ارتباط بین آنها استفاده کرد (سرنام ER از Entity-Relation گرفته شده که به معنی موجودیت - رابطه می‌باشد).



شکل ۸-۱- نمودار ER آموزشگاه

همان طور که در نمودار فوق می بینید، موجودیت‌ها با مستطیل و ویژگی‌های آنها با دایره یا بیضی نشان داده شده است و ارتباط بین موجودیت‌ها به صورت خطوطی نشان داده شده‌اند که دو موجودیت را به هم وصل می‌کنند. بعد از تحلیل و بررسی داده‌ها و به دست آمدن نمودار بالا، نوبت به پیاده‌سازی بانک اطلاعاتی می‌رسد. برای این کار نکات زیر را در نظر داشته باشید:

برای هر کدام از موجودیت‌ها یک جدول در بانک اطلاعاتی ایجاد می‌شود که ستون‌های آن همان ویژگی‌های موجودیت می‌باشد. به علاوه یک ستون اضافی به جدول اضافه می‌شود که شناسه سطر نامیده شده و به عنوان کلید اصلی استفاده خواهد شد. کلید اصلی همیشه دارای مقدار بوده و مقدار آن هم یکتا است، یعنی برای هیچ دو سطری از جدول مقدار آن تکرار نمی‌شود. به عنوان مثال برای موجودیت استاد، یک جدول با ستون‌های شناسه استاد، نام، نام خانوادگی و مدرک ایجاد می‌شود.

■ هر کدام از ویژگی‌های موجودیت‌ها دارای نوع داده مشخصی هستند. به عنوان مثال نام و نام خانوادگی دانش آموز از نوع رشته‌ای است یا معدل دانش آموز از نوع اعشاری است. هنگام ایجاد جدول باید نوع داده مناسب برای آنها تعیین شود.

■ برای هر کدام از فیلدها باید یک نام لاتین در نظر گرفته شود. به عنوان مثال برای فیلد نام دانش آموز از نام لاتین name و برای نام خانوادگی از نام لاتین family می‌توانیم استفاده کنیم. بهتر است نام فیلد مناسب با ماهیت فیلد مورد نظر باشد.

■ برای پیاده‌سازی رابطه‌ها بسته به نوع رابطه می‌توان ستون‌هایی در جداول اضافه کرد تا نشان‌دهنده رابطه بین آنها باشند یا جدول جداگانه‌ای برای رابطه ایجاد کرد. به عنوان مثال اگر هر درس مربوط به یک رشته مشخص باشد، کافی است شناسه رشته را برای هر کدام از سطرهای جدول دروس ذخیره کنیم و چون هر درس توسط چند دانش آموز انتخاب شده و نیز هر دانش آموز چند درس را اخذ می‌کند، باید جدول جداگانه‌ای برای نگه‌داری شناسه دانش آموز، شناسه درس و نمره دانش آموز داشته باشیم.

با توجه به بررسی‌های صفحه قبل می‌توان جداول بانک اطلاعاتی آموزشگاه را به صورت زیر تعیین کرد:

جدول ۸-۱ - teachers - برای نگهداری مشخصات معلمان

نام ویژگی	نام فیلد	نوع داده
شناسه معلم	Tid	عددی
نام	name	رشته‌ای
نام خانوادگی	family	رشته‌ای
تلفن	phone	رشته‌ای
مدرک	grade	رشته‌ای

جدول ۸-۲ - students - برای نگهداری مشخصات دانش‌آموزان

نام ویژگی	نام فیلد	نوع داده
شناسه دانش‌آموز	sid	عددی
نام	name	رشته‌ای
نام خانوادگی	family	رشته‌ای
معدل	average	اعشاری

جدول ۸-۳ - terms - برای نگهداری مشخصات ترم‌ها

نام ویژگی	نام فیلد	نوع داده
شناسه ترم	mid	عددی
تاریخ شروع	sdate	تاریخ
تاریخ پایان	edate	تاریخ

جدول ۸-۴ - courses - برای نگهداری مشخصات درس‌ها

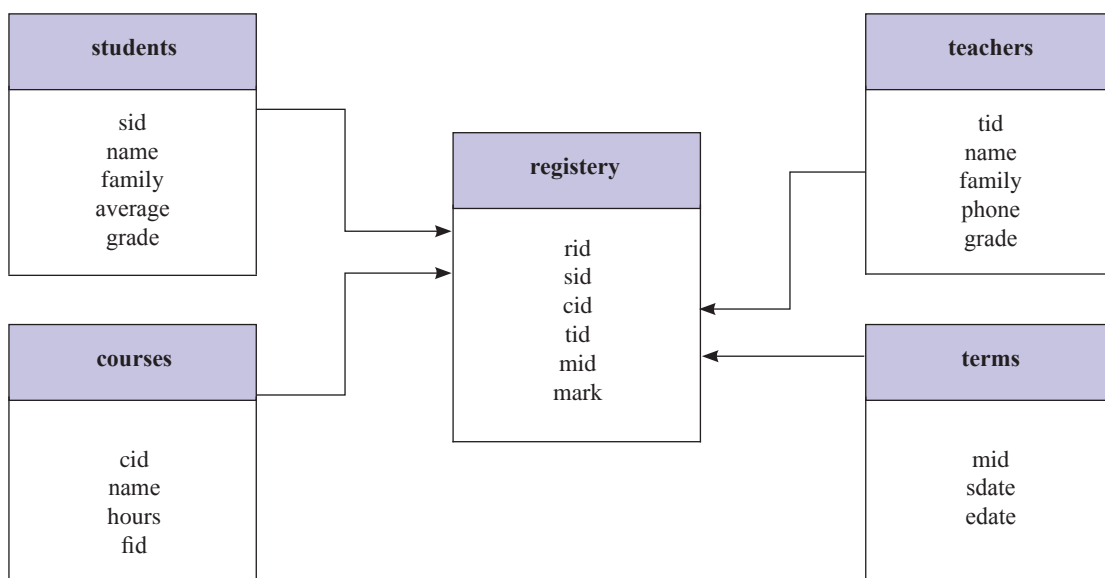
نام ویژگی	نام فیلد	نوع داده
شناسه درس	cid	عددی
نام درس	name	رشته‌ای
تعداد ساعت	hours	عددی
شناسه رشته	fid	عددی

جدول ۵-۸ - registry - برای نگه‌داری رابطه بین دانش‌آموز و درس (ثبت نام دانش‌آموزان در دروس مختلف و نمرات آنها)

نام ویژگی	نام فیلد	نوع داده
شناسه ثبت نام	rid	عددی
شناسه دانش‌آموز	sid	عددی
شناسه درس	cid	عددی
شناسه ترم	mid	عددی
شناسه معلم	tid	عددی
نمره	mark	اعشاری

در جدول registry از شناسه‌هایی استفاده شده که هرکدام از آنها کلید اصلی مربوط به یکی از جداول دیگر بانک اطلاعاتی می‌باشند. به‌عنوان مثال cid شناسه درس در جدول courses است که مشخص‌کننده درس اخذ شده می‌باشد. به این ستون‌ها، اصطلاحاً کلید خارجی گفته شده که هرکدام ارتباط بین این جدول با یک جدول دیگر را مشخص می‌کنند.

با کمک کلیدهای خارجی و ارتباطی که بین جداول برقرار می‌شود، می‌توان بعداً به اطلاعات مورد نیاز دسترسی پیدا کرد. به‌عنوان مثال با داشتن شناسه یک دانش‌آموز، لیست کل دروس اخذ شده توسط او را از بانک اطلاعاتی به‌دست آورد. از طرف دیگر با داشتن یک سطر از جدول registry می‌توان مشخصات کامل معلم، درس، ترم و دانش‌آموز مربوطه را از بانک اطلاعاتی استخراج کرد. برای نمایش بهتر ارتباط بین جداول از نمودار بانک اطلاعاتی استفاده می‌شود که برای بانک اطلاعاتی آموزشگاه در شکل ۲-۸ نشان داده شده است:



شکل ۲-۸ - نمودار بانک اطلاعاتی آموزشگاه

در اینجا به صورت خیلی ابتدایی داده‌های آموزشگاه مدل‌سازی و بانک اطلاعاتی آن طراحی گردید. در یک تحلیل واقعی و کامل موجودیت‌ها، ویژگی‌ها و رابطه‌های خیلی بیشتری باید مورد شناسایی و تحلیل قرار گیرند که بانک اطلاعاتی با جداول و ستون‌های بیشتری به دست خواهد آمد.

۳-۸- معرفی دستورات مهم SQL

SQL مخفف عبارت Structured Query Language به معنی زبان پرس‌وجوی ساخت یافته است که توسط بسیاری از سیستم‌های مدیریت بانک اطلاعاتی برای دریافت درخواست‌ها استفاده می‌شود. این زبان دارای دستورات مختلفی برای ایجاد، اصلاح یا حذف بانک اطلاعاتی، جداول و دیگر اجزای بانک اطلاعاتی و نیز دستوراتی برای بازیابی، درج، حذف یا ویرایش سطرهای جداول می‌باشد. در این درس به توضیح دستورات مهم و ضروری زبان SQL که در فصل‌های بعد به آن نیاز خواهد بود، بسنده شده است. دستوراتی که در اینجا مورد بحث قرار می‌گیرند عبارت‌اند از:

جدول ۶-۸

نام دستور	کاربرد
CREATE	ایجاد بانک اطلاعاتی و اجزای آن مثل جداول
SELECT	بازیابی اطلاعات از یک یا چند جدول
INSERT	درج سطر جدید در جدول
DELETE	حذف سطرهای جدول
UPDATE	به‌روزرسانی (اصلاح) سطرها

۱-۳-۸- دستور ایجاد اجزای بانک اطلاعاتی

دستور CREATE برای ایجاد بانک اطلاعاتی یا هر کدام از اجزای آن استفاده می‌شود.

ایجاد بانک اطلاعاتی: برای ایجاد بانک اطلاعاتی از دستور CREATE به شکل زیر استفاده می‌شود:

نام بانک اطلاعاتی CREATE

به‌عنوان مثال با اجرای دستور زیر بانک اطلاعاتی به نام bookshop ایجاد می‌شود:

CREATE bookshop;

درحالتی که تنها یک دستور درخواست شود، گذاشتن علامت سمیکالن (;) بعد از دستور اجباری نیست ولی برای اجرای چند دستور طی یک درخواست باید بعد از هر کدام علامت سمیکالن گذاشته شود. همچنین توجه داشته باشید که زبان SQL نسبت به کوچک یا بزرگ بودن حروف کلمات خود حساس نیست درحالی که نام بانک اطلاعاتی، نام جداول و نام ستون‌ها باید دقیقاً به همان شکلی که تعریف و نامگذاری شده‌اند در دستورات نوشته شوند.

ایجاد جدول: برای ایجاد جدول بانک اطلاعاتی از دستور CREATE به شکل زیر استفاده می‌شود:
 CREATE TABLE (مشخصات ستون‌های جدول) نام جدول ;
 مشخصات ستون‌های جدول شامل نام ستون، نوع داده و ویژگی‌های دیگر است که ابتدا نام ستون نوشته شده و در ادامه آن نوع داده و بعد از آن در صورت نیاز ویژگی‌های دیگر نوشته می‌شوند. دستور زیر برای ایجاد جدول books که پیش‌تر نشان داده شد، استفاده شده است:

```
CREATE TABLE books(
bid int primary key ,
bname varchar(100) ,
sid int ,
author varchar(100),
price double ) ;
```

در دستور فوق ستون bid از نوع عددی صحیح و به‌عنوان کلید اصلی، ستون‌های bname و author از نوع رشته‌ای به طول حداکثر ۱۰۰ کاراکتر و ستون price از نوع اعشاری تعریف شده‌اند.
 انواع داده‌های مختلفی می‌توان برای ستون‌های جدول استفاده کرد که در فصل بعد با مهم‌ترین آنها آشنا می‌شوید. همچنین برای هر کدام از ستون‌ها می‌توان یک یا چند نمونه از ویژگی‌های زیر را تعیین کرد:

جدول ۷-۸

نام ویژگی	کاربردی
unique	منحصر به فرد کردن ستون
primary key	تعیین ستون به‌عنوان کلید اصلی جدول
default	تعیین مقدار پیش‌فرض برای ستون
not null	پوچ ناپذیر کردن ستون یا به عبارتی اجباری کردن مقداردهی ستون

در مثال زیر جدول معلمان ایجاد شده به‌گونه‌ای که:

- ستون tid کلید اصلی باشد.
- ستون phone منحصر به فرد باشد.
- ستون grade دارای مقدار پیش‌فرض یک باشد.
- مقداردهی ستون‌های tname و phone در زمان درج اطلاعات معلمان اجباری باشد و نتوانند خالی بمانند.

```
CREATE TABLE teachers(
tid int primary key ,
tname varchar(100) not null ,
phone varchar(11) unique not null,
grade int default 1) ;
```

۲-۳-۸- دستور خواندن از بانک اطلاعاتی

دستور SELECT برای بازیابی اطلاعات از یک یا چند جدول مورد استفاده قرار می‌گیرد و شکل کلی آن به صورت زیر است:

; عبارت شرطی WHERE نام جدول FROM نام ستون‌ها SELECT
نتیجه اجرای دستور select همیشه یک جدول دارای سطر و ستون است. در ادامه چند مثال از دستور select روی جدول زیر اجرا شده است.

جدول ۸-۸- جدول books مشخصات کتاب‌ها

bid	bname	sid	Author	price
۱۰۰۱	بینوایان	۲	ویکتور هوگو	۴۲۰۰۰۰
۱۰۰۲	اقلیم هشتم	۱	محسن هجری	۲۸۰۰۰۰
۱۰۰۳	کلیله و دمنه	۱	آذر یزدی	۲۸۰۰۰۰
۱۰۰۴	مبانی سیاست	۴	جعفر محسنی	۲۵۰۰۰۰

مثال ۱: برای بازیابی ستون‌های شناسه و نام از جدول books دستور زیر را درخواست می‌کنیم:

SELECT bid , bname FROM books ;

جدول ۸-۹

نتیجه اجرای دستور فوق یک جدول به صورت جدول ۸-۹ است:

bid	bname
۱۰۰۱	بینوایان
۱۰۰۲	اقلیم هشتم
۱۰۰۳	کلیله و دمنه
۱۰۰۴	مبانی سیاست

مثال ۲: برای بازیابی مشخصات کامل کتاب با شناسه مساوی ۱۰۰۲ دستور زیر درخواست می‌شود:

SELECT * FROM books WHERE bid=1002;

علامت * به معنی همه ستون‌های جدول است و نتیجه اجرای دستور فوق به صورت زیر می‌باشد:

bid	bname	sid	author	price
۱۰۰۲	اقلیم هشتم	۱	محسن هجری	۲۸۰۰۰۰

مثال ۳: برای خواندن مشخصات همه کتاب‌های با موضوع ۱ دستور زیر درخواست می‌شود:

```
SELECT * FROM books WHERE sid=1;
```

دستور فوق خروجی زیر را تولید می‌کند:

Bid	bname	sid	author	price
۱۰۰۲	اقلیم هشتم	۱	محسن هجری	۲۸۰۰۰۰
۱۰۰۳	کلیله و دمنه	۱	آذر یزدی	۲۸۰۰۰۰

برای تعیین عبارت شرطی از عملگرهای محاسباتی، مقایسه‌ای و منطقی در زبان SQL استفاده می‌شود که تقریباً مشابه عملگرهای زبان php هستند. در نتیجه می‌توان با استفاده از عملگرهای ارائه شده در فصل دوم کتاب، عبارت شرطی مورد نظر را تعیین کرد. به‌عنوان مثال برای بازیابی کتاب‌های دارای قیمت بین ۲۰۰۰۰ و ۳۰۰۰۰ می‌توان از دستور زیر استفاده کرد:

```
SELECT * FROM books WHERE price >= 20000 and price <= 30000;
```

علاوه بر عملگرهای معمولی که قبلاً یاد گرفته‌اید در زبان SQL می‌توان از عملگرهای زیر نیز استفاده کرد:

عملگر like

از این عملگر برای مقایسه ستون از جدول با یک عبارت با بخشی از آن استفاده می‌شود.

به‌عنوان مثال برای بازیابی کتاب‌هایی که نام آنها شامل کلمه php باشد، دستور زیر اجرا می‌شود:

```
SELECT * FROM books WHERE bname like '%php%';
```

که علامت % مشخص کننده هیچ یا چند کاراکتر است.

عملگر between

از این عملگر برای بررسی قرارداد داشتن مقدار یک ستون بین دو مقدار معین استفاده می‌شود.

به‌عنوان مثال برای بازیابی مشخصات کتاب‌هایی که قیمت بین ۲۰۰۰۰ و ۳۰۰۰۰ دارند می‌توان دستور زیر را درخواست کرد:

```
SELECT * FROM books WHERE price between 20000 and 30000;
```

عملگر in

این عملگر بررسی می‌کند که مقدار یک ستون در بین چند مقدار معین وجود دارد یا خیر؟

به‌عنوان مثال برای بازیابی مشخصات کتاب‌ها دارای موضوعات ۱، ۲ یا ۳ می‌توان دستور زیر را درخواست کرد:

```
SELECT * FROM books WHERE sid in (1,2,3);
```


امکانات دیگر دستور select

با استفاده از توابع آماری مثل count در مقابل دستور select می‌توان محاسباتی روی ستون‌های جدول انجام داد و نتایج را به دست آورد. به عنوان مثال دستور زیر تعداد سطرهای کتاب‌های موجود در جدول books را بازیابی می‌کند:

```
SELECT count(*) as cnt FROM books;
```

نتیجه اجرای دستور فوق با فرض داشتن چهار سطر کتاب جدولی مشابه زیر خواهد بود:

cnt
4

برای مرتب‌سازی اطلاعات بازیابی شده براساس یک یا چند ستون از جدول، از عبارت order by در انتهای دستور select استفاده می‌شود.

به عنوان مثال دستور زیر لیست کتاب‌ها را براساس نام به صورت صعودی برمی‌گرداند:

```
SELECT * FROM books ORDER BY bname;
```

برای به دست آوردن لیست کتاب‌ها به صورت مرتب شده براساس قیمت و به شکل نزولی از دستور زیر استفاده می‌شود:

```
SELECT * FROM books ORDER BY price DESC;
```

کلمه DESC برای تعیین ترتیب نزولی و کلمه ASC برای تعیین ترتیب صعودی استفاده می‌شود. اگر این کلمه نوشته نشود، به طور پیش فرض ASC در نظر گرفته می‌شود.

دستور select مهم‌ترین دستور در زبان SQL است که برای دریافت گزارشات متنوع از بانک اطلاعاتی استفاده می‌شود و در اینجا فقط به ذکر مهم‌ترین قابلیت‌های آن بسنده کردیم، برای یادگیری کامل تمام قابلیت‌های این دستور می‌توانید از مراجع مختلف در اینترنت استفاده کنید.

۳-۳-۸- دستورات به روزرسانی داده‌ها

دستوراتی در SQL وجود دارند که برای درج، ویرایش یا حذف رکوردها استفاده می‌شوند.

دستور INSERT

از دستور insert برای درج یک یا چند سطر در جدول استفاده می‌شود و شکل کلی آن به صورت زیر است:

```
INSERT INTO (نام ستون‌ها) VALUES (مقادیر);
```

مقادیر از نظر تعداد و نوع داده باید متناظر با ستون‌های نوشته شده باشند. اگر قصد مقاردهی تمام ستون‌های جدول را دارید لازم نیست نام ستون‌ها ذکر شود و کافی است مقادیر به ترتیبی که ستون‌ها در جدول تعریف شده‌اند نوشته شوند.

```
INSERT INTO (مقادیر) VALUES (نام جدول);
```

به عنوان مثال برای اضافه کردن یک کتاب با مشخصات زیر:

شناسه کتاب: ۱۰۰۵

نام درس: برنامه نویسی وب

شناسه موضوع: ۳

می توان یکی از دستورات زیر را درخواست کرد:

```
INSERT INTO books( bid , bname , author , sid , price)
```

```
VALUES (1005 , 'برنامه نویسی وب' , 'محمدی' , 3 , 23000);
```

```
INSERT INTO books
```

```
VALUES (1005 , 'برنامه نویسی وب' , 'محمدی' , 3 , 23000);
```

مقادیر رشته‌ای باید حتماً داخل نقل قول نوشته شوند.

گاهی اوقات تنظیمات جدول به گونه‌ای است که مقدار کلید اصلی یا همان شناسه جدول به صورت خودکار تعیین می‌شود. در این صورت نیازی به تعیین مقدار برای کلید اصلی نیست. فرض کنید در جدول books شناسه کتاب دارای ویژگی مقداردهی خودکار باشد.

در این صورت برای درج کتاب از دستور زیر استفاده می‌شود:

```
INSERT INTO books(bname , author , sid , price)
```

```
VALUES ('برنامه نویسی وب' , 'محمدی' , 3 , 23000);
```

در این دستور برای سطر جدید، شناسه کتاب (bid) مساوی آخرین شناسه کتاب موجود در جدول به علاوه یک می‌شود. به عنوان مثال اگر آخرین شناسه کتاب عدد ۱۰۰۵ باشد، شناسه سطر جدید مساوی $1006 = 1 + 1005$ خواهد شد.

دستور DELETE

دستور delete برای حذف یک یا چند سطر از جدول استفاده می‌شود و شکل کلی آن به صورت زیر است:

DELETE FROM نام جدول WHERE عبارت شرطی

به عنوان مثال برای حذف یک کتاب با شناسه ۱۰۰۲ می‌توان دستور زیر را درخواست کرد:

```
DELETE FROM books WHERE bid=1002 ;
```

و یا برای حذف همه کتاب‌های با موضوع ۲ می‌توان دستور زیر را درخواست کرد:

```
DELETE FROM books WHERE sid=2;
```

دستور UPDATE

دستور update برای به‌روزرسانی یک یا چند سطر از جدول استفاده شده و دارای شکل کلی زیر است:

UPDATE نام جدول SET ستون ۱ = مقدار ۱ , مقدار ۲ = ستون ۲ , ...

WHERE عبارت شرطی;

به عنوان مثال برای اصلاح مشخصات کتابی با شناسه ۱۰۰۲ می توان نوشت:

```
UPDATE books
```

```
SET bname='منطقه هشتم',author='هجری',sid=3 ,price= 25600
```

```
WHERE bid=1002;
```

در دستور update همه ستون‌هایی که باید مقدار جدید بگیرند در مقابل کلمه SET نوشته می‌شوند. در مقابل کلمه where هم شرط قرار گرفته و سطرهایی که باید تغییر کنند مشخص می‌شود. اگر شرط نوشته نشود ستون‌های ذکر شده در مقابل کلمه set برای تمامی سطرهای جدول تغییر کرده و مقدار جدید در نظر گرفته شده را خواهند گرفت.

۴-۸- معرفی MySQL

MySQL یک سیستم مدیریت پایگاه داده رابطه‌ای متن باز و یکی از محبوب‌ترین سیستم‌های مدیریت پایگاه داده بوده که در سال ۱۹۹۵ ارائه شده است. ویژگی‌های اصلی MySQL عبارت‌اند از:

- دریافت و نصب آن بسیار راحت و رایگان است.
- با اکثر سیستم عامل‌های جدید از جمله ویندوز و لینوکس سازگاری دارد.
- دارای کارایی و سرعت بالا در بارگذاری اطلاعات است.
- از حافظه سیستم به خوبی استفاده کرده و کمتر دچار کمبود حافظه می‌شود.
- برای مدیریت بانک‌های اطلاعات یا حجم‌های کم تا بسیار بالا مناسب است.
- با تعریف کاربران و کنترل دسترسی آنها امنیت اطلاعات را تأمین می‌کند.

به علاوه MySQL به صورت Client/Server عمل می‌کند به این معنی که پس از نصب، نقش سرویس‌دهنده داده را داشته و برنامه‌هایی که نیاز به بانک اطلاعاتی دارند به عنوان سرویس‌گیرنده، درخواست‌های خود را به آن فرستاده و پاسخ را دریافت می‌کنند. بنابراین می‌تواند روی یک سیستم در شبکه نصب شده و برنامه‌ها از سیستم‌های دیگر درخواست‌های خود را به سرویس‌دهنده یعنی MySQL ارسال کنند.

۵-۸- مدیریت بانک‌های اطلاعاتی MySQL



شکل ۳-۸- نصب mysql

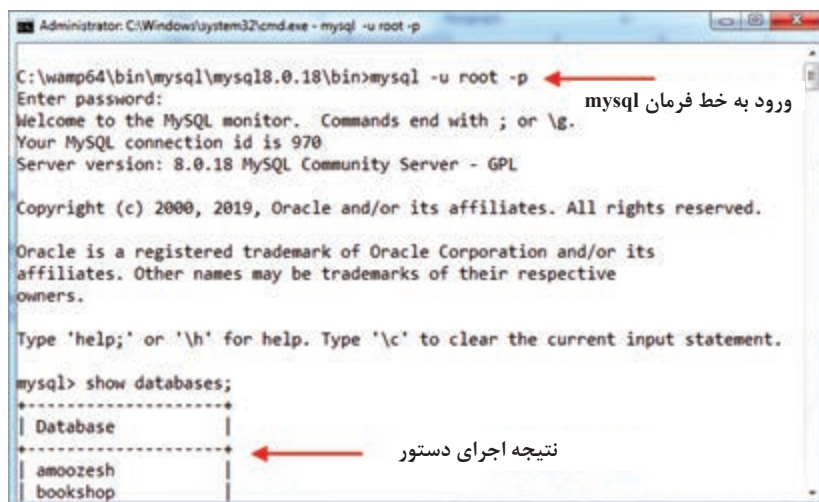
برای نصب MySQL نیاز به کار خاصی ندارید همین که یکی از بسته‌های معرفی شده در فصل اول کتاب مثل Wamp Server را روی سیستم محلی خود نصب کنید، MySQL هم نصب شده و قابل استفاده خواهد بود. به تصویر آورده شده در شکل ۳-۸ دقت کنید. در نسخه‌های جدید Wamp Server، علاوه بر MySQL، نرم‌افزار دیگری به نام MariaDB هم نصب می‌شود. هر چند MariaDB نسخه جدیدتر و کاراتر MySQL است و احتمال دارد در آینده جای MySQL را بگیرد ولی چون روش کار با MariaDB مشابه MySQL بوده و در حال حاضر MySQL رایج‌تر و کاربردی‌تر است. در این کتاب از MySQL استفاده شده است.

برای ایجاد بانک اطلاعاتی، تعریف جداول و مدیریت داده‌ها با MySQL دو روش وجود دارد:

روش اول - استفاده از خط فرمان MySQL

در پنجره خط فرمان (cmd) ویندوز، به زیرپوشه bin از محل نصب MySQL بروید، (برای این کار می‌توانید در ویندوز، پوشه محل نصب MySQL را باز کرده و همزمان با نگه داشتن کلید shift، کلیک راست کرده و گزینه open command window here را اجرا کنید) سپس برنامه mysql.exe را اجرا کنید، یک خط فرمان به صورت `mysql>` نشان داده می‌شود و می‌توان دستورات MySQL و نیز دستورات SQL برای عملیات مختلف را در آن اجرا کرد.

در شکل ۴-۸ پنجره خط فرمان را می‌بینید:



شکل ۴-۸- خط فرمان MySQL

در شکل فوق دیده می‌شود که ابتدا برای ورود به خط فرمان MySQL با استفاده از نام کاربری root دستور زیر اجرا شده است:

```
mysql -u root;
```

پس از نمایش اعلان mysql برای نمایش لیست بانک‌های اطلاعاتی دستور زیر اجرا شده است:

```
show databases;
```

این روش قدیمی بوده و کار با آن مشکل است با این حال در شرایط اضطراری که هیچ ابزار دیگری در دسترس نباشد می‌تواند استفاده شود.

روش دوم - استفاده از نرم‌افزارهای گرافیکی

نرم‌افزارهای کمکی برای مدیریت بانک‌های اطلاعاتی MySQL وجود دارند که با نصب یکی از آنها می‌توانید به راحتی و در محیط گرافیکی با استفاده از پنجره‌ها، منوها و دکمه‌ها هر کاری که لازم باشد برای ساخت بانک اطلاعاتی، نمایش لیست بانک‌های اطلاعاتی موجود، نمایش اجزای بانک اطلاعاتی و جداول، نمایش سطرهای جدول و نیز اصلاح و به‌روزرسانی را انجام دهید. با جستجوی مختصر در اینترنت، می‌توانید این نرم‌افزارها را پیدا کرده و دانلود کنید.

اصلی‌ترین نرم‌افزاری که برای مدیریت بانک‌های اطلاعاتی MySQL استفاده می‌شود، نرم‌افزار تحت وب phpMyAdmin است که با نصب بسته Wamp Server یا بسته‌های مشابه، به صورت پیش‌فرض روی سیستم شما نصب می‌شود. به علاوه در پنل مدیریت سایت‌های اینترنتی هم وجود داشته و می‌تواند برای مدیریت بانک‌های اطلاعاتی وب سایت‌ها استفاده شود. در فصل بعد مدیریت بانک اطلاعاتی با phpMyAdmin را فرا می‌گیرید.

خودآزمایی

- ۱ سیستم مدیریت بانک اطلاعاتی را تعریف کنید.
- ۲ سه نرم‌افزار برای مدیریت بانک اطلاعاتی را نام برده و در مورد آنها توضیح دهید.
- ۳ منظور از موجودیت‌ها در یک محیط عملیاتی چیست؟ چند تا از آنها را نام ببرید.
- ۴ موجودیت‌های اصلی در سیستم آزمون الکترونیکی را نام ببرید.
- ۵ نمودار ER برای سیستم آموزشگاه را رسم کنید.
- ۶ رابطه‌های یک به چند چگونه در بانک اطلاعاتی پیاده‌سازی می‌شوند.
- ۷ رابطه‌های چند به چند بین موجودیت‌ها چگونه در بانک اطلاعاتی پیاده‌سازی می‌شوند.
- ۸ شکل کلی دستور SQL برای درج رکورد را بنویسید.
- ۹ شکل کلی دستور SQL برای اصلاح رکورد را بنویسید.
- ۱۰ چهار ویژگی مهم MySQL را بنویسید.
- ۱۱ نرم‌افزار phpMyAdmin چه کاربردی دارد؟

پرسش‌های چهار گزینه‌ای

- ۱ کدام مورد از امکانات سیستم مدیریت بانک اطلاعاتی می‌باشد؟
 الف) ساخت بانک‌های اطلاعاتی
 ب) ساخت جدول
 ج) حذف و اصلاح رکوردها
 د) همه موارد
- ۲ در بانک‌های اطلاعاتی رابطه‌ای، نمایش داده‌ها به صورت می‌باشد.
 الف) روابط ریاضی
 ب) روابط منطقی
 ج) جداول اطلاعاتی
 د) رکوردهای داده‌ای
- ۳ کدام گزینه از ویژگی‌های MySQL نمی‌باشد؟
 الف) رایگان بودن
 ب) وابسته به PHP بودن
 ج) قابل حمل بودن
 د) متن باز بودن
- ۴ در یک محیط عملیاتی هر چیزی که باید در مورد آن داده‌هایی نگه‌داری شود را چه می‌گویند؟
 الف) موجودیت
 ب) رابطه
 ج) جدول
 د) نقش
- ۵ کدام گزینه تعریف بهتری از SQL را ارائه می‌دهد؟
 الف) یک نوع سیستم مدیریت بانک اطلاعاتی است.
 ب) یک زبان برای نوشتن پرس و جو است.
 ج) یک استاندارد برای ذخیره‌سازی داده‌ها است.
 د) همه موارد
- ۶ برای به‌روزرسانی داده‌ها در SQL از کدام دستور استفاده می‌شود؟
 الف) INSERT
 ب) DELETE
 ج) UPDATE
 د) ALTER
- ۷ در دستور زیر، به‌جای علامت سؤال (?) کدام گزینه می‌تواند قرار گیرد؟ (a و b ستون‌های جدول هستند)
 SELECT ? FROM tablename;
 الف) a,b
 ب) علامت *
 ج) a=b
 د) موارد الف و ب
- ۸ در دستور زیر، به‌جای ۱ و ۲ کدام کلمات می‌توانند قرار گیرند؟
 INSERT 1 tablename(a , b) 2 ('x', 'y');
 الف) INTO و VALUES
 ب) TABLE و VALUES
 ج) RECORD و VALUES
 د) VALUES و RECORD

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸
پاسخ	د	ج	ب	الف	ب	ج	د	الف

دستور کار عملی

- ۱ برای یک سیستم نوبت‌دهی درمانگاه، داده‌ها را تحلیل کرده و موجودیت‌ها، ویژگی‌ها و ارتباطات آنها را تحلیل کرده و نمودار ER را رسم کنید. سپس جداول لازم برای بانک اطلاعاتی را طراحی کنید.
- ۲ یک نرم‌افزار دلخواه برای مدیریت بانک‌های اطلاعاتی MySQL دانلود و نصب کرده و از طریق آن به MySQL متصل شوید.

فصل نهم

کار با نرم افزار phpMyAdmin



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- بانک اطلاعاتی جدید در phpMyAdmin ایجاد کند.
- عملیات ورود و صدور را برای بانک اطلاعاتی اجرا کند.
- جداول بانک اطلاعاتی را ایجاد کرده و مدیریت کند.
- سطرهای بانک اطلاعاتی را اضافه، حذف و یا ویرایش کند.
- دستورات SQL را روی یک بانک اطلاعاتی اجرا کند.

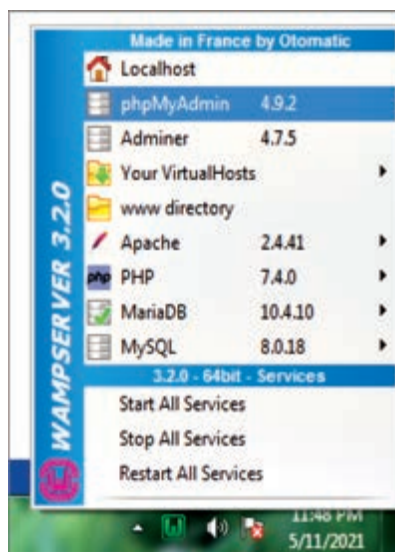
در فصل قبل روش تحلیل و طراحی بانک‌های اطلاعاتی را فراگرفتید، در این فصل مهارت پیاده‌سازی بانک اطلاعاتی MySQL با استفاده از نرم‌افزار phpMyAdmin را پیدا می‌کنید. این نرم‌افزار به صورت متن باز و رایگان بوده و با نصب بسته‌های Wamp server، XAMP یا EasyPHP به صورت خودکار روی سیستم نصب می‌شود. همچنین روی اغلب سرویس‌دهنده‌های میزبانی وب وجود دارد.

۹-۱- اجرای برنامه phpMyAdmin

روش اول و کلی کار نسخه‌های مختلف با phpMyAdmin چه بر روی سرویس‌دهنده‌های میزبانی وب نصب شده باشند یا روی سیستم‌های محلی یکسان است. بنابراین با مهارت‌هایی که در این فصل به دست می‌آورید،

می‌توانید با نسخه‌های قدیمی‌تر یا جدیدتر آن و بر روی سیستم محلی و یا سرویس‌دهنده‌های میزبانی، بانک‌های اطلاعاتی MySQL را ایجاد و مدیریت کنید. (سرویس‌دهنده‌های میزبانی وب در اینترنت قرار داشته و فایل‌ها و اطلاعات وب سایت‌های اینترنتی را نگه‌داری می‌کنند.)

در صورت نصب wamp server برای اجرای phpMyAdmin می‌توان مطابق شکل ۹-۱ از گزینه‌ای با این عنوان در منوی مدیریتی آن استفاده کرد:

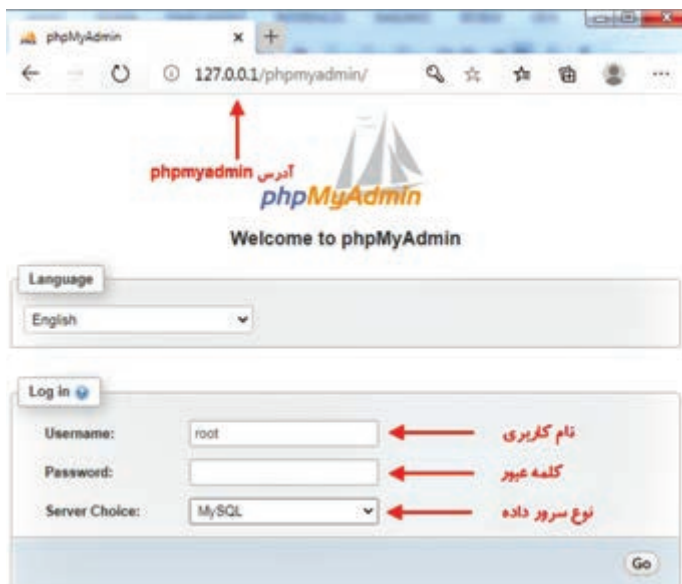


شکل ۹-۱- اجرای phpmyadmin

روش دوم برای اجرای phpMyAdmin نوشتن آدرس آن در نوار آدرس مرورگر است:

<http://127.0.0.1/phpmyadmin>

در نسخه‌های جدید phpMyAdmin بلافاصله پس از اجرا کادر ورود شکل ۹-۲ نمایش داده می‌شود که باید



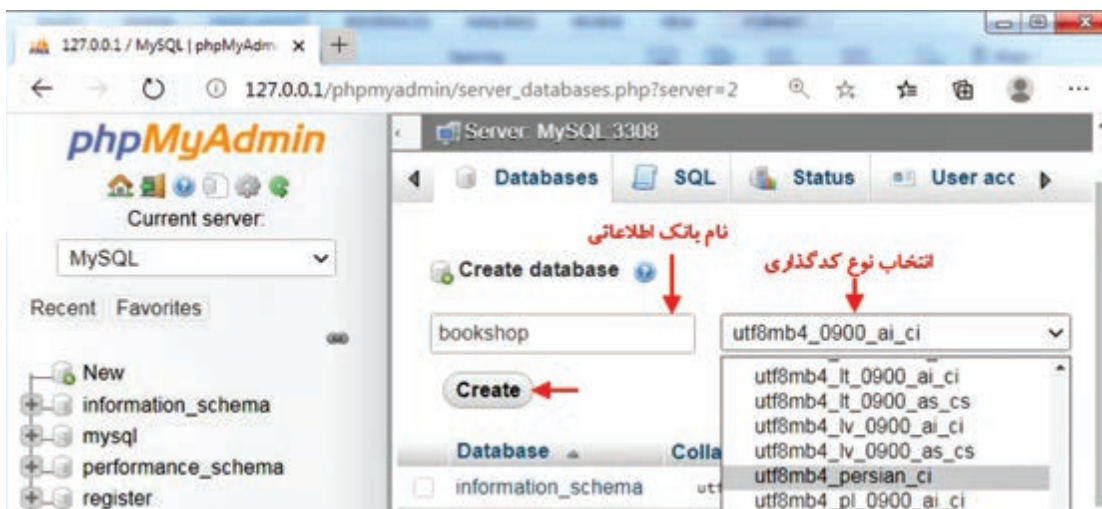
شکل ۹-۲- کادر ورود به phpMyAdmin

نام کاربری و کلمه عبور وارد شود. در شروع کار می توان نام کاربری root را وارد کرده، قسمت کلمه عبور را خالی گذاشته و در قسمت سرور MySQL را انتخاب کرد.

لازم به ذکر است در هنگام نصب MySQL به طور پیش فرض یک کاربر بنام root بدون کلمه عبور ایجاد می شود که می توان کلمه عبور آن را تعیین کرده و همچنین کاربران دیگری به MySQL اضافه کرد.

۹-۲- مدیریت بانک های اطلاعاتی

پس از ورود به phpMyAdmin مطابق شکل ۹-۳ زبانه databases را باز کنید، در این زبانه لیست بانک های اطلاعاتی موجود نمایش داده می شود که امکان انتخاب یک یا چند بانک اطلاعاتی و حذف آنها وجود دارد. برای ایجاد بانک اطلاعاتی مطابق شکل نام بانک اطلاعاتی را وارد کرده و collation آن را برای تعیین نوع کدگذاری حروف انتخاب کنید. انتخاب collation بستگی به نوع جدول و داده هایی دارد که قرار است در آن ذخیره شود. برای داده های فارسی بهتر است کدگذاری هایی که با utf8 یا utf8mb4 شروع می شوند را انتخاب کنید.



شکل ۹-۳- مدیریت بانک های اطلاعاتی

با کلیک روی نام یک بانک اطلاعاتی در ستون سمت چپ پنجره، صفحه مدیریت بانک اطلاعاتی باز می‌شود که در بالای این صفحه نام بانک اطلاعاتی نمایش داده می‌شود. این صفحه دارای زبانه‌های زیر است:

Structure: در این زبانه لیست جداول موجود در بانک اطلاعاتی نمایش داده شده و امکان حذف جداول یا انجام عملیات دیگر روی آنها فراهم می‌شود که در ادامه فصل با مهم‌ترین آنها آشنا خواهید شد.

SQL: در این زبانه یک کادر خالی برای دریافت دستورات به زبان SQL و اجرای آنها نمایش داده شده است. همچنین امکان انتخاب یک فایل متنی حاوی دستورات SQL و اجرای آنها وجود دارد.

Search: در این زبانه امکان جست‌وجوی پیشرفته در یک یا چند جدول از بانک اطلاعاتی وجود دارد.

Query: در این زبانه امکاناتی برای انتخاب ستون‌هایی از جداول مختلف، تعیین محدودیت برای نمایش سطرها و سپس تولید دستور Select مناسب وجود دارد.

Export: از این زبانه برای صدور بانک اطلاعاتی شامل جداول و اطلاعات آنها به صورت یک فایل SQL و یا فرمت‌های دیگر استفاده می‌شود. نتیجه را می‌توان نمایش داده و یا به صورت یک فایل دانلود کرد.

