بنام ایران ...

برخیز که غیر از تو مرا دادرسی نیست
گویی همه خوابند، کسی را به کسی نیست
آزادی و پرواز از آن خاک به این خاک
جز رنج سفر از قفسی تا قفسی نیست
(ه.الف.سایه)

پروژه برنامه سازی پیشرفته

(فاز دوم)

استاد درس :

دکتر علی جاویدانی

دستیاران آموزشی :

کیانا جابری - مهدیس یعقوب انصاری - مبینا صفری - پارسا دولت آبادی - متین امیرپناه فر - محمد متین سلیمانی - شهاب ثمری شمس - نیما مخملی

زمستان 1404

سلام دوباره به همه کدنویس‌های خفن!

فاز اول تمام شد و شما حالا یک موتور قدرتمند در اختیار دارید.
چه در پروژه شبکه اجتماعی Linko و چه در پروژه Chess++، شما هسته‌ی منطقی سیستم را ساخته‌اید کلاس‌هایی مانند Piece، Board، Network، User و سایر اجزای اصلی که قوانین، داده‌ها و رفتار سیستم را مدیریت می‌کنند.

اما بهترین موتور دنیا هم بدون بدنه، فرمان و صندلی، قابل استفاده نیست.
در فاز اول شما «منطق» را ساختید؛ در فاز دوم قرار است به آن «چهره» بدهید.

در این مرحله، کنسول سیاه و سفید (CLI) را کنار می‌گذاریم. دیگر خبری از تایپ کردن دستورات خشک و دیدن خروجی‌های متنی پشت سر هم نیست. کاربر باید بتواند:

- **با موس کلیک کند**
- **دکمه‌ها را فشار دهد**
- **عناصر را انتخاب کند**
- **و از رنگ‌ها، تصاویر و چیدمان بصری لذت ببرد**

اما یک نکته بسیار مهم:
هدف این فاز بازنویسی پروژه نیست.
هنر شما در این مرحله آن است که همان کلاس‌هایی که در فاز اول طراحی کرده‌اید (مثل Board یا Network) را بدون تغییر در منطق داخلی، به یک رابط گرافیکی متصل کنید.

به بیان دیگر:
به جای cout کردن خروجی‌ها، آن‌ها را روی یک پنجره گرافیکی رسم خواهید کرد.

ابزار کار — چرا Raylib؟

برای پیاده‌سازی رابط گرافیکی در زبان C++ گزینه‌های متعددی وجود دارد؛ از فریم‌ورک‌های کامل و صنعتی گرفته تا کتابخانه‌های سبک گرافیکی. با این حال، پیشنهاد اکید تیم آموزشی استفاده از raylib است.

چرا Raylib انتخاب مناسبی برای این پروژه است؟

1. سادگی و شفافیت در کدنویسی
برخلاف فریم‌ورک‌های سنگین مانند Qt که مبتنی بر Designer و Drag & Drop هستند، در raylib همه چیز مستقیماً با کدنویسی ساخته می‌شود.
شما پنجره را با کد ایجاد می‌کنید، دکمه را با کد رسم می‌کنید و رویدادها را با کد مدیریت می‌کنید. این موضوع باعث می‌شود درک عمیق‌تری از ساختار رابط کاربری و نحوه عملکرد آن پیدا کنید.

2. سبک، سریع و بدون پیچیدگی‌های اضافی
raylib یک کتابخانه کم‌حجم و مینیمال است. برای شروع کار تنها کافی است فایل هدر را #include کرده و برنامه را کامپایل کنید. نیازی به تنظیمات پیچیده، فایل‌های پروژه سنگین یا وابستگی‌های گسترده نخواهید داشت.

توابع آن بسیار مستقیم و قابل فهم هستند، مانند:

- DrawText
- DrawTexture
- DrawRectangle
- IsMouseButtonPressed

این سادگی باعث می‌شود تمرکز شما بر «طراحی رابط کاربری» و «اتصال آن به منطق پروژه» باقی بماند، نه درگیری با تنظیمات فنی محیط.

3. هماهنگ با ساختار پروژه‌های شما
از آنجا که در فاز اول تمرکز بر طراحی شیءگرا و معماری تمیز بوده است، raylib نیز به خوبی با این سبک هماهنگ می‌شود. شما می‌توانید کلاس‌های گرافیکی جداگانه تعریف کنید و آن‌ها را به هسته منطقی پروژه متصل نمایید، بدون آنکه وابسته به یک سیستم پیچیده UI باشید.

