

پروژه: عامل هوشمند دسته‌بندی ایمیل و اسناد

۱. ایده اصلی پروژه چیست؟

این پروژه یک دستیار هوشمند برای مدیریت ایمیل‌ها و اسناد است. هدف آن این است که وقتی یک ایمیل، متن PDF، یا یک سند متنی وارد سیستم می‌شود، برنامه آن را بخواند، مفهوم کلی آن را بفهمد، موارد مهم را پیدا کند، و به کاربر بگوید که چه کاری باید انجام دهد.

به زبان ساده، این پروژه مثل یک منشی هوشمند دیجیتال عمل می‌کند. یعنی به جای اینکه کاربر همه ایمیل‌ها و فایل‌ها را خودش یکی‌یکی بخواند، سیستم ابتدا آن‌ها را بررسی می‌کند و یک خلاصه قابل فهم، سطح اهمیت، کارهای لازم، مهلت‌ها، و پیشنهاد اقدام بعدی را نشان می‌دهد.

در فایل README پروژه هم هدف اصلی همین‌طور تعریف شده است: سیستم از n8n به عنوان مسیر اجرای خودکار یا همان agentic pipeline استفاده می‌کند و یک داشبورد ساده مرورگری دارد که ایمیل‌ها، PDFها و متن اسناد را دسته‌بندی و تحلیل می‌کند.

۲. این پروژه چه مشکلی را حل می‌کند؟

امروزه بسیاری از افراد، مخصوصاً دانشجویان، کارمندان، اعضای تیم‌های کوچک، و افراد حرفه‌ای، روزانه تعداد زیادی ایمیل و سند دریافت می‌کنند. بعضی از این پیام‌ها مهم هستند، بعضی فقط اطلاع‌رسانی هستند، بعضی مهلت دارند، و بعضی هم ممکن است شامل اطلاعات حساس باشند.

مشکل این است که کاربر باید برای فهمیدن اهمیت هر پیام، زمان زیادی صرف کند.

مثلاً ممکن است بین چندین ایمیل معمولی، یک ایمیل مهم وجود داشته باشد که نوشته:

فرم بورسیه شما باید تا فردا ساعت ۵ عصر ارسال شود.

اگر کاربر این ایمیل را دیر ببیند، ممکن است یک فرصت مهم را از دست بدهد.

این پروژه کمک می‌کند که چنین پیام‌هایی سریع‌تر تشخیص داده شوند. سیستم می‌تواند:

- تشخیص دهد پیام فوری است یا نه.
- مهلت‌ها و زمان‌های مهم را پیدا کند.
- کارهایی را که باید انجام شوند استخراج کند.
- متن طولانی را خلاصه کند.
- پیشنهاد دهد که قدم بعدی چیست.
- اگر موضوع حساس یا نامطمئن بود، از کاربر بخواهد خودش تأیید کند.

README پروژه نیز توضیح می‌دهد که سیستم فوریت، مهلت‌ها و کارها را تشخیص می‌دهد، خلاصه می‌سازد، و اقدام امن بعدی را پیشنهاد می‌کند.

۴. اجزای اصلی پروژه

پروژه از چند بخش اصلی تشکیل شده است:

الف) داشبورد ساده

داشبورد بخشی است که کاربر با آن تعامل می‌کند. کاربر می‌تواند نوع ورودی را انتخاب کند، مثلاً:

- ایمیل
- متن PDF
- سند متنی

سپس فرستنده، عنوان، و محتوای پیام را وارد می‌کند. در صفحه داشبورد، ورودی‌های لازم مثل 'Source type'، 'Sender or Subject or title'، 'source'، و 'Content' وجود دارند.

بعد از آن کاربر روی دکمه **Analyze with Agent** کلیک می‌کند. داشبورد نتیجه تحلیل را نمایش می‌دهد، از جمله:

- Priority یا اولویت
- Category یا دسته‌بندی
- Confidence یا میزان اطمینان
- Summary یا خلاصه
- Recommended next step یا اقدام پیشنهادی
- Deadlines یا مهلت‌ها
- Tasks یا کارهای لازم
- Safety notes یا نکات ایمنی

این بخش‌های خروجی در فایل HTML داشبورد تعریف شده‌اند.

ب) n8n workflow

n8n مغز اجرایی یا pipeline اصلی پروژه است. در این پروژه n8n یک workflow دارد که از طریق webhook ورودی می‌گیرد.

Webhook یعنی یک آدرس اینترنتی که وقتی داده‌ای به آن ارسال شود، workflow اجرا می‌شود.

در فایل workflow پروژه، node اول با نام Incoming Email or Document تعریف شده و متد آن POST است. یعنی برای ارسال داده به آن باید درخواست POST فرستاده شود، نه اینکه فقط آدرس را در مرورگر باز کنیم.