Import: از این زبانه برای ورود بانک اطلاعاتی ذخیره شده به صورت یک فایل متنی شامل دستورات SQL برای ساخت جداول و درج اطلاعات آنها استفاده می‌شود.

Designer: از این زبانه برای ایجاد ارتباط بین جداول و رسم نمودار بانک اطلاعاتی استفاده می‌شود.

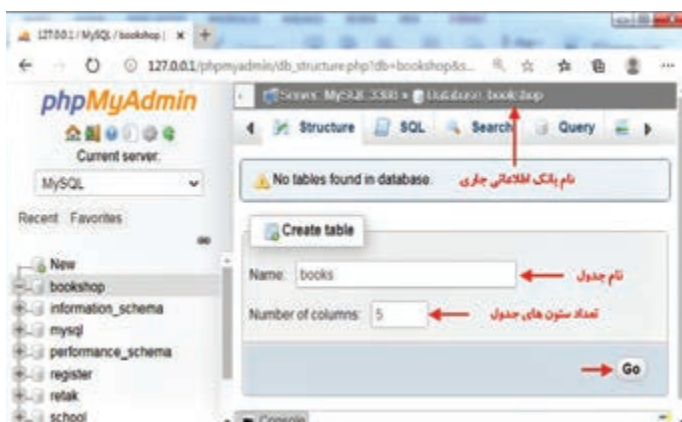
Operation: از این زبانه برای انجام عملیاتی روی بانک اطلاعاتی از جمله ایجاد جدول جدید، تغییر نام بانک اطلاعاتی، کپی کردن بانک اطلاعاتی شامل ساختار، داده‌ها و یا ساختار و داده‌ها به صورت بانک اطلاعاتی جدید و تغییر collation بانک اطلاعاتی استفاده می‌شود.

Privileges: از این زبانه برای مدیریت کاربران بانک اطلاعاتی و تعیین مجوزهای آنها استفاده می‌شود. برای هر بانک اطلاعاتی به‌طورپیش‌فرض کاربر root اضافه می‌شود که می‌توان کاربران دیگری را نیز در این بخش اضافه کرد.

Drop: از این گزینه برای حذف بانک اطلاعاتی استفاده می‌شود.

در قسمت‌های بعد برای انجام عملیات مختلف از این زبانه‌ها استفاده می‌کنید.

۹-۳- ایجاد جداول

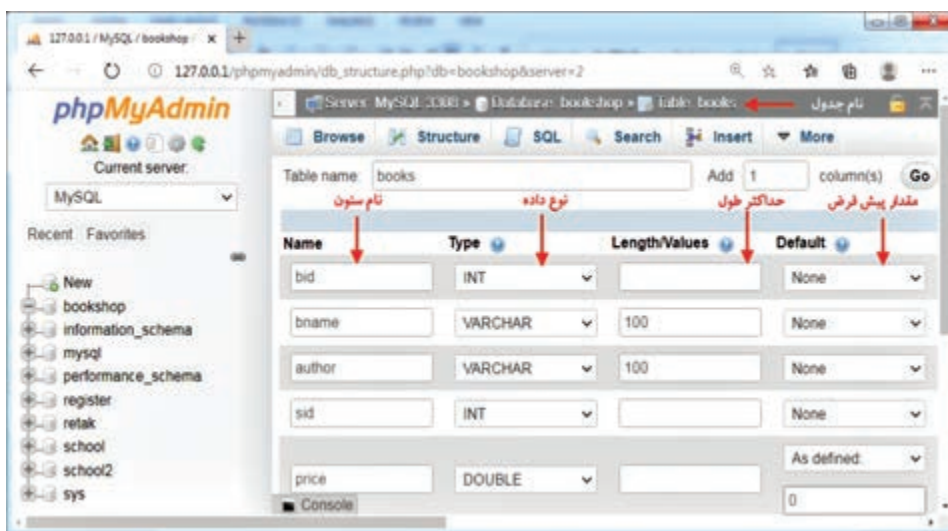


شکل ۹-۴- ساخت جدول

جداول اصلی‌ترین بخش یک بانک اطلاعاتی می‌باشند، پس از انتخاب بانک اطلاعاتی در زبانه Structure، مشابه شکل ۹-۴ لیست جداول نمایش داده شده و امکان تعریف جدول جدید فراهم می‌شود:

برای ساخت جدول نام آن و تعداد ستون‌های آن را وارد کرده و روی دکمه Go کلیک کنید. در نام‌گذاری بانک اطلاعاتی، جداول و ستون‌های جدول به نکات زیر توجه داشته باشید:

- ۱ به هیچ عنوان از حروف فارسی برای نام‌گذاری استفاده نکنید.
 - ۲ از کلمات رزرو شده در SQL (مثل set, select, into) نمی‌توانید به عنوان نام استفاده کنید.
 - ۳ MySQL نسبت به حالت حروف حساس است و باید در نام‌گذاری‌ها به حالت حروف توجه داشته باشید.
 - ۴ نام نمی‌تواند دارای فاصله باشد و اگر نام دارای چند کلمه است می‌توانید بین کلمات خط زیر (_) قرار دهید (مثل first_name) و یا هر کلمه را با حروف بزرگ شروع کنید (مثل FirstName).
 - ۵ پیشنهاد می‌شود از یک روش یکسان برای نام‌گذاری استفاده کنید به عنوان مثال می‌توانید همیشه از حروف کوچک لاتین استفاده کنید.
 - ۶ همیشه برای نام‌گذاری از کلمات با معنی و متناسب با موضوع استفاده کنید.
- پس از تأیید کادر فوق، شکل ۵-۹ را خواهید دید که باید مشخصات ستون‌های جدول را وارد کنید.



شکل ۵-۹- مشخصات ستون‌های جدول

در قسمت Length برای ستون‌هایی که نوع داده کاراکتری دارند، حداکثر طول آنها تعیین می‌شود. در قسمت Default می‌توان مقدار پیش‌فرض برای ستون تعیین کرد. در زمان درج کردن اگر مقداری برای ستون تعیین نشود، مقدار پیش‌فرض را خواهد گرفت، مقدار پیش‌فرض می‌تواند NULL (پوچ) یا یک مقدار دلخواه تعیین شود.

در قسمت Collation می‌توان روش کدگذاری حروف را تعیین کرد و اگر انتخاب نشود به‌طورپیش‌فرض همان کدگذاری جدول برای آن در نظر گرفته خواهد شد.

در قسمت Type نوع داده ستون تعیین می‌شود. پرکاربردترین انواع داده‌ها در جدول ۹-۱ نمایش داده شده‌اند:

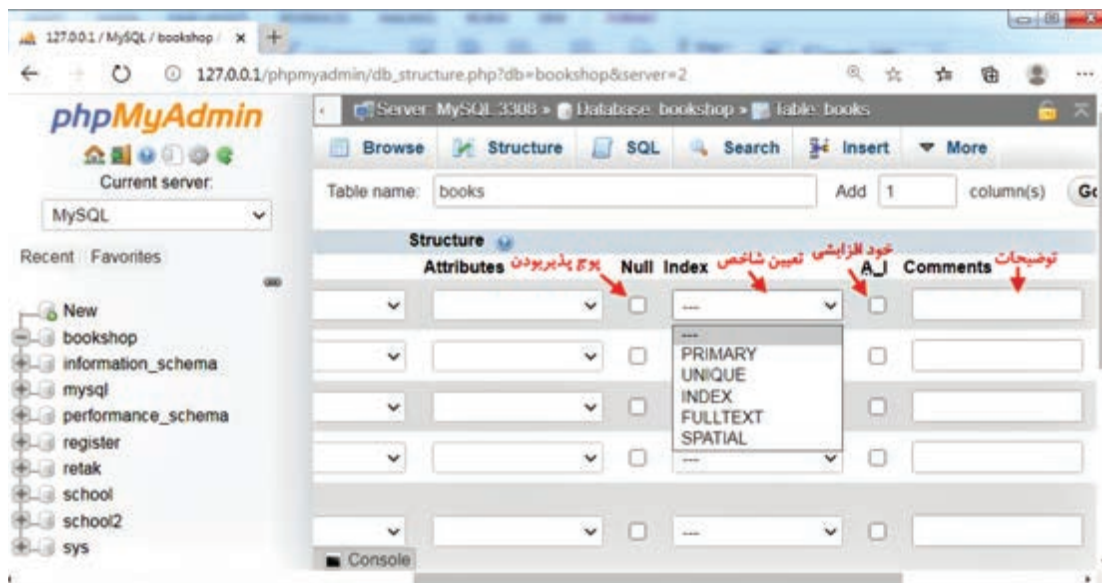
جدول ۹-۱

دسته	نوع داده	کاربرد
رشته‌ای	char	نگهداری رشته‌های به طول حداکثر ۲۵۶ کاراکتر
	varchar	نگهداری رشته‌های به طول حداکثر ۶۵۵۳۵ کاراکتر
	text	نگهداری متن بلند
	blob	نگهداری مجموعه‌ای از بایت‌ها مربوط به متن یا تصویر یا هر شیء دیگر
عددی صحیح	tinyint	نگهداری اعداد یک بایتی
	smallint	نگهداری اعداد دو بایتی
	mediumint	نگهداری اعداد سه بایتی
	int	نگهداری اعداد چهار بایتی
	bigint	نگهداری اعداد هشت بایتی
اعشاری	float	اعشاری با دقت معمولی با چهار بایت فضا
	double	اعشاری با دقت مضاعف با هشت بایت فضا
تاریخ و زمان	Date	نگهداری تاریخ با فرمت YYYY - MM - DD
	Time	نگهداری زمان با فرمت hh:mm:ss
	Datetime	نگهداری تاریخ و زمان

برای انواع داده char و varchar باید حداکثر طول رشته تعیین شود. به‌عنوان مثال اگر ستون bname برای نام کتاب از نوع char(100) تعیین شود به این معنی است که حداکثر تعداد کاراکترها برای نام کتاب می‌تواند ۱۰۰ باشد.

تفاوت بین این دو نوع داده این است که اگر از char استفاده شود، در همان ابتدا ۱۰۰ خانه از حافظه برای نام کتاب با فضای خالی (space) پر شده و نام کتاب ممکن است بخشی از آن را استفاده کند. چنانچه از varchar استفاده شود، فقط به تعداد مورد نیاز حافظه تخصیص داده می‌شود، به‌عنوان مثال اگر طول نام کتاب ۲۰ کاراکتر باشد، به همان اندازه یعنی ۲۰ خانه حافظه به آن تخصیص داده می‌شود.

در شکل ۹-۶ تنظیمات بیشتری برای ستون‌ها نشان داده شده است:



شکل ۹-۶- تنظیمات بیشتر ستون‌ها

با انتخاب گزینه null این امکان فراهم می‌شود که مقداری برای آن ستون اختیاری شده و می‌تواند هنگام درج سطر جدید خالی بماند. در قسمت Index می‌توان ستون را به عنوان کلید اصلی یا انواع دیگر کلیدها تعیین کرد. در قسمت AI می‌توان ستون کلید اصلی را به صورت خود افزایشی (Auto Increment) تعیین کرد تا هنگام درج سطر جدید، مقداری به صورت خودکار به آن داده شود. در قسمت Comments می‌توان نام فارسی ستون یا هر توضیح دیگری را وارد کرد تا بعداً در تشخیص ساختار جدول به ما کمک کند. بعد از وارد کردن ستون‌ها و انجام تنظیمات ضروری برای هر ستون، روی دکمه GO کلیک می‌کنیم تا جدول ساخته شده و باز شود.

۱-۳-۹- زبانه‌های مربوط به جدول

با باز شدن یک جدول، نام جدول در بالای صفحه بعد از نام بانک اطلاعاتی نمایش داده شده و زبانه‌های زیر برای جدول در دسترس قرار می‌گیرد که در ادامه روش کار با هر کدام از این زبانه را می‌بینید:

Browse: برای نمایش سطرهای جدول استفاده می‌شود و امکان انتخاب یک یا چند سطر از جدول و حذف یا ویرایش آنها فراهم شده است.

Structure: در این زبانه لیست ستون‌های جدول نشان داده شده و امکان حذف و اصلاح ستون و یا اضافه کردن یک یا چند ستون به جدول وجود دارد. یک ستون را می‌توان به صورت کلید اصلی (Primary Key)، یکتا (unique) و یا فهرست تعریف کرد.

SQL: دارای یک کادر برای وارد کردن دستورات SQL و اجرای آن است.

Search: برای انجام جست و جوی پیشرفته در جدول بر اساس ستون‌های مختلف و شرط‌های مختلف استفاده می‌شود.

Insert: برای درج سطرهای جدید در جدول استفاده می‌شود که هر بار می‌توان دو یا چند سطر را در آن وارد کرد.

Export: برای صدور جدول و اطلاعات آن به فرمت‌های مختلف استفاده می‌شود.

Import: برای ورود یک فایل حاوی دستورات SQL و اجرای آنها روی بانک اطلاعاتی استفاده می‌شود.

Operation: برای انجام عملیات مختلف روی جدول از جمله تغییر نام، کپی و انتقال جدول و تغییر Collation استفاده می‌شود.

Empty: برای خالی کردن جدول استفاده می‌شود. اجرای این گزینه تمام سطرهای جدول را حذف کرده و جدول را به حالت اولیه برمی‌گرداند.

Drop: برای حذف جدول استفاده می‌شود.

برای دسترسی به جداول دیگر بانک اطلاعاتی یا ایجاد جدول، ابتدا در بالای صفحه یا ستون سمت چپ صفحه، روی نام بانک اطلاعاتی کلیک کنید تا زبانه‌های مربوط به بانک اطلاعاتی نمایش داده شود و از طریق زبانه Structure بتوانید لیست جداول را مشاهده کرده و جدول جدید اضافه نمایید.

۲-۳-۹- اصلاح ستون‌های جدول

با باز کردن جدول، در زبانه Structure همانطور که در شکل ۹-۷ می‌بینید، لیست ستون‌های جدول نشان داده می‌شود. در کنار هر ستون گزینه‌هایی برای اصلاح (Change) یا حذف (Drop) ستون وجود دارد، با انتخاب ستون‌ها می‌توان گزینه‌های حذف و ویرایش پایین لیست را اجرا کرد. با اجرای این گزینه‌ها می‌توان ستون‌ها را حذف کرد یا تنظیمات آنها را تغییر داد:

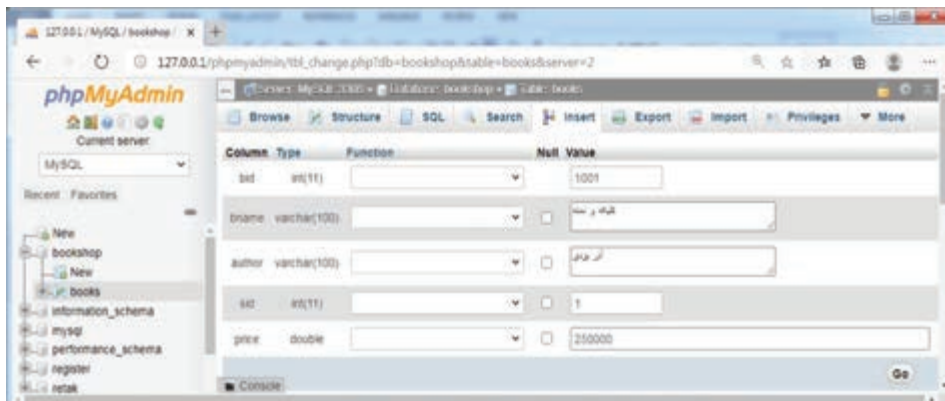


شکل ۹-۷- مدیریت ستون‌های جدول

برای اضافه کردن یک یا چند ستون جدید می توانید از قسمت Add columns در زیر لیست استفاده کنید.

۹-۳-۳- درج رکورد جدید در جدول

با بازکردن جدول، مطابق شکل ۹-۸ در زبانه Insert امکان وارد کردن اطلاعات یک یا دو رکورد و درج آنها در جدول فراهم می شود.

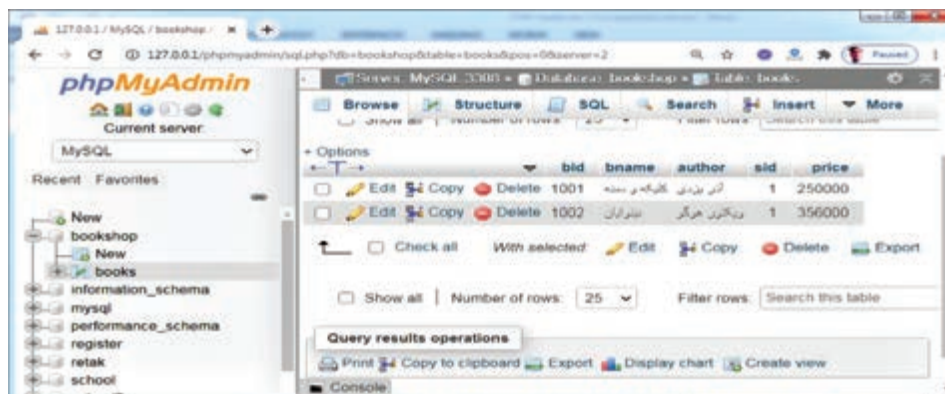


شکل ۹-۸- درج رکورد جدید

در این صفحه فقط کافی است در قسمت Value برای هر کدام از فیلدها مقدار وارد کرده و در پایان روی دکمه Go کلیک کنید. برای ستون های خودافزایشی (AI) و ستون هایی که پوچ (Null) پذیر هستند، وارد کردن مقدار اختیاری است.

۹-۳-۴- مدیریت رکوردهای جدول

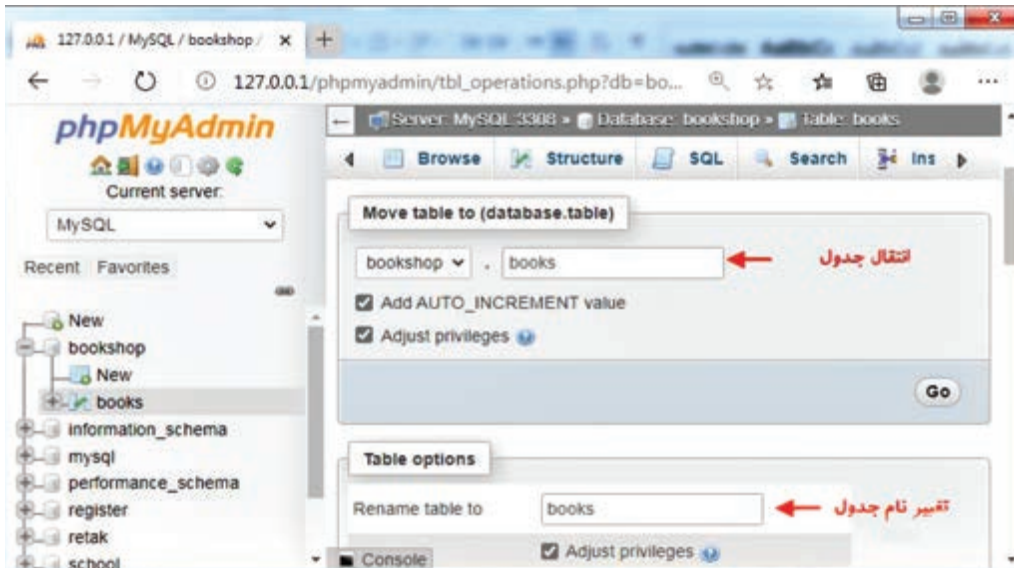
پس از انتخاب و باز کردن جدول، در زبانه Browse مربوط به آن لیست سطرهای جدول مشابه شکل ۹-۹ نشان داده می شود. در کنار هر سطر لینک هایی برای ویرایش (Edit)، حذف (Delete) یا کپی (Copy) سطر وجود دارد. همچنین با انتخاب یک یا چند سطر و استفاده از لینک های زیر لیست می توان سطرها را حذف، ویرایش یا کپی برداری کرد. به عنوان مثال با اجرای گزینه Copy، سطرهای جدیدی از روی سطرهای انتخاب شده کپی خواهد شد.



شکل ۹-۹- مدیریت رکوردهای جدول

۵-۳-۹- تغییر نام، انتقال و یا کپی جدول

مطابق شکل ۹-۱۰ در زبانه Operations از جدول باز شده می توان عملیات زیر را روی جدول فعلی انجام داد:

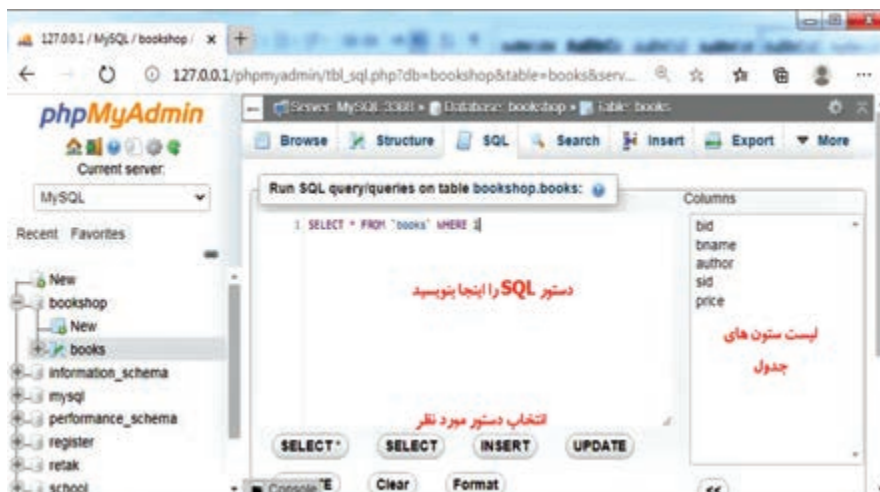


شکل ۹-۱۰ عملیات روی جدول

- ۱ انتقال (Move) جدول از بانک اطلاعاتی فعلی به بانک اطلاعاتی دیگر با نام فعلی یا نام جدید.
- ۲ تغییر نام (Rename) جدول به نام دلخواه.
- ۳ ایجاد یک کپی (Copy) از جدول در بانک اطلاعاتی جاری یا بانک اطلاعاتی دیگر. هنگام کپی برداری می توان انتخاب کرد که جدول خالی یا جدول به همراه سطرهای داخل آن کپی شوند.
- ۴ حذف کردن کامل جدول (Delete).
- ۵ حذف همه سطرهای جدول یا خالی کردن جدول (Empty).

۹-۴- اجرای دستورات SQL

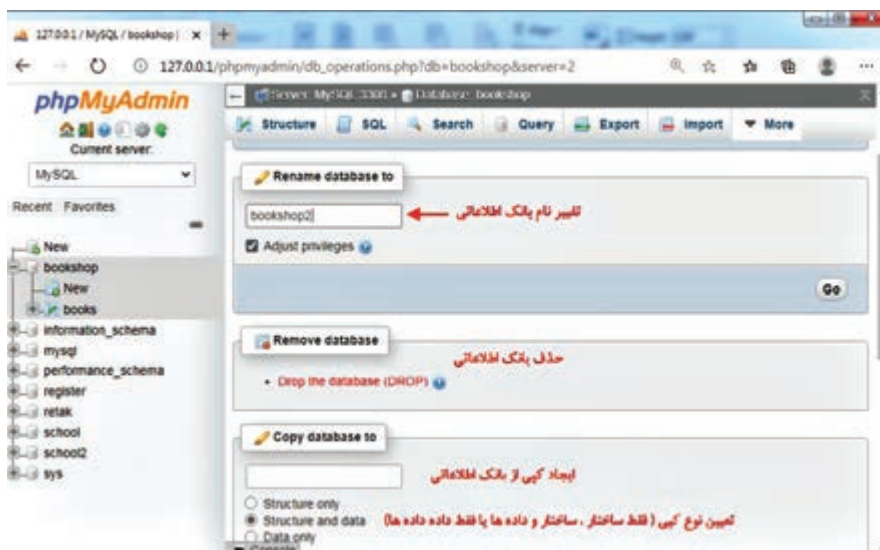
در زبانه SQL مربوط به بانک اطلاعاتی یا جدول، مشابه شکل ۹-۱۱ یک کادر متنی برای وارد کردن دستورات دلخواه به زبان SQL و اجرای آن وجود دارد. می‌توانید هر تعداد از دستورات SQL را در این کادر وارد کرده و با کلیک روی دکمه Go اجرا کنید.



شکل ۹-۱۱- اجرای دستورات SQL

۹-۵- عملیات روی بانک اطلاعاتی

در زبانه Operation مربوط به بانک اطلاعاتی مطابق شکل ۹-۱۲ می‌توان کارهایی مثل تغییر نام بانک اطلاعاتی، حذف یا نسخه‌برداری از آن را انجام داد:

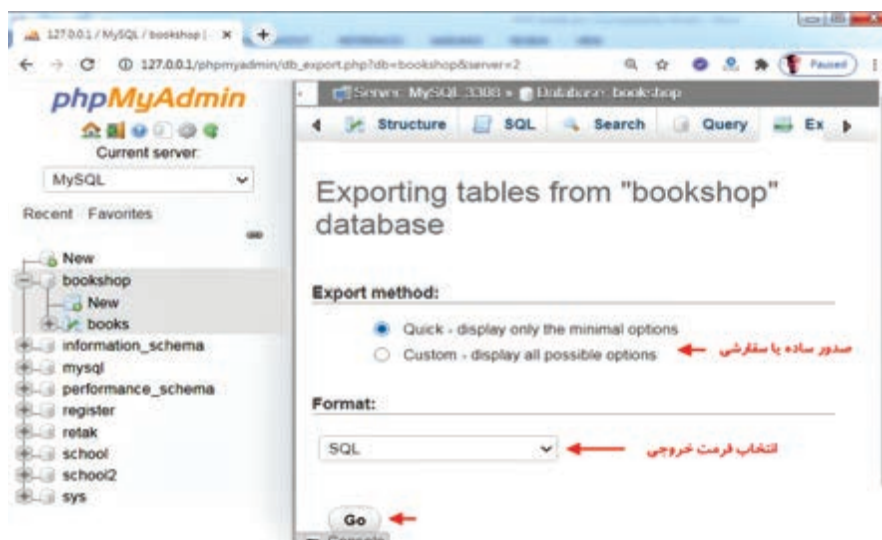


شکل ۹-۱۲- عملیات کلی روی بانک اطلاعاتی

۹-۶- ورود و صدور اطلاعات

یکی از مهم‌ترین کارهایی که لازم است روی جدول یا کل بانک اطلاعاتی انجام شود، ورود و صدور آنها است. منظور از ورود، تبدیل داده‌ها از فرمت برنامه‌های دیگر به فرمت MySQL است و منظور از صدور، تبدیل داده‌ها از فرمت MySQL به فرمت نرم‌افزارهای دیگر است.

در زبانه Export می‌توان اطلاعات یک یا چند جدول از بانک اطلاعاتی را به فرمت‌های CSV، JSON، XML، SQL و یا دیگر فرمت‌های تعریف شده و رایج صادر کرد. در شکل ۹-۱۳، چنانچه گزینه Quick انتخاب شده باشد همه جداول صادر خواهند شد و اگر custom انتخاب شود، لیست جداول را نمایش داده و می‌توان یک یا چند جدول دلخواه را برای صدور انتخاب کرد.



شکل ۹-۱۳- صدور بانک اطلاعاتی

در اکثر موارد داده‌ها را به قالب SQL صادر می‌کنند. در قالب SQL، یک فایل متنی ساخته و دانلود می‌شود که شامل دستوراتی به زبان SQL برای ساخت جداول و درج همه رکوردهای جداول در آنها است. در زبانه Import می‌توان یک فایل SQL را انتخاب کرد تا بارگذاری شده و همه دستورات داخل آن اجرا شده و در نتیجه جداول ساخته و سطرها داخل آن درج گردد.

برای پشتیبان‌گیری از بانک اطلاعاتی یا انتقال بانک اطلاعاتی از یک سیستم به سیستم دیگر از امکانات صدور و ورود استفاده می‌شود. لازم به ذکر است که می‌توان بانک اطلاعاتی روی سیستم محلی را با قالب SQL صادر کرده و فایل حاصل را روی سیستم میزبانی وب وارد کرد تا بانک اطلاعاتی از سیستم شما به سایت شما در اینترنت منتقل شود.

خودآزمایی

- ۱ در مورد ویژگی‌ها و امکانات نرم افزار phpMyAdmin توضیح دهید.
- ۲ توضیح دهید که برای اجرای نرم افزار phpMyAdmin چه روش‌هایی وجود دارد.
- ۳ زبانه‌های structure، query و export در محیط phpMyAdmin چه کاربردی دارند؟
- ۴ انواع داده اعشاری را نام برده و با هم مقایسه کنید.
- ۵ انواع داده کاراکتری char و varchar چه تفاوتی با هم دارند؟
- ۶ گزینه‌های edit، delete و copy در زبانه browse جدول در phpMyAdmin چه کاربردی دارند؟
- ۷ گزینه‌های change و drop در زبانه structure جدول در phpMyAdmin چه کاربردی دارند؟
- ۸ روش انتقال یک بانک اطلاعاتی موجود از یک سیستم به سیستم دیگر را توضیح دهید.

سوالات چهار گزینه‌ای

- ۱ با نصب کدام بسته ابزار phpMyAdmin به صورت خودکار نصب می‌شود؟
الف) wamp server (ب) xamp (ج) easyphp (د) همه موارد
- ۲ نام کاربر پیش فرض MySQL که هنگام نصب آن ایجاد می‌شود، چیست؟
الف) administrator (ب) sa (ج) root (د) user
- ۳ برای انتخاب ستون‌هایی از جداول مختلف، تعیین محدودیت برای نمایش سطرها و سپس تولید دستور Select مناسب از کدام زبانه در phpMyAdmin استفاده می‌شود؟
الف) query (ب) export (ج) SQL (د) search
- ۴ در نام گذاری بانک اطلاعاتی، جداول و ستون‌های جدول در MySQL، کدام گزینه بهتر است رعایت شود؟
الف) از حروف فارسی برای نام گذاری استفاده نشود.
ب) از کلمات رزرو شده در SQL به عنوان نام استفاده نشود.
ج) از حروف بزرگ برای نام گذاری استفاده شود.
د) موارد الف و ب
- ۵ کدام مورد برای تنظیمات ستون‌های جدول در phpMyAdmin نادرست است؟
الف) در قسمت null می‌توان مقدار اولیه ستون را تعیین کرد.
ب) در قسمت index می‌توان ستون را به عنوان کلید اصلی جدول تعیین کرد.
ج) در قسمت default می‌توان مقدار پیش فرض برای ستون تعیین کرد.
د) در قسمت comments می‌توان توضیحات برای ستون وارد کرد.
- ۶ کدام نوع داده ستون‌های جدول در MySQL برای نگه‌داری مقادیر باینری مثل فایل‌ها استفاده می‌شود؟
الف) binary (ب) blob (ج) integer (د) bigint
- ۷ کدام امکان در زبانه operation از بانک اطلاعاتی در phpMyAdmin در دسترس نمی‌باشد؟
الف) انتقال بانک اطلاعاتی
ب) کپی کردن بانک اطلاعاتی
ج) تغییر نام بانک اطلاعاتی
د) حذف بانک اطلاعاتی

۸ گزینه copy از زبانه operations جدول در phpMyAdmin برای کپی کردن چه قسمتی از جدول می تواند استفاده شود؟

الف) ساختار جدول (ب) داده های جدول (ج) ساختار و داده های جدول (د) همه موارد
۹ برای صدور بانک اطلاعاتی به صورتی که در phpMyAdmin سیستم دیگر قابل وارد کردن باشد، کدام فرمت خروجی را انتخاب می کنیم؟

الف) CSV (ب) SQL (ج) XLS (د) موارد الف و ب
۱۰ برای پشتیبان گیری از بانک اطلاعاتی در phpMyAdmin و انتقال از یک سیستم به سیستم دیگر از زبانه استفاده می شود.

الف) download (ب) backup (ج) export (د) import

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
پاسخ	د	ج	الف	د	ب	ب	الف	د	ب	ج

دستور کار عملی

۱ بانک اطلاعاتی برای فروشگاه کتاب آنلاین را در phpmyadmin ایجاد کنید و برای هر جدول آن حداقل سه سطر دلخواه درج کنید.

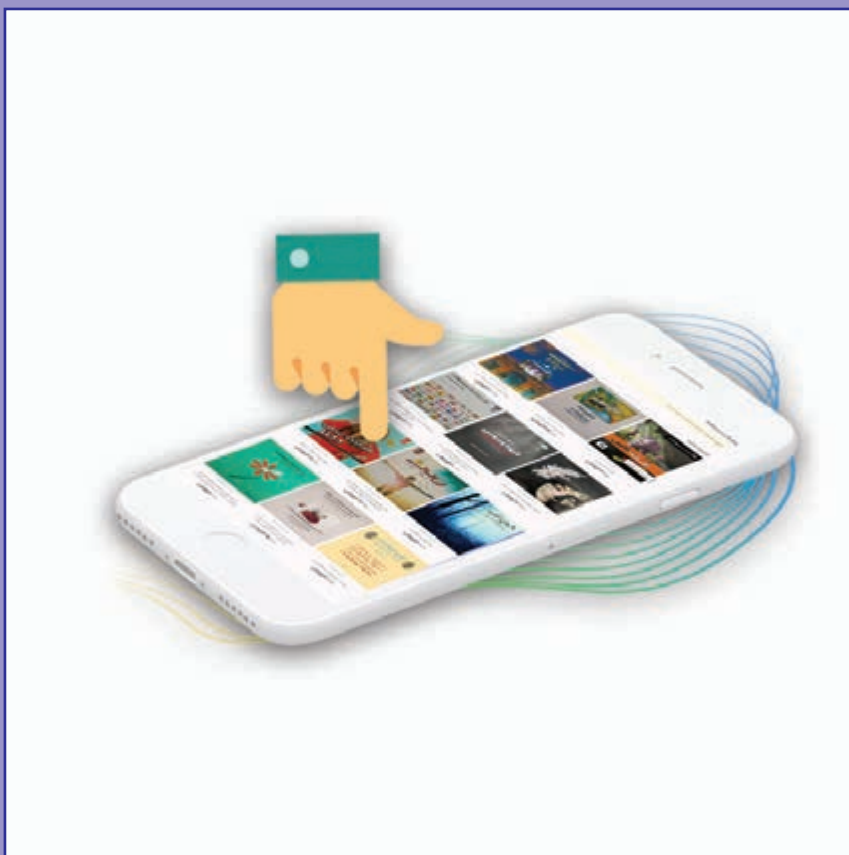
۲ بانک اطلاعاتی فروشگاه را با فرمت SQL صادر کنید.

۳ بانک اطلاعاتی جدیدی ایجاد کرده و با وارد کردن فایل SQL به دست آمده در سؤال قبل، جداول و داده های آن را ایجاد کرده و سطرهای داده ای را به آن اضافه کنید.

۴ بانک اطلاعاتی را با نام جدیدی کپی کنید.

فصل دهم

تحليل و طراحی سایت اینترنتی فروشگاه کتاب



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- عملکرد کلی سایت فروشگاه اینترنتی را شرح دهد.
 - انواع کاربران در فروشگاه اینترنتی را شناسایی کند.
 - وظایف و عملکرد کاربران در فروشگاه اینترنتی را شرح دهد.
 - موجودیت‌ها در فروشگاه اینترنتی را شرح دهد.
 - ساختار مناسب برای بانک اطلاعاتی فروشگاه را طراحی کند.
 - ساختار مناسب و ارتباط بین بخش‌های فروشگاه اینترنتی را نمایش دهد.
- در فصل‌های قبل با نحوه برنامه‌نویسی به زبان PHP آشنا شده و توانایی طراحی و پیاده‌سازی صفحات وب شامل اسکریپت‌های PHP را پیدا کردید. با استفاده از فرم‌های ورود اطلاعات، رکوردها را دریافت کرده و در برنامه استفاده کردید.
- در ادامه با دستورات استفاده از بانک اطلاعاتی و روش طراحی صفحات مختلف برای نمایش رکوردها، دریافت اطلاعات و به‌طور کلی طراحی یک نرم‌افزار کاربردی مبتنی بر وب آشنا می‌شوید. برای فهم بهتر و اجرای مثال‌های برنامه‌نویسی واقعی‌تر، یک نرم‌افزار فروشگاه اینترنتی کتاب در نظر گرفته شده و انتهای هر فصل صفحاتی از این فروشگاه طراحی و کدنویسی می‌شود.
- قبل از شروع پیاده‌سازی یک سیستم نرم‌افزاری باید آن را از جنبه‌های مختلف مورد بررسی و تجزیه و تحلیل قرار داده و طراحی کرد. در این فصل نرم‌افزار فروشگاه اینترنتی کتاب تحلیل و طراحی شده و در فصل بعد قالب کلی سایت طراحی و پیاده‌سازی شده است.

۱۰-۱- معرفی سیستم فروشگاه اینترنتی

هر سیستم برای حل یک مشکل سازمان، بهینه‌سازی یا بهبود عملکرد کل یا بخشی از سازمان ایجاد می‌شود. لذا نیاز است تا قبل از طراحی و پیاده‌سازی سیستم با حضور در محیط سازمان، با موجودیت‌ها و فرایندهای کاری آشنا شده و از طریق مصاحبه با صاحب‌نظران و کارکنان آن سازمان نسبت به موضوع شناخت کامل پیدا کرده و با آگاهی طرح اولیه سیستم آماده شود.

پس از تهیه طرح اولیه و تعیین حدود و قابلیت‌های مورد نیاز سیستم و تشکیل تیم کاری برای اجرای پروژه، مدت زمان موردنیاز و هزینه‌های مربوط به هر بخش آن از جمله طراحی، کدنویسی، نصب، رفع اشکال و آموزش کاربران، محاسبه شده و به کارفرما ارائه می‌گردد و پس از رسیدن به توافق نهایی با کارفرما و عقد قرارداد، عملیات اصلی طراحی و پیاده‌سازی شروع می‌شود.

به‌طورمشخص قبل از پیاده‌سازی یک سیستم مبتنی بر وب باید شناخت کامل نسبت به اهداف و عملکرد سیستم، انواع کاربران، وظایف و اختیارات آنها، فعالیت‌ها و قابلیت‌های اصلی سیستم داشته باشید. باید داده‌های موجود در سیستم را شناسایی کرده و جداول و بانک اطلاعاتی مناسبی برای آن طراحی کنید. همچنین ساختار کلی صفحات سایت، ارتباط بین آنها و محتوا و طرح کلی هر کدام از صفحات را تعیین کنید.

نتایج تحلیل و طراحی به‌صورت نمودارهای استاندارد نمایش داده شده و مستند می‌شود. این مستندات برای

تأیید در اختیار کارفرما قرار گرفته و نیز در طول اجرای پروژه به عنوان یک مرجع توسط افراد تیم مورد استفاده خواهد بود. حتی این مستندات می‌تواند پس از تکمیل پروژه برای استفاده از سیستم، یا توسعه امکانات آن مورد استفاده قرار گیرد.

مستندات سیستم جزء جدایی‌ناپذیر یک قرارداد پروژه نرم‌افزاری بوده به صورت دقیق نشان می‌دهد که مجری پروژه موظف به تولید چه محصولی و با چه ویژگی‌هایی است و می‌تواند به عنوان یک سند قانونی در مواقع بروز اختلاف نظر بین مجری و کارفرما مورد استفاده قرار گیرد.

فروشگاه اینترنتی یک نرم‌افزار کاربردی مبتنی بر وب است که برای ارائه کالا و خدمات به انبوهی از کاربران استفاده می‌شود. این نرم‌افزار به مشتریان اجازه می‌دهد تا محصولات موردنظر خود را از طریق جست‌وجو پیدا کرده و با مشاهده ویژگی‌های محصول و مقایسه آن با موارد مشابه و بررسی نظرات خریداران قبلی، به صورت مجازی انتخاب کرده، سفارش دهند و وجه آن را به صورت الکترونیکی پرداخت کنند. خرید از فروشگاه اینترنتی مزایای زیر را دارد:

- صرفه‌جویی در زمان
- خرید راحت محصولات
- مطالعه تمامی اطلاعات محصول قبل از خرید آن
- امکان خرید از فروشگاه‌های بزرگ و معتبر
- انتخاب بهترین محصول با استفاده از نظرات کاربران
- دریافت محصول به آسانی و درب منزل
- امکان پرداخت اینترنتی و همچنین پرداخت در محل تحویل

۲-۱۰- عملکرد فروشگاه اینترنتی کتاب

در این فصل یک سیستم فروشگاه اینترنتی کتاب مورد بررسی و تحلیل قرار گرفته است. برای این کار لازم است پس از مراجعه به فروشگاه‌های کتاب و مشاهده فرایند انتخاب و خرید کتاب و مصاحبه با فروشندگان و خریداران و همچنین با مشاهده شیوه کار فروشگاه‌های اینترنتی، شناخت کامل از عملکرد این نوع فروشگاه‌ها به دست آید.

در یک فروشگاه اینترنتی کتاب، محصول اصلی کتاب بوده که در معرض دید مراجعه‌کنندگان قرار می‌گیرد. افراد می‌توانند بر اساس دسته‌بندی موضوعی به کتاب‌ها دسترسی پیدا کرده و با جست‌وجوی کتاب موردنظر و مشاهده مشخصات کامل کتاب، آن را به سبد خرید خود اضافه کنند.

هر سفارش می‌تواند شامل یک یا چند کتاب باشد و مشتری پس از گشت و گذار در فروشگاه و تکمیل سبد خرید، در پایان با ثبت مشخصات خود و آدرس تحویل، سفارش خود را قطعی می‌نماید. همچنین مشتری می‌تواند در روزهای بعد نیز به سایت وارد شده و سفارشات قبلی خود را دیده و از وضعیت آن مطلع گردد. مدیران فروشگاه می‌توانند موضوعات و کتاب‌های هر موضوع و مشتریان سایت را مدیریت کرده و سفارشات مشتریان را مشاهده کنند. همچنین می‌توانند جزئیات هر کدام از سفارشات را نمایش داده یا ویرایش کنند و همچنین وضعیت سفارشات را تغییر دهند.