4. منابع آموزشی کامل و مثال‌های فراوان
raylib دارای مستندات رسمی، Cheat Sheet کاربردی و مثال‌های متعدد است که یادگیری را بسیار سریع می‌کند. تقریباً برای هر قابلیت اصلی، نمونه کد آماده وجود دارد.

نکته مهم

استفاده از raylib اجباری نیست و در صورت تمایل می‌توانید از کتابخانه‌هایی مانند SFML یا Qt نیز استفاده کنید.

جزئیات گرافیکی پروژه Chess++

در فاز دوم، تیم‌های پروژه **Chess++** موظف هستند یک تجربه بصری کامل و قابل‌استفاده برای بازی شطرنج طراحی کنند. هدف این بخش، تبدیل منطق ترمینالی فاز اول به یک محیط گرافیکی شفاف، منظم و کاربرپسند است؛ بدون تغییر در قوانین یا هسته منطقی بازی.

در ادامه، اجزای مورد انتظار رابط گرافیکی به‌صورت دقیق و شفاف توضیح داده شده است.

الف) صفحه بازی

صفحه شطرنج باید به‌صورت یک جدول ۸×۸ رسم شود. هر خانه باید مربع و هم‌اندازه باشد.

الزامات طراحی:

- استفاده از دو رنگ متمایز برای خانه‌ها
  (مثلاً کرم و قهوه‌ای، یا طوسی روشن و سرمه‌ای)
- صفحه باید در مرکز پنجره قرار گیرد.
- اطراف صفحه فضای خالی مناسب برای نمایش اطلاعات جانبی در نظر گرفته شود.
- اندازه پنجره باید ثابت باشد (برای مثال: 800×600 پیکسل).
  تغییر اندازه پویا (Resizable Window) الزامی نیست و توصیه نمی‌شود.

هدف این محدودیت‌ها، جلوگیری از به‌هم‌ریختگی چیدمان و ساده نگه‌داشتن طراحی است.

ب) مهره‌ها

نمایش مهره‌ها باید به‌صورت تصویری (Sprite-Based) انجام شود.

نکات مهم:

- استفاده از حروفی مانند P یا K برای نمایش مهره‌ها قابل قبول نیست.
- تصاویر باید با فرمت PNG و بدون پس‌زمینه (Transparent Background) باشند
- اندازه تصاویر باید متناسب با اندازه خانه‌ها تنظیم شود.
- هر مهره باید تصویر مخصوص به خود را داشته باشد (شاه، وزیر، رخ، اسب، فیل، پیاده).

مهره‌های خاص فاز اول

برای مهره‌های ویژه تعریف‌شده در فاز اول مانند:

- Joker
- Armored Queen
- Spy

باید آیکون متمایز در نظر گرفته شود.
نیازی به طراحی حرفه‌ای نیست؛ حتی یک تغییر رنگ ساده، اضافه کردن علامت کوچک یا تغییر جزئی در تصویر پایه نیز کافی است.

هدف، قابل تشخیص بودن مهره‌های خاص در صفحه بازی است.

ج) تعامل با کاربر (Interaction)

رابط کاربری باید مبتنی بر کلیک موس باشد و روند انتخاب و حرکت مهره‌ها به‌صورت واضح و قابل فهم نمایش داده شود.

1. انتخاب مهره

وقتی کاربر روی یک مهره کلیک می‌کند:

- دور خانه آن مهره باید یک کادر رنگی (Highlight) کشیده شود.
- این کادر باید واضح اما ساده باشد (مثلاً سبز یا آبی روشن).

2. نمایش حرکات مجاز

پس از انتخاب مهره:

- خانه‌هایی که مهره می‌تواند به آن‌ها حرکت کند باید مشخص شوند.
- این نمایش می‌تواند به یکی از روش‌های زیر انجام شود:
  - رسم یک دایره کوچک در مرکز خانه
  - تغییر رنگ پس‌زمینه خانه
  - نیمه‌شفاف کردن رنگ خانه

الزامی نیست که نمایش پیچیده باشد؛ فقط باید قابل فهم و دقیق باشد.