این workflow سه بخش کلی دارد:

1. دریافت ایمیل یا سند از طریق webhook
2. تحلیل متن و تصمیم‌گیری
3. برگرداندن نتیجه به داشبورد به صورت JSON

اتصال بین node‌ها هم در فایل workflow مشخص شده است. ابتدا داده از webhook وارد node تحلیل می‌شود و بعد نتیجه به node پاسخ‌دهنده فرستاده می‌شود.

ج) منطق تحلیل یا triage logic

منطق تحلیل یعنی سیستم چگونه تصمیم می‌گیرد که یک پیام مهم است یا نه.

در این پروژه سیستم به دنبال چند نوع نشانه در متن می‌گردد:

1. کلمات مربوط به فوریت
مثل 'urgent'، 'asap'، 'immediately'، 'overdue'، 'final notice'
2. عبارات‌های مربوط به مهلت
مثل 'deadline'، 'by Friday'، 'due tomorrow'، یا زمان‌هایی مثل 'PM 5'

3. عبارت‌های مربوط به کار یا وظیفه
مثل `please submit`، `review`، `sign`، `reply`، `schedule`
4. موضوعات حساس
مثل مسائل مالی، حقوقی، پزشکی، HR، یا `confidential`
5. پیام‌های کم‌اهمیت
مثل `FYI only` یا `newsletter`

README پروژه همین موارد را به عنوان منطق اصلی دسته‌بندی توضیح داده است.

در کد داشبورد نیز همین منطق دیده می‌شود. مثلاً سیستم کلمات فوری، کلمات حساس، `deadline` ها و `task` ها را از متن استخراج می‌کند.

۵. جریان کاری پروژه از ابتدا تا انتها

فرض کنید یک ایمیل وارد سیستم می‌شود:

Subject: Urgent: Scholarship renewal due tomorrow at 5 PM

.Content: Please submit the signed form and updated transcript

این دقیقاً شبیه نمونه‌ای است که در پروژه قرار داده شده است. نمونه ایمیل فوری داخل فایل `samples/urgent_email.txt` شامل یک یادآوری نهایی برای ارسال فرم بورسیه تا فردا ساعت ۵ عصر است.

حالا پروژه این کارها را انجام می‌دهد:

مرحله ۱: کاربر متن را وارد داشبورد می‌کند

کاربر متن ایمیل یا سند را در داشبورد وارد می‌کند. داشبورد یک فرم ساده دارد که اطلاعات را از کاربر می‌گیرد.

مرحله ۲: داشبورد داده را به n8n می‌فرستد

وقتی کاربر روی دکمه تحلیل کلیک می‌کند، داشبورد یک درخواست `POST` به n8n می‌فرستد. در کد جاوااسکریپت، درخواست با `Content-Type: application/json` و `'method': 'POST'` ارسال می‌شود.

مرحله ۳: n8n workflow اجرا می‌شود

n8n متن را دریافت می‌کند و `node` تحلیل را اجرا می‌کند. این `node` متن را بررسی می‌کند و سعی می‌کند موارد مهم را پیدا کند.

مرحله ۴: سیستم اولویت را مشخص می‌کند

اگر متن شامل کلمات فوری یا `deadline` باشد، اولویت پیام بالا می‌رود. اگر پیام فقط `newsletter` یا `FYI` باشد، اولویت پایین می‌آید. این رفتار در منطق محلی داشبورد هم مشخص است.

مرحله ۵: سیستم خروجی ساختاریافته تولید می‌کند

خروجی پروژه یک JSON ساختاریافته است. README نشان می‌دهد خروجی شامل `priority`، `category`، `summary`، `confidence`، `requiresHumanApproval`، `suggestedAction`، `tasks`، `deadlines` و `safetyNotes` است.

مثلاً ممکن است خروجی چنین مفهومی داشته باشد:

Priority: high

Category: deadline

Summary: این ایمیل درباره تمدید بورسیه است و باید تا فردا ساعت ۵ عصر انجام شود.

Deadlines: tomorrow at 5 PM

Tasks: submit the signed form and updated transcript

Suggested Action: یک یادآور بساز، اما قبل از انجام اقدام نهایی از کاربر تأیید بگیر.

Requires Human Approval: true

۶. چرا پروژه ایمن طراحی شده است؟

یکی از نکات مهم پروژه این است که سیستم خودش کارهای خطرناک انجام نمی‌دهد.

مثلاً سیستم:

- خودش ایمیل ارسال نمی‌کند.
- خودش فایل حذف نمی‌کند.
- خودش فرم امضا نمی‌کند.
- خودش رویداد تقویم ایجاد نمی‌کند، مگر اینکه کاربر تأیید کند.
- اگر پیام حساس باشد، از انسان می‌خواهد آن را بررسی کند.