۳-۱۰- نقش‌های کاربران

در ادامه انواع کاربران سیستم شناسایی شده و عملکرد هر کدام از آنها مورد تحلیل و بررسی قرار می‌گیرد. با توجه به شرح عملکرد سیستم فروشگاه، کاربران می‌توانند دارای یکی از سه نقش زیر باشند:

- ۱ بازدیدکننده: کاربرانی که به سایت وارد شده و می‌توانند کتاب‌ها را جست‌وجو کرده و مشخصات آنها را مشاهده کرده و کتاب‌ها را به سبد خرید خود اضافه کنند.
 - ۲ مشتریان: کاربرانی که از سایت خرید داشته و مشخصات خود را وارد کرده و عضو سایت شده‌اند.
 - ۳ مدیران: کاربرانی که از مسئولین فروشگاه هستند و می‌توانند موضوعات، کتاب‌ها، مشتریان و سفارشات را مدیریت کنند.
- به‌طور کلی هر کدام از نقش‌های کاربری دارای امکانات زیر خواهند بود:

بازدیدکننده

- ۱ مشاهده صفحه اول سایت
- ۲ جست‌وجوی کتاب‌ها
- ۳ نمایش کتاب‌های مربوط به یک موضوع
- ۴ نمایش جزئیات کامل یک کتاب
- ۵ اضافه کردن کتاب به سبد خرید
- ۶ نمایش سبد خرید
- ۷ حذف کتاب از سبد خرید
- ۸ ثبت قطعی خرید

مشتری

- ۱ ورود به پنل شخصی
- ۲ تغییر مشخصات فردی
- ۳ تغییر کلمه عبور
- ۴ مشاهده لیست سفارشات قبلی
- ۵ نمایش جزئیات هر سفارش

مدیر فروشگاه

- ۱ مدیریت موضوعات سایت (نمایش، اضافه، حذف و ویرایش کتاب)
 - ۲ مدیریت کتاب‌ها (نمایش، اضافه، حذف و ویرایش کتاب)
 - ۳ مدیریت مشتریان (نمایش، اضافه، حذف و ویرایش مشتری)
- از طرفی دقت داشته باشید که مدیر فروشگاه، تمام امکانات مشتری و مشتری نیز تمام امکانات بازدیدکننده را در اختیار دارد.

۴-۱۰- طراحی بانک اطلاعاتی فروشگاه

برای طراحی بانک اطلاعاتی، با حضور در محیط فروشگاه کتاب و مشاهده قفسه‌ها و دسته‌بندی موضوعی کتاب، مشخصات کتاب‌ها، لیست فروش، دفترچه آدرس مشتریان و مشاهده سیستم‌های نرم‌افزاری موجود، موجودیت‌ها را شناسایی کرده و ویژگی‌های هر کدام و ارتباط بین آنها را مشخص می‌کنند. در یک فروشگاه کتاب اینترنتی، موجودیت‌های زیاد با ویژگی‌های متنوع و ارتباطات پیچیده وجود دارد که پرداختن به همه آنها از حوصله این کتاب خارج است. بنابراین برای راحتی و فهم بهتر، تنها چند موجودیت اصلی شامل کتاب‌ها، مشتریان و سفارشات و برای هر کدام چند ویژگی مهم در نظر گرفته شده است. همان‌گونه که در فصل نهم با روش تحلیل داده‌ها و طراحی بانک اطلاعاتی آشنا شدید، در این قسمت می‌توان برای یک فروشگاه اینترنتی کتاب، موجودیت‌ها، ویژگی‌های هر موجودیت و ارتباط بین موجودیت‌ها را شناسایی کرده و نمودار ER را رسم کنید.

موجودیت‌ها می‌توانند کتاب‌ها، مشتریان و سفارشات باشند:

■ موجودیت کتاب دارای ویژگی‌های نام کتاب، موضوع، نام مؤلف، قیمت، توصیف، تصویر روی جلد و وضعیت است. (موضوع کتاب باید از بین چند مقدار مشخص انتخاب شود و وضعیت، نمایش یا عدم نمایش کتاب به مراجعه‌کنندگان را تعیین می‌کند و دارای مقدار ۰ یا ۱ است.)

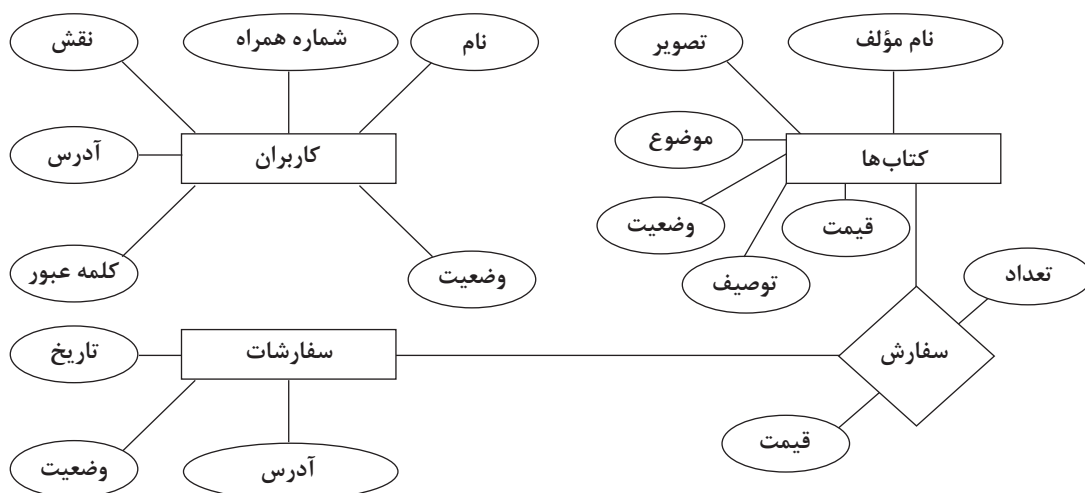
■ موجودیت کاربر یا مشتری دارای ویژگی‌های نام، شماره همراه، آدرس، نقش، کلمه عبور و وضعیت است. (نقش تعیین می‌کند که کاربر از مشتریان است یا کارکنان یا مدیران فروشگاه)

■ موجودیت سفارش دارای ویژگی‌های تاریخ سفارش و آدرس تحویل است. رابطه بین موجودیت‌ها به شرح زیر می‌باشد:

■ هر کتاب در چند سفارش درخواست می‌شود و هر سفارش شامل چند کتاب می‌باشد.

■ هر مشتری، می‌تواند چند سفارش داشته باشد.

برای داشتن تصویر خواناتر و فهم بهتر داده‌ها می‌توان از نمودار ER شکل ۱-۱۰ برای نمایش موجودیت‌ها و ارتباط آنها استفاده کرد:



شکل ۱-۱۰- نمودار ER فروشگاه کتاب

با توجه به نمودار صفحه قبل می‌توان جداول بانک اطلاعاتی فروشگاه اینترنتی کتاب را تعیین کرد. ابتدا یک جدول برای نگه‌داری موضوعات (مثل داستانی، روانشناسی، تاریخی و...) تعریف می‌کنیم که دارای ستون‌های شناسه و نام موضوع باشد:

جدول ۱-۱۰- موضوعات subs

نام ویژگی	نام فیلد	نوع داده
شناسه	sid	عددی
نام	sname	رشته‌ای

جدول ۱-۲- books: برای نگه‌داری مشخصات کتاب‌ها

نام ویژگی	نام فیلد	نوع داده
شناسه کتاب	bid	عددی
نام	bname	رشته‌ای
نام مؤلف	author	رشته‌ای
قیمت	price	عددی
شناسه موضوع	sid	عددی
توصیف کتاب	des	متن بلند
نام تصویر روی جلد	cover	رشته‌ای
وضعیت	status	عددی

جدول ۱-۳- users: برای نگه‌داری مشخصات کاربران و مشتریان

نام ویژگی	نام فیلد	نوع داده
شناسه کاربر	uid	عددی
نام	uname	رشته‌ای
شماره همراه	mobile	رشته‌ای
آدرس	addr	رشته‌ای
کلمه عبور	pass	رشته‌ای
نقش	role	رشته‌ای
وضعیت (فعال یا غیرفعال)	status	عددی

جدول ۴-۱۰: orders برای نگه‌داری سفارشات

نام ویژگی	نام فیلد	نوع داده
شناسه سفارش	oid	عددی
تاریخ	odate	تاریخ
شناسه کاربر مشتری	uid	عددی
آدرس پستی	addr	رشته‌ای
شناسه وضعیت	sid	عددی

برای وضعیت سفارش (sid) یکی از چند گزینه زیر در نظر گرفته شده است:

۱ منتظر پرداخت

۲ پرداخت شده

۳ در حال پردازش

۴ ارسال شده

۵ تحویل شده

۶ مرجوع شده

می‌توان همه این گزینه‌ها را در یک جدول جداگانه به نام states ذخیره کرده و شناسه آنها را در جدول سفارشات استفاده نمود.

جدول ۵-۱۰: states برای نگه‌داری وضعیت‌ها

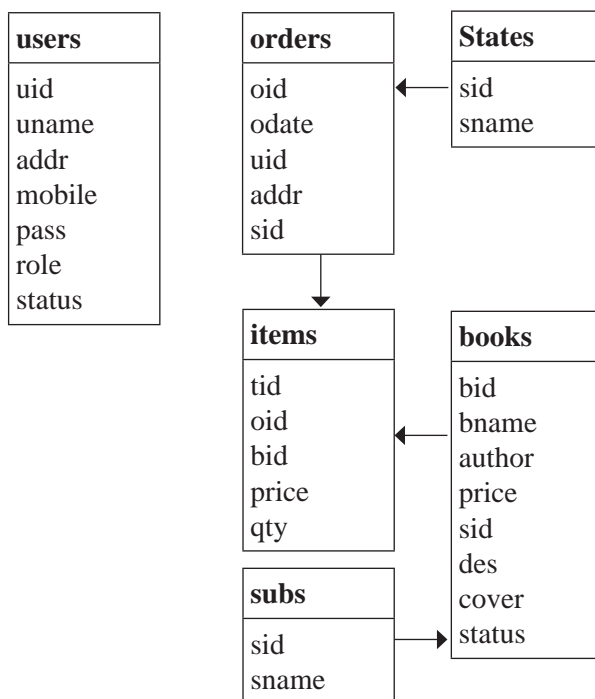
نام ویژگی	نام فیلد	نوع داده
شناسه وضعیت	sid	عددی
عنوان	sname	رشته‌ای

جدول ۶-۱۰: items برای نگه‌داری مشخصات اقلام سفارش

نام ویژگی	نام فیلد	نوع داده
شناسه آیتم	tid	عددی
شناسه سفارش	oid	عددی
شناسه کتاب	bid	عددی
قیمت واحد	price	عددی
تعداد	qty	عددی

چون رابطه بین کتاب و سفارش چند به چند است باید از یک جدول جداگانه برای نگهداری آن استفاده شود. به عبارت دیگر باید اطلاعات اقلام همه سفارشات را در یک جدول مشابه items ذخیره کرده و با استفاده از کلید خارجی oid در این جدول مشخص شود که هر آیتم مربوط به کدام سفارش می باشد. در جدول سفارشات و اقلام سفارش کلیدهای خارجی متناظر با کلیدهای اصلی در دیگر جداول قرار داده شده است. به عنوان مثال ستون uid یک کلید خارجی بوده و با ستون کلید اصلی uid در جدول users در ارتباط است. با استفاده از این کلید خارجی با داشتن شناسه یک کاربر می توان به لیست سفارشات او دسترسی پیدا کرد.

نمودار بانک اطلاعاتی فروشگاه در شکل ۱۰-۲ نشان داده شده است:



شکل ۱۰-۲- نمودار بانک اطلاعاتی فروشگاه

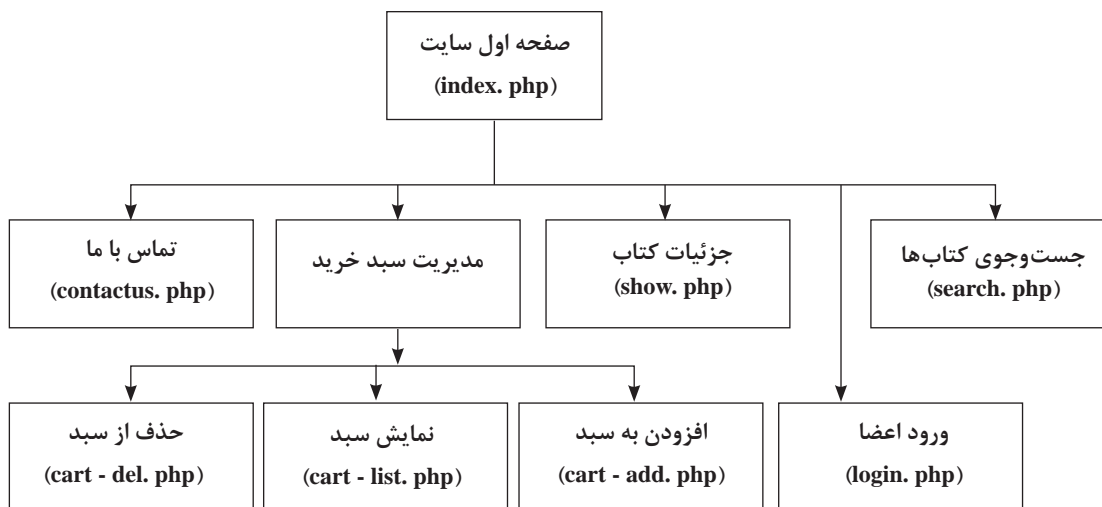
۱۰-۵- طراحی ساختار سایت

با توجه به امکانات سایت فروشگاه اینترنتی و نقش کاربران، مشخص می شود که سایت دارای دو مجموعه از صفحات است:

صفحات عمومی: صفحاتی که هر بازدیدکننده به آنها دسترسی دارد و نقش ویتترین فروشگاه را دارد. در این صفحات لیست موضوعات نشان داده شده و برای هر موضوع لیست کتابها نشان داده خواهد شد. همچنین امکان جست و جوی کتاب و کار با سبد خرید را فراهم خواهد کرد.

صفحات خصوصی یا مدیریتی: صفحاتی هستند که کاربر پس از ورود و احراز هویت به آنها دسترسی

پیدا می‌کند یا همان صفحاتی که در پنل شخصی هر کاربر نمایش داده شده که البته تعداد و نحوه نمایش آنها برای هر کاربر به نقش آن کاربر بستگی دارد. در این قسمت صفحات مختلف سایت تعیین شده و ارتباط آنها به صورت درختی نمایش داده می‌شود. در شکل ۱۰-۳ ساختار ارتباطی صفحات بخش عمومی سایت را مشاهده می‌کنید:



شکل ۱۰-۳-۱- ساختار درختی صفحات عمومی سایت

هر کاربر با ورود به سایت، ابتدا «صفحه اول» سایت را می‌بیند. در صفحه اول متن معرفی فروشگاه و لیست جدیدترین کتاب‌های اضافه شده به فروشگاه و لیست پر فروش ترین کتاب‌ها را مشاهده می‌کند. با کلیک روی هر کتاب «صفحه جزئیات کتاب» باز خواهد شد و در آن مشخصات کامل کتاب به همراه دکمه افزودن به سبد را خواهد دید.

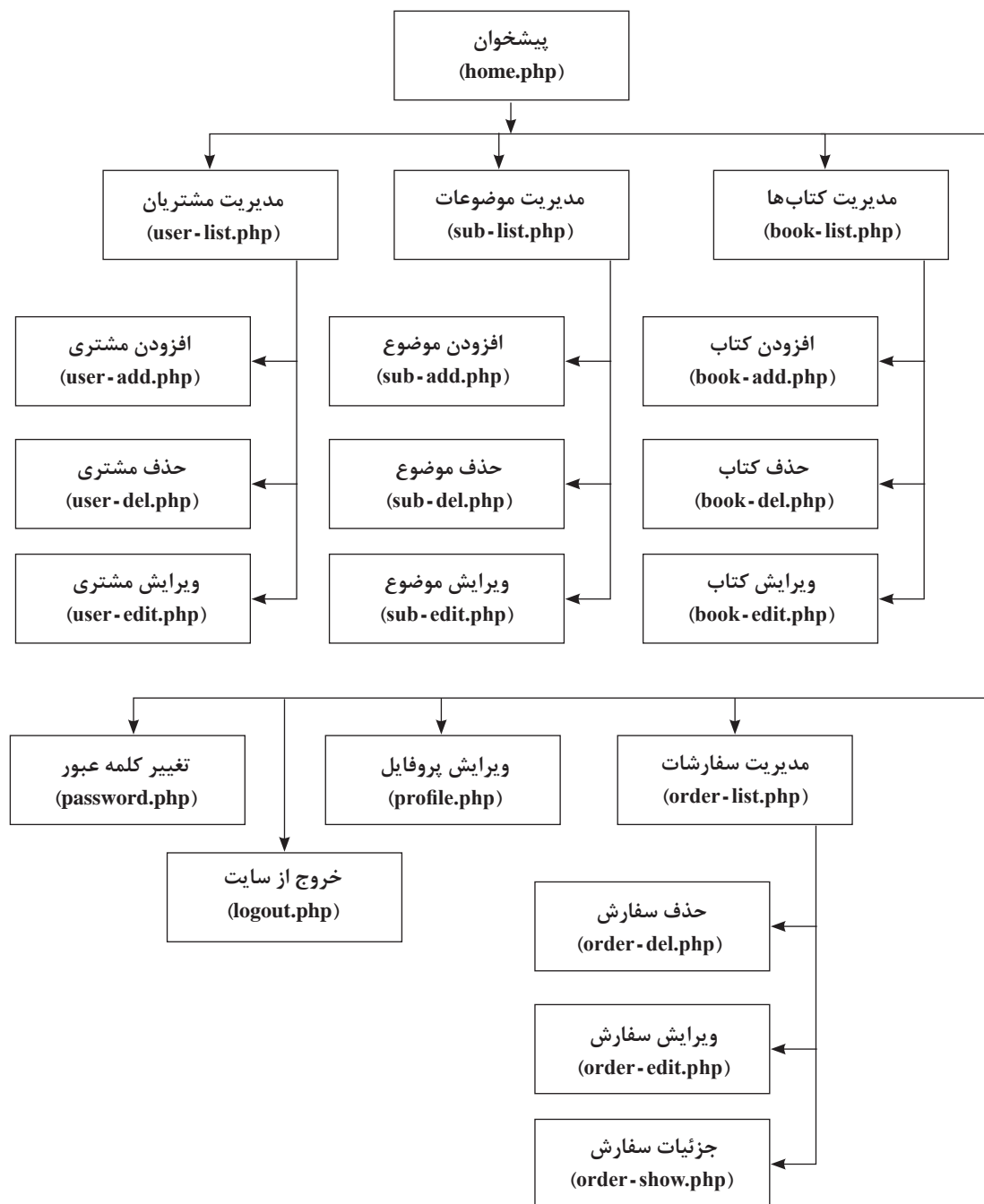
در هر جای سایت با کلیک روی «افزودن به سبد»، کتاب به سبد یا همان لیست خرید اضافه می‌شود. در بالای همه صفحات لینکی برای «مشاهده سبد خرید» وجود خواهد داشت که با کلیک روی آن، جزئیات سبد خرید را خواهند دید. در صفحه نمایش سبد خرید، امکان حذف یک قلم از سبد وجود خواهد داشت. برای قطعی کردن سبد خرید، لازم است تا کاربر به سیستم وارد شده یا در آن ثبت نام کند که این موضوع در صفحه نمایش خرید به او اعلام می‌شود.

در صفحه «جست و جوی کتاب»، فرمی برای جست و جوی کتاب بر اساس نام یا توصیف و یا بازه قیمتی نشان داده شده و در ذیل آن هم لیست کتاب‌های پیدا شده نمایش می‌شود.

در صفحه «تماس با ما»، روش‌های تماس با مسئولین فروشگاه از جمله، آدرس، تلفن و آدرس‌ها در شبکه‌های اجتماعی نشان داده می‌شود.

در بالای همه صفحات، لوگو و نام فروشگاه به همراه پیوندهایی برای دسترسی به صفحات مختلف قرار داده شده و در پایین همه صفحات، آدرس فروشگاه، تلفن، نمادها و مجوزهای قانونی فروشگاه برای ارتباط و نیز جلب اعتماد مشتریان نمایش داده می‌شود.

در شکل ۱۰-۴ صفحات بخش مدیریتی سایت و ارتباط بین آنها به صورت درختی نشان داده شده است:



شکل ۱۰-۴- ساختار درختی صفحات مدیریتی سایت

۱۰-۶- طراحی صفحات سایت

پس از تعیین صفحات مختلف فروشگاه و ارتباط بین آنها و نمایش آن به صورت درختی می‌بایست برای هر کدام از صفحات شکل کلی صفحه، اجزای آن و محل قرار گرفتن اجزا با رسم شکل تعیین شود. در شکل ۱۰-۵ طرح کلی صفحه اول سایت نشان داده شده است:

پیوندها به صفحات مختلف	لوگو و نام فروشگاه
لیست موضوعات	
جدیدترین کتاب‌ها	
نماد اعتماد و مجوزها	آدرس فروشگاه
حق کپی رایت	

شکل ۱۰-۵- طراحی صفحه اول سایت

در شکل ۱۰-۶ نمای کلی «صفحه جست‌وجوی کتاب» را مشاهده می‌کنید:

پیوندها به صفحات مختلف	لوگو و نام فروشگاه
لیست موضوعات	
فرم جست‌وجوی کتاب	
لیست کتاب‌ها	
نماد اعتماد و مجوزها	آدرس فروشگاه
حق کپی رایت	

شکل ۱۰-۶- طراحی صفحه جست‌وجوی کتاب

تصویر روی جلد کتاب
عنوان کتاب

در صفحات مختلف لیست کتاب‌ها دیده خواهد شد، برای نمایش هر کتاب از طراحی شکل ۷-۱۰ استفاده می‌شود:

شکل ۷-۱۰- طراحی جعبه نمایش کتاب

با توجه به طراحی صفحات عمومی سایت مثل صفحه اول و صفحه جست‌وجوی کتاب متوجه می‌شوید که همه صفحات عمومی دارای طرح کلی یکسان بوده و بخش‌های بالا و پایین صفحه برای همه صفحات مشابه هم است.

در شکل ۸-۱۰، طراحی صفحه مدیریت کتاب‌ها (book-list.php) نشان داده شده است.

<table border="1"> <tr> <td>عنوان صفحه</td> </tr> <tr> <td>فرم جست‌وجوی کتاب</td> </tr> <tr> <td>لیست کتاب‌ها</td> </tr> <tr> <td>نام طراح و تلفن پشتیبانی</td> </tr> </table>	عنوان صفحه	فرم جست‌وجوی کتاب	لیست کتاب‌ها	نام طراح و تلفن پشتیبانی	<table border="1"> <tr> <td>ستون ثابت کناری</td> </tr> <tr> <td>لوگو و نام فروشگاه</td> </tr> <tr> <td>لینک به صفحات مختلف مدیریتی</td> </tr> </table>	ستون ثابت کناری	لوگو و نام فروشگاه	لینک به صفحات مختلف مدیریتی
عنوان صفحه								
فرم جست‌وجوی کتاب								
لیست کتاب‌ها								
نام طراح و تلفن پشتیبانی								
ستون ثابت کناری								
لوگو و نام فروشگاه								
لینک به صفحات مختلف مدیریتی								

شکل ۸-۱۰- طراحی صفحه مدیریت کتاب‌ها

برای طراحی دیگر صفحات مدیریتی هم از همین طرح کلی استفاده کرده و فقط به جای فرم جست‌وجو و لیست کتاب‌ها محتوای هر صفحه قرار می‌گیرد.

در این فصل، یک سایت فروشگاه اینترنتی به صورت نمونه تحلیل و طراحی و مستندسازی گردید. این مستندات باید مورد توافق تیم طراحی سایت و نیز کارفرما قرار گرفته و به عنوان یک سند، در حین پیاده‌سازی، نصب و توسعه‌های بعدی سایت، مورد استفاده قرار گرفته و در صورت نیاز تغییر داده شده و به‌روز گردد.

خودآزمایی

- ۱ برای سایت خبرگزاری، موجودیت‌ها، ویژگی‌ها و ارتباط بین آنها را نوشته و نمودار ER آن را رسم کنید.
- ۲ با استفاده از نمودار ER سؤال قبل، بانک اطلاعاتی سایت خبرگزاری را طراحی کنید.
- ۳ برای سایت خبرگزاری، نقش‌های کاربری را تعیین کرده و امکانات هر کدام را لیست کنید.
- ۴ ساختار درختی صفحات عمومی برای سایت خبرگزاری را رسم کرده و طرح کلی هر کدام از آنها را رسم کنید.



فصل یازدهم

طراحی قالب برای صفحات فروشگاه کتاب



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- تنظیمات stylesheet را در صفحه استفاده کند.
 - فایل‌های چارچوب bootstrap را در صفحه معرفی کند.
 - از خواص stylesheet برای تنظیم اجزای صفحه استفاده کند.
 - از کلاس‌های bootstrap برای تنظیم اجزای صفحه استفاده کند.
 - یک صفحه قالب برای سایت طراحی کند.
 - در طراحی صفحات سایت از قالب آماده استفاده نماید.
- در فصل قبل عملکرد سیستم فروشگاه اینترنتی مورد بررسی قرار گرفته، انواع کاربران و فعالیت‌های هر کدام شناسایی شد و پس از بررسی داده‌های فروشگاه و رسم نمودار ER، جداول بانک اطلاعاتی طراحی گردید. همچنین ساختار صفحات عمومی و مدیریتی سایت مشخص شده و طرح کلی صفحات سایت با رسم شکل نشان داده شد. در این فصل دو قالب جداگانه برای استفاده در صفحات عمومی و مدیریتی سایت طراحی می‌شود.

۱۱- طراحی قالب سایت

قبل از شروع طراحی و پیاده‌سازی صفحات سایت می‌بایست قالب مشترک برای صفحات طراحی و آماده شود. یک قالب مجموعه‌ای از فایل‌ها شامل فایل‌های CSS، جاوا اسکریپت، فونت‌ها، تصاویر و طراحی بخش‌های مشترک صفحات است که می‌تواند به صورت آماده تهیه شده و یا توسط تیم پروژه طراحی شود. استفاده از قالب مشترک برای صفحات سایت باعث طراحی یکنواخت و سریع‌تر صفحات می‌شود. با توجه به طرح کلی برای صفحات عمومی و صفحات مدیریت برای سایت فروشگاه اینترنتی که در فصل قبل نشان داده شد، نیاز به دو قالب جداگانه است. قالب عمومی برای صفحات عمومی که در معرض دید مراجعه‌کنندگان سایت هستند و قالب خصوصی برای صفحات مدیریت رکوردها که در معرض دید مدیران سایت هستند.

برای طراحی قالب سایت به صورت کاربر پسند و واکنش‌گرا که بتواند روی دستگاه‌های مختلف تغییر شکل پیدا کرده و به خوبی نمایش داده شود، می‌توان از تنظیمات CSS و چارچوب طراحی واسط کاربری bootstrap استفاده کرد.

از CSS برای تعیین ویژگی‌هایی مثل اندازه‌ها، رنگ‌ها و ترازبندی برای هر کدام از اجزا استفاده می‌شود و چارچوب bootstrap، مجموعه‌ای از فایل‌ها شامل تنظیمات stylesheet و دستورات جاوا اسکریپت است که در طراحی قالب کلی سایت، چیدمان عناصر صفحه و اضافه کردن عناصر آماده مثل منوها، زبانه‌ها و کادرها به طراح کمک می‌کنند.

استفاده از چارچوب bootstrap کمک می‌کند تا بدون درگیر شدن با تنظیمات جزئی و خسته‌کننده، روی بخش‌های اصلی و کدنویسی صفحات تمرکز شده و صفحات حرفه‌ای در زمان کوتاه طراحی شوند. برای یادگیری CSS و bootstrap می‌توانید از سایت‌های آموزشی مختلف از جمله w3schools.com استفاده کرد.

۱۱-۲ طراحی قالب برای صفحات مدیریتی

در این فصل این دو قالب مجزا برای صفحات عمومی و صفحات مدیریتی طراحی شده و در فصل های بعد از آنها استفاده خواهد شد. برای رعایت سادگی و اختصار از قالب ساده‌ای مشابه شکل ۱۱-۱ برای صفحات عمومی سایت استفاده می‌شود:



شکل ۱۱-۱ قالب صفحات عمومی فروشگاه اینترنتی

۱۱-۲-۱ ساختار قالب

قالب فوق، دارای ۵ قسمت یا نوار افقی به صورت زیر است که برای ساخت هر کدام از یک تگ `div` به همراه تنظیمات `CSS` و `bootstrap` استفاده شده است:

- ۱ عنوان شامل لوگو و نام فروشگاه و لینک‌های مربوط به صفحات جست‌وجو، سبد خرید و...
 - ۲ نوار پیمایش سایت شامل لینک مربوط به موضوعات کتاب‌ها
 - ۳ ناحیه اصلی برای نمایش محتوای هر صفحه
 - ۴ پاصفحه اول شامل آدرس فروشگاه
 - ۵ پاصفحه دوم شامل توضیح حقوق سایت
- برای طراحی این قالب، از زیرپوشه‌ها و فایل‌های زیر استفاده شده است:
- ۱ پوشه `fonts`: شامل فایل‌های فونت که در صفحات استفاده شده‌اند.
 - ۲ پوشه `bs`: شامل فایل‌های چارچوب `bootstrap`
 - ۳ پوشه `images`: شامل تصاویر مثل لوگوی فروشگاه است.
 - ۴ فایل `pstyle.css`: شامل تنظیمات استایل شیت است.
 - ۵ فایل `pheader.php`: شامل طراحی سرصفحه و ستون کناری سایت است.
 - ۶ فایل `pfooter.php`: شامل طراحی پاصفحه سایت است.

در ادامه جزئیات طراحی و کدنویسی هر کدام از فایل‌ها نشان داده شده است. همان‌طور که در فصل پنجم بیان شد، برای طراحی قالب می‌توان ابتدا یک صفحه کامل طراحی کرده و سپس بخش‌هایی از آن که بین صفحات مختلف سایت مشترک است را در فایل‌های جداگانه با اسامی مثل `header` و `footer` ذخیره کرد. در ادامه برای رعایت اختصار، طراحی دو بخش `header` و `footer` ساخته شده از روی قالب اصلی بررسی شده است.

۲-۱۱- طراحی فایل `header`

در صفحه `pheader.php`، فایل‌های مربوط به چارچوب `bootstrap` و نیز فایل `pstyle.css` معرفی شده و طراحی بخش‌های بالای سایت (نوار عنوان و نوار پیمایش) انجام شده است:

صفحه `pheader.php`

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>فروشگاه کتاب اینترنتی</title>
<script src="bs/js/jquery.min.js"></script>
<script src="bs/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="bs/css/bootstrap.min.css">
<link rel="stylesheet" href="bs/css/bootstrap-rtl.min.css">
<link rel="stylesheet" href="pstyle.css"/>

</head>

<body>
<div class="container-fluid header">
  <div class="container">
    <div class="row">
      <div class="col-5">
        
      </div>
      <div class="col-7" style="text-align:left">
        <a href=" " >سید خرید</a> |
        <a href="search.php">جست‌وجو</a> |
        <a href="contactus.php">تماس با ما</a> |
        <a href="login.php">ورود اعضا</a>
      </div>
    </div>
  </div>
</body>
```

```

</div>
</div>
<div class="container-fluid navbar">
<div class="container">
<ul>
<li><a href=" " >صفحه اول</a></li>
<li> | <a href=" " >تاریخی</a></li>
<li> | <a href=" " >سیاسی</a></li>
<li> | <a href=" " >اقتصادی</a></li>
</ul>
</div>
</div>
<div class="container-fluid body">
<div class="container">

```

در صفحه pheader.php :

1 در قسمت header صفحه فایل های کتابخانه ای bootstrap و jquery معرفی شده اند:

```

<script src="bs/js/jquery.min.js"></script>
<script src="bs/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="bs/css/bootstrap.min.css">
<link rel="stylesheet" href="bs/css/bootstrap-rtl.min.css">

```

فایل jquery.js یک کتابخانه از جاوا اسکریپت است که برای پویا کردن صفحه کاربرد دارد و توسط bootstrap استفاده می شود.

فایل bootstrap.min.js شامل دستوراتی از bootstrap است که در ایجاد امکاناتی مثل منوها و کادرهای پاپ آپ استفاده می شود. فایل bootstrap.min.css تنظیمات استایل شیت و تعریف کلاس های bootstrap را شامل می شود و فایل bootstrap-rtl-min.css دارای تعاریف و کلاس هایی است که صفحه و اجزای صفحه را راست به چپ می کند و برای صفحات فارسی باید استفاده شود.

2 برای ایجاد نوارهای عنوان و پیمایش از تگ div با کلاس container-fluid از bootstrap استفاده شده که از عرض صفحه نمایش به طور کامل استفاده می کند و در داخل آن هم تگ div با کلاس container از بوت استرپ تا محتوا دارای حاشیه هایی از دو طرف صفحه باشد:

```

<div class="container-fluid">
<div class="container">

</div>
</div>

```

۲ برای نمایش لیست موضوعات در نوار پیمایش، از لیست غیرترتیبی استفاده شده و به ازای هر کدام از موضوعات یک تگ `li` و یک تگ `a` استفاده شده است:

```
<ul>
<li><a href="">صفحه اول</a></li>
<li> | <a href="">تاریخی</a></li>
<li> | <a href="">سیاسی</a></li>
<li> | <a href="">اقتصادی</a></li>
</ul>
```

۱۱-۲-۳ طراحی فایل footer

در نهایت فایل `pheader.php` به صورت زیر پایان یافته است که بدنه اصلی صفحه را شروع می کند:

```
<div class="container-fluid body">
<div class="container">
```

فایل `pfooter.php` دارای طراحی زیر است:

صفحه pfooter.php
<pre></div> </div> <div class="container-fluid footer1"> <div class="container"> <h5>آدرس فروشگاه</h5> تهران، منطقه ۵۰، محله ۴۳، خیابان یکم، تلفن ۰۱۲۳۴۵۶ </div> </div> <div class="container-fluid footer2"> <div class="container"> تمام حقوق این سایت متعلق به فروشگاه کتاب است. </div> </div> </body> </html></pre>

در صفحه `pfooter.php`، ابتدا تگ‌هایی که در انتهای `pheader.php` باز شده بود، بسته شده و دو نوار پاصفحه اول و پاصفحه دوم طراحی شده‌اند. در انتها هم تگ‌های اصلی `body` و `html` بسته شده‌اند. در صفحه `pheader.php` از فایل `stylesheet` خارجی با نام `pstyle.css` استفاده شده است. این فایل دارای

مجموعه‌ای از کلاس‌های stylesheet است که در طراحی فایل‌های سرصفحه و پاصفحه استفاده شده‌اند:

فایل pstyle.css

```
@font-face{
  font-family:IranSans;
  src:url('fonts/iransans.ttf') ;
}
body *{
  font-family: IranSans;
  font-size: 16px;
}

.header{
  padding: 10px 0px;
  background-color:#eee;
}

.navbar{
  height: 50px;
  background-color:#082d2a ;
  color:#fff;
}

.navbar ul {
  padding-right: 0px;
}

.navbar ul li{
  float: right;
  margin-left: 10px;
  list-style-type: none;
}

.navbar ul li a{
  color: #fff;
}

.body{
  padding: 20px 0px;
  min-height: 300px;
  background-color:#fff;
}
```

```

.footer1 {
  min-height: 130px;
  background: #082d2a;
  color:#fff;
  padding: 10px 0px;
}
.footer2 {
  height: 40px;
  padding: 10px 0px;
  background: #041818;
  color:#aaa;
}
.box {
  margin-top: 10px;
  padding: 10px;
  height: 320px;
  border : 1px solid #ddd;
}
.box img {
  width: 100%;
  height: 200px;
}

```

در ابتدای فایل `pstyle.css`، فونت جدیدی به نام `IranSans` تعریف شده و مسیر و نام فایل‌های فونت برای آن تعیین شده است.

```

@font-face {
font-family:IranSans;
src:url('fonts/iransans.ttf') ;
}

```

می‌توان برای استفاده از فونت دلخواه در طراحی سایت، فایل‌های فونت را در پوشه مشخصی قرار داده و با این روش نام فونت جدیدی تعریف کرده و در تنظیمات سایت استفاده نمود. در این صورت هنگام نمایش صفحات، فایل‌های فونت هم بارگیری شده و در صفحه استفاده خواهد شد.

بعد از طراحی فایل‌های قالب سایت، نوبت به استفاده از آنها می‌رسد. صفحه temp.php که نمایش آن را در شکل ۱۱-۱ شاهد بودید، دارای طراحی زیر می‌باشد:

صفحه temp.php
<pre><?php include('pheader.php'); ?></pre> <p>بدنه صفحه</p> <pre><?php include('pfooter.php'); ?></pre>

در صفحه فوق، ابتدا فایل pheader.php ضمیمه شده، در ادامه محتوای صفحه قرار می‌گیرد و در پایان هم فایل pfooter.php ضمیمه می‌شود. در نتیجه در زمان اجرای temp.php مفسر PHP ابتدا محتوای فایل pheader.php سپس طراحی بدنه صفحه و در انتها هم محتوای فایل pfooter.php را در خروجی قرار می‌دهد. بنابراین صفحه‌ای که تحویل مرورگر شده و نمایش داده می‌شود از کنار هم قرار گرفتن محتوای فایل سرصفحه، محتوای صفحه و فایل پاصفحه به دست می‌آید و این در حالی است که مرورگر و کاربری که صفحه را می‌بیند، متوجه این موضوع نخواهد شد.

۳-۱۱- طراحی قالب برای صفحات مدیریتی

در این قسمت یک قالب کامل با استفاده از تگ‌های div و تنظیمات CSS و bootstrap برای استفاده در صفحات خصوصی یا مدیریتی سایت فروشگاه طراحی کرده و از روی آن فایل‌های جداگانه برای نمایش سرصفحه و پاصفحه ایجاد می‌شود.

۱-۳-۱۱- ساختار قالب

قالب خصوصی طراحی شده برای پانل مدیریت فروشگاه کتاب دارای ظاهر مشابه شکل ۱۱-۲ می‌باشد:



شکل ۱۱-۲- قالب برای صفحات مدیریتی فروشگاه

برای طراحی قالب فوق، از زیرپوشه‌ها و فایل‌های زیر استفاده شده است:

۱ پوشه **fonts**: شامل فایل‌های فونت که در صفحات استفاده شده‌اند.

۲ پوشه **bs**: شامل فایل‌های چارچوب bootstrap

۳ پوشه **images**: شامل تصاویر

۴ فایل **style.css**: شامل تنظیمات استایل شیت

۵ **header.php**: شامل طراحی سرصفحه و ستون کناری سایت

۶ **footer.php**: شامل طراحی حاوی پاصفحه

در ادامه محتوای هر کدام از فایل‌ها نشان داده شده است:

۲-۳-۱۱- طراحی header

در صفحه **header.php**، فایل‌های مربوط به چارچوب bootstrap و نیز فایل **style.css**، معرفی شده و طراحی بخش بالایی و ستون کناری صفحه انجام شده است:

صفحه **header.php**

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>فروشگاه کتاب اینترنتی</title>
<script src="bs/js/jquery.min.js"></script>
<script src="bs/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="bs/css/bootstrap.min.css">
<link rel="stylesheet" href="bs/css/bootstraprtl.min.css">
<link rel="stylesheet" href="style.css"/>

</head>

<body>
<div class="container-fluid">
<div class="row">
<div class="col-sm-2 sidebar">
<h3>
<br>
فروشگاه کتاب
</h3>
<ul >
<li><a href="home.php">پیشخوان</a></li>
<li><a href="profile.php">پروفایل</a></li>
```

```

<li><a href="password.php">تغییر کلمه عبور</a></li>
<li><a href="my-orders.php">سفارشات من</a></li>
<li><a href="book-list.php">مدیریت کتابها</a></li>
<li><a href="sub-list.php">مدیریت موضوعات</a></li>
<li><a href="user-list.php">مدیریت مشتریان</a></li>
<li><a href="order-list.php">مدیریت سفارشات</a></li>
</ul>
</div>
<div class="col-sm-10 main">
<div class="header">
<?=$page_title ?>
<span class="user">نام کاربری</span>
</div>
<div class="content">

```

در طراحی صفحه فوق:

۱ در قسمت header صفحه، فایل‌های کتابخانه‌ای bootstrap و jquery معرفی شده‌اند:

```

<script src="bs/js/jquery.min.js"></script>
<script src="bs/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="bs/css/bootstrap.min.css">
<link rel="stylesheet" href="bs/css/bootstrap-rtl.min.css">

```

۲ در طراحی کلی صفحه از یک تگ div با کلاس container_fluid از bootstrap استفاده شده که عرض صفحه نمایش را به‌طور کامل استفاده می‌کند. برای ایجاد ستون کناری و بدنه صفحه برای استفاده از شبکه‌بندی bootstrap، از تگ‌های div با کلاس‌های row و col استفاده شده است:

```

<div class="container-fluid">
<div class="row">
<div class="col-sm-2">ستون کناری</div>
<div class="col-sm-8">بدنه صفحه</div>
</div>
</div>

```

۳ برای طراحی منو در ستون کناری از لیست غیرترتیبی استفاده شده و با کمک قواعد stylesheet، تنظیمات آن تعیین شده است. این منو دارای لینک‌هایی برای صفحات مختلف بخش مدیریتی فروشگاه می‌باشد:

```

<ul >
<li><a href="home.php">پیشخوان</a></li>
<li><a href="profile.php">پروفایل</a></li>
<li><a href="password.php">تغییر کلمه عبور</a></li>
<li><a href="my-orders.php">سفارشات من</a></li>

```

```

<li><a href="book-list.php">مدیریت کتاب‌ها</a></li>
<li><a href="sub-list.php">مدیریت موضوعات</a></li>
<li><a href="user-list.php">مدیریت مشتریان</a></li>
<li><a href="order-list.php">مدیریت سفارشات</a></li>
</ul>

```

۴ در بالای صفحه، متغیر `$page_title`، که حاوی عنوان صفحه است نمایش داده شده و سپس یک فضا برای نمایش نام کاربر جاری، ایجاد شده است.