3. انجام حرکت

با کلیک روی خانه مقصد:

- **اگر حرکت مجاز باشد، مهره باید به موقعیت جدید منتقل شود.**
- **وضعیت صفحه باید بلافاصله به‌روزرسانی گردد.**

4. مدیریت خطا

اگر کاربر حرکتی غیرمجاز انجام دهد:

- **باید بازخورد بصری داده شود (مثلاً قرمز شدن لحظه‌ای خانه)**
- **یک صدای کوتاه خطا پخش شود (اختیاری)**

پیام متنی بزرگ یا پنجره اضافی لازم نیست؛ یک بازخورد ساده کافی است.

د) پنل اطلاعات (Side Panel)

در کنار صفحه شطرنج باید یک بخش جانبی (Side Panel) طراحی شود که اطلاعات و کنترل‌های اصلی بازی را نمایش دهد.

این پنل می‌تواند در سمت راست یا چپ صفحه قرار گیرد.

اطلاعات مورد نیاز:

1. نمایش نوبت بازیکن

- **مشخص شود نوبت سفید است یا سیاه.**
- **می‌توان از یک دایره سفید یا سیاه کوچک در کنار متن استفاده کرد.**
- **یا یک متن ساده مانند:**

White's Turn / Black's Turn

2. دکمه‌های کنترل بازی

دکمه‌های زیر باید وجود داشته باشند:

- **New Game (شروع بازی جدید)**
- **Save Game (ذخیره بازی)**
- **Load Game (بارگذاری بازی)**
- **Exit (خروج)**

طراحی این دکمه‌ها می‌تواند ساده باشد (یک مستطیل با متن داخل آن).
نیازی به انیمیشن پیشرفته نیست.

3. قبرستان (Graveyard)

مهره‌هایی که در جریان بازی حذف می‌شوند باید:

- **به‌صورت کوچک‌تر**
- **در کنار صفحه**
- **و به تفکیک رنگ (سفید و سیاه)**

نمایش داده شوند.

چیدمان ساده و مرتب کافی است. هدف، نمایش وضعیت بازی است نه طراحی پیچیده گرافیکی.

🌐 جزئیات گرافیکی پروژه Linko

در فاز دوم، تیم‌های پروژه **Linko** باید تمرکز اصلی خود را بر طراحی چیدمان (Layout)، خوانایی رابط کاربری و تجربه کاربری ساده و تمیز قرار دهند.

هدف، ساخت یک رابط گرافیکی سبک و کاربردی است؛ نه طراحی یک شبکه اجتماعی پیچیده و حرفه‌ای در سطح صنعتی. بنابراین طراحی باید:

- ساده و منظم باشد
- رنگ‌بندی ملایم و خوانا داشته باشد
- بدون انیمیشن‌های سنگین و افکت‌های پیچیده پیاده‌سازی شود

در ادامه، اجزای مورد انتظار رابط گرافیکی به‌صورت شفاف و بدون ابهام توضیح داده شده است.

صفحه ورود و ثبت‌نام (Authentication Page)

اولین صفحه برنامه باید مربوط به ورود یا ثبت‌نام کاربر باشد.

ساختار کلی صفحه:

- یک پس‌زمینه ساده (رنگ ثابت یا گرادیان ملایم)
- دو کادر ورودی (Input Box):
  - نام کاربری
  - رمز عبور
- دو دکمه بزرگ و خوانا:
  - «ورود»
  - «ثبت‌نام»

الزامات پیاده‌سازی:

- کاربر باید بتواند در کادرها تایپ کند.
  (مدیریت ورودی کیبورد در **raylib** یکی از چالش‌های آموزشی این فاز است.)
- کادر فعال باید با رنگ یا کادر متفاوت مشخص شود.
- در صورت اشتباه بودن رمز عبور:
  - یک متن کوچک قرمز رنگ زیر کادر نمایش داده شود.
  - نیازی به پنجره هشدار جداگانه نیست.
- رمز عبور می‌تواند با کاراکتر * نمایش داده شود (اختیاری اما توصیه‌شده).

طراحی باید ساده و مرتب باشد؛ نیازی به افکت‌های گرافیکی پیچیده وجود ندارد.

صفحه اصلی

پس از ورود موفق، کاربر وارد صفحه اصلی (Feed) می‌شود.

1. نوار بالا (Header)

در بالای صفحه باید یک نوار افقی وجود داشته باشد که شامل موارد زیر است:

- **لوگوی ساده برنامه (می‌تواند یک متن گرافیکی باشد)**
- **نام کاربر واردشده**
- **دکمه «خروج»**

این بخش باید ثابت باشد و با اسکرول جابه‌جا نشود.

2. لیست پست‌ها

پست‌ها باید به‌صورت عمودی و زیر هم نمایش داده شوند.

هر پست باید شامل موارد زیر باشد:

- **آواتار نویسنده**
  (می‌تواند یک دایره رنگی پیش‌فرض باشد؛ نیازی به سیستم آپلود تصویر نیست)
- **نام نویسنده**
- **متن پست**
  - **متن‌های طولانی باید به‌صورت خودکار شکسته شوند (Line Wrapping)**
- **تاریخ یا زمان انتشار**
- **بخش تعامل**

دکمه‌های تعامل

زیر هر پست باید:

- آیکون ❤️ برای لایک
- آیکون 💬 برای کامنت

قرار داشته باشد.

الزامات عملکردی:

- با کلیک روی لایک:
  - آیکون باید تغییر وضعیت دهد (توپر یا قرمز شود)
  - عدد لایک افزایش یابد
- کامنت می‌تواند فقط شمارش نمایش دهد (پیاده‌سازی کامل صفحه کامنت الزامی نیست، مگر در صورت تمایل)

طراحی باید تمیز، با فاصله مناسب بین پست‌ها و بدون شلوغی باشد.

ایجاد پست جدید

باید یک دکمه مشخص برای ایجاد پست جدید وجود داشته باشد.

پیشنهاد:

- یک دکمه «+» بزرگ در گوشه پایین صفحه

با کلیک روی این دکمه:

- یک پنجره کوچک (Modal) در مرکز صفحه باز شود.
- کاربر بتواند متن پست را وارد کند.
- دکمه «ارسال» وجود داشته باشد.

پس از ارسال:

- پنجره بسته شود.
- پست جدید در بالای فید نمایش داده شود.

نیازی به انیمیشن باز و بسته شدن پنجره نیست؛ نمایش ساده کافی است.

صفحه پروفایل کاربری

با کلیک روی نام هر کاربر (در فید):

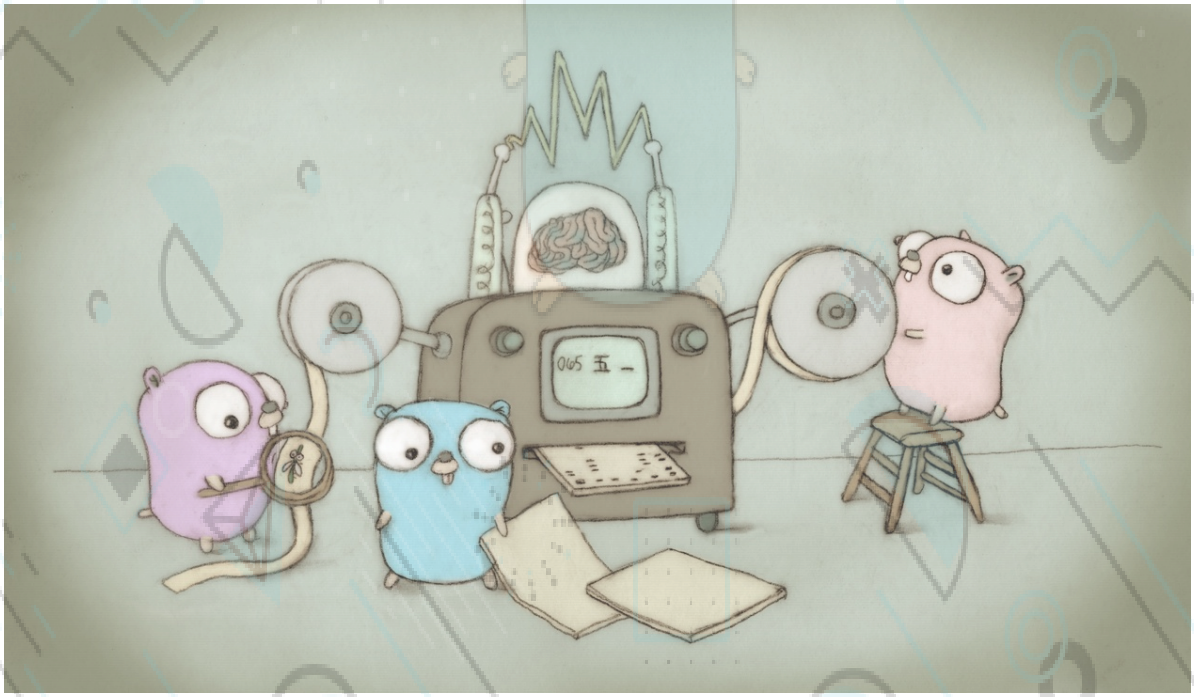
- **صفحه پروفایل آن کاربر باز شود.**

این صفحه باید شامل موارد زیر باشد:

- **نام کاربر**
- **بیوگرافی (Bio)**
- **لیست پست‌های همان کاربر (فقط پست‌های خودش)**

طراحی این صفحه می‌تواند مشابه فید باشد، اما فقط شامل پست‌های همان فرد.