۵ در پایان برای نمایش محتوای صفحه یک تگ `div` با کلاس `content` باز شده است.

۳-۱۱- طراحی footer

فایل `footer.php` دارای طراحی زیر است:

صفحه footer.php
<pre> </div> <div class="row footer"> این سایت برای فروش کتاب طراحی شده است </div> <div> </div> </div> </body> </html> </pre>

در صفحه فوق ابتدا تگ با کلاس `content` که در `header.php` باز شده بود، بسته شده و سپس تگ `div` با کلاس `footer` به صورت زیر قرار می‌گیرد:

```

<div class="row footer">
<div class="col-sm-12">
    این سایت برای فروش کتاب طراحی شده است
</div>
</div>

```

در ادامه هم دو تگ‌های `div` و تگ‌های اصلی `body` و `html` که قبلاً در فایل `header.php` باز شده‌اند، بسته می‌شوند.

بعد از طراحی فایل‌های قالب سایت نوبت به استفاده از آنها می‌رسد. صفحه temp.php که نمایش آن را در شکل ۱۱-۲ شاهد بودید، دارای طراحی زیر می‌باشد:

صفحه temp.php
<pre><?php \$page_title="عنوان صفحه"; include('header.php');?> بدنه صفحه <?php include('footer.php');?></pre>

در صفحه فوق، ابتدا متغیر \$page_title مساوی عنوان صفحه قرار داده شده تا در داخل header نمایش داده شود سپس فایل header.php ضمیمه شده در ادامه محتوای صفحه قرار می‌گیرد و در پایان هم فایل footer.php ضمیمه می‌شود. پس از این در تمام صفحات فروشگاه به همین ترتیب عمل خواهد شد.

۱۱-۳-۴ طراحی استایل شیت

در فایل style.css مجموعه‌ای از کلاس‌های استایل شیت نوشته شده‌اند که در صفحات header و footer و نیز محتوای صفحات استفاده می‌شوند.

فایل style.css
<pre>@font-face{ font-family:IranSans; src:url('fonts/iransans.ttf') ; } body *{ font-family: IranSans; font-size:16px; } .sidebar{ background-color:#555; color:#fff; min-height: 900px; text-align: center; } .sidebar h3{ font-size:30px; font-weight: bold; } .sidebar.logo{ width:100px; height:100px;</pre>

```
}  
.sidebar ul{  
text-align: right;  
padding-right: 10px;  
margin-top: 50px;  
}  
  
.sidebar ul li {  
list-style-type: none;  
padding: 7px 0px 7px 0px;  
border-bottom: 1px solid #aaa;  
}  
.sidebar ul li a{  
color: #fff;  
}  
  
.main.header{  
padding: 10px;  
font-size: 30px;  
color:#555;  
font-weight: bold;  
}  
.main.header.user{  
float: left;  
font-size: 16px;  
font-weight: normal;  
}  
.main.content{  
padding: 10px;  
border-top: 1px solid #eee;  
padding-bottom: 50px;  
}  
.main.footer{  
padding: 10px;  
background-color: #eee;  
border-top: 1px solid #eee;  
position: fixed;  
bottom: 0px;  
width: 100%;  
color:#999;  
}  
form.row{  
padding-bottom: 10px;  
}
```


برای استفاده از فایل‌های قالب در صفحات مدیریتی مشابه صفحه `home.php` عمل شده است:

صفحه `home.php`

```
<?php
$page_title="عنوان صفحه";
include('header.php'); ?>
محتوای صفحه
<?php include('footer.php'); ?>
```

برای سازماندهی بهتر فایل‌ها می‌توان فایل‌های قالب عمومی به علاوه همه صفحات عمومی سایت را در پوشه اصلی سایت و فایل‌های قالب مدیریتی و همه صفحات مدیریتی را در زیر پوشه دلخواهی از سایت قرار داد.

محل قرارگیری فایل‌های قالب عمومی + صفحات عمومی `site_folder`
محل قرارگیری فایل‌های قالب مدیریتی + صفحات مدیریتی `admin_folder`

تذکر



در این فصل قالب‌های کلی برای استفاده در طراحی صفحات عمومی و مدیریتی سایت فروشگاه اینترنتی طراحی گردید. در فصل‌های بعد برای طراحی صفحات مختلف مربوط به این فروشگاه از این فایل‌های قالب استفاده خواهد شد. بنابراین توصیه می‌شود قبل از شروع فصل‌های بعد از درک کامل فایل‌های قالب صفحات سایت و نحوه استفاده از آنها اطمینان حاصل کنید.

خودآزمایی

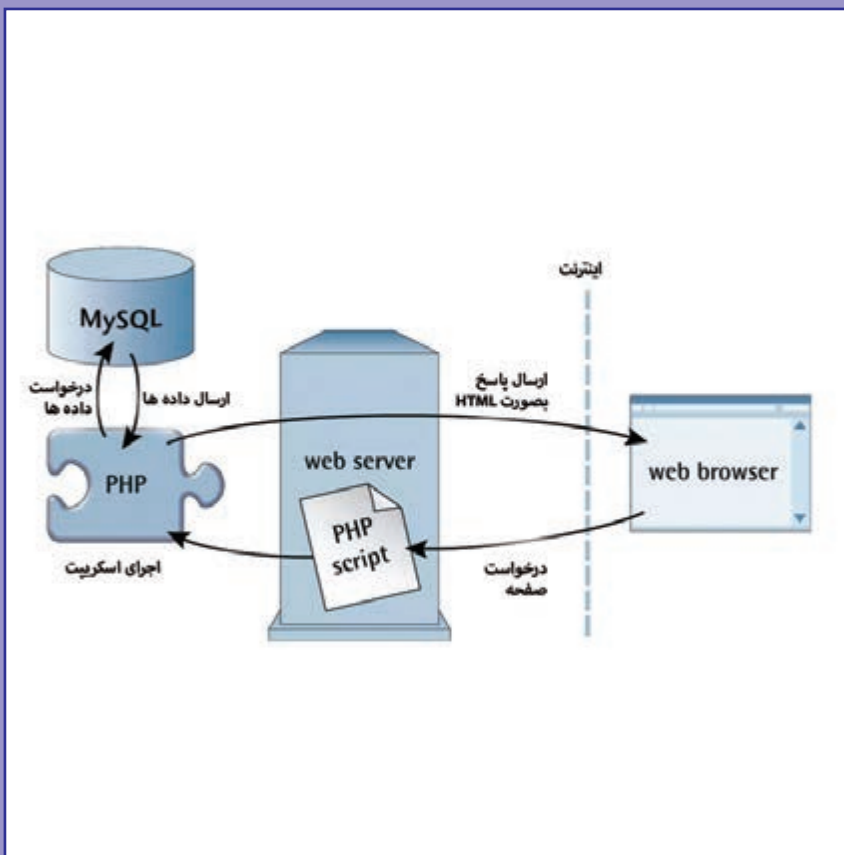
- ۱ دلایل استفاده از قالب یکسان برای صفحات سایت را شرح دهید.
- ۲ برای استفاده از فونت دلخواه در طراحی سایت و امکان بارگیری آن در زمان نمایش صفحه باید چه تنظیماتی در stylesheet انجام دهیم؟
- ۳ کاربرد CSS و bootstrap در طراحی صفحات وب چیست؟
- ۴ چه فایل‌هایی از bootstrap باید در بخش head صفحه معرفی شوند، تگ‌های مربوطه را بنویسید.

دستور کار عملی

- ۱ یک قالب دلخواه برای صفحات سایت طراحی کنید و فایل‌های header.php و footer.php را برای آن ایجاد کنید.
- ۲ با جست‌وجو در اینترنت قالب‌های رایگان برای طراحی سایت با استفاده از bootstrap پیدا کرده و از روی آنها فایل‌های header و footer را ایجاد کنید. (راهنمایی: برای این کار کافی است در سایت گوگل عبارت free bootstrap templates را جست‌وجو کنید)
- ۳ با گشت‌وگذار در اینترنت چند سایت دلخواه را بررسی کرده و با توجه به قالب استفاده شده برای صفحات، قسمت‌های مشترک بین صفحات سایت را پیدا کنید.

فصل دوازدهم

اتصال به بانک اطلاعاتی MySQL و نمایش اطلاعات



هدفهای رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- تکنولوژی‌های اتصال به بانک اطلاعاتی MySQL را توضیح دهد.
 - با تکنولوژی MySQLi به بانک اطلاعاتی متصل شود.
 - رکوردهای جدول بانک اطلاعاتی MySQL را خوانده و نمایش دهد.
- در فصل هشتم و نهم روش طراحی بانک‌های اطلاعاتی و مدیریت آنها با استفاده از نرم‌افزار phpMyAdmin آموزش داده شد. همچنین در فصل دهم و یازدهم یک سایت فروشگاه اینترنتی تحلیل و طراحی شد و فایل‌های قالب برای استفاده در صفحات مختلف آن طراحی گردید.
- در همه مثال‌های برنامه‌نویسی این فصل و فصل‌های بعد از جداول بانک اطلاعاتی فروشگاه اینترنتی استفاده می‌شود و برای پیاده‌سازی و اجرای مثال‌ها لازم است، ابتدا بانک اطلاعاتی با نام bookshop در محیط phpMyAdmin به صورت کامل ایجاد شود و از نظر نام بانک اطلاعاتی، نام جداول و نام ستون‌ها به صورت دقیق با طراحی ارائه شده در فصل دهم مطابقت داشته باشد.

۱۲-۱- دسترسی به بانک اطلاعاتی MySQL

زبان PHP دارای مجموعه‌ای از دستورات برای ارتباط با هر سیستم خارجی از جمله MySQL می‌باشد. در این فصل به شرح دستوراتی پرداخته می‌شود که با بانک اطلاعاتی موجود در MySQL ارتباط برقرار کرده و با ارسال درخواست‌ها به آن، عملیات مورد نظر روی بانک اطلاعاتی انجام می‌دهند.

به طور کلی در هر کدام از صفحات برای کار با یک سیستم بانک اطلاعاتی مثل MySQL مراحل زیر انجام می‌گیرد:

- ۱ یک اتصال به سرویس دهنده بانک اطلاعاتی (MySQL) ایجاد می‌شود.
 - ۲ بانک اطلاعاتی مورد نظر انتخاب می‌شود.
 - ۳ درخواست (دستور با زبان SQL) به بانک اطلاعاتی ارسال می‌شود.
 - ۴ اگر درخواست خواندن رکوردها (دستور select) اجرا شده باشد، نتایج به دست آمده از آن استخراج می‌گردد.
 - ۵ اتصال به سرویس دهنده بانک اطلاعاتی بسته می‌شود.
- برای اتصال به بانک اطلاعاتی MySQL، باید توابع مشخصی از PHP اجرا گردد که در نسخه‌های مختلف PHP این توابع ممکن است دچار تغییراتی شوند.
- در جدیدترین نسخه PHP (نسخه ۷) دو روش برای اتصال به سرویس دهنده MySQL وجود دارد:

- ۱ روش PDO
- ۲ روش MySQLi

روش PDO (یا PHP Data Objects): یک روش عمومی برای اتصال به انواع مختلف بانک‌های اطلاعاتی است. در حالی که MySQLi فقط می‌تواند با بانک اطلاعاتی MySQL کار کند. بنابراین اگر احتمال می‌رود که در آینده از بانک اطلاعاتی دیگری استفاده شود، PDO انتخاب بهتری خواهد بود در عوض استفاده از MySQLi دارای سرعت و کارایی بالاتری می‌باشد.

روش MySQLi: دارای دو نسخه شیء‌گرا و روال‌گرا است که در این کتاب برای راحتی کار از نسخه روال‌گرا استفاده شده است.

همه روش‌های فوق، دستورات SQL را برای اجرا به MySQL می‌فرستند و قبل از اجرای آن از نظر نداشتن کدهای مخفی و مخرب بررسی کرده و دارای امنیت بالا هستند.

۲-۱۲- دستورات اتصال به بانک اطلاعاتی

می‌دانید که اولین مرحله برای دسترسی به بانک اطلاعاتی، ایجاد اتصال به بانک اطلاعاتی است. در این قسمت دستور لازم برای اتصال به بانک اطلاعاتی با روش‌های مختلف نشان داده شده است.

قطعه کد زیر از MySQLi شیء‌گرا برای اتصال به سرور بانک اطلاعاتی استفاده می‌کند:

```
$link = new mysqli('servername:port', 'username', 'password');
```

قطعه کد زیر از MySQLi روال‌گرا برای اتصال به سرور بانک اطلاعاتی استفاده می‌کند:

```
$link = mysqli_connect('servername:port', 'username', 'password', ['dbname']);
```

نوشتن نام بانک اطلاعاتی (database) در دستور فوق اختیاری است. همچنین نوشتن شماره پورت در صورتی که MySQL از پورت پیش‌فرض با شماره ۳۳۰۶ استفاده کند، اختیاری است.

از قطعه کد زیر برای اتصال به سرور بانک اطلاعاتی MySQL با روش PDO استفاده شده است:

```
$link = new PDO('mysql:host=servername;dbname=mydb', 'username', 'password');
```

توجه کنید که مثال PDO بالا از بانک اطلاعاتی mydb استفاده می‌کند.

PDO یک بانک اطلاعاتی لازم دارد تا به آن متصل شود و اگر این بانک اطلاعاتی نباشد خطا رخ خواهد داد. در این بخش از روش MySQLi روال‌گرا برای اتصال به بانک اطلاعاتی و کار با آن استفاده شده و دستورات آن توضیح داده می‌شود.

۳-۱۲- مراحل لازم برای کار با بانک اطلاعاتی

برای اتصال به بانک اطلاعاتی و انجام عملیات روی آن نیاز است تا همه یا تعدادی از مراحل زیر انجام شود:

مرحله ۱- ایجاد اتصال به سرویس دهنده اطلاعاتی

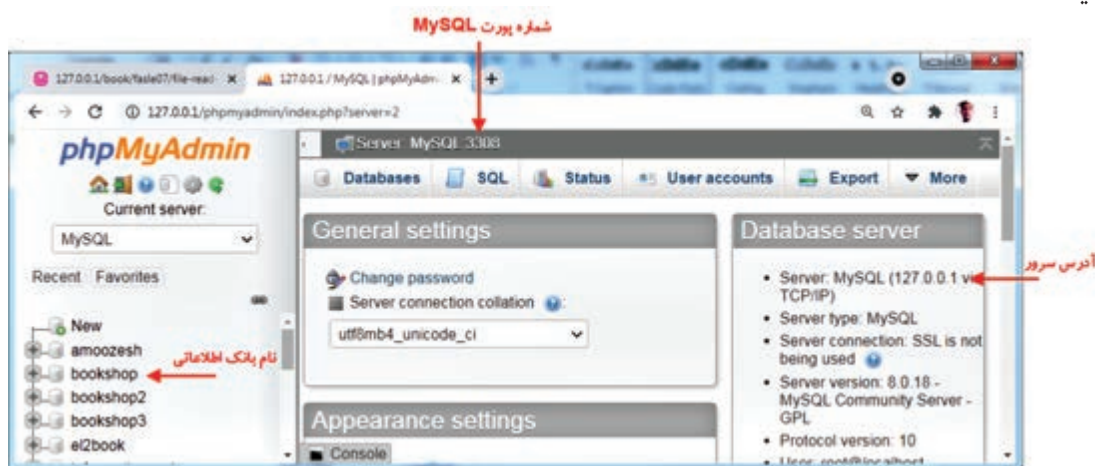
```
$link = mysqli_connect(servername,port,username,password,dbname);
```

پارامترهای استفاده شده در این تابع در جدول ۱۲-۱ توضیح داده شده‌اند:

جدول ۱۲-۱

پارامتر	کاربرد
servername	نام سرویس دهنده بانک اطلاعاتی است و برای اتصال به MySQL موجود روی سیستم محلی از localhost یا 127.0.0.1 استفاده می‌شود.
port	شماره پورت MySQL که به طور پیش فرض ۳۳۰۶ می‌باشد.
username	نام کاربر بانک اطلاعاتی است و می‌توان از نام کاربر پیش فرض root استفاده کرد.
password	کلمه عبور کاربر است، به طور پیش فرض برای کاربر root، پوچ "" تعیین می‌شود.
dbname	نام بانک اطلاعاتی که اختیاری است و می‌تواند تعیین نشود.

برای به دست آوردن نام سرویس دهنده و شماره پورت از برنامه phpMyAdmin مطابق شکل ۱۲-۱ استفاده کنید:



شکل ۱۲-۱- مشخصات سرویس دهنده داده

لازم به یادآوری است، هنگام نصب MySQL به طور پیش فرض یک کاربر به نام root و بدون کلمه عبور برای آن تعریف می‌شود.

مثال: اتصال به بانک اطلاعاتی bookshop روی سرویس دهنده بانک اطلاعاتی localhost با پورت شماره ۳۳۰۶ با نام کاربری root بدون کلمه عبور

```
$link = mysqli_connect("localhost3308:", "root", "", "bookshop");
```

اگر این دستور درست اجرا شود، یک شیء اتصال و در غیر این صورت مقدار پوچ برگشت داده شده و در متغیر \$link قرار می گیرد. برای بررسی درستی یا خطا بعد از این دستور می توان \$link را بررسی کرد:

```
if(!$link)
```

```
die("خطا در اتصال به بانک اطلاعاتی");
```

تابع die باعث می شود تا پیغام نمایش داده شده و بلافاصله اجرای برنامه قطع شود.

مرحله ۲- انتخاب بانک اطلاعاتی

در تابع اتصال به سرویس دهنده بانک اطلاعاتی می توان نام بانک اطلاعاتی را هم تعیین کرد. اگر در آن تابع نام بانک اطلاعاتی تعیین نشود بعد از اتصال باید با دستور زیر بانک اطلاعاتی (مثلاً بانک اطلاعاتی bookshop) انتخاب شود:

```
$db = mysqli_select_db($link, "bookshop");
```

اگر بانک به درستی انتخاب نشود مقدار پوچ در متغیر \$db قرار می گیرد که می توان آن را بررسی کرده و در صورت پوچ بودن، پیغام خطا نمایش داد.

```
if(!$db)
```

```
die("بانک اطلاعاتی نمی تواند انتخاب شود");
```

مرحله ۳- اجرای درخواست

برای انجام عملیات روی بانک اطلاعاتی باید درخواست خود را با زبان SQL تعیین کرده و با کمک تابع mysqli_query به MySQL ارسال کنیم تا نتیجه برگشت داده شده و در متغیر قرار گیرد:

```
$result = mysqli_query($link, "دستور SQL");
```

در فصل های قبلی با دستورات مهم زبان SQL آشنا شده اید و می دانید که برای مثال دستور select برای خواندن سطرهای جدول استفاده می شود.

مثال: خواندن تمام سطرهای جدول books

```
$result = mysqli_query($link, "SELECT * FROM books");
```

مرحله ۴- استخراج نتایج پرس و جو

چنانچه دستور SELECT روی بانک اطلاعاتی اجرا شود، نتیجه آن یک شیء است که از طریق آن می توان سطرها را به دست آورد. برای به دست آوردن اطلاعات هر کدام از سطرها می توان از دستورات زیر استفاده کرد:

```
$row = mysqli_fetch_array($result);
```

دستور فوق، یک سطر از نتیجه را استخراج کرده در آرایه شمارشی \$row قرار می دهد.

```
$row = mysqli_fetch_assoc( $result );
```

دستور فوق یک سطر از نتیجه را استخراج کرده در آرایه انجمنی \$row قرار می‌دهد که کلیدهای آن، اسامی ستون‌های جدول هستند.

همچنین این دستورات نشانگر سطر جاری را به بعدی منتقل می‌کنند و چنانچه دوباره اجرا شوند، اطلاعات سطر بعدی را استخراج خواهند کرد. برای استخراج همه سطرها دستورات فوق را در حلقه تکرار قرار می‌دهند.

مرحله ۵- بستن یک اتصال

بعد از انجام عملیات روی بانک اطلاعاتی بهتر است اتصال بسته شود:

```
mysqli_close($link);
```

هر چند هنگام پایان یافتن اجرای اسکریپت به صورت خودکار اتصال بسته خواهد شد.

تذکر



زبان SQL نسبت به کوچک و بزرگی حروف حساس نیست، به عنوان مثال نوشتن select یا SELECT برای آن فرقی نکرده و هر دو را اجرا می‌کند ولی برای نوشتن نام کاربری، کلمه عبور و اسامی بانک اطلاعاتی، جداول و ستون‌ها باید حروف از نظر کوچکی و بزرگی به صورت دقیق، همان‌طور که در بانک اطلاعاتی تعریف شده‌اند، نوشته شوند.

۴-۱۲- توابع کار با بانک اطلاعاتی

توابع PHP برای کار با بانک اطلاعاتی MySQL در جدول ۲-۱۲ نشان داده شده‌اند، این توابع در این فصل و فصل‌های بعد استفاده شده و عملکرد آنها به صورت کامل تشریح خواهد شد.

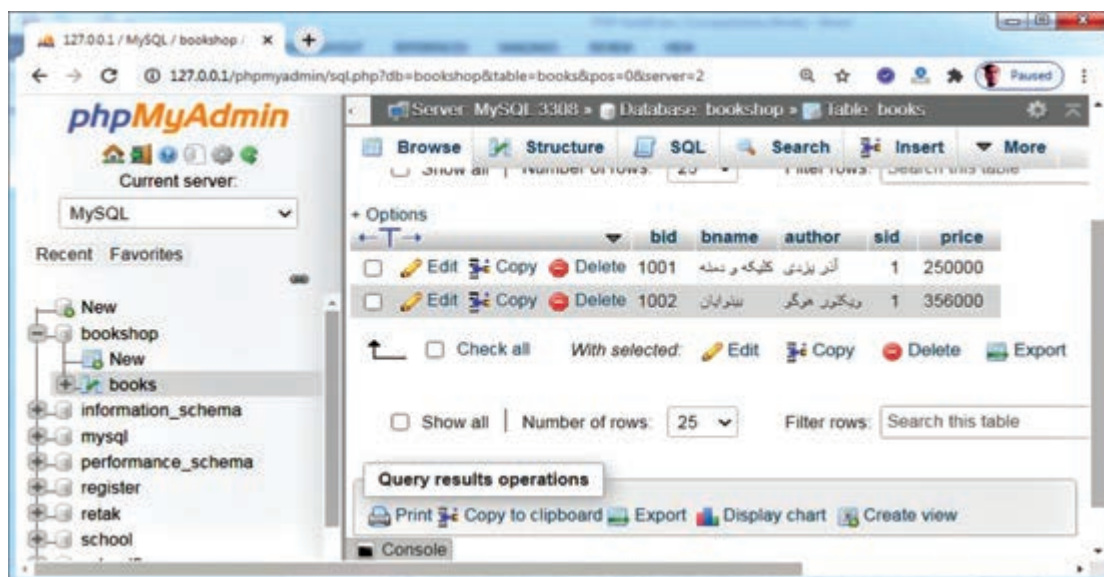
جدول ۲-۱۲

کاربرد	نام تابع
ایجاد اتصال به بانک اطلاعاتی	mysqli_connect
تعیین بانک اطلاعاتی جاری	mysqli_select_db
اجرای دستورات SQL روی بانک اطلاعاتی	mysqli_query
واکشی یک رکورد از جدول و قرار دادن آن در آرایه شمارشی	mysqli_fetch_array
واکشی یک رکورد از جدول و قرار دادن آن در آرایه انجمنی	mysqli_fetch_assoc
نمایش پیغام خطای بانک اطلاعاتی	mysqli_error
بستن یک اتصال به بانک اطلاعاتی	mysqli_close
برگرداندن آخرین شناسه خود افزایشی ایجاد شده	mysqli_insert_id

۵-۱۲- نمایش اطلاعات جدول

پیش تر روش اتصال به بانک اطلاعاتی MySQL، اجرای درخواست‌ها روی آن و استخراج نتایج را فرا گرفتید. در این قسمت با استفاده از دستورات فوق برنامه‌ای برای نمایش اطلاعات جداول از بانک اطلاعاتی می‌نویسید. برای پیاده‌سازی برنامه‌های بعدی از بانک اطلاعاتی bookshop مربوط به فروشگاه کتاب که در فصل‌های قبل ساختار آن طراحی شد، استفاده می‌شود. بنابراین قبل از هر کاری باید در محیط phpMyAdmin این بانک اطلاعاتی ایجاد شده، جداول مربوطه به آن اضافه شوند و برای هر جدول چند سطر اطلاعات دلخواه در آنها درج شود.

مثال ۱: با فرض اینکه مشابه شکل ۲-۱۲، جدول books در بانک اطلاعاتی bookshop، حاوی اطلاعات کتاب‌ها می‌باشد، صفحه‌ای برای نمایش لیست کتاب‌ها پیاده‌سازی کنید.



شکل ۲-۱۲- جدول کتاب‌ها در بانک اطلاعاتی

در صفحه book-list.php دستوراتی برای اتصال به بانک اطلاعاتی، اجرای دستور SELECT روی جدول books، خواندن اطلاعات و نمایش نام و قیمت کتاب‌ها در صفحه وب نوشته شده است:

برنامه book-list.php

```
<html>
<body>
<?php
$link=mysqli_connect("localhost:3308","root","","bookshop");
if(! $link ) die("خطا در اتصال به بانک");
$result=mysqli_query($link,"SELECT * FROM books");
```

```

while($row = mysqli_fetch_assoc( $result )) {
    echo ("<br>NAME:". $row['bname']);
    echo ("<br>PRICE:". $row['price']);
}
?>
</body>
</html>

```

خروجی کد فوق به صورت زیر است:

```

NAME: کلیله و دمنه
PRICE:250000
NAME: بینوایان
PRICE:356000
NAME: اقلیم هشتم
PRICE:250000
NAME: مبانی سیاست
PRICE:230000

```

در کد فوق، از تابع `mysqli_connect` برای اتصال به سرویس دهنده داده (MySQL) استفاده شده است:

```
$link =mysqli_connect("localhost:3308", "root", " ", "bookshop");
```

در این تابع آدرس سرویس دهنده به صورت `localhost:3308` نوشته شده که عدد `۳۳۰۸` شماره پورت MySQL است. (با دقت در شکل ۲-۱۲، می توانید در بالای صفحه آن را ببینید. چنانچه MySQL از شماره پورت پیش فرض (`۳۳۰۶`) استفاده کند، این شماره نمایش داده نشده و نیازی به نوشتن آن در دستور اتصال هم نخواهید داشت).

در ادامه از نام کاربری پیش فرض `root` و بدون کلمه عبور برای اتصال به MySQL استفاده شده و در پایان هم نام بانک اطلاعاتی `bookshop` تعیین شده تا به عنوان بانک اطلاعاتی جاری، انتخاب شده و دستورات بعدی روی آن اجرا شوند.

همچنین نتیجه تابع `mysqli_connect` در متغیر `$link` قرار گرفته و بلافاصله این متغیر بررسی شده است، اگر مقدار آن پوچ باشد، تابع `die` فراخوانی شده که پیغام مناسب را نمایش داده و اجرای برنامه را قطع می کند:

```
if( ! $link ) die("خطا در اتصال به بانک");
```

مثال ۲: صفحه‌ای برای نمایش لیست کاربران فروشگاه کتاب، که جدول آن مشابه زیر ایجاد شده است، طراحی کنید:

جدول ۳-۱۲- مشتریان (users)

uid	name	mobile	addr
۱۰۱	علی رضایی	۰۹۱۱۱۲۳	تهران
۱۰۲	فاطمه احمدی	۰۹۱۲۲۳۲	شیراز
۱۰۳	محمد اکبری	۰۹۱۳۵۵۵	اصفهان

برای استفاده از امکانات dreamweaver ابتدا در محیط این نرم‌افزار یک صفحه جدید به نام user-list.php ایجاد کرده و یک جدول دارای ۲ سطر و ۴ ستون در آن درج کنید (اگر از ویرایشگر دیگری استفاده می‌کنید ممکن است نیاز باشد تا تگ‌های مربوط به جدول را تایپ کنید):

```

user-list.php
<html>
<body>
<h1> لیست کاربران </h1> <hr>
< table width="600" border="1">
<tbody>
<tr>
<td>آدرس</td><td>همراه</td><td>نام</td><td>شناسه</td>
</tr><tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
</body>
</html>

```

سپس می‌توانید قبل از سطر خالی دوم که برای نمایش اطلاعات کاربر ایجاد شده است، کدنویسی کرده و دستوراتی برای اتصال به بانک اطلاعاتی و اجرای دستور SELECT روی جدول مشتریان نوشته و یک حلقه تکرار برای استخراج رکوردهای کاربران پیاده‌سازی کنید به گونه‌ای که سطر خالی در داخل حلقه قرار گیرد:

```

<?php
$link = mysqli_connect("localhost3308:", "root", " ", "bookshop");
if( ! $link ) die("خطا در اتصال به بانک اطلاعاتی");
$result = mysqli_query($link , "SELECT * FROM users");
while($row = mysqli_fetch_assoc( $result )) { ?>

```

سطری که باید مشخصات کاربر را نمایش دهد

```

<?php } ?>

```

سپس هر کدام از فیلدهای رکورد کاربر را در یکی از سلول‌های سطر تکرار شونده نمایش دهید:

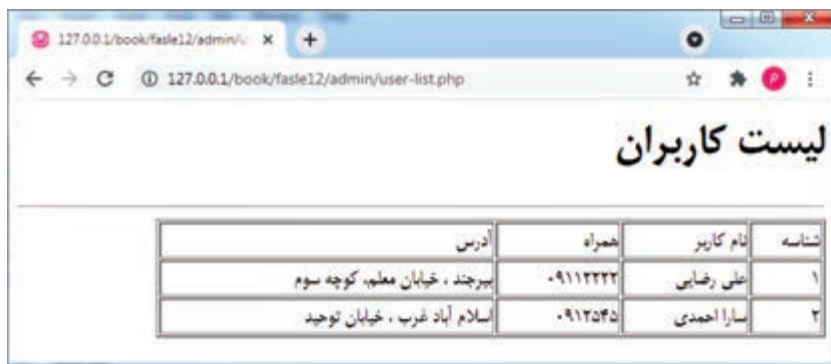
صفحه کامل user-list.php

```

<html>
<body>
<h1>لیست کاربران</h1><hr>
<table width="600" border="1" >
<tbody>
<tr>
<td>نام کاربر</td><td>شناسه</td>
<td>آدرس</td><td>همراه</td>
</tr>
<?php
$link = mysqli_connect("localhost:3308", "root", " ", "bookshop");
if( ! $link ) die("خطا در اتصال به بانک اطلاعاتی");
$result = mysqli_query($link , "SELECT * FROM users");
while($row = mysqli_fetch_assoc( $result )) { ?>
<tr>
<td><?= $row['uid'] ?></td>
<td><?= $row['uname'] ?></td>
<td><?= $row['mobile'] ?></td>
<td><?= $row['addr'] ?></td>
</tr>
<?php } ?>
</tbody>
</table>
</body>
</html>

```

بعد از این که صفحه `user-list.php` را ذخیره کرده و از طریق مرورگر درخواست کنید، صفحه مشابه شکل ۱۲-۳ را مشاهده خواهید کرد:



شکل ۱۲-۳- نمایش لیست مشتریان

۱۲-۶- نمایش کتاب‌ها در فروشگاه اینترنتی


در این قسمت دو صفحه برای نمایش لیست کتاب‌ها و نمایش جزئیات کتاب در صفحه جداگانه طراحی و پیاده‌سازی می‌کنید:



این دو صفحه در بخش عمومی سایت فروشگاه کتاب استفاده شده و در طراحی آنها از فایل‌های قالب طراحی شده در فصل یازدهم استفاده می‌شود.

۱۲-۶-۱- طراحی صفحه نمایش لیست کتاب‌ها

فایل `book-index.php` برای نمایش لیست کتاب‌ها استفاده شده به گونه‌ای که برای هر کدام از کتاب‌ها یک جعبه مشابه شکل زیر نمایش می‌دهد:

طراحی جعبه	نمایش
<pre><div class="box"> <h2 align="center"> نام کتاب </h2> </div></pre>	 <div style="border: 1px solid black; padding: 5px; text-align: center;">نام کتاب</div>

پیاده‌سازی کامل فایل book-index.php به صورت زیر می‌باشد:

```
صفحه book-index.php
<?php include("pheader.php");?>
<div class="row">
<?php
$link =
mysqli_connect("localhost:3308","root","","bookshop");
$result = mysqli_query($link , "SELECT * FROM books ");
while($book = mysqli_fetch_assoc( $result )) { ?>
<div class="col-sm-3">
<div class="box">
<a href="book-show.php?bid=<?=$book['bid']?>">

</a>
<a href="book-show.php?bid=<?=$book['bid']?>">
<h2 align="center"><?=$book['bname']?></h2>
</a>
</div>
</div>
<?php } ?>
</div>
<?php include("pfooter.php");?>
```

در صفحه book-index.php ابتدا فایل pheader.php ضمیمه شده سپس تگ‌ها و دستورات بدنه صفحه قرار گرفته و در انتها هم فایل pfooter.php ضمیمه شده است. در بدنه صفحه ابتدا یک تگ div با کلاس row از bootstrap ایجاد شده، سپس به بانک اطلاعاتی متصل شده و رکوردهای کتاب از جدول خوانده می‌شود و طی یک حلقه تکرار برای هر کدام از آنها یک تگ div با کلاس col-sm-3 از bootstrap ایجاد می‌شود و جعبه طراحی شده در داخل آن قرار می‌گیرد.

در این برنامه با استفاده از امکان شبکه‌بندی bootstrap هر سطر به چهار ستون‌های سه واحدی تقسیم شده است. همچنین تگ img و نام کتاب در داخل تگ a قرار گرفته‌اند و تبدیل به پیوند شده‌اند، بنابراین در صورت کلیک روی تصویر یا نام کتاب، صفحه book-show.php درخواست شده و شناسه کتاب به‌عنوان پارامتر به آن ارسال خواهد شد تا مشخصات کامل همان کتاب نمایش داده شود.

برای ارسال پارامتر به یک صفحه از طریق پیوند، بعد از نوشتن آدرس، نام و پسوند صفحه، علامت سؤال نوشته و نام پارامتر را مساوی مقدار موردنظر قرار می‌دهیم. در این جا فیلد شناسه کتاب به‌عنوان مقدار

پارامتر چاپ می شود:

```
<a href="book_show.php?bid=<?=$book['bid']?>">
<?=$book['bname']?>
</a>
```

ارسال پارامترها به صفحات دیگر در بسیاری از قسمت‌های سایت لازم بوده و در فصل‌های آینده بیشتر از آن استفاده می‌شود.

نمایش صفحه book-index.php را در شکل ۱۲-۴ مشاهده می‌کنید. در این صفحه برای همه کتاب‌ها از یک تصویر ثابت استفاده شده است.



شکل ۱۲-۴- نمایش لیست کتاب‌ها

۲-۶-۱۲- طراحی صفحه نمایش جزئیات کتاب

در ادامه صفحه book-show.php طراحی می‌شود. در این صفحه شناسه کتاب به عنوان پارامتر دریافت شده، رکورد مربوط به آن از بانک اطلاعاتی خوانده و نمایش داده می‌شود:

صفحه book-show.php

```
<?php include("pheader.php");?>
<?php
$bid= $_GET['bid'];
$link = mysqli_connect("localhost:3308","root","","bookshop");
$result = mysqli_query($link , "SELECT * FROM books where bid=$bid ");
$book = mysqli_fetch_assoc( $result ) ;?>
<div class="row">
<div class="col-sm-4">

<h3>مؤلف : <?=$book['author']?></h3>
<h3>قیمت : <?=$book['price']?></h3>
```

```

</div>
<div class="col-sm-8">
<h1><?=$book['bname']?></h1>
<?=nl2br($book['des'])?>
</div>
</div>
<?php include("pfooter.php");?>

```

در کد فوق برای دریافت پارامتر شناسه کتاب دستور زیر نوشته شده است. برای دریافت پارامترهای ارسال شده به صفحه از آرایه `$_GET` و کلید نام پارامتر استفاده می‌شود:

```
$bid= $_GET['bid'];
```

به همین صورت در دستور `SELECT` در بخش `WHERE` شرط `bid=$bid` قرار گرفته که باعث خواندن یک رکورد با شناسه مورد نظر می‌شود:

```
$result = mysqli_query($link , "SELECT * FROM books where bid=$bid");
```

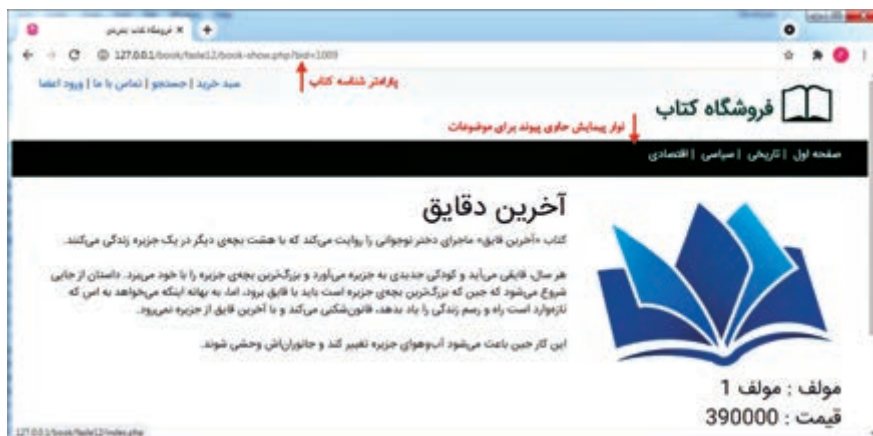
یک رکورد خوانده شده از بانک اطلاعاتی با یک بار اجرای تابع زیر واکنشی شده و در آرایه `$book` انجمنی قرار می‌گیرد:

```
$book = mysqli_fetch_assoc( $result );
```

تابع مفید دیگری به نام `nl2br()` که برای نمایش توصیف کتاب استفاده شده باعث می‌شود تا خط هر جای متن که خط شکسته شده به جای آن تگ `br` قرار گرفته و پاراگراف‌های متن به درستی نمایش داده شوند.

در ضمن برای نمایش دو ستون کنار هم امکان شبکه بندی `bootstrap` استفاده شده که یک ستون با عرض ۴ واحد برای نمایش مشخصات کتاب و ستون دیگر به عرض ۸ واحد برای نمایش توصیف کتاب نمایش داده می‌شود.

در شکل ۵-۱۲ اجرای صفحه `book-show.php` هنگام کلیک روی کتاب «کلیله و دمنه» با شناسه ۱۰۰۱ نشان داده شده است:



شکل ۵-۱۲- نمایش جزئیات کتاب

۷-۱۲- نمایش لیست موضوعات

در فصل یازدهم برای ساخت نوار پیمایش قالب صفحات عمومی سایت از تعدادی پیوند ثابت به صورت زیر استفاده شد :

```
<ul>
<li><a href="">صفحه اول</a></li>
<li> | <a href="">تاریخی</a></li>
<li> | <a href="">سیاسی</a></li>
<li> | <a href="">اقتصادی</a></li>
</ul>
```

حال که با نحوه دسترسی به بانک اطلاعاتی و خواندن و نمایش رکوردها آشنا شدید می توانید به جای استفاده از پیوندهای ثابت، پیوندها را با توجه به موضوعات موجود در بانک اطلاعاتی و به صورت پویا نمایش دهید تا با تغییر موضوعات در بانک اطلاعاتی نوار پیمایش سایت هم به صورت خودکار تغییر کند. برای این منظور در صفحه pheader.php به جای تگ های قبلی می توانید از طراحی زیر استفاده کنید:

```
<?php
$link = mysqli_connect('127.0.0.1:3308','root','','bookshop2');
$res =
mysqli_query($link , "SELECT * FROM subs ");
?>
<ul>
<li><a href="index.php">صفحه اول</a></li>
<?php while($row=mysqli_fetch_assoc($res) ) { ?>
<li> | <a href="search.php?sid=<?=$row['sid']?>"><?=$row['sname']?></a></li>
<?php } ?>
</ul>
```

در کد فوق بعد از خواندن رکوردهای موضوعات از جدول subs از بانک اطلاعاتی، برای هر کدام از آنها تگ های زیر نمایش داده شده است:

```
<li> | <a href="search.php?sid=<?=$row['sid']?>"><?=$row['sname']?></a></li>
```

در خاصیت href از تگ a بعد از نام برنامه search.php یک علامت سؤال (?) قرار گرفته و سپس sid مساوی مقدار شناسه موضوع شده است. در نتیجه وقتی کاربر روی پیوند مربوط به موضوعی مثلاً با شناسه ۳ کلیک کند، درخواست زیر می شود؟ این شناسه به عنوان پارامتر به برنامه search.php ارسال شده و می تواند در جست و جوی کتاب های دارای این موضوع استفاده شود:

```
http://127.0.0.1/book/false10/search.php?sid=3
```

در این فصل روش اتصال به بانک اطلاعاتی و اجرای درخواست های خواندن از جداول را فرا گرفته و صفحاتی برای نمایش رکوردهای موجود در بانک اطلاعاتی به صورت جدول HTML یا نوار پیمایش سایت پیاده سازی کردید. در فصل بعد با برنامه های دیگری برای انجام عملیات حذف، درج و ویرایش رکوردهای بانک اطلاعاتی آشنا خواهید شد.

خودآزمایی

- ۱ دو روش برای اتصال به بانک اطلاعاتی MySQL را نام ببرید.
- ۲ روش PDO و MySQLi برای ارتباط با بانک اطلاعاتی را با هم مقایسه کنید.
- ۳ دستور اتصال به بانک اطلاعاتی با روش MySQLi روال گرا را نوشته و پارامترهای آن را توضیح دهید.
- ۴ یک دستور برای اتصال به بانک اطلاعاتی bank1 موجود روی سرور با آدرس 23.121.457.25 و شماره پورت 3306 و با استفاده از نام کاربری user1 و کلمه عبور 123 بنویسید.
- ۵ قطعه برنامه‌ای برای خواندن اطلاعات دانش‌آموزان از جدول studs با ستون‌های name و avgr و نمایش آنها در یک جدول بنویسید (بانک اطلاعاتی school روی سرور محلی قرار دارد).

سوالات چهار گزینه‌ای

- ۱ کدام گزینه روش اتصال به بانک اطلاعاتی در PHP را نشان می‌دهد؟
الف) MySQLi روال گرا
ب) MySQLi شیء‌گرا
ج) PDO
د) همه موارد
- ۲ در روش MySQLi از کدام تابع برای اتصال به بانک اطلاعاتی MySQL استفاده می‌شود؟
الف) connect
ب) mysql_connect
ج) mysqli_connect
د) موارد ب و ج
- ۳ کدام ویژگی تنها مربوط به روش PDO برای اتصال به بانک اطلاعاتی است؟
الف) بررسی امنیتی دستور قبل از اجرا
ب) امکان اتصال به بانک‌های اطلاعاتی مختلف
ج) سرعت بالا در دسترسی به داده‌ها
د) همه موارد
- ۴ کدام گزینه عملکرد تابع mysqli_query را دقیق‌تر نشان می‌دهد؟
الف) اتصال به بانک اطلاعاتی
ب) اجرای دستورات روی بانک اطلاعاتی
ج) خواندن رکوردها از بانک اطلاعاتی
د) جست‌وجوی رکوردها در بانک اطلاعاتی
- ۵ تابع mysqli_fetch_assoc دقیقاً به چه منظور استفاده می‌شود؟
الف) واکنشی یک رکورد از نتیجه پرس و جو
ب) خواندن رکورد از بانک اطلاعاتی
ج) ذخیره اطلاعات رکوردها
د) پیمایش رکوردهای جدول
- ۶ تابع mysqli_insert_id به منظور..... به کار می‌رود؟
الف) درج رکورد جدید
ب) درج شناسه برای رکورد
ج) مقداردهی شناسه رکورد
د) برگرداندن آخرین شناسه تولید شده

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶
پاسخ	د	ج	ب	ب	الف	د

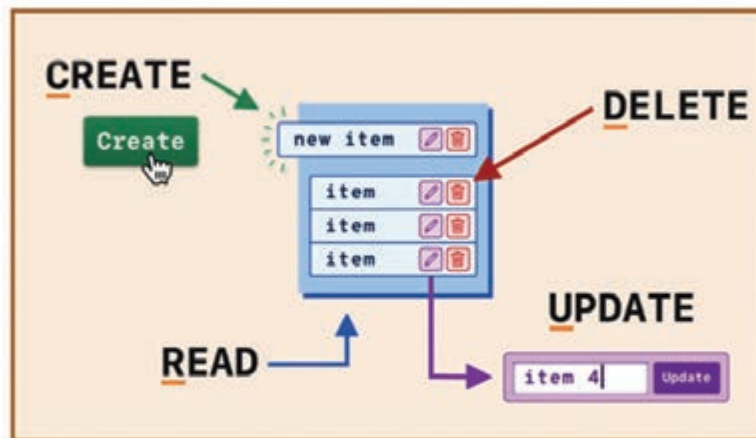
دستور کار عملی

در محیط phpMyAdmin، بانک اطلاعاتی bookshop را ایجاد کنید. سپس جداول موضوعات و کتاب‌ها و مشتریان را در آن ایجاد کرده و برای هر جدول چند سطر دلخواه وارد کنید. سپس دستورات زیر را اجرا کنید: مشابه صفحه نمایش مشتریان، یک صفحه برای نمایش کتاب‌های موجود در بانک اطلاعاتی طراحی کنید و در طراحی آن از فایل‌های قالب استفاده کنید. صفحه‌ای برای نمایش لیست مشتریان با استفاده از فایل‌های قالب طراحی کنید.



فصل سیزدهم

افزودن رکورد به بانک اطلاعاتی



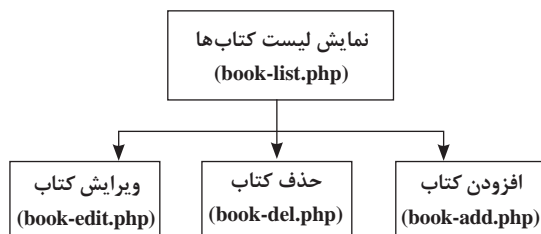
C R U D

هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- رکورد جدید در بانک اطلاعاتی درج کند.
 - رکورد جدید از طریق فرم دریافت کرده و در بانک اطلاعاتی درج کند.
 - در فرم افزودن رکورد از لیست‌های انتخابی پویا استفاده کند.
 - به همراه فرم، فایل‌ها را دریافت کرده و روی سایت بارگذاری کند.
 - کنترل درستی و اعتبار داده‌های فرم و نوع و اندازه فایل‌ها را قبل از درج در بانک اطلاعاتی، انجام دهد.
 - در طراحی صفحات از فایل‌های قالب آماده استفاده کند.
 - با کمک کلاس‌های `bootstrap`، فرم پیشرفته طراحی کند.
- در فصل قبل با روش کلی و دستورات زبان PHP برای ایجاد اتصال به بانک‌های اطلاعاتی MySQL و اجرای درخواست روی بانک اطلاعاتی آشنا شدید. همچنین توانستید لیست رکوردها را از بانک اطلاعاتی خوانده

و به صورت‌های مختلف در صفحه وب نمایش دهید. در این فصل و فصل بعد برای جدول کتاب‌ها از بانک اطلاعاتی فروشگاه اینترنتی، صفحاتی مطابق شکل ۱-۱۳ برای اضافه کردن، حذف و ویرایش رکوردها طراحی و کدنویسی می‌کنید:



شکل ۱-۱۳- ساختار درختی صفحات سایت

۱-۱۳- درج رکورد به بانک اطلاعاتی

در این قسمت روش دریافت اطلاعات از فرم و ذخیره کردن در جدول بانک اطلاعاتی آموزش داده می‌شود. برای اضافه کردن رکورد جدید به بانک اطلاعاتی لازم است مشابه روشی که در فصل قبل بیان شد، دستوراتی برای اتصال به بانک اطلاعاتی MySQL و اجرای درخواست درج رکورد اجرا شود. برای توضیح بهتر و هدفمند جدول کتاب‌ها از بانک اطلاعاتی فروشگاه کتاب (`bookshop`) به صورت زیر در نظر گرفته شده و مثال‌های این فصل بر اساس آن پیاده‌سازی شده‌اند:

جدول ۱-۱۳- `books` برای نگهداری مشخصات کتاب‌ها

نام ویژگی	نام فیلد	نوع داده
شناسه کتاب	bid	عددی
نام	bname	رشته‌ای
نام مؤلف	author	رشته‌ای
قیمت	price	عددی
شناسه موضوع	sid	عددی
توصیف	des	رشته‌ای بلند
نام تصویر جلد کتاب	cover	رشته‌ای
وضعیت نمایش	status	عددی

قبل از نوشتن دستورات PHP یادآوری می‌شود، برای ایجاد درخواست درج رکورد روی بانک اطلاعاتی MySQL از دستور INSERT دارای شکل کلی زیر در زبان SQL استفاده می‌شود:

(مقدار ۱ و مقدار ۲ و مقدار ۳ و...) values (ستون ۱ و ستون ۲ و...) نام جدول insert into

مثال: فرض کنید مشخصات کتاب در متغیرهایی در برنامه قرار گرفته باشد، سپس دستوراتی برای درج یک کتاب جدید بنویسید.

برای این کار صفحه book-add1.php به صورت زیر پیاده‌سازی شده است:

صفحه book-add1.php

```
<?php
$name = 'برنامه نویسی پایتون';
$author = 'رضا';
$price = 230000;
$sid = 4;
$sql = "INSERT INTO
    books($name,$author,$price,$sid)
    VALUES ('$name','$author','$price', '$sid)'";

$link = mysqli_connect('127.0.0.1:3308', 'root', '', 'bookshop');
$res = mysqli_query ($link , $sql );

if ( $res )
    echo 'یک رکورد اضافه شد';
else echo mysqli_error($link);

mysqli_close($link);
?>
```

در کد فوق، عملیات زیر انجام شده است:

- ۱ ابتدا مشخصات کتاب در متغیرها قرار گرفته اند.
- ۲ دستور SQL برای درج ایجاد شده و در متغیر \$sql قرار گرفته است و چون از "" برای نوشتن رشته ثابت استفاده شده، متغیرهای داخل رشته ثابت تشخیص داده شده و مقدار آنها مورد استفاده قرار می‌گیرد.
- ۳ با فراخوانی تابع mysqli_connect() یک اتصال به بانک اطلاعاتی bookshop ایجاد می‌شود و در متغیر \$link قرار می‌گیرد.
- ۴ با فراخوانی تابع mysqli_query() دستور SQL قرار گرفته در متغیر \$sql روی بانک اطلاعاتی اجرا شده و نتیجه برگشتی در متغیر \$res قرار می‌گیرد. نتیجه می‌تواند مقدار منطقی true یا false باشد.

۵ متغیر \$res بررسی می‌شود اگر دارای مقدار منطقی درست بود، پیغام مناسب نمایش داده و در غیر این صورت با کمک تابع `mysqli_error()` پیغام خطا نمایش داده می‌شود.

۶ با فراخوانی تابع `mysqli_close()` اتصال بسته می‌شود.

توجه داشته باشید، چون ستون شناسه کتاب (`bid`) از نوع خودافزایشی است و به صورت خودکار مقدار می‌گیرد، هنگام درج رکورد می‌توان برای آن مقدار تعیین نکرد که در این حالت بعد از اجرای تابع `mysqli_query()` و اطمینان از درج صحیح رکورد می‌توان مقدار تولید شده برای شناسه کتاب را با استفاده از تابع `mysqli_insert_id()` به دست آورد.

دستورات زیر بعد از بررسی متغیر \$res و اطمینان از درج رکورد شناسه کتاب را به دست آورده و همراه با پیغام نمایش می‌دهد:

```
if( $res ) {  
    $bid = mysqli_insert_id( $link );  
    echo " درج شد " . $bid . " رکورد جدید با شناسه " ;  
}else echo mysqli_error($link);
```

همان‌طور که در مثال بالا و مثال‌های فصل قبل دیدید، برای انجام هر عملیاتی روی بانک اطلاعاتی باید دستور اتصال به بانک اجرا شود. از طرفی در این دستور باید آدرس سرویس‌دهنده، شماره پورت، نام کاربری، کلمه عبور و نام بانک اطلاعاتی نوشته شود. تصور کنید سایت شما دارای چندین صفحه باشد که این دستور در همه آنها نوشته شده باشد حال در صورت تغییر هر کدام از پارامترهای فوق باید همه صفحات را ویرایش کرده و تنظیمات جدید را بنویسید.

برای پرهیز از نوشتن دستورات تکراری و راحت‌تر شدن تغییرات بعدی می‌توان دستور اتصال به بانک و دستور تعیین کدگذاری اتصال را در فایل جداگانه (مثلاً با نام `config.php`) مشابه زیر قرار داده و در همه صفحات سایت آن را ضمیمه کرد:

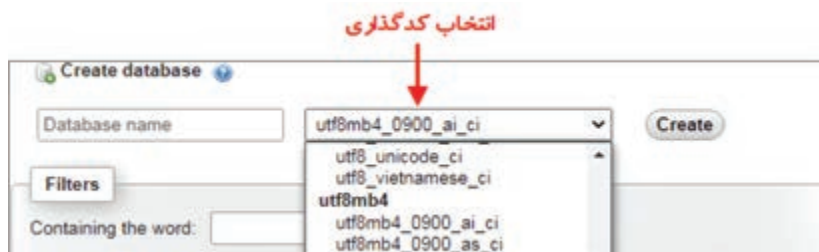
صفحه `config.php`

```
<?php  
$link = mysqli_connect("127.0.0.1:3308", "root", "", "bookshop");  
mysqli_query($link , "SET NAMES utf8" );  
?>
```

در کد فوق علاوه بر ایجاد اتصال درخواستی روی بانک اجرا شده تا کدگذاری اتصال را به `utf8` تنظیم کند:
`mysqli_query($link , "SET NAMES utf8");`

این تنظیم برای خواندن یا نوشتن اطلاعات فارسی لازم است. تنظیم روش کدگذاری بستگی به `collation` انتخاب شده برای بانک اطلاعاتی در محیط `phpMyAdmin` دارد.

در شکل ۲-۱۳، نحوه انتخاب `collation` برای بانک اطلاعاتی نشان داده شده است. همین ویژگی برای جداول و ستون‌های آنها نیز قابل انتخاب است.



شکل ۲-۱۳- انتخاب collation برای بانک اطلاعاتی

در مثال فوق فرض شد که ویژگی Collation بانک اطلاعاتی، یکی از گزینه‌های انتخاب شده است که با utf8 شروع شده است. در صفحات بعد به جای نوشتن دستورات ایجاد اتصال به بانک اطلاعاتی از فایل config.php استفاده شده است.

۲-۱۳- دریافت اطلاعات رکورد جدید از کاربر

در بخش قبل یک اسکریپت برای درج یک رکورد با اطلاعات از پیش تعریف شده در بانک اطلاعاتی ایجاد شد. واضح است که برنامه باید امکان دریافت اطلاعات رکورد جدید از کاربر و درج آن در بانک اطلاعاتی را داشته باشد. در این قسمت برای درج رکورد جدید از فرم وب کمک می‌گیریم که در فصل‌های قبل با روش طراحی و پردازش اطلاعات آنها آشنا شده‌اید.

مثال: صفحه‌ای طراحی کنید که یک فرم خالی برای دریافت مشخصات کتاب به کاربر نمایش داده و پس از تکمیل و ارسال، رکورد جدید در بانک اطلاعاتی درج گردد. برای این کار ابتدا یک صفحه به نام book-add2.php ایجاد کرده و یک فرم برای دریافت مشخصات کتاب در آن قرار دهید:

صفحه book-add2.php

```
<html>
<body dir="rtl">
<form action="book-add3.php" method="post">
نام کتاب: <input type="text" name="bname" /><br>
مؤلف: <input type="text" name="author" /><br>
قیمت: <input type="text" name="price" /><br>
موضوع: <input type="text" name="sid" /><br>
<input type="submit" value="افزودن کتاب" />
</form>
</body>
</html>
```

فرم فوق، دارای چند ورودی جعبه متن با اسامی `price`، `author`، `aname` و `sid` است که به ترتیب برای دریافت نام، مؤلف، قیمت و شناسه موضوع کتاب استفاده خواهند شد. چون فرض شده ستون `bid` از نوع خودافزایشی است پس نیازی به دریافت آن از کاربر نبوده و خود سیستم به آن مقدار خواهد داد. همان طور که دیده می شود در نام گذاری ورودی های فرم از همان اسامی ستون های جدول استفاده شده است. این کار باعث خوانایی بهتر برنامه می شود. برای راحتی کار و فهم بهتر موضوع، از فرمی بسیار ساده و ابتدایی استفاده شده است که البته می توان با استفاده از تگ ها و تنظیمات بیشتر، فرم های کامل تر و زیباتری طراحی کرد. این فرم پس از تکمیل توسط کاربر و ارسال با روش `post` که در ویژگی `method` آن تعیین شده و به برنامه ای با نام `book-add3.php` که در ویژگی `action` آن تعیین شده فرستاده می شود. در مرحله بعد فایل `book-add3.php` را ایجاد کرده و دستورات زیر را در آن قرار دهید تا اطلاعات فرم را دریافت کرده و رکورد جدیدی را در بانک اطلاعاتی درج کند. برای راحتی می توانید یک نسخه از دستورات `book-add1.php` که در بخش قبل ایجاد شد را در این فایل قرار داده و تغییر دهید.

برنامه `book-add3.php`

```
<?php
$aname = $_POST['aname'];
$author= $_POST['author'];
$price = $_POST['price'];
$sid = $_POST['sid'];
$sql= "INSERT INTO books (aname,author,price,sid)
VALUES ('$aname','$author',$price, $sid) " ;
$link = mysqli_connect('127.0.0.1:3308', 'root', '', 'bookshop');
$res = mysqli_query ($link , $sql );
if( $res ) {
    $bid = mysqli_insert_id( $link );
    echo " درج شد " . $bid . ". رکورد جدید با شناسه " ;
} else echo mysqli_error($link);
mysqli_close($link);
?>
```

در کد فوق چون فرم با روش `post` ارسال شده ابتدا مشخصات کتاب از آرایه `$_POST` به دست آمده و سپس در دستور `insert` مورد استفاده قرار گرفته اند. چنانچه فرم با متد `get` فرستاده شود به جای آرایه `$POST` از آرایه `$GET` استفاده می شود.

۳-۱۳- استفاده از انواع ورودی‌ها در فرم

در این بخش برای جدول کتاب‌ها فرم کامل‌تری طراحی می‌شود که دارای فیلدهای لیست انتخابی، جعبه انتخاب و انتخاب فایل باشد. ضمن اینکه در پردازش اطلاعات فرم دستوراتی برای بررسی خالی نبودن فیلدهای ضروری، درستی اندازه و پسوند فایل و ذخیره کردن فایل اضافه می‌شود. یکی از اجزای مهم فرم‌های ورود اطلاعات لیست‌های انتخابی هستند که معمولاً آیتم‌های لیست را در یک جدول بانک اطلاعاتی ذخیره کرده و برای نمایش لیست، رکوردها را از جدول خوانده و گزینه‌های لیست ساخته می‌شوند.

در طراحی بانک اطلاعاتی فروشگاه کتاب، شناسه موضوع (ستون sid) در جدول کتاب‌ها (books)، یکی از شناسه‌های موجود در جدول موضوعات (subs) است که اصطلاحاً به آن کلید خارجی می‌گویند. برای دیگر جداول هم ممکن است چنین ارتباطی وجود داشته باشد. چنانچه ستونی از جدول دارای این ویژگی باشد در فرم‌های افزودن و ویرایش بهتر است به جای جعبه متن ساده از لیست کشویی برای انتخاب آن استفاده شود. با فرض اینکه جدول موضوعات به صورت زیر طراحی شده و در بانک اطلاعاتی ایجاد شده باشد:

جدول ۲-۱۳- موضوعات subs

نام ویژگی	نام فیلد	نوع داده
شناسه	sid	عددی
نام	sname	رشته ای

ورودی دیگری که به فرم اضافه می‌شود، یک جعبه انتخاب برای انتخاب وضعیت نمایش یا عدم نمایش کتاب است. فرض کنید به دلایلی مثل ناقص بودن اطلاعات یک کتاب، در یک زمان تصمیم گرفتید که اطلاعات آن کتاب در سایت نمایش داده نشود تا در آینده بعد از تکمیل مشخصات دوباره نمایش آن ادامه پیدا کند. در این حالت به جای حذف کتاب از بانک اطلاعاتی می‌توان از فیلد مخصوصی استفاده کرد تا وضعیت کتاب را نگه‌داری کرده و اگر مقدار یک داشته باشد، کتاب نمایش داده شده و در غیر این صورت نمایش داده نشود. ورودی‌های دیگری که لازم است به فرم اضافه شود، جعبه متن چند سطری برای دریافت توصیف کتاب و جعبه متن دریافت فایل تصویر روی جلد کتاب است.

برای اضافه کردن موارد فوق فرم افزودن کتاب دوباره با نام book-add5.php، به صورت زیر طراحی شده است:

صفحه book-add5.php

```
<html>
<body dir="rtl">
<h1>افزودن کتاب</h1>
<hr>
<form action="book-add6.php" method="post" enctype="multipart/form-data">
نام کتاب: <input type="text" name="bname" /><br>
```

```

مؤلف : <input type="text" name="author" /><br>
قیمت : <input type="text" name="price" /><br>
قابل نمایش :
<input type="checkbox" name="status" /><br>
موضوع :
<select name='sid' >
<option value="0" >---</option>
<?php
require('config.php');
$res = mysqli_query($link , 'SELECT * FROM subs' );
while($row = mysqli_fetch_assoc($res)) { ?>
<option value="<?=$row['sid']; ?>" >
<?=$row['sname']; ?>
</option>
<?php } ?>
</select>
<br>
تصویر جلد : <input type="file" name="cover" /><br>
توصیف کتاب : <br><textarea name="des" cols="30"></textarea><br><br>
<input type="submit" value="افزون کتاب" />
</form>

</body>
</html>

```

در صفحه `book-add5.php`، برای دریافت شناسه موضوع، به جای استفاده از تگ `INPUT`، از تگ `SELECT` استفاده شده است به طوری که لیست موضوعات موجود در بانک اطلاعاتی را نمایش می‌دهد. برای این کار، یک تگ `select` با نام `sid` شامل یک آیتم پیش فرض (با مقدار صفر و عنوان ...) و لیستی پویا از آیتم‌ها ایجاد شده است.

برای خواندن لیست موضوعات، به بانک اطلاعاتی متصل شده و درخواست `select` اجرا می‌شود سپس طی یک حلقه تکرار، همه رکوردهای جدول موضوعات خوانده شده و به ازای هر کدام یک تگ `option` نمایش داده می‌شود به طوری که مقدار آن مساوی شناسه موضوع و عنوان آن مساوی نام موضوع باشد. برای ورودی «قابل نمایش بودن» از تگ `input` با نوع `"checkbox"` استفاده شده است:

```
<input type="checkbox" name="status" />
```

برای انتخاب فایل تصویر جلد کتاب از تگ `input` با نوع `"file"`، استفاده شده است. علاوه بر این، برای تگ `form` خاصیت `enctype` به `"multipart/form-data"` مقداردهی شده و خاصیت

action آن هم دارای مقدار book-add6.php می‌باشد. پس اطلاعات فرم به برنامه book-add6.php ارسال خواهد شد. در شکل ۳-۱۳ نمایش فرم فوق را مشاهده می‌کنید:



شکل ۳-۱۳- فرم دریافت مشخصات کتاب

در ادامه صفحه book-add6.php برای دریافت اطلاعات فرم، پیاده‌سازی شده است تا پس از انجام بررسی‌های لازم، رکورد را در بانک اطلاعاتی درج کند. در این فرم، یک فایل تصویری به‌عنوان تصویر جلد کتاب انتخاب و ارسال می‌شود تا برای رکورد جدید ذخیره شود. یک روش رایج برای ضمیمه کردن فایل‌ها به همراه رکوردهای بانک اطلاعاتی، نگهداری فایل در یک زیرپوشه مخصوص و ذخیره کردن نام فایل در جدول بانک اطلاعاتی است. با این روش برای نمایش فایل‌های ضمیمه، نام فایل از جدول خوانده شده و با مراجعه به زیرپوشه مخصوص، خود فایل به‌دست می‌آید.

صفحه book-add6.php

```
<?php
include('config.php');
$name = $_POST['bname'];
$author = $_POST['author'];
$price = intval($_POST['price']); // دریافت قیمت به صورت عددی
$cid = $_POST['cid'];
$des = $_POST['des'];
$status = 0 ;
if (isset($_POST['status'])) $status=1;

$errors=array();
if ( $name==' ')
```

```

        $errors[] = 'نام کتاب تعیین نشده است';
if ( $price==0)
        $errors[] = 'قیمت کتاب نامعتبر است';
$cover=' ';
$fsize = $_FILES['cover']['size'];
if($fsize>0)
{
        $fname = $_FILES['cover']['name'];
        $tname = $_FILES['cover']['tmp_name'];
        $sarr = explode('.', $fname);
        $sxt = end($sarr);
        if($sxt !='jpg' && $sxt!='png')
                $errors[] = 'فرمت فایل غیر مجاز است';
        if($fsize> 100 * 1024 )
                $errors[] = 'اندازه فایل بیشتر از حد مجاز است';
}
if ( count($errors)==0)
{
        $cover = time().'.'.$fname;
        move_uploaded_file($tname,"covers/$cover");
        $sql= "INSERT INTO books (bname,author,price,sid,des,status,cover)
        VALUES ('$bname','$author',$price, $sid,'$des',$status,'$cover' )" ;
        $res = mysqli_query ($link , $sql );
        if( $res )
                echo "رکورد جدید اضافه شد" ;
        else echo mysqli_error($link);
}else {
        echo "<ul>";
        foreach($errors as $err)
                echo "<li>$err</li>";
        echo "</ul>";
}
?>

```

در برنامه صفحه قبل:

۱ در زمان ارسال فرم، ورودی‌های جعبه انتخاب (checkbox) تنها در صورتی که انتخاب شده باشند، ارسال می‌شوند. پس برای استفاده از این ورودی‌ها، ابتدا فرض شده که جعبه انتخاب نشده باشد و `$status` مساوی صفر قرار گرفته است، سپس بررسی می‌شود اگر `$_POST['status']` در حافظه وجود داشته باشد به معنی ارسال ورودی و در نتیجه انتخاب جعبه است، پس `$status` مقدار یک می‌گیرد. برای دریافت «وضعیت نمایش کتاب» از دستورات زیر استفاده شده است:

```
$status = 0 ;
if (isset($_POST['status'])) $status=1;
```

۲ برای نگهداری همه پیغام‌های خطا از یک آرایه به نام `$errors` استفاده شده است که در شروع کار خالی است و در صورت تشخیص خطا، یک پیغام خطا به آن اضافه می‌شود:

```
$errors=array();
```

```
if ( $bname=='')
```

```
    $errors[] = 'نام کتاب تعیین نشده است';
```

۳ برای به دست آوردن مشخصات فایل تصویر جلد کتاب از آرایه `$_FILES` استفاده شده است که یک آرایه دو بعدی بوده و برای هر فایل ارسالی دارای یک سطر است و ستون‌های آن ویژگی‌های مختلف فایل را نگهداری می‌کنند:

```
$fsize = $_FILES['cover']['size']; // اندازه فایل
```

```
$fname = $_FILES['cover']['name']; // نام فایل
```

```
$name = $_FILES['cover']['tmp_name']; // نام موقت فایل
```

۴ برای بررسی اینکه فایل ارسال شده یا خیر، اندازه آن با صفر مقایسه شده است، اگر اندازه آن بزرگ‌تر از صفر بود به معنی انتخاب و ارسال فایل می‌باشد.

۵ برای به دست آوردن پسوند فایل، ابتدا نام فایل بر اساس علامت نقطه به چند بخش شکسته شده، در آرایه قرار گرفته و سپس آخرین خانه آرایه به عنوان پسوند استفاده می‌شود.

```
$arr = explode('.', $fname);
```

```
$ext = end($arr); // آخرین خانه آرایه را بر می‌گرداند
```

۶ پسوند فایل با دو مقدار `jpg` و `png` مقایسه شده، اگر غیر از این دو بود، به آرایه خطاها پیغام مناسب اضافه شده است:

```
if($ext !='jpg' && $ext!='png')
```

```
    $errors[] = 'فرمت فایل غیر مجاز است';
```

۷ اندازه فایل با ۱۰۰ کیلوبایت (۱۰۰*۱۰۲۴ بایت) مقایسه شده، اگر از آن بزرگ‌تر بود، پیغام خطای مناسب به آرایه خطاها اضافه شده است:

```
if($fsize > 100 * 1024 )
```

```
    $errors[] = 'اندازه فایل بیشتر از حد مجاز است';
```

۸ در پایان، تعداد خطاها بررسی شده، اگر صفر بود، یعنی هیچ خطایی وجود نداشته و دستورات ذخیره کردن رکورد جدید در بانک اطلاعاتی اجرا شده است، در غیر این صورت با کمک حلقه `foreach` همه پیغام‌های

خطا به صورت یک لیست نمایش داده می‌شود:

```
if ( count($errors)==0)
{
    // دستورات ذخیره‌سازی فایل و داده‌ها
} else {
    echo "<ul>";
    foreach($errors as $err)
        echo "<li>$err</li>";
    echo "</ul>";
}
```

۹ برای ذخیره‌سازی فایل از دستور زیر استفاده شده که فایل را در زیرپوشه covers سایت ذخیره می‌کند. این زیرپوشه باید قبل از اجرای برنامه ایجاد شود در غیر این صورت اجرای برنامه با خطا مواجه می‌شود. برای پرهیز از رونویسی فایل‌های هم نام، مهر زمان جاری به ابتدای نام فایل اضافه می‌شود:

```
$cover = time().'!'.$fname;
```

```
move_uploaded_file($fname,"covers/$cover");
```

۱۰ دستور درج ساخته شده و نام جدید فایل که از مهر زمان + نام اصلی فایل تشکیل شده، در ستون cover جدول ذخیره می‌شود:

```
$sql= "INSERT INTO books (bname,author,price,sid,des,status,cover)
```

```
VALUES ('$bname','$author',$price, $sid,'$des',$status,'$cover' )" ;
```

```
$res = mysqli_query ($link , $sql );
```

مثال: استفاده از یک صفحه برای نمایش و پردازش فرم

در مثال‌های قبلی همیشه از دو صفحه جداگانه استفاده شد: صفحه اول برای نمایش فرم و صفحه دوم برای دریافت اطلاعات فرم و ثبت رکورد در بانک اطلاعاتی. در حالی که می‌توان برای طراحی خود فرم و دستورات پردازش آن از یک صفحه مشترک استفاده کرد. به عنوان مثال برای اضافه کردن کتاب می‌توان از روش زیر استفاده کرد:

صفحه book-add7.php

```
<?php
if (isset($_POST['bname'])) {
    // دستورات مربوط به دریافت داده‌ها و بررسی‌ها و ذخیره کردن رکورد در بانک اطلاعاتی
}
?>
<form action="book-add7.php" >
<!-- تگ‌های طراحی فرم -->
</form>
```


در این صفحه، ابتدا با استفاده از تابع `isset()`، بررسی می‌شود، اگر `$_POST['bname']` در حافظه وجود داشت، به معنی این است که فرم تکمیل شده و ارسال شده است و دستورات اتصال به بانک و ثبت رکورد انجام می‌شود. در غیر این صورت بخش بالای صفحه اجرا نشده و بخش دوم که طراحی فرم است، اجرا می‌گردد.

۴-۱۳- طراحی فرم افزودن کتاب با استفاده از قالب

پیش‌تر با روش دریافت اطلاعات رکورد و درج آن در بانک اطلاعاتی آشنا شدید، همچنین در بخش قبل یک قالب مشترک برای صفحات مختلف سایت فروشگاه طراحی کردید. بنابراین وقت آن رسیده است که از یک صفحه حرفه‌ای‌تر و زیباتر برای دریافت مشخصات کتاب استفاده کرده و جزئیات بیشتری را در طراحی فرم، مورد توجه قرار دهید.

صفحه `book-add.php`، شامل طراحی فرم با استفاده از کلاس‌های `bootstrap` و نیز بخش پردازش فرم می‌باشد:

صفحه `book-add.php`

```
<?php
include('config.php');
if (isset($_POST['bname'])) {
    // بخش پردازش داده‌های فرم
    $bname = $_POST['bname'];
    $author= $_POST['author'];
    $price = intval($_POST['price']);
    $sid = $_POST['sid'];
    $des = $_POST['des'];
    $status = 0 ;
    if (isset($_POST['status'])) $status=1;
    $errors=array();
    if ( $bname==' ' )
        $errors[] = 'نام کتاب تعیین نشده است';
    if ( $price==0)
        $errors[] = 'قیمت کتاب نامعتبر است';
    $cover=' ';
    $fsize = $_FILES['cover']['size'];
    $fname = ' ';
    if($fsize>0)
    {
        $fname = $_FILES['cover']['name'];
```

```

$name = $_FILES['cover']['tmp_name'];
$arr = explode('.', $fname);
$ext = end($arr);
if($ext != 'jpg' && $ext != 'png')
    $errors[] = 'فرمت فایل غیر مجاز است';
if($fsize > 100 * 1024 )
    $errors[] = 'اندازه فایل بیشتر از حد مجاز است';
}
if ( count($errors)==0)
{
if ($fname != ''){
$cover = time().'.'.$fname;
move_uploaded_file($name, "covers/$cover");
}
$sql= "INSERT INTO books (bname,author,price,sid,des,status,cover)
VALUES ('$bname','$author','$price', $sid,'$des','$status','$cover' )" ;
$res = mysqli_query ($link , $sql );
if( $res ) {
    echo "رکورد جدید اضافه شد" ;
    exit;
}
else echo mysqli_error($link);
}
}
// بخش طراحی فرم
$page_title='افزودن کتاب';
include('header.php');
include('show-errors.php');
?>
</div>
<div class="form-group">
    <label for="bname">نام کتاب</label>
    <input type="text" maxlength="100" name="bname" class="form-control" />
</div>
<div class="form-group">
    <label for="author">نام مؤلف</label>
    <input type="text" name="author" class="form-control" />

```

```

</div>
<div class="form-group">
    <label for="price">قیمت</label>
    <input type="number" name="price" class="form-control" />
</div>
<div class="form-group">
<label for="sid">موضوع کتاب</label>
<select name="sid" class="form-control">
<option value="0" >---</option>
<?php
$res2 = mysqli_query($link , 'SELECT * FROM subs' );
while($row2 = mysqli_fetch_assoc($res2)) { ?>
<option value="<?=$row2['sid']; ?>" >
<?=$row2['sname']; ?>
</option>
<?php } ?>
</select>
</div>
<div class="form-group">
    <label for="cover">تصویر جلد</label>
    <input type="file" name="cover" accept=".jpg,.png" class="form-control" />
</div>
<div class="form-group">
    <label for="des">توصیف کتاب</label>
    <textarea name="des" rows="5" class="form-control" ></textarea>
</div>
<div class="form-group">
<input type="submit" value="ارسال فرم" class="btn btn-info" />
<input type="reset" value="بازنویسی" class="btn btn-warning" />
</div>
</form>
<?php include('footer.php');

```

در بخش طراحی فرم صفحه **book-add.php**: در ابتدا فایل `header.php` ضمیمه شده تا بخش بالای سایت و ستون کناری صفحه ساخته شود. سپس برای نمایش لیست پیغام‌های خطاهای احتمالی، فایل `show-errors.php` ضمیمه شده است.

در مثال فوق، فایل `show-errors.php` حاوی دستوراتی برای نمایش پیغام‌های خطای موجود در آرایه `$errors` است که در بخش پردازش فرم مقداردهی می‌شود:

فایل `show-errors.php`

```
<?php
if ( isset($errors) && count($errors)>0) { ?>
<div class="alert alert-danger">
<?php
echo "<ul>";
foreach($errors as $err)
    echo "<li>$err</li>";
echo "</ul>";
?>
</div>
<?php } ?>
```

در طراحی فروشگاه اینترنتی در تمام صفحات پیغام‌های خطا را در آرایه `$errors` قرار داده و در ابتدای هر صفحه، فایل `show-errors.php` ضمیمه شده تا در صورت وجود خطا، آنها را در یک کادر قرمز رنگ نمایش دهد. برای نمایش هر کدام از ورودی‌های فرم یک تگ `div` دارای کلاس `bootstrap` با نام `form-group` استفاده شده که در داخل آن یک برجسب برای نمایش عنوان و تگ ورودی فرم با کلاس `form-control` از `bootstrap` قرار گرفته‌اند:

```
<div class="form-group">
    <label for="author">نام مؤلف</label>
    <input type="text" name="author"
        class="form-control" />
</div>
```

برای ورودی نام کتاب از خاصیت `maxlength` استفاده شده تا حداکثر تعداد کاراکترهای ورودی محدود شود:

```
<input type="text" name="author" class="form-control" />
```

برای ورودی فایل تصویر جلد کتاب، از خاصیت `accept` برای تعیین فرمت فایل‌های قابل پذیرش استفاده شده است:

```
<input type="file" name="cover" accept=".jpg,.png" class="form-control" />
```

برای دکمه‌های ارسال و بازنویسی از کلاس‌های btn از bootstrap استفاده شده است.
<input type="submit" value=" ارسال فرم " class="btn btn_info" />
و نهایتاً در پایان صفحه، فایل footer.php، ضمیمه شده است.
نمایش بخشی از صفحه book-add.php را در شکل ۱۳-۴ مشاهده می‌کنید:



شکل ۱۳-۴- نمایش صفحه افزودن کتاب

در صورتی که اطلاعات کتاب به صورت ناقص ارسال شده یا فایل تصویر ارسال شده دارای فرمت یا اندازه مورد نظر نباشد، در بخش پردازش داده‌های فرم پیام‌های خطا در آرایه‌ی \$errors قرار گرفته و توسط فایل show_errors.php به صورت شکل ۱۳-۵ نمایش داده خواهد شد:



شکل ۱۳-۵- نمایش پیام‌های خطا

در این فصل توانستید اطلاعات را از فرم‌های وب دریافت کرده و در بانک اطلاعاتی ذخیره کنید. در این فرم‌ها از انواع ورودی‌ها مثل جعبه متن ساده و چندسطری، جعبه انتخاب، لیست انتخابی (کشویی) و فایل استفاده کردید و برای نمایش رکوردهای جدول بانک اطلاعاتی در یک لیست انتخابی پویا دستوراتی در فرم قرار دادید. همچنین با استفاده از قالب طراحی شده در فصل‌های قبل، صفحه‌ای نسبتاً کامل برای دریافت مشخصات کتاب و اعتبارسنجی داده‌های فرم پیاده‌سازی کردید. در فصل بعد مهارت طراحی و پیاده‌سازی صفحاتی برای حذف و ویرایش رکوردها را پیدا خواهید کرد.

۱ یک دستور INSERT مناسب برای درج اطلاعات زیر در جدول users بنویسید.

uname	Reza
role	customer
mobile	09121231234
status	1

۲ دستورات لازم برای اتصال به بانک اطلاعاتی bookshop و درج رکورد ذکر شده در سؤال ۱ را در جدول users بنویسید.

۳ کاربرد تابع mysqli_insert_id را شرح دهید.

سوالات چهار گزینه‌ای

۱ در دستور زیر، به جای علامت ? باید کدام گزینه قرار گیرد؟

Insert into table_name(?) values ...

الف) مقدار (ب) ستون و مقدار

ج) تعداد ستون‌ها (د) اسامی ستون‌ها

۲ برای اجرای دستور insert روی بانک اطلاعاتی از کدام تابع php استفاده می‌شود؟

الف) –mysqli_query (ب) mysqli_connect

ج) mysqli_insert (د) mysqli_insert_row

۳ پس از اجرای موفق دستور زیر، محتوای متغیر \$result چه مقداری خواهد بود؟

`$res = mysqli_query($link , "delete ...");`

الف) ۱ (ب) true

ج) رکورد جدید (د) شناسه رکورد

۴ برای نمایش پیغام خطای برگشته از mysql طی آخرین درخواست اجرا شده، کدام دستور زیر مناسب است؟

الف) mysqli_error() (ب) echo mysqli_error()

ج) mysqli_errors() (د) echo mysqli_errors()

۵ دستور SET NAMES در زبان SQL چه کاربردی دارد؟

الف) تنظیم کدگذاری اتصال به بانک اطلاعاتی

ب) تنظیم روش کدگذاری بانک اطلاعاتی

ج) تغییر نام بانک اطلاعاتی

د) تغییر نام جداول بانک اطلاعاتی

۶ دستورات زیر برای استفاده از یک ورودی به نام status در فرم ورود اطلاعات نوشته شده است. نوع این ورودی کدام است؟

```
$status = 0 ;
if (isset($_POST['status'])) $status=1;
```

- الف) لیست انتخابی
 ب) جعبه انتخاب
 ج) دکمه رادیویی
 د) جعبه متن
 ۷ خروجی دستورات زیر کدام است؟

```
$a = explode('.', 'date.inc.php');
$b = end( $a );
```

- الف) date
 ب) inc
 ج) php
 د) نمایش خطا
 ۸ در صورتی دستورات داخل if اجرا می شوند که....؟

```
if (isset($_POST['bname'])) {
    دستورات
}
```

- الف) ورودی bname دارای مقدار باشد.
 ب) فرمی با روش post فرستاده شده باشد.
 ج) فرم دارای ورودی با نام bname باشد.
 د) موارد ب و ج

۹ از کلاس form-control مربوط به بوت استرپ در طراحی کدام تگ ها استفاده می شود؟

الف) form ب) input ج) table د) option

۱۰ برای به دست آوردن آدرس و نام موقت فایل آپلود شده با نام ورودی cover، در دستور زیر، به جای علامت سؤال (?) کدام گزینه قرار می گیرد؟

```
$a = $_FILES['cover'][' ? '];
tmp_name      د      temp      ج      path      ب      name      الف
```

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
پاسخ	د	الف	ب	ب	الف	ب	ج	د	ب	د

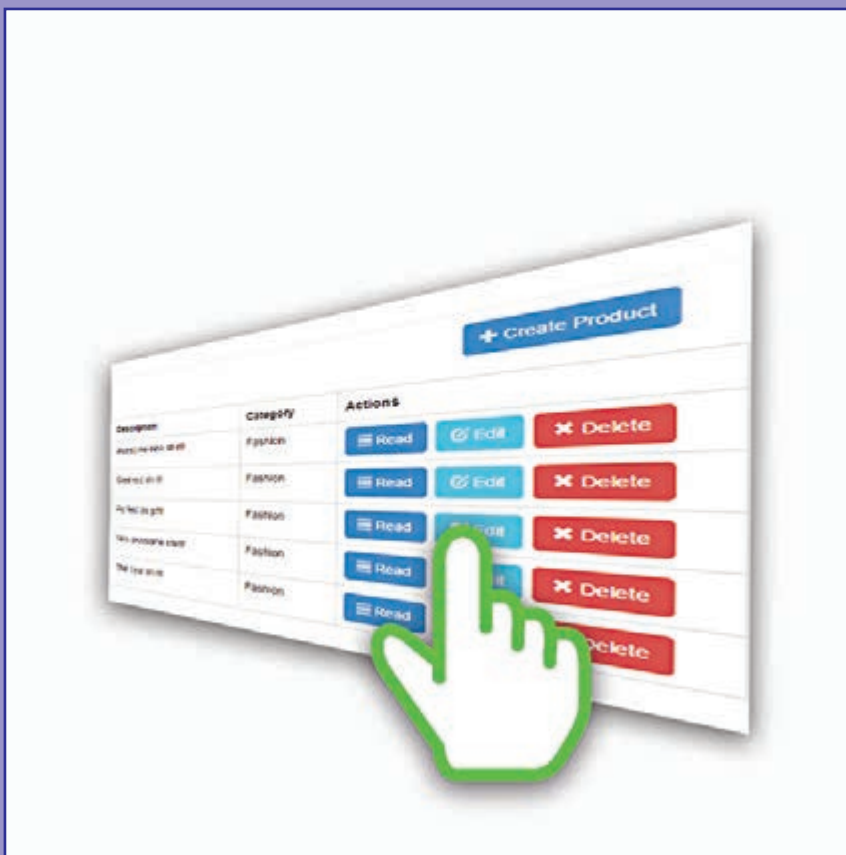
دستور کار عملی

- ۱ در سایت فروشگاه اینترنتی کتاب، صفحه‌ای برای افزودن مشتری، طراحی و پیاده‌سازی کنید.
- ۲ برای طراحی صفحه افزودن مشتری از قالب آماده استفاده کنید.