یک دکمه «بازگشت» برای برگشت به فید اصلی کافی است.



ویژگی‌های امتیازی (Bonus)

بخش‌های امتیازی این فاز الزامی نیستند؛ اما پیاده‌سازی آن‌ها می‌تواند باعث افزایش نمره نهایی پروژه شود و حتی بخشی از نمرات از دست‌رفته در فاز قبل را جبران کند.

موارد پیشنهادی عمداً ساده انتخاب شده‌اند تا بدون پیچیدگی زیاد و بدون درگیر شدن با مفاهیم پیشرفته گرافیکی، قابل پیاده‌سازی باشند. تمرکز همچنان بر سادگی، خوانایی کد و اتصال صحیح به منطق فاز اول است.

در ادامه، پیشنهادهای امتیازی برای هر دو پروژه آورده شده است.

1. صداگذاری (Sound Effects)

پیاده‌سازی صدا در raylib بسیار ساده است و تنها با چند تابع قابل انجام می‌باشد.

موارد پیشنهادی:

- صدای کلیک دکمه‌ها
- صدای حرکت مهره در Chess++
- صدای لایک کردن پست در Linko

صداها می‌توانند بسیار کوتاه و ساده باشند (فرمت WAV یا MP3).
نیازی به طراحی حرفه‌ای یا میکس پیچیده نیست.

2. موسیقی پس‌زمینه (Background Music)

می‌توان یک موزیک ملایم در:

- منوی اصلی
- صفحه ورود
- یا هنگام اجرای بازی

پخش کرد.

الزامی نیست که در تمام برنامه فعال باشد. یک موزیک ساده و سبک کافی است.
قابلیت قطع/وصل کردن صدا نیز می‌تواند امتیاز اضافی محسوب شود.

3. Hover Effect (افکت قرارگیری موس روی دکمه)

برای بهبود تجربه کاربری، زمانی که موس روی یک دکمه قرار می‌گیرد:

- رنگ دکمه کمی روشن‌تر شود
- اندازه آن اندکی بزرگ‌تر شود

این تغییر باید ظریف و ساده باشد.
نیازی به انیمیشن پیچیده نیست.

4. حالت تم تاریک / روشن (Dark / Light Mode)

می‌توان یک دکمه کوچک در برنامه قرار داد که:

- رنگ‌بندی کل رابط کاربری را تغییر دهد
- بین حالت روشن (پس‌زمینه سفید) و حالت تاریک (پس‌زمینه تیره) جابه‌جا شود

در این حالت باید:

- رنگ متن‌ها
- رنگ دکمه‌ها
- رنگ پس‌زمینه

همگی هماهنگ تغییر کنند.

پیاده‌سازی این قابلیت نشان‌دهنده مدیریت مناسب رنگ‌ها و ساختار منظم در طراحی UI است.

5 . کلیدهای میانبر (Keyboard Shortcuts)

افزودن چند کلید میانبر ساده می‌تواند امتیاز اضافی داشته باشد.

مثال‌ها:

- **کلید Space برای شروع بازی جدید در Chess++**
- **کلید Esc برای خروج از برنامه**
- **کلید Enter برای ارسال پست در Linko**

تعداد محدود و کاربردی کافی است؛ نیازی به تعریف میانبرهای زیاد نیست.

6. انیمیشن ساده (Simple Animation)

افزودن یک انیمیشن سبک می‌تواند پروژه را حرفه‌ای‌تر نشان دهد، بدون آنکه پیچیدگی زیادی ایجاد کند.

در Chess++

- **مهره به جای جابه‌جایی ناگهانی،**
  به صورت نرم (Slide) به خانه مقصد حرکت کند.

این حرکت می‌تواند در مدت زمان کوتاه (مثلاً ۰.۲ تا ۰.۳ ثانیه) انجام شود.