فصل چهاردهم

حذف و ویرایش رکوردهای بانک اطلاعاتی



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- در صفحه نمایش لیست رکوردها برای هر سطر لینک حذف و ویرایش نشان دهد.
 - رکورد دلخواه از بانک اطلاعاتی حذف کند.
 - رکورد بانک اطلاعاتی را خوانده و در فرم ویرایش قرار دهد.
 - تغییرات رکورد اصلاح شده را در بانک اطلاعاتی درج کند.
 - در طراحی صفحه ویرایش رکورد از فایل‌های قالب سایت استفاده کند.
- در فصل‌های قبل روش اتصال به بانک‌های اطلاعاتی MySQL، خواندن و نمایش اطلاعات و درج رکورد در بانک اطلاعاتی را فرا گرفتید، در این فصل مهارت طراحی صفحات برای حذف و ویرایش رکوردهای بانک اطلاعاتی را پیدا می‌کنید.
- در تمام مثال‌های این فصل از جداول بانک اطلاعاتی فروشگاه‌های اینترنتی و فایل‌های قالب فروشگاه‌ها که در فصل‌های قبل طراحی شده است، استفاده می‌شود.

۱-۱۴- حذف رکوردهای بانک اطلاعاتی

همان‌گونه که در فصل‌های قبل آموختید، برای حذف رکوردهای بانک اطلاعاتی از دستور delete در زبان SQL استفاده می‌شود که دارای شکل کلی زیر است:

[شرط مورد نظر where] نام جدول delete from

«شرط مورد نظر» تعیین می‌کند که کدام رکورد یا رکوردهای جدول حذف شوند.

بیشتر اوقات از دستور delete برای حذف یک رکورد مشخص استفاده می‌شود و برای این منظور بهتر است شناسه یا کلید اصلی رکورد برای تعیین شرط استفاده شود. به عنوان مثال برای حذف رکورد از جدول کتاب‌ها (books) با شناسه مساوی ۱۰۰۱ بهتر است دستور زیر اجرا شود.

```
delete from books where bid=1001;
```

گاهی اوقات هم لازم می‌شود، تعدادی از رکوردها حذف شود. به عنوان مثال برای حذف همه کتاب‌های با شناسه موضوع مساوی ۲، دستور زیر اجرا می‌شود:

```
delete from books where sid=2;
```

اگر هیچ شرطی برای دستور تعیین نشود، نیازی به نوشتن بخش where نبوده و دستور زیر همه رکوردهای جدول books را حذف می‌کند:

```
delete from books;
```

مثال: فایل `book-del1.php` برای حذف کتاب با شناسه مساوی ۱۰۰۱ نوشته شده است.

فایل `book-del1.php`

```
<?php
$bid=1001;
include('config.php');
$sql="DELETE FROM books WHERE bid=$bid " ;
$res = mysqli_query( $link , $sql);
if ( $res )
    echo "رکورد با موفقیت حذف شد"
else echo mysqli_error($link); ?>
```

در کد فوق، برای اتصال به بانک اطلاعاتی، فایل `config.php` ضمیمه شده است که در فصل قبل روش ایجاد و استفاده از آن شرح داده شده است.

برنامه `book-del1.php` همیشه کتاب با شناسه ۱۰۰۱ را حذف می‌کند. برای حذف کتاب با شناسه دلخواه می‌توان هنگام درخواست این برنامه، شناسه کتاب را به صورت پارامتر به این صفحه ارسال کرد تا آن را دریافت کرده و در متغیر `$bid` قرار دهد:

فایل `book-del.php`

```
<?php
$bid=$_GET['bid'];
include('config.php');
$sql="delete from books where bid=$bid " ;
$res=mysqli_query( $link , $sql);
if ( $res )
    echo "رکورد با موفقیت حذف شد"
else echo mysqli_error($link);
?>
```

قبل از اجرای برنامه فوق باید شناسه کتاب تعیین شود. برای این منظور می‌توان در صفحه‌ای که لیست کتاب‌ها نمایش داده می‌شود، برای هر کتاب، مشابه زیر، یک پیوند «حذف» به فایل `book-del.php` برقرار کرده و شناسه کتاب را به آن اضافه نمود:

`<a href="book-del2.php?bid=<?=$row['bid']?>">حذف`

در این قسمت روش حذف کتاب بر اساس شناسه کتاب نشان داده شد، با همین شیوه می‌توان برای حذف رکورد در هر جدولی از بانک اطلاعاتی عمل کرد.

۲-۱۴- ویرایش رکوردهای بانک اطلاعاتی

یکی از عملیات مهم روی بانک اطلاعاتی، ویرایش رکوردها است. برای اصلاح یا ویرایش یک رکورد جدول بانک اطلاعاتی از دستور UPDATE در زبان SQL استفاده می‌شود که دارای شکل کلی زیر است:

نام جدول UPDATE

... , مقدار ۲ = ستون ۲ , مقدار ۱ = ستون ۱ SET

عبارت شرطی WHERE

اگر عبارت شرطی نوشته نشود، همه سطرهای جدول اصلاح می‌شوند و اگر بر اساس کلید اصلی یا همان شناسه نوشته شود، تنها یک رکورد اصلاح می‌گردد. به عنوان مثال برای اصلاح مشخصات کتاب زیر می‌توان دستور زیر را روی بانک اطلاعاتی اجرا کرد:

مقدار	ویژگی
1001	bid
PHP	bname
500	price

```
UPDATE books
```

```
SET bname='PHP',price=500
```

```
WHERE bid=1001
```

برای طراحی بخش ویرایش رکوردها در یک وب سایت، بهتر است مراحل زیر انجام شود:

مرحله ۱: بر اساس شناسه تعیین شده با اتصال به بانک اطلاعاتی، اطلاعات رکورد خوانده و در یک فرم وب نمایش داده شود.

مرحله ۲: بعد از اینکه کاربر اطلاعات فرم را تغییر داد و آن را ارسال کرد، با اتصال به بانک اطلاعاتی و اجرای دستور UPDATE، تغییرات در بانک اطلاعاتی ثبت گردد.

در ادامه برای ویرایش رکورد از جدول کتاب‌ها این مراحل اجرا شده است:

در مرحله ۱، برای تعیین شناسه رکورد می‌توان مشابه کاری که برای حذف کتاب انجام شد، در صفحه نمایش لیست رکوردها، به ازای هر رکورد، یک پیوند «ویرایش» به برنامه ویرایش قرار داده و شناسه رکورد را به عنوان پارامتر به آن ارسال کرد. به عنوان مثال برای پیاده‌سازی ویرایش کتاب‌ها: در صفحه نمایش لیست کتاب‌ها، پیوند زیر را قرار دهید:

```
</a href="book-edit1.php?bid=?" ?=$row['bid']?>> حذف </a>
```

پیوند فوق، صفحه book-edit1.php را درخواست کرده و شناسه کتاب را به آن ارسال می‌کند.

در صفحه book-edit1.php، قبل از شروع طراحی فرم، دستوراتی برای به دست آوردن شناسه کتاب با روش GET و اتصال به بانک اطلاعاتی و خواندن مشخصات کتاب مورد نظر بنویسید.

در زمان طراحی فرم برای هر کدام از ورودی‌ها، مقدار اولیه آن‌ها را مساوی فیلد مربوطه از رکورد کتاب قرار دهید.

در برنامه book-edit1.php، دستورات لازم نشان داده شده است:

برنامه book-edit1.php

```
<html>
<body dir="rtl">
<?php
$bid = $_GET['bid']; // دریافت پارامتر شناسه کتاب
include('config.php');
$result = mysqli_query($link,"SELECT * FROM books WHERE bid=$bid");
$row = mysqli_fetch_assoc( $result );
?>
<form action="book-edit2.php?bid=<?=$row['bid']?>" method="post">
نام : <input type="text" name="bname"
value="<?=$row['bname']?>" /><br>
مؤلف : <input type="text" name="author"
value="<?=$row['author']?>" /><br>
قیمت : <input type="text" name="price"
value="<?=$row['price']?>" /><br>
<input type="submit" value=" ثبت تغییرات " />
</form>
</body>
</html>
```

در کد فوق، شناسه کتاب از طریق GET خوانده شده و در دستور SELECT برای خواندن مشخصات کتاب استفاده می‌شود.

برای فرم هم خاصیت action مساوی `book-edit2.php?bid=<?=$row['bid']?>` و متد آن مساوی `post` قرار داده شده است تا اطلاعات فرم با روش `post` به برنامه `book-edit2.php` ارسال شده و پارامتر `bid` هم با روش `get` به این برنامه ارسال گردد.

برای هر یک از تگ‌های `input`، مقدار فعلی فیلد کتاب به خاصیت `value` آن‌ها نسبت داده شده تا به‌طورپیش‌فرض نمایش داده شود. در ضمن برای ورودی شناسه کتاب (`bid`)، خاصیت `readonly` نوشته شده تا فقط خواندنی بوده و کاربر نتواند شناسه را تغییر دهد.

فرم ویرایش کتاب در نتیجه اجرای صفحه `book-edit1.php` در شکل ۱۴-۱ نشان داده شده است:



شکل ۱۴-۱- فرم ویرایش کتاب

پس از اینکه کاربر مشخصات کتاب را تغییر داد و فرم را ارسال کرد، اطلاعات فرم به برنامه `book-edit2.php` دارای کد زیر تحویل داده می‌شود تا اطلاعات ارسالی را از فرم دریافت کرده و با اتصال به بانک اطلاعاتی، تغییرات را به آن منتقل کند.

برنامه `book-edit2.php`

```
<?php
// دریافت پارامتر شناسه کتاب
$bid = $_GET['bid'];
// دریافت مشخصات کامل کتاب
$name = $_POST['bname'];
$author = $_POST['author'];
$price = $_POST['price'];
// ساخت دستور آپدیت
$sql = "
UPDATE books
SET bname='$name',author='$author',price=$price
WHERE bid=$bid " ;

include('config.php');
$res = mysqli_query ($link , $sql );
if ( $res )
    header("location:book-list.php");
else echo mysqli_error($link);
?>
```

در برنامه صفحه قبل ابتدا پارامتر شناسه کتاب و اطلاعات کامل کتاب در متغیرها قرار می‌گیرد. سپس دستور UPDATE به زبان SQL برای تغییر مشخصات کتاب ساخته شده و در متغیر \$sql نوشته می‌شود. در ادامه دستور update با استفاده از تابع mysqli_query روی بانک اطلاعاتی اجرا شده و نتیجه در متغیر \$res قرار می‌گیرد. سپس نتیجه بررسی می‌شود، اگر درست بود به صفحه book-list.php تغییر مسیر می‌دهد و در غیر این صورت پیغام خطا نمایش داده می‌شود. در پایان هم اتصال بسته می‌شود.

۳-۱۴- تغییر مسیر درخواست‌ها

هنگامی که مرورگر صفحه وبی را درخواست می‌کند، سرور دهنده صفحه را اجرا کرده و نتیجه را برمی‌گرداند. گاهی اوقات نیاز است تا سرور دهنده، صفحه درخواستی را اجرا کند و پس از آن صفحه دیگری را هم اجرا کند و آنگاه نتیجه برگردانده شود. به این فرایند تغییر مسیر گفته می‌شود. به عنوان مثال وقتی کاربر در حال مشاهده صفحه نمایش لیست رکوردها است و روی پیوند حذف رکورد کلیک می‌کند، انتظار دارد بعد از حذف رکورد دوباره همان صفحه لیست رکوردها را ببیند. در اینجا نیاز است که سرور دهنده بعد از اجرای برنامه حذف، دوباره صفحه نمایش لیست کتاب‌ها را اجرا نماید. تابع header() برای ارسال اطلاعات اضافه مثل نوع و اندازه پاسخ، از سرور دهنده به سرور گیرنده استفاده می‌شود. از این تابع همچنین می‌توان برای تغییر مسیر درخواست‌ها استفاده کرد. به عنوان مثال در صفحه page1.php می‌توانید دستور زیر را بنویسید:

صفحه page1.php
<pre>// دستورات برنامه‌نویسی header("location:page2.php"); // ادامه برنامه</pre>

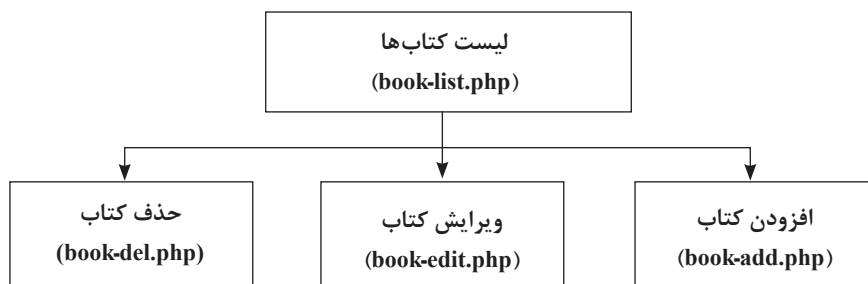
در برنامه فوق، اگر مرورگر صفحه page1.php را درخواست کند، سرور دهنده پس از اجرای دستورات این صفحه و رسیدن به دستور header از ادامه برنامه page1.php صرف نظر کرده و صفحه page2.php را درخواست خواهد کرد. در نهایت خروجی صفحه page2.php به مرورگر برگردانده می‌شود. باید توجه داشت که هیچ‌گاه نباید تگ‌های HTML و یا دستورات خروجی PHP قبل از اجرای تابع header() قرار گرفته و اجرا شوند. به عبارت دیگر نباید قبل از اجرای دستور header هیچ نوع خروجی تولید شده باشد.

با کمک تابع header() می‌توان بعد از اجرای موفق عملیات اضافه کردن، حذف یا ویرایش رکورد به جای نمایش پیغام موفقیت به صفحه نمایش لیست رکوردها تغییر مسیر داد.

<pre>if (\$res) header("location:book-list.php"); else echo mysqli_error(\$link);</pre>

۴-۱۴- تکمیل مدیریت کتاب‌ها در فروشگاه اینترنتی

برای مدیریت کتاب‌ها در فروشگاه اینترنتی حداقل باید صفحات شکل ۲-۱۴ طراحی شوند و در طراحی این صفحات از فایل‌های قالب خصوصی که در فصل قبل ایجاد شده، استفاده خواهد شد.



شکل ۲-۱۴- صفحات مدیریت کتاب در فروشگاه اینترنتی

صفحه افزودن کتاب (book-add.php) در فصل قبل آموزش داده شد. در ابتدای این فصل هم برنامه حذف کتاب با نام book_del.php پیاده‌سازی گردید. در ادامه برای تکمیل بخش مدیریت کتاب فروشگاه صفحه نمایش لیست کتاب‌ها (book-list.php) و صفحه ویرایش کتاب (book_edit.php) طراحی می‌شود.

۱-۴-۱۴- طراحی صفحه نمایش لیست کتاب‌ها

ابتدا صفحه نمایش لیست کتاب‌ها با نام book-list.php به شکل ۳-۱۴ طراحی شده است:

شناسه	نام کتاب	مؤلف	قیمت	موضوع	افزودن کتاب
1001	سرمایه گذاری خطر پذیر	مؤلف دوم	150000	5	حذف ویرایش
1009	آخرین دقائق	مؤلف 1	390000	1	حذف ویرایش
1010	آینده‌ی کهن	آرمان آریز	120000	1	حذف ویرایش
1011	بازگشت به شهر زمرد	ل فرانک باوم	100000	1	حذف ویرایش

شکل ۳-۱۴- نمایش لیست کتاب‌ها

در شکل فوق، پیوندهای افزودن، حذف و ویرایش کتاب را مشاهده می‌کنید که پیش‌تر روش ساخت آنها توضیح داده شد.

صفحه book_list.php به صورت زیر پیاده سازی شده است:

صفحه book-list.php

```
<?php
$page_title='مدیریت کتابها';
include('header.php');
?>
<table class="table table-bordered" >
<tr>
<th>نام کتاب</th><th>شناسه</th>
<th>قیمت</th><th>مؤلف</th>
<th>موضوع</th>
<th><a href="book-add.php">افزودن کتاب</a></th>
</tr>
<?php
include('config.php');
$result = mysqli_query($link,"SELECT * FROM books ");
while($row = mysqli_fetch_assoc( $result )) { ?>
<tr>
<td><?=$row['bid'] ?></td>
<td><?=$row['bname'] ?></td>
<td><?=$row['author'] ?></td>
<td><?=$row['price'] ?></td>
<td><?=$row['sid'] ?></td>
<td>
<a href="book-del.php?bid=<?=$row['bid']?>">
حذف</a> |
<a href="book-edit.php?bid=<?=$row['bid']?>">
ویرایش</a>
</td>
</tr>
<?php } ?>
</table>
<?php include('footer.php');
```

همان طور که می بینید در صفحه book_list.php، فایل های header.php و footer.php که در فصل قبل پیاده سازی شد، ضمیمه شده اند.

۲-۴-۱۴- طراحی صفحه ویرایش کتاب

در طراحی فرم ویرایش کتاب جزئیات بیشتر شامل انواع ورودی‌ها و استفاده از bootstrap را مورد استفاده قرار داده و در پردازش فرم کنترل درستی داده‌ها، فرمت و اندازه فایل، ثبت و نمایش پیغام‌های خطا پیاده‌سازی می‌شود.

در شکل ۴-۱۴ بخشی از صفحه ویرایش کتاب را مشاهده می‌کنید:



شکل ۴-۱۴- صفحه ویرایش کتاب

صفحه book_edit.php شامل دو بخش پردازش فرم ارسالی و بخش طراحی فرم می‌شود. پیاده‌سازی بخش اول صفحه book_edit.php به صورت زیر است:

بخش اول صفحه book-edit.php

```
<?php
$bid = $_GET['bid'];
include('config.php');
if (isset($_POST['bname'])) {
// پردازش داده‌های فرم
$bname = $_POST['bname'];
$author= $_POST['author'];
$price = intval($_POST['price']);
$sid = $_POST['sid'];
$des = $_POST['des'];
$cover1= $_POST['cover1'];
$cover = $_POST['cover1'];
$status= 0 ;
if (isset($_POST['status'])) $status=1;
$errors=array();
if ( $bname=='')
```

```

        $errors[] = 'نام کتاب تعیین نشده است';
if ( $price==0)
        $errors[] = 'قیمت کتاب نامعتبر است';
$fsize = $_FILES['cover']['size'];
$fname = ' ';
if($fsize>0)
{
        $fname = $_FILES['cover']['name'];
        $name = $_FILES['cover']['tmp_name'];
        $arr = explode('.', $fname);
        $ext = end($arr);
        if($ext !='jpg' && $ext!='png')
                $errors[] = 'فرمت فایل غیر مجاز است';
        if($fsize > 100 * 1024 )
                $errors[] = 'اندازه فایل بیشتر از حد مجاز است';
}
if ( count($errors)==0)
{
        if ($fname!=' ') {
                @unlink("covers".DIRECTORY_SEPARATOR.$cover1);
                $cover = time().'.'.$fname;
                move_uploaded_file($name,"covers/$cover");
        }
        $sql= "UPDATE books
        SET bname='$bname',author='$author'
        ,price=$price ,sid=$sid,des='$des'
        ,status=$status,cover='$cover'
        WHERE bid=$bid " ;
        $res = mysqli_query ($link , $sql );
        if( $res )
                header("location:book-list.php");
        else echo mysqli_error($link);
}
}
// بخش طراحی فرم

```

با بررسی کد فوق متوجه می‌شوید که بیشتر دستورات آن مشابه صفحه افزودن رکورد است که در فصل قبل به‌طور کامل توضیح داده شده است.

در بخش طراحی فرم، همان‌طور که قبلاً برای ویرایش کتاب گفته شد، ابتدا مشخصات کتاب از بانک اطلاعاتی خواننده و در آرایه‌ای به نام \$row قرار می‌گیرد:

```
$result = mysqli_query($link,"SELECT * FROM books  
WHERE bid=$bid");  
$row = mysqli_fetch_assoc( $result );
```

فایل show_errors.php که در فصل قبل، طراحی و توضیح داده شد، برای نمایش لیست پیغام‌های خطایی که احتمالاً در بخش اول صفحه (بخش پردازش داده‌های فرم) تولید شده‌اند، ضمیمه شده است.

برای مقداردهی اولیه هر نوع از ورودی‌های فرم به صورت زیر عمل شده است:

برای جعبه متن‌های ساده مثل جعبه متن مؤلف، فیلد مربوطه از کتاب \$row[author] به خاصیت value تگ input نسبت داده شده تا به عنوان مقدار پیش فرض نمایش داده شده و ویرایش شود:

```
<div class="form-group">  
<label for="author">نام مؤلف</label>  
<input type="text" name="author" value="<?=$row['author'] ?>" class="form-control" />  
</div>
```

برای مقداردهی جعبه انتخاب، مثل انتخاب قابل نمایش بودن کتاب، در تگ input از نوع checkbox، بررسی شده اگر فیلد مربوطه از کتاب مساوی ۱ بود، خاصیت checked برای آن اضافه شده تا به‌طور پیش فرض انتخاب شود:

```
<div class="form-group">  
<input type="checkbox" name="status"  
<?php if($row['status']==1) echo 'checked'; ?> />  
<label for="status">  
قابل نمایش  
</label>  
</div>
```

برای نمایش لیست انتخابی پویا برای انتخاب موضوع کتاب، لیست موضوعات از بانک اطلاعاتی خواننده شده و طی یک حلقه تکرار، آیتم‌های لیست نمایش داده شده‌اند. برای انتخاب پیش فرض موضوع قبلی کتاب برای هر آیتم بررسی شده اگر شناسه موضوع آن مساوی فیلد شناسه موضوع کتاب بود، خاصیت selected برای آن اضافه شده است:

```
<div class="form-group">  
<label for="sid">موضوع کتاب</label>  
<select name="sid" class="form-control">
```

```
<option value="0" >---</option>
<?php
$res2 = mysqli_query($link , 'SELECT * FROM subs' );
while($row2 = mysqli_fetch_assoc($res2)) { ?>
<option value="<?=$row2['sid']; ?>"
<?php if($row['sid']==$row2['sid']) echo 'selected'; ?>>
<?=$row2['sname']; ?>
</option>
<?php } ?>
</select>
</div>
```

برای نمایش تصویر قبلی کتاب از تگ `img` استفاده شده که آدرس منبع آن شامل زیر پوشه `covers` به علاوه نام تصویر قبلی که در فیلد `cover` کتاب قرار دارد، تعیین می شود:



همچنین یک ورودی مخفی به نام `cover1` به فرم اضافه شده و نام قبلی تصویر در آن قرار می گیرد. ورودی های مخفی از دید کاربر هستند ولی همراه دیگر ورودی های فرم به سرویس دهنده ارسال می شوند:

```
<div class="form-group">
<label for="cover">تصویر جلد</label>
<?php if($row['cover']!= ' '): ?>

<?php endif; ?> <br><br>
<input type="hidden" name="cover1" value="<?=$row['cover']?>" />
<input type="file" name="cover" accept=".jpg,.png" class="form-control" />
</div>
```

همان گونه که در پیاده سازی بخش اول صفحه ویرایش کتاب دیدید، در بخش پردازش داده های فرم، ابتدا مقدار ورودی `cover1` در متغیر `$cover` و هم متغیر `$cover1` قرار می گیرد.

```
$cover1= $_POST['cover1'];
$cover = $_POST['cover1'];
```

در ادامه اگر فایل ارسال شده باشد و اندازه و فرمت آن هم درست باشد، نام فایل جدید و در غیر این صورت همان نام تصویر قبلی در بانک اطلاعاتی ذخیره می‌شود. ضمن اینکه در صورت ارسال فایل مناسب، با استفاده از دستور زیر فایل تصویر قبلی حذف می‌گردد:

```
@unlink("covers".DIRECTORY_SEPARATOR.$cover1);
```

تابع unlink() برای حذف فایل استفاده می‌شود و علامت @ قبل آن باعث می‌شود که اگر حذف فایل با خطا مواجه شد از نمایش پیام خطا صرف نظر شده و اجرای برنامه مختل نشود.

همچنین ثابت از پیش تعریف شده DIRECTORY_SEPARATOR به جای جداکننده فهرست‌ها استفاده شده است. با توجه به اینکه در سیستم عامل ویندوز از علامت \ و در سیستم عامل لینوکس از علامت / برای جداکردن فهرست‌ها استفاده می‌شود، استفاده از این ثابت باعث می‌شود تا بر اساس سیستم عامل مورد استفاده، علامت مناسب بین نام پوشه و نام فایل قرار گیرد.

برای ورودی‌های متن طولانی از textarea استفاده شده و متن پیش فرض داخل بدنه این تگ قرار می‌گیرد:

```
<div class="form-group">
```

```
<label for="des">توصیف کتاب</label>
```

```
<textarea name="des" rows="5" class="form-control" ><?=$row['des']?></textarea>
```

```
</div>
```

۳-۴-۱۴- ویرایش گروهی رکوردها

گاهی اوقات در طراحی سایت‌ها لازم می‌شود تا در یک صفحه وب لیستی از رکوردها نمایش داده شده و برای همه آنها اطلاعاتی وارد شود. به عنوان مثال در سایت فروشگاه اینترنتی مشابه شکل ۵-۱۴ مدیر فروشگاه بتواند در یک صفحه، لیست کالاها را مشاهده کرده و قیمت همه آنها را تغییر داده و همزمان ارسال کند.



شکل ۵-۱۴- ویرایش گروهی قیمت کتاب‌ها

با استفاده از روش‌هایی که تا حالا آموختید و نیز استفاده از آرایه‌ها در طراحی فرم و سپس پردازش آنها، می‌توان این گونه امکانات را به سایت اضافه کرد.

در ادامه صفحه `book-price.php` برای ویرایش گروهی قیمت کتاب‌ها در سایت فروشگاه اینترنتی پیاده‌سازی شده است:

صفحه `book-price.php`

```
<?php
require('config.php');
if(isset($_POST['books'])) {
// بخش پردازش داده‌های فرم
$books= $_POST['books'];
foreach($books as $bid=>$price) {
    $price=intval($price);
    $sql = "UPDATE books SET price=$price WHERE bid=$bid";
    $res=mysqli_query($link , $sql);
}
}
// بخش نمایش فرم
$page_title='اصلاح قیمت کتاب‌ها';
include('header.php'); ?>
<form action="book-price2.php" method="post">
<table class="table table-bordered" >
<tr>
<td>شناسه</td><td>نام کتاب</td>
<td>مؤلف</td><td>قیمت</td>
</tr>
<?php
$sql ="SELECT * FROM books";
$result = mysqli_query( $link , $sql );
while($row = mysqli_fetch_assoc( $result )) { ?>
<tr>
<td><?= $row['bid'] ?></td>
<td><?= $row['bname'] ?></td>
<td><?= $row['author'] ?></td>
<td width="300">
<input type="text" size="10"
name="books[<?=$row['bid']?>]"
```

```

        value="<?= $row['price']?>"
        class="form-control"
    /></td>
</tr>
<?php } ?>
<tr>
<td colspan="4" >
        <input type="submit" class="btn btn-info" value="ثبت قیمت‌ها" />
        <input type="reset" class="btn btn-warning" value="بازنویسی" />
</td>
</tr>
</table>
</form>
<?php include('footer.php');?>

```

در صفحه `book_price.php` ابتدا بررسی می‌شود اگر `$_POST['books']` در حافظه وجود دارد، فرم ارسال شده و بخش پردازش داده‌ها اجرا می‌شود در غیر این صورت از اجرای این بخش صرف نظر می‌شود. در بخش طراحی فرم، با اتصال به بانک اطلاعاتی مشخصات کتاب‌ها خوانده شده و در سطرهای جدول نمایش داده می‌شود. برای هر سطر یک تگ `input` نشان داده شده که نام آن، آرایه `books` با اندیس شناسه کتاب و مقدار پیش فرض آن، قیمت کتاب است:

```

<input type="text" size="10"
        name="books[<?=$row['bid']?>]"
        value="<?= $row['price']?>"
        class="form-control"
    />

```

به عنوان مثال برای کتاب با شناسه ۱۰۰۱ و قیمت ۲۵۰۰۰۰، تگ زیر در صفحه قرار خواهد گرفت:

```

<input type="text" size="10"
        name="books[1001]"
        value="250000"
        class="form-control"
    />

```

هنگام ارسال فرم، مجموعه این ورودی‌ها به صورت یک آرایه به سرویس دهنده فرستاده می‌شود. در بخش پردازش فرم ابتدا آرایه `books` از فرم دریافت شده و طی یک حلقه تکرار `foreach` کلیدها و مقادیر آن به دست آمده و هر جفت از آنها در یک دستور `update` قرار گرفته و روی بانک اطلاعاتی اجرا می‌شود:


```
$books= $_POST['books'];  
foreach($books as $bid=>$price) {  
    $price=intval($price);  
    $sql = "UPDATE books SET price=$price WHERE bid=$bid";  
    $res=mysqli_query($link , $sql);  
}
```

در این فصل با روش انجام عملیات حذف و ویرایش روی رکوردهای بانک اطلاعاتی آشنا شده و صفحات مربوط به حذف و ویرایش کتابها در فروشگاه اینترنتی را با استفاده از فایل های قالب پیاده سازی کردید. به طور مشابه لازم است برای هر کدام از جداول بانک اطلاعاتی مثل جدول مشتریان یا سفارشات صفحاتی برای مدیریت رکوردها طراحی گردد. در فصل بعد با عملیات پیشرفته تری روی بانک اطلاعاتی آشنا خواهید شد.

خودآزمایی

- ۱ برای حذف یک رکورد از جدول users با وضعیت (status) مسای ۵، دستور مناسب را بنویسید.
- ۲ مجموع دستوراتی برای حذف رکورد از جدول users در بانک اطلاعاتی bookshop، با توجه به پارامتر دریافتی به نام uid پیاده‌سازی کنید.
- ۳ منظور از تغییر مسیر درخواست توسط سرویس‌دهنده چیست؟ توضیح دهید.
- ۴ دستوراتی برای دریافت مشخصات نام فایل دریافت شده از فرم دارای ورودی با نام pic و نمایش پسوند آن بنویسید.
- ۵ فرض کنید یک فرم دارای ورودی‌های name (نام) و natcode (کد ملی) باشد، دستوراتی برای دریافت این داده‌ها از فرم و بررسی درستی آن‌ها بنویسید به طوری که در صورت نادرست بودن داده‌ها پیغام‌های خطای مناسب را نمایش دهد. (نام باید مقدار داشته باشد و طول کد ملی بین ۸ تا ۱۰ رقم باشد)
- ۶ دستور update را برای تغییر مشخصات کاربر به صورت زیر، روی جدول users بنویسید. (uid کلید اصلی است)

uid	10
uname	Amin
role	Customr

- ۷ برای طراحی صفحه‌ای که اطلاعات کاربر یا شناسه مساوی ۱۰ را از بانک اطلاعاتی خوانده و برای ویرایش در فرم نشان می‌دهد، دستورات و تگ‌های لازم را بنویسید.

سوالات چهارگزینه‌ای

- ۱ کدام دستور از SQL برای حذف سطرهای جدول استفاده می‌شود؟
الف) delete ب) drop ج) remove د) موارد ب و ج
- ۲ برای حذف تنها یک رکورد از جدول بانک اطلاعاتی، بهتر است از کدام گزینه برای تعیین شرط در دستور حذف استفاده کنیم؟
الف) ستون نام ب) کلید اصلی ج) چند کلید د) کلید خارجی
- ۳ اگر هیچ شرطی برای دستور حذف تعیین نشود، این دستور...
الف) اجرا نشده و خطا می‌گیرد. ب) هیچ رکوردی را حذف نمی‌کند.
ج) اولین سطر جدول را حذف می‌کند. د) همه رکوردهای جدول را حذف می‌کند.
- ۴ برای ایجاد پیوند به فایل delete.php و ارسال پارامتر id با مقدار ۱۰۰ به جای علامت ستاره (*) در تگ زیر کدام گزینه قرار می‌گیرد؟

 delete

- الف) delete.php?100 ب) delete.php=100
ج) delete.php?id= 100 د) delete.php&id=100

- ۵ در مقابل کلمه SET از دستور UPDATE در زبان SQL، کدام گزینه قرار می‌گیرد؟
 الف) نام ستون‌های جدول (ب) نام ستون‌ها با مقادیر هر کدام
 ج) نام جدول با مقادیر (د) نام ستون کلید اصلی و مقدار
- ۶ کدام خاصیت از تگ input مربوط به جعبه متن برای تعیین مقدار پیش فرض استفاده می‌شود؟
 الف) value (ب) content (ج) default (د) init
- ۷ برای انتخاب پیش فرض یک گزینه از لیست انتخابی، کدام خاصیت را استفاده می‌کنیم؟
 الف) checked (ب) selected (ج) current (د) موارد الف و ب
- ۸ برای تغییر مسیر درخواست توسط سرویس دهنده از کدام تابع php استفاده می‌شود؟
 الف) redirect (ب) path (ج) include (د) header
- ۹ کاربرد تابع unlink() از php چیست؟
 الف) حذف پیوند (ب) قطع ارتباط (ج) حذف فایل (د) هیچ کدام

پاسخنامه

سؤال	۱	۲	۳	۴	۵	۶	۷	۸	۹
پاسخ	الف	ب	د	ج	ب	الف	ب	د	ج

دستور کار عملی

- ۱ یک وب سایت دارای صفحاتی برای مدیریت کتاب‌ها شامل نمایش لیست کتاب‌ها، افزودن، حذف و ویرایش کتاب دارای قالب یکسان (با استفاده از فایل‌های سرصفحه و پاصفحه) طراحی و کدنویسی کنید.
- ۲ تمرین شماره ۱ را یک بار دیگر برای مدیریت مشتریان، تکرار کنید. (بر اساس جدول مشتریان بانک اطلاعاتی ارائه شده در فصل نهم)
- ۳ یک صفحه شامل لیست کتاب‌ها طراحی کنید، طوری که در کنار هر کتاب یک جعبه انتخاب (checkbox) نمایش دهد و امکان حذف گروهی کتاب‌های انتخاب شده وجود داشته باشد.



فصل پانزدهم

جست و جو و محدود کردن نمایش رکوردها



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- نمایش رکوردها را بر اساس کلید خارجی محدود کند.
- رکوردها را بر اساس ترکیبی از پارامترها، جست‌وجو کند.
- نتایج جست‌وجوی رکوردها را به شکل مناسب نمایش دهد.
- رکوردها را به صورت صفحه‌بندی شده نمایش دهد.

در فصل‌های قبل روش انجام عملیات اصلی روی بانک اطلاعاتی را فرا گرفتید. همچنین قبلاً مهارت پیدا کردید تا با طراحی قالب مناسب برای سایت فروشگاه اینترنتی صفحات نمایش کتاب‌ها، افزودن، حذف و ویرایش کتاب‌ها را پیاده‌سازی کنید. در این فصل، روش‌های جست‌وجوی رکوردها و محدودسازی تعداد رکوردها را آموزش می‌بینید. مشابه فصل‌های قبل مثال‌های ارائه شده در این فصل هم روی جداول بانک اطلاعاتی فروشگاه اینترنتی شامل جدول کتاب‌ها و جدول موضوعات پیاده‌سازی شده و برای طراحی ظاهری صفحات نیز از فایل‌های قالب طراحی شده در فصل‌های قبل کمک گرفته شده است.

۱۵-۱- بازیابی و نمایش رکوردها بر اساس کلید خارجی

بیشتر اوقات در برنامه‌های کاربردی نیاز است تا رکوردها بر اساس کلید خارجی از بانک اطلاعاتی خوانده و نمایش داده شوند. این کار باعث محدود شدن رکوردهای نمایش داده شده و مدیریت بهتر آنها می‌شود. در فروشگاه اینترنتی مورد بحث هر کتاب دارای یک ویژگی به نام موضوع است و جدول books در بانک اطلاعاتی هم دارای یک کلید خارجی به نام sid است که با کمک این کلید خارجی می‌توان همه کتاب‌های مربوط به یک موضوع را شناسایی کرده و از جدول استخراج کرد.

در صفحه مدیریت کتاب‌ها می‌توان در بالای صفحه یک لیست انتخابی شامل موضوعات نشان داد تا کاربر مدیر بتواند موضوع موردنظر را انتخاب کرده و فقط کتاب‌های مربوط به آن موضوع را نمایش دهد. همین محدودسازی می‌تواند در صفحات عمومی سایت به این صورت باشد که به عنوان مثال، لیست موضوعات در نوار پیمایش بالای سایت نشان داده شده و کاربر با کلیک روی هر موضوع فقط کتاب‌های مربوط به همان موضوع را در صفحه مشاهده کند.

از طرف دیگر در صفحه مدیریت موضوعات می‌توان در کنار هر موضوع علاوه بر پیوندهای حذف و ویرایش پیوندی با عنوان «کتاب‌ها» اضافه کرد تا با کلیک روی آن تنها کتاب‌های آن موضوع نمایش داده شود.

در شکل ۱۵-۱ صفحه نمایش لیست موضوعات (sub-list.php) به همراه پیوندهای حذف، ویرایش و کتاب‌ها دیده می‌شود که با کلیک روی پیوند «کتاب‌ها» صفحه book-list.php درخواست شده و شناسه موضوع هم

برای آن ارسال می‌شود:



شکل ۱۵-۱- صفحه نمایش لیست موضوعات

هنگام پیاده‌سازی صفحه sub_list.php برای ایجاد پیوند «کتاب‌ها» از تگ a به صورت زیر استفاده می‌شود:

```
<a href="book-list.php?sid=<?=$row['sid']?>">
کتاب‌ها
</a>
```

برای نمایش لیست محدود شده کتاب‌ها در صفحه book-list.php که در فصل قبل طراحی شد، قسمت اتصال به بانک اطلاعاتی و خواندن رکوردهای کتاب از بانک به صورت زیر اصلاح می‌شود:

```
اصلاحات در صفحه book-list.php
<?php
//...
$cid=$_GET['cid']
include('config.php');
if( isset($_GET['cid']))
    $sql ="SELECT * FROM books WHERE /cid=$cid";
else $sql ="SELECT * FROM books";
$result = mysqli_query($link , $sql );
//...
```

در کد فوق بررسی می‌شود، اگر پارامتر cid با روش GET ارسال شده، آن را در دستور SELECT استفاده می‌کند تا رکوردها به صورت محدود شده از بانک اطلاعاتی خوانده شود، در غیر این صورت همه رکوردها خوانده می‌شود. صفحه نمایش لیست کتاب‌ها با رکوردهای محدود شده برای موضوع ۱ را در شکل ۱۵-۲ مشاهده می‌کنید:



شکل ۱۵-۲- نمایش کتاب‌های با موضوع مساوی ۱

۱۵-۲- جست‌وجوی رکوردها

یکی از امکانات مهم هر سایت جست‌وجوی اطلاعات است تا کاربران بتوانند اطلاعات موردنظر خود را راحت و سریع پیدا کنند چرا که با گذشت زمان، اطلاعات سایت بیشتر و بیشتر و پیدا کردن اطلاعات موردنظر مشکل می‌شود. معمولاً همه سایت‌ها در بالای صفحه جعبه متنی برای جست‌وجو قرار می‌دهند. در این قسمت به عنوان نمونه امکان جست‌وجو بر اساس نام یا توصیف کتاب و بازه قیمتی برای صفحه

مدیریت کتاب اضافه شده است تا با روش کلی انجام جست‌وجو در مطالب سایت آشنا شده و بتوانید در صفحات مختلف سایت‌ها آن را استفاده کنید.

مطابق شکل ۳-۱۵ در صفحه book-search.php برای اضافه کردن امکان جست‌وجو در صفحه نمایش کتاب‌ها، یک فرم قبل از نمایش جدول قرار می‌گیرد که دارای سه جعبه متن برای دریافت عبارت جست‌وجو، حداقل قیمت، حداکثر قیمت و یک دکمه ارسال باشد:



شکل ۳-۱۵- صفحه جست‌وجوی کتاب

در زیر روش پیاده‌سازی صفحه book-search.php را مشاهده می‌کنید:

```

صفحه book-search.php
<?php
$page_title='جست‌وجوی کتاب‌ها';
include('header.php');
$srch = '';
$price1 = '';
$price2 = '';
if( isset($_POST['srch'])) {
    $srch = $_POST['srch'];
    $price1 = $_POST['price1'];
    $price2 = $_POST['price2'];
}
$sql="SELECT * FROM books WHERE 1=1 ";
if($price1!=' ') $sql.=" AND price >= $price1 ";
if($price2!=' ') $sql.=" AND price <= $price2 ";
if ($srch!=' ')
    $sql.=" AND ( bname LIKE '%$srch%' OR des LIKE '%$srch%')";
?>
<form sction="book-search.php" method="post">

```



```

<div class="row">
    <div class="col-sm-1">عبارت</div>
    <div class="col-sm-3">
        <input type="text" name="srch" value="<?=$srch?>" class="form-control">
    </div>
    <div class="col-sm-1">از قیمت</div>
    <div class="col-sm-2">
        <input type="text" name="price1" size="10"
        value="<?=$price1?>" class="form-control" />
    </div>
    <div class="col-sm-1">تا قیمت</div>
    <div class="col-sm-2">
        <input type="text" name="price2" size="10" value="<?=$price2?>" class="form-control" />
    </div>
    <div class="col-sm-2">
        <input type="submit" value="جست و جو" class="btn btn-info"/>
    </div>
</div>
</form>
<table class="table table-bordered" >
<tbody>
<tr>
<td>شناسه</td><td>نام کتاب</td>
<td>مؤلف</td><td>قیمت</td>
<td>موضوع</td>
<td><a href="book-add.php">افزودن کتاب</a></td>
</tr>
<?php
include('config.php');
$result = mysqli_query( $link , $sql );
// ادامه کدنویسی و طراحی

```

جزئیات کامل این صفحه و صفحه قبل قبلاً شرح داده شده است بنابراین در این جا تنها قسمت‌های جدید آن شرح داده می‌شود:

فرم جست‌وجو دارای ورودی‌هایی به ازای هر کدام از پارامترهای جست‌وجو بوده و اطلاعات فرم با متد post به همین صفحه یعنی book_search.php ارسال می‌شود:

```
<form action="book-search.php" method="post">
```

اجزای فرم اینجا قرار گرفته‌اند

```
</form>
```

هر کدام از ورودی‌های فرم با مقدار قبلی پارامتر جست‌وجو، مقداردهی اولیه شده‌اند و برای فرمت دهی آن‌ها از کلاس form_control از چارچوب bootstrap استفاده شده است:

```
<input type="text" name="srch" value="<?=$srch?>" class="form-control">
```

در ابتدای صفحه قطعه کد زیر ابتدا متغیرهای جست‌وجو را با پوچ مقداردهی کرده و سپس بررسی می‌کند، اگر `$_POST['srch']` در حافظه وجود داشت به معنی ارسال فرم جست‌وجو است. پس داده‌های ارسال شده از فرم را در متغیرهای جست‌وجو قرار می‌دهد:

```
$srch = '';
```

```
$price1 = '';
```

```
$price2 = '';
```

```
if( isset($_POST['srch'])) {
```

```
    $srch = $_POST['srch'];
```

```
    $price1 = $_POST['price1'];
```

```
    $price2 = $_POST['price2'];
```

```
}
```

سپس دستور SELECT به صورت زیر در متغیر `$sql` قرار می‌گیرد:

```
$sql="SELECT * FROM books WHERE 1=1 ";
```

و هر کدام از پارامترهای جست‌وجو بررسی می‌شوند اگر دارای مقدار باشند، عبارت شرطی مناسبی به ادامه `$sql` اضافه می‌شود:

```
if($price1!=' ') $sql.=" AND price >= $price1 ";
```

```
if($price2!=' ') $sql.=" AND price <= $price2 ";
```

```
if ($srch!=' ')
```

```
    $sql.=" AND ( bname LIKE '%$srch%' OR des LIKE '%$srch%' );
```

برای عبارت جست‌وجو (متغیر `$srch`) عبارت شرطی طوری نوشته می‌شود که کتاب‌هایی که نام یا توصیف آن‌ها شامل عبارت موجود در `$srch` باشد را جست‌وجو کند. یادآوری می‌شود: در زبان SQL از عملگر LIKE برای بررسی مشابه بودن و از علامت‌های % برای تعیین هیچ یا چند کاراکتر در ابتدا یا انتهای عبارت استفاده می‌شود.

برای اتصال به بانک اطلاعاتی و خواندن رکوردها از دستور `$sql` ساخته شده در ابتدای صفحه استفاده می‌شود:

```
include('config.php');
```

```
$result = mysqli_query( $link , $sql );
```

۳-۱۵- محدود کردن تعداد رکوردها

در بعضی صفحات سایت نیاز است تا تعداد مشخصی از رکوردها نمایش داده شود. برای این کار می‌توان هنگام اتصال به بانک اطلاعاتی و اجرای دستور SELECT روی آن با کمک بخش LIMIT، تعداد رکوردها را محدود نمود. از طرفی بعضی اوقات نیاز به نمایش رکوردها به صورت مرتب شده بر اساس یک یا چند ستون از جدول است.

شکل کلی دستور SELECT برای محدود کردن رکوردها به صورت زیر است:

[تعداد رکورد، رکورد شروع LIMIT] نام جدول SELECT * FROM

به عنوان مثال دستور زیر برای خواندن ۲۰ رکورد با شروع از ردیف ۴۰ از جدول کتاب‌ها اجرا می‌شود:

Select * From books LIMIT 40 , 20 ;

همچنین دستور SELECT برای مرتب سازی رکوردها بر اساس یک یا چند ستون جدول به شکل زیر نوشته می‌شود:

[ASC|DESC ... , ستون ۲ , ستون ۱ ORDER BY] نام جدول SELECT * FROM

کلمات ASC و DESC نوع مرتب‌سازی را تعیین می‌کنند: ASC برای مرتب سازی صعودی استفاده می‌شود که به صورت پیش فرض بوده و می‌تواند نوشته نشود و DESC برای مرتب سازی نزولی استفاده می‌شود. به عنوان مثال دستور زیر لیست کتاب‌ها را به صورت مرتب شده بر اساس عنوان و به صورت صعودی می‌خواند:

SELECT * FROM books ORDER BY bname;

دستور زیر نیز لیست کتاب‌ها را به صورت مرتب شده بر اساس قیمت و به صورت نزولی (از بیشتر به کمتر) از بانک اطلاعاتی می‌خواند :

SELECT * FROM books ORDER BY price DESC;

با ترکیب محدودیت و مرتب سازی می‌توان با استفاده از دستور زیر لیست ۱۰ کتاب با کمترین قیمت را از بانک اطلاعاتی به دست آورد:

SELECT * FROM books ORDER BY price LIMIT 0,10;

مثال: صفحه اول سایت فروشگاه اینترنتی کتاب به نام index.php را طراحی کنید که مشابه شکل ۴-۱۵ یک لیست هشت تایی از جدیدترین کتاب‌ها را نمایش دهد:



شکل ۴-۱۵- نمایش صفحه اول سایت فروشگاه

برای طراحی این صفحه می‌توان از فایل‌های قالب عمومی سایت و فایل‌های pheader.php و pfooter.php که در فصل‌های قبلی طراحی شد، استفاده کرد و بخشی از پیاده‌سازی صفحه فوق به صورت زیر می‌باشد:

بخشی از پیاده‌سازی صفحه اول سایت index.php

```
<?php include('pheader.php'); ?>
<h1 class="title2">تازه‌ترین کتاب‌ها</h1>
<div class="row">
<?php
require_once('config.php');
$sql = "SELECT * FROM books WHERE status=1 ORDER By bid DESC LIMIT 0,8";
$res = mysqli_query($link, $sql);
while($row=mysqli_fetch_assoc($res)){ ?>
<div class="col-sm-3">
    <div class="box">
        <a href="book-show.php?bid=<?=$row['bid']?>">
            
            <strong><?=$row['bname']?></strong>
            </a><br>
            <span><?=$row['price']?>ریال</span>
        </div>
    </div>
<?php } ?>
</div>
<?php include('pfooter.php'); ?>
```

در کد فوق در دستور SELECT از شرط status=1 استفاده شده تا فقط کتاب‌هایی که وضعیت نمایش آنها مساوی یک است نمایش داده شود. یادآوری می‌شود: ستون status در جدول کتاب‌ها تعیین می‌کند که کتاب می‌تواند به بینندگان سایت نشان داده شود یا خیر. همچنین در این صفحه از قالب صفحات عمومی سایت استفاده شده است.

۴-۱۵- صفحه بندی نمایش رکوردها

در یک سیستم واقعی معمولاً تعداد رکوردها بسیار زیاد است و نمایش همه آنها در صفحه امکان پذیر نمی باشد. چون زمان بارگیری صفحه طولانی شده و بررسی همه رکوردها توسط کاربر مشکل می گردد. بنابراین برنامه های کاربردی باید رکوردها را به صورت صفحه بندی شده نمایش داده و امکان حرکت بین صفحات را برای کاربر فراهم نمایند.

با استفاده از روش هایی که تا به حال بیان شد، می توان رکوردها را به صورت صفحه بندی شده نمایش داد. در ادامه صفحه ای برای نمایش لیست رکوردهای جدول کتاب ها در فروشگاه اینترنتی طراحی و کدنویسی شده است. روش کلی برای اجرای صفحه بندی در صفحه نمایش لیست رکوردها این است که باید تعداد رکوردهای قابل نمایش در هر صفحه را تعیین کرده و با اتصال به بانک اطلاعاتی، تعداد کل رکوردها را به دست آورده و تعداد صفحات را محاسبه کرد.

سپس با توجه به شماره صفحه جاری که به عنوان پارامتر دریافت می شود، محدوده رکوردها تعیین شده و از جدول خوانده و نمایش داده شود. همچنین پیوندهایی برای نمایش صفحه اول، صفحه قبل، صفحه بعد یا صفحه آخر نمایش داده شود.

صفحه `book_pages.php` برای نمایش صفحه بندی رکوردهای کتاب به صورت زیر پیاده سازی شده است. با توجه به کم بودن تعداد رکوردهای جدول، $\$max=3$ قرار داده شده تا در هر صفحه سه رکورد نمایش داده شود. با تغییر مقدار آن می توان هر تعداد رکورد را در یک صفحه نمایش داد.

صفحه `book-pages.php`

```
<?php
$page_title='مدیریت کتاب ها';
include('header.php');
// تعیین صفحه جاری
if (isset($_GET['page'])) {
    $page = intval ( $_GET['page'] );
} else {
    $page = 1;
}

// محاسبه ردیف شروع
$max = 3;
$offset = ($page-1) * $max;

// محاسبه تعداد صفحات
include('config.php');
```

```

$ssql = "SELECT COUNT(*) as total FROM books";
$res = mysqli_query($link,$ssql);
$row = mysqli_fetch_assoc($res);
$total = $row['total'];
$pages = ceil($total / $max);

// تعیین دستور برای انتخاب رکوردها از بانک
$ssql = "SELECT * FROM books LIMIT $offset, $max ";

// نمایش صفحه بندی
?>
<ul class="pagination" style="text-align: center">
<li><a class="page-link" href="?page=1">صفحه اول</a></li>
<li class="page-item <?php if($page <= 1){ echo 'disabled'; } ?>">
<a class="page-link" href="<?php if($page > 1){ echo "?page=".( $page - 1); } ?>">صفحه قبل</a>
</li>
<li class="page-item disabled">
<a class="page-link">صفحه <?=$page?> از <?=$pages ?></a>
</li>
<li class="page-item <?php if($page >= $pages){ echo 'disabled'; } ?>">
<a class="page-link" href="<?php if($page < $pages) { echo "?page=".( $page + 1); } ?>">
صفحه بعد</a>
</li>
<li><a class="page-link" href="?page=<?php echo $pages; ?>">صفحه آخر</a></li>
</ul>

<table class="table table-bordered" >
<tr>
<th>شناسه</th><th>نام کتاب</th>
<th>مؤلف</th><th>قیمت</th>
<th>موضوع</th>
<th><a href="book-add.php">افزودن کتاب</a></th>
</tr>
<?php

```

```

$res = mysqli_query($link , $sql );
while($row = mysqli_fetch_assoc( $res )) { ?>
<tr>
<td><?=$row['bid'] ?></td>
<td><?=$row['bname'] ?></td>
<td><?=$row['author'] ?></td>
<td><?=$row['price'] ?></td>
<td><?=$row['sid'] ?></td>
<td>
<a href="book-del.php?bid=<?=$row['bid']?>">حذف</a> |
<a href="book-edit.php?bid=<?=$row['bid']?>">ویرایش</a>
</td>
</tr>
<?php } ?>
</table>
<?php include('footer.php');

```

در صفحه فوق مراحل زیر برای صفحه‌بندی لیست کتاب‌ها انجام شده است:

۱ شماره صفحه جاری از طریق پارامتر ارسال شده به دست می‌آید. در دستورات زیر بررسی می‌شود اگر پارامتر page ارسال شده باشد به عنوان شماره صفحه استفاده شده در غیر این صورت شماره صفحه مساوی ۱ در نظر گرفته می‌شود:

```

if (isset($_GET['page'])) {
    $page = intval ( $_GET['page'] ) ;
} else {
    $page = 1;
}

```

۲ با فرض اینکه تعداد رکوردهای قابل نمایش در هر صفحه در متغیر \$max قرار داشته باشد، شماره ردیف شروع (\$offset) برای خواندن رکوردها از جدول محاسبه می‌شود:

```

$max = 3;
$offset = ($page-1) * $max;

```

۳ با اتصال به بانک اطلاعاتی تعداد رکوردهای کتاب خوانده شده و در متغیر دلخواه \$total قرار می‌گیرد و با تقسیم آن بر \$max (تعداد رکوردها در هر صفحه)، تعداد صفحات محاسبه می‌شود:

```

$sql = "SELECT COUNT(*) as total FROM books";
$res = mysqli_query($link,$sql);
$row = mysqli_fetch_assoc($res);

```

```
$total = $row['total'];
```

```
$pages = ceil($total / $max);
```

از تابع `ceil()` برای گرد کردن عدد اعشار به سمت پایین استفاده می‌شود. به عنوان مثال در دستور زیر مقدار ۳ در متغیر `$k` قرار می‌گیرد:

```
$k = ceil( 3.7 );
```

۴ دستور `sql` برای خواندن رکوردهای مربوط به صفحه جاری با شروع از ردیف `$offset` و به تعداد `$max` ایجاد می‌شود:

```
$sql = "SELECT * FROM books LIMIT $offset, $max ";
```

۵ پیوندهایی برای نمایش صفحات مختلف نمایش داده می‌شود. در نمایش لیست پیوند صفحات از کلاس‌های `bootstrap` استفاده شده است.

```
<ul class="pagination">
```

```
<li><a class="page-link" href="?page=1">صفحه اول</a></li>
```

```
<li class="page-item <?php if($page <= 1){ echo 'disabled'; } ?>">
```

```
<a class="page-link" href="<?php if($page > 1){ echo "?page=".( $page - 1); } ?>">صفحه قبل</a>
```

```
</li>
```

```
<li class="page-item disabled">
```

```
<a class="page-link">صفحه <?=$page?> از <?=$pages ?></a>
```

```
</li>
```

```
<li class="page-item <?php if($page >= $pages){ echo 'disabled'; } ?>">
```

```
<a class="page-link" href="<?php if($page < $pages) { echo "?page=".( $page + 1); } ?>">صفحه بعد</a>
```

```
</li>
```

```
</li>
```

```
<li><a class="page-link" href="?page=<?php echo $pages; ?>">صفحه آخر</a></li>
```

```
</ul>
```

۶ در پایان هم با استفاده از دستور `sql` که قبلاً ساخته شد، لیست رکوردها از بانک اطلاعاتی خوانده شده و نمایش داده می‌شود.

در شکل ۵-۱۵ اجرای صفحه `book_pages.php` را می‌بینید که در هر صفحه ۳ کتاب را نمایش می‌دهد:



شکل ۵-۱۵- نمایش صفحه‌بندی شده رکوردها

در این فصل با روش‌های کلی برای جست‌وجوی رکوردها، محدود کردن و صفحه‌بندی رکوردها آشنا شدید. با استفاده از این نمونه کدها در کنار کدهایی که در فصل‌های قبل پیاده‌سازی کردید می‌توانید صفحات کامل و کاربردی را طراحی نمایید.

خودآزمایی

با در نظر گرفتن جدول کاربران (users) به صورت زیر، به سؤالات پاسخ دهید:

نام ستون	کاربرد
uid	شناسه
uname	نام کاربر
status	وضعیت (۰- غیرفعال، ۱- فعال)

- ۱ یک دستور SQL برای خواندن مشخصات کاربران فعال از ردیف ۳۰ تا ۴۰ و به صورت مرتب‌شده براساس نام بنویسید.
- ۲ دستوراتی برای خواندن اطلاعات کاربران فعال و نمایش آن‌ها در یک جدول بنویسید.
- ۳ دستوراتی برای دریافت یک کلمه با روش post و نمایش رکوردهایی از جدول کاربران که نام آن‌ها شامل آن کلمه باشد، بنویسید.

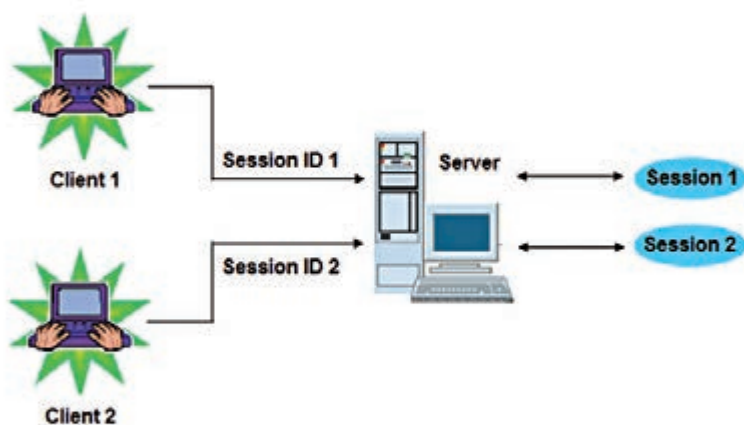
دستور کار عملی

- ۱ یک صفحه برای نمایش کتاب‌ها طراحی کنید که در بالای صفحه، یک لیست انتخابی پویا شامل همه موضوعات و یک دکمه نمایش، داشته باشد و با انتخاب یک موضوع بتوانید لیست کتاب‌های آن موضوع را نمایش دهید.
- ۲ با استفاده از فایل‌های قالب، صفحه‌ای برای جست‌وجوی مشتریان براساس نام یا آدرس آن‌ها طراحی و کدنویسی کنید.
- ۳ با استفاده از قالب بخش عمومی سایت که در فصل‌های قبل طراحی شد، یک صفحه برای جست‌وجوی کتاب‌ها طراحی کنید که بتواند کتاب‌ها را بر اساس موضوع، بخشی از عنوان یا توصیف جست‌وجو کند. برای نمایش موضوعات از لیست انتخابی پویا استفاده کنید و کتاب‌ها را به صورت جعبه‌های کنار هم نمایش دهید.
- ۴ صفحه‌ای برای نمایش لیست کتاب‌ها طراحی کنید به طوری که کتاب‌ها را به صورت صفحه‌بندی شده نمایش دهد.



فصل شانزدهم

کار با sessionها



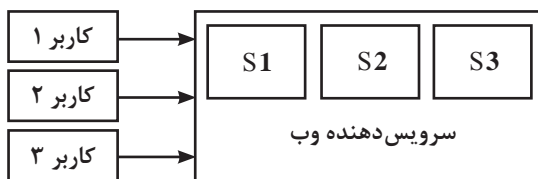
هدف‌های رفتاری

- انتظار می‌رود هنرجو در پایان این فصل بتواند:
- مفهوم sessionها را شرح دهد.
- دستورات کار با sessionها را در برنامه استفاده کند.
- با کمک sessionها، اطلاعات را بین صفحات مختلف به اشتراک بگذارد.
- با کمک sessionها سبد خرید یک فروشگاه اینترنتی را پیاده‌سازی کند.

۱-۱۶- تعریف Session

Session یا جلسه (نشست) از نظر لغوی مجموعه‌ای از ارتباطات و تبادل اطلاعات موقت بین دو یا چند دستگاه ارتباطی، رایانه یا کاربر است که در زمان مشخص شروع شده و پس از گذشت زمان مشخصی پایان می‌یابد. به‌عنوان مثال در دنیای واقعی یک جلسه آموزشی را در نظر بگیرید که در زمان مشخصی شروع شده، اطلاعاتی بین دانش‌آموزان و معلم رد و بدل شده و در زمان مشخصی به پایان می‌رسد. همچنین وقتی کاربر یک برنامه را در محیط ویندوز اجرا کرده و یا از طریق مرورگر یک سایت را باز می‌کند، یک جلسه را شروع کرده است. در طی این جلسه با برنامه یا وب سایت تعامل داشته، اطلاعاتی از آن به دست آورده یا به آن می‌دهد و در نهایت از آن خارج شده و به جلسه پایان می‌دهد.

از دید برنامه‌نویسی مبتنی بر وب، session یک فضای حافظه روی سرور می‌دهنده است که به هر کاربر داده می‌شود. وقتی کاربر وارد یک سایت می‌شود، این فضا به او تخصیص داده شده و یک سری اطلاعات مربوط



شکل ۱-۱۶- نحوه کار Session

به کاربر در آن ذخیره می‌شود. به این صورت که این فضا فقط به یک کاربر اختصاص دارد و اگر چندین کاربر همزمان وارد سایت شده باشند، همان‌طور که در شکل ۱-۱۶ می‌بینید برای هر کاربر یک session مخصوص ایجاد شده و در اختیار او قرار می‌گیرد:

گفته شد هر کاربر دارای یک session اختصاصی بوده که از دسترس کاربران دیگر محافظت شده است. در اینجا منظور از کاربر همان برنامه مرورگر است. با اجرای مرورگر، کاربر به‌وجود آمده، برای او session تخصیص داده شده و با بسته شدن مرورگر کاربر از بین رفته و به‌طور پیش‌فرض session او هم می‌تواند پاک شود. اطلاعات قرار داده شده در session مربوط به یک کاربر در تمام صفحات آن سایت قابل دسترس و استفاده هستند. پس session می‌تواند یک فضای مشترک بین صفحات مختلف سایت، جهت به اشتراک گذاری اطلاعات یک کاربر باشد و می‌توان از آن برای اهداف مختلفی استفاده کرد. نکته آخر اینکه session یک ابزار در اختیار برنامه‌نویس سایت است تا از آن برای ذخیره کردن اطلاعات یا به اشتراک گذاشتن داده‌ها بین صفحات مختلف یک سایت برای هر کاربر استفاده کند. کاربران سایت از وجود session اطلاعی نداشته و نیازی هم به دانستن آن ندارند.

۲-۱۶- دستورات PHP برای کار با Session

هر زبان برنامه‌نویسی مبتنی بر وب از جمله PHP امکاناتی برای کار با sessionها فراهم می‌کند. در این قسمت روش دسترسی به session و نوشتن داده‌ها در آن و یا خواندن داده‌های آن را در زبان PHP فرا می‌گیرید. در زبان PHP فضای session را به صورت یک آرایه انجمنی پیش ساخته با نام \$_SESSION در اختیار برنامه‌نویس قرار می‌گیرد (همه حروف به صورت بزرگ نوشته می‌باشد) و اطلاعات به صورت جفت کلید - مقدار در آن ذخیره می‌شوند. همچنین دارای توابعی برای کار با session می‌باشد که در ادامه شرح داده شده‌اند.

۱-۲-۱۶- شروع به کار Session

در هر صفحه‌ای از سایت، قبل از اینکه بتوان از session استفاده نموده و یا اطلاعاتی در آن ذخیره کرد، باید session شروع به کار کند. برای شروع به کار آن از تابع session_start() به صورت زیر استفاده می‌شود:

شروع به کار Session
<pre><?php session_start(); ?> <html> <body> ... </body> </html></pre>

توجه داشته باشید که تابع session_start() همواره باید قبل از تگ <html> قرار گیرد و هیچ گاه نباید قبل از اجرای آن، اطلاعاتی به خروجی فرستاده شده باشد.

۲-۲-۱۶- نوشتن داده‌ها در Session

بعد از شروع به کار session، می‌توان اطلاعات را در آن ذخیره کرد. همان طور که پیش تر گفته شد، اطلاعات باید در آرایه انجمنی \$_SESSION با تعیین یک کلید دلخواه نوشته شوند. به عنوان مثال می‌توان برای ذخیره کردن نام کاربر از کلید user استفاده کرد:

شروع به کار Session و ذخیره کردن نام کاربر در آن
<pre><?php session_start(); // ذخیره کردن نام کاربر \$_SESSION['user']='ali'; ?> <html><body> // ادامه طراحی صفحه</pre>

۳-۲-۱۶ خواندن داده‌ها از Session

برای خواندن داده‌ها، بعد از شروع به کار session، می‌توان خانه‌های آرایه \$_SESSION را خواند، البته به شرط اینکه این خانه‌ها از قبل مقداردهی شده باشند. توجه داشته باشید که می‌توان داده‌ها را در یک صفحه از سایت در session ذخیره کرده و در صفحات دیگر از session خواند.

خواندن و نمایش داده‌های Session

```
<?php
session_start();
?>
<html>
<body>
<h1>user name : <?=$_SESSION['user'];?></h1>
</body>
```

۴-۲-۱۶ از بین بردن Session

بعضی مواقع لازم می‌شود تا داده‌های session پاک شوند. به‌عنوان مثال هنگام خروج کاربر از سایت باید داده‌های شخصی او از session پاک شود. برای حذف خانه‌ای با کلید مشخص از session می‌توان از تابع unset() به‌صورت زیر استفاده کرد:

حذف کردن خانه‌ای از Session

```
<?php
session_start();
unset($_SESSION['user']);
?>
```

برای از بین بردن کل اطلاعات session می‌توان از تابع session_destroy() به‌صورت زیر استفاده کرد:

از بین بردن کامل Session

```
<?php
session_start();
Session_destroy();
?>
```

همان‌طور که در کد صفحه قبل می‌بینید، برای از بین بردن session، باید اول آن را شروع به کار کرده و سپس حذف نمود.

مثال: با استفاده از session برای صفحه یک شمارنده ایجاد کنید که با هر بار نمایش صفحه، یک واحد به آن اضافه شود.

برای این کار در آرایه \$_SESSION، یک خانه با کلید count در نظر گرفته و دستوراتی بنویسید که در ابتدای صفحه وجود آن را بررسی کند. اگر این خانه وجود داشت، یک واحد به آن اضافه کرده و در غیر این صورت آن را با ۱ مقداردهی کند تا ایجاد شود. در پایان مقدار آن را چاپ نماید. با هر بار درخواست و نمایش این صفحه، عدد جدیدی برای شمارنده خواهید دید و با بستن مرورگر و اجرای مجدد صفحه، شمارنده از ۱ شروع خواهد شد.

پیاده‌سازی شمارنده برای صفحه با استفاده از Session

```
<?php
session_start();
if ( isset($_SESSION['count']))
    $count = $_SESSION['count'];
else $count = 0;
$count ++ ;
$_SESSION['count'] = $count;
?>
<html>
<body>
<h1>Counter : <?=$count ?></h1>
</body>
</html>
```

۳-۱۶- کاربرد Session در نمایش پیغام بازخورد

از session می‌توان برای به اشتراک‌گذاری اطلاعات بین صفحات یک سایت استفاده کرد. به‌عنوان مثال فرض کنید در یک صفحه عمل اضافه، ویرایش یا حذف کتاب را انجام داده و با تابع header به صفحه لیست کتاب‌ها تغییر مسیر می‌دهید، می‌خواهید در بالای صفحه لیست کتاب‌ها، پیغام مناسب نمایش دهید تا کاربر از انجام موفق یا خطای احتمالی آن آگاه شود که به این پیغام‌ها بازخورد می‌گویند.

برای نمایش بازخورد، بعد از انجام عمل اضافه، حذف یا ویرایش می‌توانید پیغام مورد نظر را در یک خانه از session ذخیره کرده و در صفحه بعدی آن را نمایش دهید.

مثال: قطعه کد صفحه بعد عمل حذف کتاب را انجام می‌دهد و پیغام مناسب را در session ذخیره می‌کند:

پیاده‌سازی book-del.php با ارسال بازخورد

```
<?php
session_start();
$bid=$_GET['bid'];
require('config.php');
$sql= " DELETE FROM books WHERE bid=$bid " ;
$res = mysqli_query( $link , $sql );
if ( $res ){
    $_SESSION['fb'] = 'رکورد با موفقیت حذف شد';
    header('location:book-list.php');
} else echo mysqli_error($link);
?>
```

در کد فوق بعد از اجرای دستور حذف روی بانک اطلاعاتی، نتیجه بررسی می‌شود، اگر حذف رکورد موفق بود، پیغام موفقیت در خانه‌ای از session با کلید fb ذخیره شده و به صفحه book-list.php تغییر مسیر داده می‌شود. بخشی از پیاده‌سازی صفحه book-list.php در زیر نشان داده شده است:

فایل stud-list.php همراه با نمایش پیغام

```
<?php
session_start();
$page_title='مدیریت کتاب‌ها';
include('header.php');
if (isset($_SESSION['fb'])) { ?>
<div class="alert alert-success">
    <?php
        echo $_SESSION['fb'];
        unset($_SESSION['fb']);
    ?>
</div>
<?php } ?>
<table class="table table-bordered">
```

در کد فوق در بالای صفحه وجود `$_SESSION['fb']` بررسی می‌شود: اگر در حافظه وجود داشت مقدار آن در یک کادر مناسب نمایش داده شده و این خانه session از حافظه پاک می‌شود تا دوباره استفاده نشود.

در شکل ۱۶-۲ اجرای صفحه نمایش لیست کتاب‌ها را پس از حذف کتاب مشاهده می‌کنید:



شکل ۱۶-۲- نمایش بازخورد پس از حذف کتاب

۴-۱۶- پیاده‌سازی سبد خرید فروشگاه

یک فروشگاه اینترنتی امکانی برای کاربران فراهم می‌کند تا با گردش در سایت و مشاهده کالاها و خدمات مختلف، در صورت تمایل آنها را به سبد خرید خود اضافه کنند و در پایان با مشاهده سبد خرید و تأیید آن، خرید خود را قطعی نمایند.

همان‌طور که پیش‌تر گفته شد، یکی از کاربردهای session به اشتراک گذاشتن داده‌های بین صفحات مختلف سایت است و از آنجا که سبد خرید باید در صفحات مختلف در دسترس کاربر باشد و به عبارت دیگر همه صفحات سایت باید بتوانند به صورت مشترک به آن دسترسی داشته باشند، می‌توان برای پیاده‌سازی آن از session استفاده کرد.

در ادامه برای فروشگاه اینترنتی کتاب طی چند مرحله سبد خرید با کمک session پیاده‌سازی شده است:

مرحله اول: طراحی دکمه افزودن به سبد

در صفحه نمایش جزئیات کتاب که در فصل‌های قبل طراحی شد، برای کتاب یک پیوند با عنوان «افزودن به سبد»، به شکل ۱۶-۳ اضافه شده است:



شکل ۱۶-۳- دکمه افزودن به سبد برای کتاب

برای ساخت دکمه «افزودن به سبد» پیوند زیر اضافه شده تا صفحه cart-add.php را درخواست کرده و شناسه کتاب را به عنوان پارامتر به آن ارسال کند:

```
<a class="btn btn-success"
href="cart-add.php?bid=?=$book['bid']?>>
```

افزودن به سبد

```
</a>
```

برای نمایش پیوند به صورت دکمه سبز رنگ از کلاس های btn و btn-success از bootstrap استفاده شده است.

مرحله دوم: افزودن کتاب به سبد خرید

کلیدها	مقادیر
1001	2
1003	1

برای نگه داری لیستی از کتاب‌ها، از یک آرایه انجمنی به نام cart استفاده می‌شود که کلیدهای آن را شناسه‌های کتاب‌ها و مقادیر آن را تعداد کتاب‌ها برای خرید تشکیل می‌دهند. به عنوان مثال در آرایه روبه‌رو ۲ عدد کتاب با شناسه ۱۰۰۱ و ۱ عدد کتاب با شماره ۱۰۰۳ ذخیره شده است. این آرایه در session نگه داری می‌شود.

صفحه cart-add.php را مشابه زیر طراحی کنید تا پارامتر شناسه کتاب را دریافت کرده و در session ذخیره کند:

```

برنامه cart-add.php
<?php
session_start();
$bid=$_GET['bid'];
if ( isset($_SESSION['cart']) )
    $cart = $_SESSION['cart'] ;
else $cart = array();
if( isset($cart[$bid]) )
    $cart[$bid] ++;
else $cart[$bid]=1;
$_SESSION['cart'] = $cart;
header("location:book-show.php?bid=$bid");
?>
```

در این صفحه، session آغاز به کار کرده و شناسه کتاب دریافتی به عنوان پارامتر را در متغیر \$bid قرار می‌دهد.

سپس بررسی می‌شود اگر `$_SESSION['cart']` در حافظه وجود دارد آن را به متغیر `$cart` نسبت می‌دهد در غیر این صورت `$cart` را به صورت یک آرایه خالی ایجاد می‌کند:

```
if ( isset($_SESSION['cart']) )
```

```
    $cart = $_SESSION['cart'] ;
```

```
else $cart = array();
```

در ادامه دوباره بررسی می‌کند اگر آرایه `$cart` خانه‌ای با کلید مساوی شناسه کتاب دارد به آن خانه از آرایه یک واحد اضافه می‌کند تا تعداد کتاب‌های درخواستی یکی بیشتر شود در غیر این صورت یک خانه جدید با کلید مساوی شناسه این کتاب با مقدار به آرایه اضافه می‌نماید:

```
if( isset($cart[$bid]) )
```

```
    $cart[$bid] ++;
```

```
else $cart[$bid]=1;
```

در پایان آرایه `$cart` را در `session` قرار داده و به صفحه نمایش جزئیات کتاب تغییر مسیر می‌دهد. در این تغییر مسیر، شناسه کتاب هم به عنوان پارامتر به صفحه بعدی ارسال می‌شود:

```
$_SESSION['cart'] = $cart;
```

```
header("location:book-show.php?bid=$bid");
```

بنابراین اگر در صفحه نمایش جزئیات کتاب روی دکمه «افزودن به سبد» کلیک شود، کتاب به سبد خرید اضافه شده و دوباره همین صفحه نمایش داده می‌شود.

مرحله سوم: نمایش اقلام سبد خرید

همان‌گونه که در شکل ۱۶-۳ هم دیدید، یک پیوند به صفحه «سبد خرید» در بالای صفحه وب قرار داده شده تا کاربر در هر زمان بتواند جزئیات سبد خرید خود را مشاهده نموده و ادامه مراحل خرید را تکمیل نماید:

```
<a href="cart-show.php">سبد خرید</a>
```

پیاده‌سازی صفحه `cart-show.php` را می‌توان به صورت زیر انجام داد:

صفحه `cart-show.php`

```
<?php
session_start();
if(isset($_POST['cart']))
    $_SESSION['cart']= $_POST['cart'];

include('pheader.php');
if (! isset($_SESSION['cart'] )
    || count($_SESSION['cart'])==0 )
    echo '<h5 class="title1">سبد خرید شما خالی است</h5>';
else {
```

```

// نمایش محتوای سبد
?>
<h1 class="title2">سبد خرید</h1>
<form action="cart-show.php" method="post">
<table class="table table-bordered" >
<tr>
<td>شناسه</td><td>نام کتاب</td>
<td>تعداد</td><td>قیمت</td>
</tr>
<?php
$cart = $_SESSION['cart'] ;
$total=0;
foreach($cart as $bid=>$qty )
{
$sql = "SELECT * FROM books where bid=$bid";
$res = mysqli_query( $link , $sql );
$book = mysqli_fetch_assoc( $res );
$total += intval($qty) * $book['price'];
?>
<tr>
<td><?= $book['bid'] ?></td>
<td><?= $book['bname'] ?></td>
<td><?= $book['price'] ?></td>
<td width="300">
<input type="text" size="10" name="cart[<?=$bid?>]"
value="<?=$qty?>"class="form-control" /></td>
</tr>
<?php } ?>
<tr>
<td colspan="3" >
ریال : <?=$total?> جمع کل
</td>
<td>
<input type="submit" class="btn btn-info" value="ثبت تغییرات" />

```

```

</td>
</tr>
</table>
</form>
<?php }
include('pfooter.php');
?>

```

در این صفحه هم مشخصات کامل کتاب‌های موجود در سبد نشان داده می‌شود و دارای یک فرم است هم تعداد کالاها قابل ویرایش بوده و کاربر می‌تواند تغییرات را انجام داده و فرم را ارسال کند. در ابتدای صفحه بررسی می‌شود اگر سبد خرید خالی است پیغام مناسب نشان می‌دهد در غیر این صورت محتوای سبد نشان داده می‌شود:

```

if (! isset($_SESSION['cart'] )
    || count($_SESSION['cart'])==0 )
    echo '<h5 class="title1">سبد خرید شما خالی است</h5>';
else {
    // نمایش محتوای سبد
?>

```

در ادامه، آرایه کالاها از session به دست می‌آید که یک آرایه انجمنی است:

```

$cart = $_SESSION['cart'] ;

```

این آرایه با کمک حلقه تکرار foreach پیمایش می‌شود. برای هر عنصر آن کلید (شناسه کتاب) در متغیر \$bid و مقدار (تعداد کتاب) در متغیر \$qty قرار می‌گیرد و مشخصات کتاب براساس شناسه کتاب از بانک اطلاعاتی خوانده شده و در یک سطر جدول نمایش داده می‌شود:

```

<?php
$total=0;
foreach($cart as $bid=>$qty )
{
    $sql = "SELECT * FROM books where bid=$bid";
    $res = mysqli_query( $link , $sql );
    $book = mysqli_fetch_assoc( $res );
    $total += intval($qty) * $book['price'];
    //...

```

کل جدول در داخل یک تگ form قرار گرفته و برای هر کدام از کتاب‌ها از یک ورودی از نوع جعبه متن طراحی شده که نام آن آرایه cart با کلید شناسه کتاب است:

```
<input type="text" size="10" name="cart[<?=$bid?>]"
value="<?=$qty?>"class="form-control" />
```

چنانچه کاربر تعداد کتاب‌ها را تغییر داده و روی دکمه «ثبت تغییرات» کلیک کند، آرایه cart که شامل ورودی‌های تعداد برای همه ردیف‌ها است، ارسال شده و در سطر دوم برنامه تشخیص داده شده و در session ذخیره می‌شود.

از متغیر \$total برای نگهداری جمع کل سفارش استفاده می‌شود. در شروع کار مقدار صفر به آن نسبت داده شده و برای هر قلم از سفارش تعداد در قیمت واحد ضرب شده و به آن اضافه می‌گردد و در پایان جمع مبلغ سفارش نمایش داده می‌شود:

```
<td colspan="3" > جمع کل : <?=$total?>ریال</td>
```

در شکل ۴-۱۶ نمایش صفحه سبد خرید (cart-show.php) را مشاهده می‌کنید:



شکل ۴-۱۶- نمایش سبد خرید

تا اینجا مراحل اولیه خرید از سایت فروشگاه را پیاده‌سازی کرده و به دلیل رعایت اختصار و سنگین نشدن مطالب از اجرای مراحل بعدی صرف‌نظر می‌کنیم. واضح است که برای تکمیل خرید باید مراحل بعدی آن از جمله دریافت مشخصات خریدار و آدرس تحویل سفارش، اتصال به درگاه پرداخت بانک و نمایش تأییدیه ثبت سفارش پیاده‌سازی شود.

در این فصل با مفهوم session و نحوه استفاده از آن در برنامه‌ها آشنا شده و مهارت استفاده از آن در نرم‌افزارهای مبتنی بر وب را کسب کردید. در فصل بعدی به تشریح کاربرد دیگری از session برای احراز هویت و کنترل دسترسی کاربران به صفحات سایت پرداخته می‌شود.

خودآزمایی

- ۱ مفهوم session را شرح دهید.
- ۲ مفهوم session را از دید برنامه‌نویسی مبتنی بر وب شرح دهید.
- ۳ توابع مهم PHP برای کار با session را نوشته و با ذکر مثال توضیح دهید.
- ۴ برای از بین بردن یک خانه از session، چه دستوری اجرا می‌کنیم، با مثال نشان دهید.
- ۵ مراحل لازم برای احراز هویت و کنترل دسترسی کاربران با استفاده از session را بنویسید.
- ۶ برای انتقال اطلاعات از یک صفحه به صفحه دیگر با کمک session چگونه عمل می‌شود؟ با ذکر مثال توضیح دهید.

سؤالات چهارگزینه‌ای

- ۱ از دید برنامه‌نویسی PHP، کدام گزینه از ویژگی‌های session نمی‌باشد؟
 - الف) بین همه کاربران سایت مشترک است.
 - ب) بین همه صفحات سایت مشترک است.
 - ج) در برنامه‌نویسی از آن استفاده می‌شود.
 - د) داده‌ها را به صورت زوج کلید - مقدار ذخیره می‌کند.
- ۲ اطلاعات session در چه صورت از بین می‌رود؟
 - الف) باز کردن نسخه جدیدی از مرورگر
 - ب) بسته شدن مرورگر
 - ج) اجرای تابع `session_destroy`
 - د) موارد ب و ج
- ۳ در زبان PHP برای نگهداری داده‌های session از کدام آرایه استفاده می‌شود؟
 - الف) `$session`
 - ب) `$_session`
 - ج) `$SESSION`
 - د) `$_SESSION`
- ۴ برای حذف همه اطلاعات session از کدام دستور استفاده می‌شود؟
 - الف) `unset($_SESSION)`
 - ب) `session_destroy()`
 - ج) `session_delete()`
 - د) `delete($_SESSION)`
- ۵ استفاده از session برای نگهداری کدام یک از موارد زیر توصیه نمی‌شود؟
 - الف) مشخصات کاربر جاری سایت
 - ب) اقلام موجود در سبد خرید
 - ج) آمار بازدیدهای سایت
 - د) پیغام بازخورد هنگام تغییر مسیر درخواست

سؤال	۱	۲	۳	۴	۵
پاسخ	الف	د	د	ب	ج

دستور کار عملی

- ۱ یک صفحه طراحی کنید و با استفاده از session، شمارنده‌ای در آن قرار دهید که با هر بار تازه‌سازی (refresh) صفحه، یک واحد به آن اضافه شود و پس از رسیدن به ۱۰ دوباره از یک شروع کند.
- ۲ در فروشگاه اینترنتی کتاب بعد از عملیات حذف، اضافه و ویرایش کتاب‌ها، پیغام بازخورد تولید کرده و در صفحه لیست کتاب‌ها نمایش دهید.
- ۳ برای یک فروشگاه اینترنتی دلخواه، صفحات زیر را طراحی کنید:
نمایش لیست کالاها، نمایش جزئیات یک کالا، افزودن به سبد خرید، نمایش جزئیات سبد خرید.