در Linko

- **باز شدن نرم پنجره ایجاد پست**
- **ظاهر شدن تدریجی منو**
- **یا Fade ساده هنگام تغییر صفحه**

انیمیشن‌ها باید کوتاه، سبک و بدون تأثیر منفی بر عملکرد برنامه باشند.

نکات مهم درباره بخش امتیازی

- **هیچ‌کدام از این موارد اجباری نیستند.**
- **کیفیت پیاده‌سازی مهم‌تر از تعداد قابلیت‌هاست.**
- **یک یا دو ویژگی تمیز و درست، بهتر از چند ویژگی ناقص است.**
- **منطق فاز اول نباید برای اضافه کردن این قابلیت‌ها تخریب شود.**

معیار ارزیابی و بارم‌بندی (100 نمره)

ارزیابی فاز دوم بر اساس چهار بخش اصلی انجام می‌شود. معیارها به‌گونه‌ای طراحی شده‌اند که فرآیند تصحیح شفاف، منصفانه و بدون ابهام باشد.

عملکرد صحیح (40 نمره)

- اجرای برنامه بدون کرش یا خطای بحرانی
- کارکرد صحیح تمام قابلیت‌های منطقی فاز اول در محیط گرافیکی
  - در **Chess++**: حرکت‌ها، قوانین، ذخیره/بارگذاری
  - در **Linko**: لاگین، ثبت‌نام، پست، لایک، فالو و...
- هماهنگی کامل بین وضعیت منطقی سیستم و نمایش گرافیکی
- مدیریت مناسب خطاهای کاربر

رابط کاربری و کیفیت بصری (30 نمره)

- انتخاب رنگ‌های هماهنگ و خوانا
- چیدمان منظم و تراز بودن عناصر
- خوانایی متن‌ها و اندازه مناسب فونت
- طراحی تمیز، ساده و کاربرپسند

سادگی حرفه‌ای امتیاز دارد؛ شلوغی و بی‌نظمی باعث کسر نمره می‌شود.

کیفیت کدنویسی و معماری (20 نمره)

- جداسازی صحیح لایه گرافیک از منطق برنامه
- عدم انتقال منطق به کلاس‌های **UI**
- ساختار منظم پروژه و پوشه‌بندی مناسب
- عدم تمرکز تمام کد در تابع **main**
- نام‌گذاری صحیح و کدنویسی خوانا

مستندات و گیت (10 نمره)

- تاریخچه **Commit** منظم در **GitHub**
- ایجاد **Tag** نهایی مطابق دستورالعمل
- وجود فایل **README.md** شامل:
  - نحوه بیلد و اجرا
  - نحوه لینک کردن کتابخانه گرافیکی
  - توضیح ساختار پروژه

زمان‌بندی و نحوه تحویل

شما تا پایان بهمن‌ماه فرصت دارید تا پروژه‌ی خود را از یک نسخه ترمینالی به یک نرم‌افزار گرافیکی کامل تبدیل کنید.

📌 **ددلاین نهایی**

11 اسفند 1404 — ساعت23:59

پس از این زمان، مخزن‌ها بر اساس آخرین Tag ثبت‌شده بررسی خواهند شد.

محل تحویل : کونرا

مراحل گام‌به‌گام تحویل

1. **اطمینان حاصل کنید پروژه به‌طور کامل اجرا می‌شود.**
2. **پوشه assets شامل تصاویر، فونت‌ها و فایل‌های صوتی (در صورت وجود) در کنار پروژه قرار داشته باشد.**
3. **تغییرات نهایی را Commit و سپس روی GitHub Push کنید.**
4. **روی آخرین Commit یک Tag با یکی از نام‌های زیر ایجاد کنید:**
   - **v2.0**
   - **Phase2**
5. **نماینده تیم لینک ریپازیتوری را از طریق ایمیل ارسال کند.**

سخن پایانی

کار با گرافیک شاید در ابتدا چالش‌برانگیز به نظر برسد، اما با اولین پنجره‌ای که باز می‌کنید و اولین عنصری که رسم می‌کنید، جذاب‌ترین بخش برنامه‌نویسی را تجربه خواهید کرد.

این فاز فرصتی است برای ترکیب منطق مهندسی‌شده فاز اول با خلاقیت بصری شما.

منتظر دیدن پروژه‌های رنگی و خلاقانه‌تان هستیم.
موفق باشید ✌️