فصل هفدهم

احراز هویت و کنترل دسترسی کاربران

ورود اعضا

کد ملی


رمز عبور

~~auafz~~

عبارت امنیتی

ورود به سایت

رمز عبور خود را فراموش کرده اید



هدف‌های رفتاری

انتظار می‌رود هنرجو در پایان این فصل بتواند:

- مفهوم احراز هویت و کنترل دسترسی را توضیح دهد.
- صفحه ورود کاربر به سایت را طراحی کند.
- با استفاده از sessionها، احراز هویت کاربر را پیاده‌سازی کند.
- دسترسی کاربران به صفحات سایت را محدود کند.
- صفحه تغییر رمز عبور برای کاربر جاری را پیاده‌سازی کند.

در فصل‌های قبل روش تحلیل، طراحی و پیاده‌سازی بخش‌های مختلف سیستم‌های مبتنی بر وب بیان شده و عملیات مختلف روی بانک اطلاعات پیاده‌سازی شد. همچنین در فصل قبل مفهوم session و دستورات مربوط به آن ارائه گردید.

در این فصل روشی معرفی می‌شود تا دسترسی کاربران به بخش‌های مختلف سیستم کنترل شده و برای این کار از بانک اطلاعاتی برای نگهداری مشخصات کاربران و از session برای کنترل ورود و خروج کاربران استفاده شود.

۱۷-۱- کنترل دسترسی کاربران

وقتی با نرم‌افزارهای مبتنی بر ویندوز کار می‌کنید، کل امکانات نرم‌افزار به صورت به هم پیوسته و هم‌زمان با هم روی سیستم اجرا شده و سیستم می‌داند که شما چه کسی هستید. همچنین می‌داند که چه وقت برنامه را شروع کرده و چه وقت پایان داده‌اید.

حال نرم‌افزار تحت وب را در نظر بگیرید، که از صفحات کاملاً مستقل تشکیل شده‌اند. این صفحات که روی سرورس دهنده وب قرار گرفته‌اند و با هر درخواست یکی از صفحات اجرا می‌شود همچنین ممکن است هزاران کاربر هم از سرتاسر اینترنت به طور هم‌زمان در حال درخواست صفحات آن باشند.

هر نرم‌افزار مبتنی بر وب یا وب سایت معمولاً دارای دو مجموعه صفحه می‌باشد: مجموعه صفحات عمومی سایت که برای هر مراجعه کننده‌ای قابل اجرا و نمایش هستند و مجموعه صفحات خصوصی یا مدیریتی که دارای امکاناتی برای مدیریت رکوردها یا گزارشات مهم هستند و تنها برای کاربران عضو یا مدیران وب سایت، پس از تأیید هویت و ورود به سایت، براساس نقش و اختیارات از قبل تعیین شده کاربر قابل مشاهده هستند. بنابراین برای جلوگیری از دسترسی‌های غیر مجاز افراد باید دو اقدام اصلی انجام شود.

۱ احراز هویت: اینکه تنها کاربران مجاز بتوانند به سیستم وارد شده و برنامه‌ها را اجرا کنند که روش معمول، استفاده از فرم ورود کاربر و دریافت و بررسی نام کاربری و کلمه عبور است.

۲ مجاز شماری: اینکه کاربران وارد شده به سیستم، تنها براساس مجوزهای از قبل تعیین شده به امکانات دسترسی داشته باشند که یک روش اختصاص نقش به هر کاربر و فراهم کردن دسترسی به صفحات براساس نقش او می‌باشد.

۲-۱۷- احراز هویت کاربران

درخواست صفحات وب و پاسخ دهی به آنها با استفاده از پروتکل HTTP انجام می‌شود که به صورت state-less عمل می‌کند. یعنی سرویس‌دهنده‌های وب حالت را نگه‌داری نمی‌کنند. به عبارت دیگر برای هر کاربر سابقه درخواست‌های او را ثبت نمی‌کنند.

فرض کنید یک کاربر صفحه ورود را درخواست کرده و نام کاربری و کلمه عبور خود را وارد می‌کند و هویت او تأیید می‌شود. همین کاربر وقتی چند لحظه بعد درخواست صفحه دیگر را می‌کند، سرویس‌دهنده ورود او را به خاطر نیاورده و مشابه یک کاربر تازه وارد با او برخورد می‌کند. برای حل این مشکل، یک روش این است که در صفحه ورود به سایت پس از تأیید هویت کاربر، یک نشانه به‌عنوان مثال نام کاربری یا حتی همه مشخصات کاربر در session ذخیره شود و از آنجا که session بین همه صفحات سایت مشترک است، قبل از اجرای هر صفحه بررسی صورت گیرد: اگر نشانه در session ذخیره شده باشد به معنای تأیید هویت کاربر است و در غیر این صورت نشان‌دهنده عدم تأیید هویت او می‌باشد. در ادامه احراز هویت کاربران با استفاده از session به صورت مرحله به مرحله نشان داده می‌شود:

مرحله ۱: ایجاد جدول کاربران در بانک اطلاعاتی

برای احراز هویت کاربران، قبل از هر کاری باید مشخصات کاربران مجاز را در جایی نگه‌داری کرد که می‌توان یک جدول از بانک اطلاعاتی به این کار اختصاص داد. جدول ۱-۱۷ می‌تواند دارای ستون‌های زیر باشد.

جدول ۱-۱۷

نام ستون	کاربرد	ملاحظات
uid	شناسه کاربر	کلید اصلی و خودافزایی
uname	نام اصلی کاربر	رشته‌ای
mobile	شماره همراه	رشته‌ای
pass	کلمه عبور	رشته‌ای

نام کاربری می‌تواند به صورت یک ستون رشته‌ای جداگانه تعریف شود یا اینکه یکی از مشخصات منحصر به فرد کاربر مثل کد ملی، شماره همراه یا آدرس پست الکترونیکی به‌عنوان نام کاربری استفاده شود. در مثال‌های این فصل از شماره همراه به عنوان نام کاربری استفاده شده است.

مرحله ۲: پیاده‌سازی صفحه ورود به سایت

فرم ورود به سایت یکی از صفحات عمومی محسوب شده و هر مراجعه‌کننده به سایت می‌تواند آن را اجرا نماید. در این صفحه معمولاً دو کلمه به‌عنوان نام کاربری (شماره همراه) و رمز عبور از کاربر دریافت شده و ارسال می‌شود تا در جدول کاربران بانک اطلاعاتی جست‌وجو شده و درستی یا نادرستی آنها تعیین گردد. در صفحه login.php، فرم ورود کاربر طراحی شده است:

طراحی login1.php

```
<html>
<body dir="rtl">
<form action="login2.php" method="post">
<h1>ورود کاربر</h1><hr>
نام کاربری: <br><input name="mobile" type="text" /><br>
کلمه عبور: <br><input name="pass" type="password" /><br>
<input type="submit" value="ورود" />
<input type="reset" value="بازنویسی" />
</form>
</body>
</html>
```

اطلاعات فرم فوق به صفحه login2.php ارسال می‌شود. در صفحه login2.php پس از دریافت اطلاعات فرم ورود، به بانک اطلاعاتی متصل شده و وجود کاربری با این نام کاربری و کلمه عبور بررسی می‌شود:

پیاده‌سازی صفحه login2.php

```
<?php
session_start();
$user = $_POST['user'];
$pass = $_POST['pass'];
require('config.php');
$sql = "select * from `users` where `user`='".$user'
      and `pass`='".$pass' ";
$res = mysqli_query( $link , $sql );
$row = mysqli_fetch_assoc($res);
if ( $row ) {
    $_SESSION['user'] = $row['name'];
    header('location:home.php');
} else echo "نام کاربری یا کلمه عبور نامعتبر است ";
?>
```

در برنامه فوق یک پرس‌وجو برای یافتن کاربران دارای نام کاربری و کلمه عبور دریافت شده از فرم، روی بانک اطلاعاتی اجرا می‌شود. اگر کاربری با این مشخصات پیدا شد، نام کاربری در متغیر `$_SESSION['uname']`

ذخیره شده و به صفحه home.php تغییر مسیر داده می‌شود. در غیر این صورت پیغام خطا نمایش داده می‌شود. نام جدول users و ستون‌های user و pass در داخل نقل قول `` قرار داده شده‌اند (که با دکمه بالای tab در سمت چپ صفحه کلید تایپ شده است). اگر نام جدول یا ستون دارای فاصله بوده یا از کلمات رزرو شده باشد، باید داخل این نقل قول قرار گیرد. در این قسمت برای راحت‌تر شدن فهم برنامه، از دو صفحه login1.php و login2.php استفاده شده است، می‌توان با استفاده از روش‌هایی که در فصل‌های قبل بیان شد، این دو را با هم ادغام کرده و تنها یک صفحه با نام login.php داشت.

مرحله ۳: کنترل دسترسی کاربر به صفحه

برای نمایش صفحات خصوصی و مدیریتی، نیاز است تا ورود کاربران بررسی شود. می‌دانید که اطلاعات موجود در session در همه صفحات سایت در دسترس بوده و می‌تواند استفاده شود. در صفحه پردازش فرم ورود، بعد از اطمینان از درست بودن نام کاربری و کلمه عبور، نام کاربری او در session قرار می‌گیرد.

حال برای نمایش هر صفحه خصوصی، در ابتدای صفحه بررسی می‌شود که نام کاربری در session ذخیره شده است یا خیر؟ برای این کار وجود `$_SESSION['uname']` بررسی می‌شود. اگر این خانه وجود داشت به معنی تأیید نام کاربری و کلمه عبور در صفحه پردازش فرم ورود است و کاربر می‌تواند ادامه صفحه را اجرا کند، در غیر این صورت کاربر اجازه نمایش این صفحه را نداشته و باید ابتدا فرم ورود را پر کرده و وارد شود. پس به صفحه ورود تغییر مسیر صورت می‌گیرد:

قطعه کد بررسی اعتبار کاربر

```
<?php
session_start();
if ( ! isset($_SESSION['user']) )
    header('location:login1.php');
?>
ادامه طراحی صفحه//
...
```

مرحله ۴: خروج کاربر از سیستم

وقتی کاربری از طریق فرم ورود تأیید هویت شده و مجاز شناخته شده تا صفحات مدیریتی را مشاهده کند و عملیات مدیریت اطلاعات مثل درج، حذف و ویرایش انجام دهد، لازم است تا در بالای همه صفحات مدیریتی نام کاربری (یا نام واقعی) و لینک «خروج از سیستم» به او نشان داده شود تا در صورت تمایل بتواند از سیستم خارج شود.

برای اضافه کردن پیوند خروج در بالای هر صفحه از قطعه کد زیر استفاده می‌شود:

کد نمایش پیغام خوش آمدگویی و گزینه خروج
Welcome <?php echo \$_SESSION['user']; ?> [Logout]

در قطعه کد فوق یک پیوند به فایل logout.php نشان داده شده است. در فایل logout.php می‌توان session را شروع کرده و آن را از بین برده و سپس کاربر را به صفحه ورود هدایت کرد:

کد برای خروج کاربر
<?php session_start(); session_destroy(); header("location:login1.php"); ?>

۳-۱۷- پیاده‌سازی احراز هویت برای فروشگاه کتاب

در فصل‌های قبل با طراحی تعدادی از صفحات عمومی و خصوصی در فروشگاه اینترنتی آشنا شدید. در فروشگاه اینترنتی کتاب فرض شده است که هر کدام از کاربران سایت می‌توانند دارای نقش‌های زیر باشند: نقش **مراجعه‌کننده**: افرادی که به صفحات عمومی سایت از جمله صفحه اول سایت، لیست کتاب‌ها، مشخصات هر کتاب و ارتباط با ما دسترسی داشته و می‌توانند کتاب‌ها را جست‌وجو کرده و به سبد خرید خود اضافه کنند. نقش **مشتری**: کاربرانی که از سایت خرید داشته‌اند و قبلاً در طی مراحل خرید، مشخصات خود را وارد کرده و در سایت ثبت نام شده‌اند. این افراد می‌توانند پس از ورود به پنل شخصی، مشخصات خود و وضعیت سفارشات قبلی خود را مشاهده کنند.

نقش **مدیر فروشگاه**: کاربرانی که پس از ورود به پنل شخصی، امکان مدیریت رکوردهای کتاب، مشتریان، سفارشات را دارند. همچنین می‌توانند به‌عنوان یک مشتری وضعیت سفارشات خود را ببینند.

در این بخش صفحه عمومی برای ورود کاربران فروشگاه به صفحات خصوصی را پیاده‌سازی کرده و در بخش‌های بعدی صفحات خصوصی برای ویرایش پروفایل و تغییر کلمه عبور را برای کاربران مشتری و مدیر را به سایت فروشگاه اضافه می‌کنیم.

۱-۳-۱۷- ایجاد جدول کاربران فروشگاه

ابتدا در بانک اطلاعاتی فروشگاه، جدول زیر برای نگهداری همه کاربران فروشگاه شامل مشتریان و کاربران مدیر ایجاد می‌شود.

جدول ۱۷-۲- جدول کاربران (users)

نام ستون	کاربرد	ملاحظات
uid	شناسه کاربر	کلید اصلی و خودافزایشی
uname	نام اصلی کاربر	رشته‌ای
role	نقش کاربر	رشته‌ای
mobile	شماره همراه	رشته‌ای
pass	کلمه عبور	رشته‌ای
avatar	تصویر کاربر	رشته‌ای
addr	آدرس	رشته‌ای
status	وضعیت (فعال یا غیرفعال)	عددی

از ویژگی role برای نگهداری نقش کاربر استفاده می‌شود. مقدار این فیلد برای مشتریان مساوی customer و برای مدیران فروشگاه، مقدار admin خواهد بود.

۱۷-۳-۲ پیاده‌سازی صفحه ورود کاربر

برای طراحی صفحه ورود از فایل‌های قالب عمومی سایت طراحی شده در فصل‌های قبل استفاده می‌شود. در شکل ۱۷-۱ نمای ظاهری صفحه ورود به سایت را مشاهده می‌کنید:



شکل ۱۷-۱- صفحه ورود به سایت

پیاده‌سازی صفحه ورود به سایت (login.php) به صورت زیر است:

صفحه login.php

```
<?php
$errors=[];
if ( isset($_POST['mobile'])) {
    $mobile=$_POST['mobile'];
    $pass=$_POST['pass'];
    if ($mobile==' ') $errors[]='شماره همراه وارد نشده است';
    if ( count($errors)==0)
    {
        require('config.php');
        $sql="SELECT * FROM users WHERE mobile='$mobile' AND pass='$pass' ";
        $res=mysqli_query($link , $sql);
        $row=mysqli_fetch_assoc($res);
        if ( $row )
        {
            session_start();
            $_SESSION['user']=$row;
            header('location:admin/home.php');
        } else $errors[]='شماره همراه یا کلمه عبور نادرست است';
    }
}
include('pheader.php');
include('show-errors.php'); // نمایش خطاها
?>
<form action="login.php" method="post" >
<h2 class="title2">ورود به سایت</h2>
<div class="form-group">
    <label for="mobile">شماره همراه</label>
    <input type="text" name="mobile" class="form-control" />
</div>
<div class="form-group">
    <label for="pass">رمز عبور</label>
    <input type="password" name="pass" class="form-
```



```
control" />
</div>
<div class="form-group">
    <input type="submit" value="ورود به سایت" class="btn btn-info" />
</div>
</form>
<?php include('pfooter.php'); ?>
```

در بخش پردازش فرم صفحه فوق در صورتی که شماره همراه و نام کاربری در جدول کاربران وجود داشته باشد، رکورد مربوط به کاربر خوانده شده و در متغیر \$row قرار می‌گیرد سپس دستورات زیر اجرا می‌شود:

```
if ( $row )
{
    session_start();
    $_SESSION['user']=$row;
    header('location:admin/home.php');
} else $errors[]="شماره همراه یا کلمه عبور نادرست است";
```

در دستورات فوق بررسی می‌شود اگر \$row دارای اطلاعات است، session شروع به کار کرده و کل رکورد کاربر با کلید user در session ذخیره می‌شود و پس از آن به صفحه home.php که صفحه پیشخوان کاربر است تغییر مسیر داده می‌شود.

توجه کنید که صفحه home.php در زیرپوشه admin قرار گرفته است. با ذخیره کردن کل رکورد کاربر در session در هر کدام از صفحات سایت می‌توان به همه اطلاعات کاربر دسترسی داشت. به‌عنوان مثال برای به‌دست آوردن نام یا نقش کاربر می‌توان دستورات زیر را استفاده کرد:

```
$uname=$_SESSION['user']['uname'];
$role=$_SESSION['user']['role'];
```

در صفحه فوق از شماره همراه به‌عنوان نام کاربری استفاده شده است. با این کار کاربران سایت که اکثراً مشتریان فروشگاه هستند نیازی ندارند تا نام کاربری مخصوص این سایت را به‌خاطر بسپارند و هر از چند گاهی که به سایت مراجعه کنند می‌توانند تنها با به‌خاطر آوردن کلمه عبور به آن وارد شوند. همچنین در سایت‌ها گزینه‌ای با نام «فراموشی رمز عبور» قرار داده می‌شود که با کمک آن می‌توانند رمز عبور فراموش شده را بازنشانی کنند.

۳-۳-۱۷- کنترل دسترسی به صفحات سایت

هر کاربر پس از تأیید هویت و ورود به پنل شخصی، تنها براساس اختیارات خود باید به گزینه‌های آن دسترسی داشته باشد. برای سایت فروشگاه کاربران دارای نقش «مشتری» گزینه‌های زیر را در دسترس دارند:

۱ نمایش صفحه پیشخوان

۲ ویرایش پروفایل خودشان (مشخصات کاربر جاری)

۳ تغییر کلمه عبور خودشان

۴ نمایش لیست سفارشات خودشان

۵ خروج از پنل شخصی

کاربر مدیر فروشگاه علاوه بر گزینه‌های فوق گزینه‌های زیر را هم در دسترس خواهد داشت:

۱ مدیریت موضوعات

۲ مدیریت کتاب‌ها

۳ مدیریت مشتریان

۴ مدیریت سفارشات

۵ ویرایش گروهی قیمت‌ها

در نتیجه هنگام نمایش لیست گزینه‌ها در فایل قالب (header.php) نقش کاربر بررسی می‌شود و تنها در صورتی که admin بود گزینه‌های کاربر مدیر به او نمایش داده می‌شود.

```
<ul >
<li><a href="home.php">پیشخوان</a></li>
<li><a href="profile.php">پروفایل</a></li>
<li><a href="password.php">تغییر کلمه عبور</a></li>
<?php if($_SESSION['user']['role']=='admin'){ ?>
<li><a href="book-list.php">مدیریت کتاب‌ها</a></li>
<li><a href="sub-list.php">مدیریت موضوعات</a></li>
<li><a href="user-list.php">مدیریت مشتریان</a></li>
<li><a href="order-list.php">مدیریت سفارشات</a></li>
<li><a href="book-price.php">ویرایش گروهی قیمت‌ها</a></li>
<?php } ?>
</ul>
```

علاوه بر این باید در ابتدای هر صفحه اعتبار کاربر و نقش او بررسی شده و در صورت داشتن مجوز، اجازه نمایش صفحه داده شود چرا که ممکن است کاربری بدون استفاده از لینک‌های سایت بلکه با تایپ مستقیم آدرس صفحات در نوار آدرس مرورگر قصد اجرای آنها را داشته باشد.

روش‌های مختلف و پیشرفته‌ای برای بررسی اعتبار و مجوز دسترسی به صفحات وجود دارد که در این جا ساده‌ترین روش استفاده شده است. برای این منظور یک صفحه به نام check.php به صورت زیر ایجاد می‌شود:

صفحه check.php

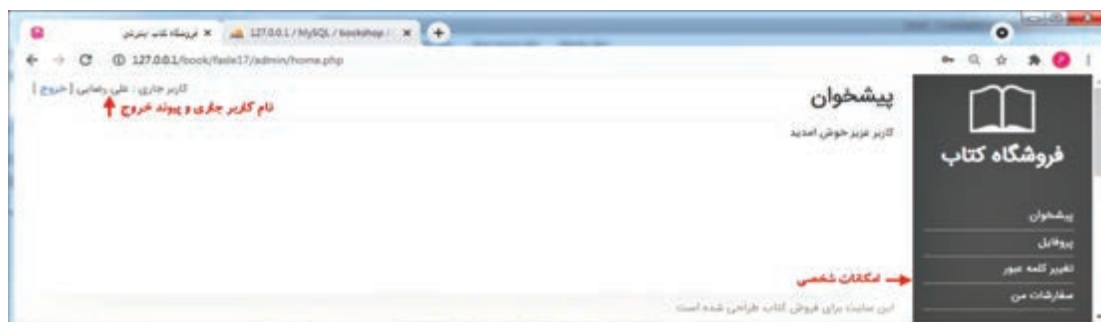
```
<?php
session_start();
if( !isset($_SESSION['user']))
    header('location:login.php');
if(isset($role) && $role!=" && $_SESSION['user']['role']!=$role)
    header('location:home.php');
?>
```

در صفحه فوق ابتدا session شروع به کار می کند، سپس بررسی می شود که کاربر وارد شده است یا خیر؟ اگر متغیر `$_SESSION['user']` وجود نداشته باشد نشان دهنده این است که کاربر به سایت وارد نشده است. بنابراین به صفحه ورود (`login.php`) به سایت تغییر مسیر داده می شود. در ادامه بررسی می شود اگر متغیر `$role` وجود دارد و مقدار آن غیر پوچ است و مقدار آن با فیلد `role` کاربر جاری مساوی نیست پس کاربر مجوز دسترسی به این صفحه را نداشته و به صفحه پیشخوان (`home.php`) تغییر مسیر داده می شود. برای اجرای دستورات `check.php` در ابتدای صفحاتی که مربوط به مدیریت فروشگاه هستند، متغیر `$role` را مساوی `admin` قرار داده و برای دیگر صفحات که مربوط به مشتریان است، متغیر `$role` را مساوی پوچ قرار داده یا اصلاً مقدار نداده و سپس فایل `check.php` ضمیمه می شود. به عنوان مثال صفحه پیشخوان (`home.php`) که برای مشتریان و مدیران فروشگاه قابل نمایش است به صورت زیر پیاده سازی می شود:

```

صفحه home.php
<?php
$role=' ';
include('check.php');
$page_title = 'پیشخوان';
include('header.php'); ?>
<h1>
    کاربر عزیز خوش آمدید
</h1>
<?php include('footer.php'); ?>
    
```

در شکل ۲-۱۷، صفحه پیشخوان کاربر به همراه نام کاربر و پیوند خروج را مشاهده می کنید. همچنین همه کاربران در پنل کاربری خود گزینه هایی برای ویرایش مشخصات و کلمه عبور خود دارند:



شکل ۲-۱۷- نمایش صفحه پیشخوان

و به عنوان مثال صفحه نمایش لیست کتاب‌ها (book-list.php) که فقط برای مدیران فروشگاه قابل نمایش است، به صورت زیر عمل می‌شود:

```
بخش اول صفحه book-list.php

<?php
$role='admin';
include('check.php');
$page_title='مدیریت کتاب‌ها';
include('header.php');
// ادامه طراحی صفحه
```

۴-۳-۱۷- خروج کاربر از سایت

معمولاً بعد از ورود کاربر به سایت، در گوشه‌ای از همه صفحات، اطلاعات نام کاربر و پیوند خروج از سایت به او نمایش داده می‌شود. برای این منظور در فایل قالب header.php مربوط به صفحات مدیریتی فروشگاه اینترنتی که طراحی سرصفحه را در خود دارد، می‌توان به صورت زیر طراحی انجام داد تا در گوشه چپ بالای صفحه نام کاربر جاری و پیوند خروج را نمایش دهد.

```
بخشی از طراحی header.php

<div class="header">
    <?=$page_title ?>
    <span class="user">
        کاربر جاری : <?=$_SESSION['user']['uname']?>
        [ <a href="logout.php">خروج</a> ]
    </span>
</div>
```

پس از اینکه کاربر روی پیوند خروج کلیک کرد، صفحه logout.php اجرا شده که می‌تواند به صورت زیر پیاده‌سازی شود که پس از پاک کردن اطلاعات session، به صفحه اول فروشگاه تغییر مسیر می‌دهد.

```
صفحه logout.php

<?php
session_start();
session_destroy();
header('location:index.php');
?>
```

۴-۱۷- پیاده‌سازی صفحه تغییر کلمه عبور

هر کاربر باید بتواند کلمه عبور خود را به دلخواه تغییر دهد. برای این منظور یک صفحه به نام password.php طراحی می‌کنیم که دارای یک فرم برای دریافت کلمه عبور قبلی، کلمه عبور جدید و تکرار کلمه عبور جدید بوده و دستوراتی برای پردازش این فرم نوشته می‌شود. در بخش پردازش فرم ابتدا موارد زیر بررسی می‌شود و اگر مشکلی وجود نداشت کلمه عبور در جدول کاربران اصلاح می‌شود.

■ طول کلمه عبور از عدد معینی کمتر نباشد.

■ کلمه عبور و تکرار آن مساوی باشد.

■ کلمه عبور قبلی درست وارد شده باشد.

صفحه password.php

```
<?php
$role=' ';
include('check.php');
require('config.php');
$errors=[];
$uid = $_SESSION['user']['uid'];
if ( isset($_POST['pass'])) {
    $pass = $_POST['pass'];
    $pass1 = $_POST['pass1'];
    $pass2 = $_POST['pass2'];

    if ($pass != $_SESSION['user']['pass'])
        $errors[]='رمز عبور قبلی نادرست است';
    if ( strlen($pass1)<6 )
        $errors[]='طول رمز عبور نباید از ۶ کاراکتر کمتر باشد';
    if ( $pass1!=$pass2 )
        $errors[]='رمز عبور و تکرار آن یکسان نیست';
    if(count($errors)==0) {
        $sql="UPDATE `users` SET pass='$pass1' WHERE uid=$uid ";
        $res = mysqli_query($link , $sql );
        if($res){
            $_SESSION['fb']='رمز عبور شما با موفقیت تغییر کرد';
            header('location:home.php');
```

```

        }else $errors[] ='تغییر رمز عبور با خطا مواجه شد';
    }
}
$page_title = 'تغییر کلمه عبور';
include('header.php');
include('show-errors.php');
$res= mysqli_query($link , "SELECT * FROM users WHERE uid=$uid " );
$row= mysqli_fetch_assoc($res);
?>
<form action="password.php" method="post" >
<div class="form-group">
    <label for="author">شماره همراه</label>
    <input readonly type="text" name="mobile" value="<?=$row['mobile']?>" class=
"form-control" />
</div>
<div class="form-group">
    <label for="pass1">رمز عبور قبلی</label>
    <input type="password" name="pass" class="form-control" />
</div>
<div class="form-group">
    <label for="pass">رمز عبور جدید</label>
    <input type="password" name="pass1" class="form-control" />
</div>
<div class="form-group">
    <label for="pass2">تکرار رمز عبور</label>
    <input type="password" name="pass2" class="form-control" />
</div>
<div class="form-group">
    <input type="submit" value="تغییر رمز عبور" class="btn btn-info" />
</div>
</form>
<?php include('footer.php'); ?>

```

در صفحه قبل ابتدا شناسه کاربر جاری از session به دست آمده و براساس آن، مشخصات کامل کاربر از بانک اطلاعاتی خوانده شده و در متغیر \$user قرار می‌گیرد:

```
$uid = $_SESSION['user']['uid'];
```

```
$res = mysqli_query($link , "SELECT * FROM `users` WHERE uid=$uid " );
```

```
$user= mysqli_fetch_assoc($res);
```

سپس از \$user برای بررسی درستی رمز عبور قبلی و یا نمایش شماره همراه کاربر در فرم استفاده می‌شود. پس از بررسی اطلاعات فرم، پیغام‌های خطا در آرایه \$errors ذخیره شده و سرانجام با ضمیمه کردن فایل show-errors.php که در فصل‌های قبل طراحی شد، نمایش داده می‌شود.

۵-۱۷- پیاده‌سازی امکانات شخصی کاربران

کاربران فروشگاه با هر نقشی که داشته باشند پس از ورود به سایت و احراز هویت می‌توانند به امکانات شخصی زیر دسترسی داشته باشند:

■ ویرایش پروفایل

■ تغییر کلمه عبور

■ پیگیری سفارشات

■ خروج

در ادامه دو صفحه ویرایش پروفایل و پیگیری سفارشات قبلی را پیاده‌سازی می‌کنیم:

ویرایش پروفایل: هر کاربر پس از ورود به صفحه شخصی خود می‌تواند مشخصات خود را مشاهده کرده و در صورت تمایل آنها را ویرایش نماید. برای این منظور صفحه profile.php ایجاد شده است که در شکل ۳-۱۷ اجرای آن را مشاهده می‌کنید:



شکل ۳-۱۷- صفحه ویرایش پروفایل

برای پیاده‌سازی صفحه profile.php مشابه ویرایش کتاب که در فصل‌های قبل مشاهده کردید، عمل می‌شود با این تفاوت که به‌جای استفاده از پارامتر ورودی برای شناسه رکورد، شناسه کاربر جاری را session به‌دست می‌آید:

```

بخشی از پیاده‌سازی profile.php

<?php
include('check.php');
require('config.php');
$errors=[];
$uid=$_SESSION['user']['uid']; // خواندن شناسه کاربر جاری از سشن
if ( isset($_POST['uname'])) {
// دستورات دریافت مشخصات و فایل تصویر از فرم و ذخیره کردن آنها
}
$page_title = 'ویرایش پروفایل';
include('header.php');
include('show-errors.php');
$res= mysqli_query($link , "SELECT * FROM users WHERE uid=$uid " );
$row= mysqli_fetch_assoc($res);?>
<form action="profile.php" ... >
طراحی اجزای فرم
</form>
<?php include('footer.php'); ?>

```

۶-۱۷- صفحه پیگیری سفارشات قبلی

جدول ۳-۱۷- جدول orders برای نگه‌داری سفارشات

نام و ویژگی	نام ستون	نوع داده
شناسه سفارش	oid	عددی
تاریخ	odate	تاریخ
شناسه کاربر مشتری	uid	عددی
آدرس پستی	addr	رشته‌ای
شناسه وضعیت	sid	عددی

هر کدام از مشتریان سایت باید بتوانند پس از ورود به پنل کاربری خود لیست سفارشات قبلی را مشاهده کرده و از وضعیت انجام آن مطلع گردند. همچنین باید بتوانند جزئیات هر سفارش را مشاهده نمایند. با توجه به طراحی جدول orders که در روبه‌رو مشاهده می‌کنید:

شناسه کاربر، کلید خارجی مرتبط با جدول کاربران است. با کمک این شناسه می‌توان سفارشات مربوط به کاربر جاری را از بانک اطلاعاتی استخراج کرد. همچنین شناسه وضعیت کلید خارجی مرتبط با جدول states می‌باشد که همه وضعیت‌های ممکن برای سفارش از آغاز تا پایان را نگهداری می‌کند:

جدول ۴-۱۷. جدول states برای نگهداری وضعیت‌ها

نام ویژگی	نام فیلد	نوع داده
شناسه وضعیت	sid	عددی
عنوان	sname	رشته‌ای

در صفحه my-orders.php ابتدا شناسه کاربر جاری را از session به دست آورده و سپس با اجرای یک پرس‌وجو در جدول orders، لیست رکوردهای با uid مساوی شناسه کاربر جاری خوانده می‌شود. در شکل ۴-۱۷ اجرای صفحه پیگیری سفارشات (my-orders.php) را مشاهده می‌کنید:



شکل ۴-۱۷. صفحه پیگیری سفارشات

در صفحه my-orders.php امکان جست‌وجوی سفارشات براساس وضعیت یا بین دو تاریخ مورد نظر فراهم شده است ضمن اینکه در هر سطر به جای شناسه وضعیت، عنوان آن نمایش داده شده و یک پیوند به صفحه نمایش جزئیات سفارش هم وجود دارد. در ادامه بخش‌هایی از پیاده‌سازی صفحه my-orders.php را مشاهده می‌کنید:

```

صفحه my-orders.php
<?php
include('check.php');
$page_title = 'سفارشات من';
include('header.php');
require('config.php');
$states = getStates(); // فراخوانی تابع برای به دست آوردن لیست وضعیت‌ها
$uid = $_SESSION['user']['uid']; // به دست آوردن شناسه کاربر جاری از سشن
    
```

```

$ssql= "SELECT * FROM orders WHERE uid=$uid ";
$ssid = ' ';
$date1 = ' ';
$date2 = ' ';
if (isset($_POST['date1']))    {
    $ssid =intval($_POST['sid']);
    $date1=intval($_POST['date1']);
    $date2=intval($_POST['date2']);
    if ($ssid!=0) $ssql.=" AND sid=$ssid ";
    if ($date1!=0) $ssql .=" AND odate>=$date1 ";
    if ($date2!=0) $ssql .=" AND odate<=$date2 ";
}
?>
<form action="my-orders.php" method="post" >
    <div class="row">
        <div class="col-sm-1">وضعيت</div>
        <div class="col-sm-2">
<select name="sid" class="form-control" />
            <option value="0">---</option>
            <?php
foreach($states as $id=>$name)    { ?>
<option value="<?=$id?" <?php if($id==$sid) echo "selected"; ?>>
                <?=$name?>
            </option>
            <?php } ?>
        </select>
    </div>
    <div class="col-sm-1">تاريخ</div>
    <div class="col-sm-2">
<input value="<?=$date1?" type="text" name="date1" class="form-control"/>
    </div>
    <div class="col-sm-1">تا تاريخ</div>
    <div class="col-sm-2">
<input value="<?=$date2?" type="text" name="date2" class="form-control"/>
    </div>

```

```

<div class="col-sm-3">
<input type="submit" value="جست و جو" class="btn btn-info"/>
</div>
</div>
</form>
<table class="table table-bordered">
<thead>
<tr>
<th scope="col">شناسه</th>
<th scope="col">تاریخ</th>
<th scope="col">آدرس</th>
<th scope="col">وضعیت</th>
<th scope="col"></th>
</tr>
</thead>
<tbody>
<?php
$res = mysqli_query($link , $sql);
echo mysqli_error($link);
while($row=mysqli_fetch_assoc($res))    {?>
<tr>
<td><?=$row['oid']?></td>
<td><?=$row['odate']?></td>
<td><?=$row['addr']?></td>
<td><?=$states[$row['sid']]?></td>
<td>
<a href="my-order-show.php?oid=<?=$row['oid']?>">جزئیات</a>
</td>
</tr>
<?php } ?>
</tbody>
</table>
<?php include('footer.php'); ?>

```

در کد فوق فرض شده است تابع `getStates()` در داخل فایل `config.php` به صورت صفحه بعد پیاده سازی شده و لیست وضعیت‌ها را از بانک اطلاعاتی خوانده، در یک آرایه انجمنی قرار داده و برمی گرداند. چون در جاهای مختلف سایت به لیست وضعیت‌ها نیاز داریم و استفاده از این تابع، با توجه به اینکه تنها یک بار اطلاعات از بانک اطلاعاتی خوانده شده و چندین بار استفاده می‌شود، باعث صرفه جویی در زمان اجرای صفحات می‌گردد.

در ادامه محتوای فایل config.php نشان داده شده است:

فایل config.php جدید

```
<?php
$link = mysqli_connect('127.0.0.1:3308','root','','bookshop2');
mysqli_query($link , 'SET NAMES utf8');

function getStates(){
    global $link;
    $res = mysqli_query($link , "SELECT * FROM states ");
    $arr=[];
    while($row=mysqli_fetch_assoc($res)) {
        $arr[$row['sid']] = $row['sname'];
    }
    return $arr;
}
?>
```

همچنین در صفحه سفارشات، امکان جست‌وجوی سفارش براساس وضعیت یا بین دو تاریخ دلخواه قرار داده شده است.

در این فصل با روش احراز هویت با استفاده از session و کنترل دسترسی کاربران در یک نرم‌افزار مبتنی بر وب آشنا شده و مهارت لازم برای پیاده‌سازی فرم ورود کاربر و کنترل دسترسی کاربران به هر صفحه براساس نقش و اختیارات از قبل تعیین شده در سایت فروشگاه اینترنتی را به دست آوردید. همچنین توانستید صفحه ویرایش پروفایل و صفحه نمایش لیست سفارشات قبلی مشتری را برای کاربر جاری پیاده‌سازی کنید. در فصل‌های قبل نیز صفحات مختلفی برای فروشگاه کتاب طراحی کردید.

در اینجا جهت رعایت اختصار از طراحی دیگر صفحات فروشگاه خودداری می‌شود. با مهارت‌هایی که تاکنون به دست آورده‌اید، می‌توانید صفحات این فروشگاه را کامل کرده و یا حتی یک سیستم دیگر را از صفر تا صد طراحی و کدنویسی کنید. با این حال مجموعه فیلم آموزشی برای پیاده‌سازی کامل فروشگاه، ضمیمه کتاب شده است که می‌تواند برای تکمیل صفحات دیگر فروشگاه و یادگیری راحت تر برنامه‌نویسی به شما کمک نماید.

خودآزمایی

- ۱ منظور از state-less بودن پروتکل HTTP چیست؟ توضیح دهید.
- ۲ مراحل لازم برای احراز هویت را شرح دهید.
- ۳ منظور از کنترل دسترسی کاربران چیست؟ توضیح دهید.
- ۴ روش کنترل دسترسی کاربران به گزینه‌های مختلف مدیریتی را با ذکر مثال توضیح دهید.
- ۵ دستورات لازم برای کنترل دسترسی کاربر با توجه به نقش آن در ابتدای صفحات خصوصی را بنویسید.

سوالات چهارگزینه‌ای

- ۱ کدام ویژگی سرویس‌دهنده وب منجر به این رفتار می‌شود که سابقه درخواست کاربران را نگهداری نکند؟
 الف) connection-less (ب) state-less
 ج) connection-full (د) state-full
- ۲ برای نگهداری یک نشانه مشترک بین صفحات سایت، جهت اثبات احراز هویت کاربر از چه وسیله‌ای استفاده می‌شود؟
 الف) session (ب) file
 ج) بانک اطلاعاتی (د) متغیر
- ۳ کدام گزینه می‌تواند به عنوان نام کاربری برای ورود به سایت استفاده شود؟
 الف) کد ملی (ب) شماره همراه
 ج) پست الکترونیکی (د) همه موارد
- ۴ دستورات زیر در چه شرایطی تغییر مسیر به صفحه home.php را انجام نمی‌دهند؟

```
if(isset($role) && $role!=" && $_SESSION['user']['role']!=$role)
    header('location:home.php');
```

- الف) متغیر \$role تعریف نشده باشد.
- ب) متغیر \$role مقدار پوچ داشته باشد.
- ج) متغیر \$role در session قرار داشته باشد.
- د) موارد الف و ب

پاسخنامه

سؤال	۱	۲	۳	۴
پاسخ	ب	الف	د	د

دستور کار عملی

- ۱ صفحه ورود به کاربر را برای سایت فروشگاه اینترنتی پیاده‌سازی کنید.
- ۲ در طراحی صفحه ورود از فایل‌های قالب طراحی شده در فصل‌های قبل کمک بگیرید.
- ۳ در جای مناسبی از قالب سایت نام کاربری و پیوند خروج را نمایش داده و صفحه خروج از سایت را پیاده‌سازی کنید.
- ۴ برای کاربر جاری صفحات تغییر کلمه عبور و ویرایش پروفایل را طراحی و کدنویسی کنید.

- 1 www.php.net
- 2 www.w3schools.com
- 3 www.getbootstrap.com
- 4 www.codenevisan.com
- 5 An Introduction to Database Systems by C.J. Date



سازمان پژوهش و برنامه‌ریزی آموزشی جهت ایفای نقش خطیر خود در اجرای سند تحول بنیادین در آموزش و پرورش و برنامه درسی ملی جمهوری اسلامی ایران، مشارکت هنرآموزان را به‌عنوان یک سیاست اجرایی مهم دنبال می‌کند. به‌منظور تحقق این امر مهم، اعتبارسنجی کتاب‌های درسی را در دستور کار خود قرار داده است تا با دریافت نظرات هنرآموزان درباره کتاب‌های نو نگاشت، کتاب‌های درسی را در اولین سال چاپ، با کمترین اشکال به هنرجویان و هنرآموزان ارجمند تقدیم نماید. در اجرای مطلوب این فرایند، همکاران گروه تحلیل محتوای آموزشی و پرورشی استان‌ها و گروه‌های آموزشی نقش سازنده‌ای را بر عهده داشتند. ضمن ارج نهادن به تلاش تمامی این همکاران، اسامی هنرآموزانی که تلاش مضاعفی را در این زمینه داشته و با ارائه نظرات خود سازمان را در بهبود محتوای این کتاب یاری کرده‌اند به شرح زیر اعلام می‌شود.

اسامی دبیران و هنرآموزان شرکت‌کننده در اعتبارسنجی کتاب طراحی صفحات وب پیشرفته - کد ۳۱۲۲۵۱

ردیف	نام و نام خانوادگی	استان محل خدمت
۱	سحر اسماعیلی	تهران
۲	زهرا کردی	شهرستان‌های تهران
۳	لیلا سعید	تهران
۴	مریم شفیعی	تهران
۵	لیلا داودی ثانی	خراسان رضوی
۶	بیتا رهنمای زربیحاری	شهرستان‌های تهران
۷	حسن جعفریه	تهران
۸	همتا بیداریان	تهران
۹	نگار نصر	تهران
۱۰	زهرا سلیمی‌زاده	تهران