فایل حذف نمی‌کند، فرم امضا نمی‌کند، و رویداد تقویم ایجاد نمی‌کند مگر با تأیید. همچنین موارد حساس یا کم‌اطمینان مقدار `requiresHumanApproval` را `true` می‌کنند.

این بخش برای ارائه به استاد خیلی مهم است، چون نشان می‌دهد پروژه فقط هوشمند نیست، بلکه مسئولانه و ایمن هم طراحی شده است.

۷. اگر n8n کار نکند چه می‌شود؟

یک ویژگی خوب این پروژه این است که اگر n8n در دسترس نباشد، داشبورد همچنان می‌تواند با منطق محلی خودش کار کند.

یعنی اگر workflow فعال نباشد یا webhook درست اجرا نشود، سیستم کاملاً از کار نمی‌افتد. در عوض، داشبورد از تابع `local fallback` استفاده می‌کند و نتیجه تقریبی تولید می‌کند. README هم توضیح می‌دهد که اگر n8n اجرا نشود، داشبورد از همان منطق `heuristic` محلی استفاده می‌کند تا پروژه برای کلاس قابل نمایش باقی بماند.

در کد داشبورد، ابتدا سیستم تلاش می‌کند نتیجه را از n8n بگیرد. اگر خطا رخ دهد، تابع `trriageLocally(payload)` اجرا می‌شود.

۸. نقش Docker در پروژه چیست؟

Docker برای اجرای n8n استفاده می‌شود. به جای اینکه کاربر n8n را دستی نصب کند، پروژه یک فایل `docker-compose.yml` دارد.

این فایل مشخص می‌کند که سرویس n8n باید با image رسمی `n8nio/n8n` اجرا شود و روی پورت 5678 در دسترس باشد. به زبان ساده، Docker کمک می‌کند محیط اجرای n8n برای همه تقریباً یکسان باشد. بنابراین دانشجو لازم نیست نگران نصب پیچیده n8n باشد.

۹. ورودی و خروجی پروژه به زبان ساده

ورودی

ورودی پروژه یک پیام یا سند است. مثلاً:

```
}

, "sourceType": "email"

, "sender": "registrar@example.edu"

, "subject": "Scholarship form due tomorrow"

".content": "Please submit the scholarship renewal form by Friday at 5 PM"

{
```

README همین قالب ورودی را نشان داده است.

خروجی

خروجی پروژه یک گزارش کوتاه و قابل فهم است. مثلاً:

```
}

, "priority": "high"

, "category": "deadline"

, ".summary": "Scholarship renewal form must be submitted by the stated deadline"

, ["deadlines": ["Friday at 5 PM"]

, ["tasks": ["Submit the scholarship renewal form"]
```

```
suggestedAction": "Create a reminder and ask the user to confirm submission"
, ".details"

, "requiresHumanApproval": true"

, "confidence": 0.78"

safetyNotes": ["Deadline detected; confirmation is required before creating or"
["sending anything

{
```

این قالب خروجی در README آمده است.

۱۰. مثال قابل ارائه به استاد

می‌توانید این مثال را در ارائه خود بگویید:

فرض کنید یک دانشجو روزانه ایمیل‌های زیادی از دانشگاه، استادها، بخش مالی، کتابخانه، و انجمن‌های دانشجویی دریافت می‌کند. ممکن است بین این ایمیل‌ها یک پیام مهم درباره بورسیه یا ثبت‌نام وجود داشته باشد که **deadline** دارد. پروژه من این پیام‌ها را دریافت می‌کند، متن آن‌ها را تحلیل می‌کند، **deadline** و **task** را پیدا می‌کند، سطح اهمیت را مشخص می‌کند، و به کاربر می‌گوید قدم بعدی چیست. اگر پیام حساس یا مهم باشد، سیستم خودش کاری انجام نمی‌دهد و از کاربر تأیید می‌خواهد. نمونه ایمیل فوری داخل پروژه دقیقاً درباره تمدید بورسیه و ارسال فرم تا فردا ساعت ۵ عصر است.

۱۱. نمونه پیام کم‌اهمیت

همه پیام‌ها مهم نیستند. مثلاً یک **newsletter** دانشگاهی ممکن است فقط اطلاع‌رسانی باشد و نیاز به اقدام خاصی نداشته باشد. پروژه یک نمونه سند **routine** هم دارد که در آن نوشته شده این فقط یک اطلاع‌رسانی است و نیازی به اقدام نیست. در چنین حالتی سیستم می‌تواند **priority** را پایین‌تر در نظر بگیرد و پیشنهاد کند که پیام فقط برای اطلاع نگهداری یا آرشیو شود.

۱۲. چرا از n8n استفاده شده است؟

n8n یک ابزار **automation** و **workflow** است. یعنی می‌توان چند مرحله را به شکل تصویری یا نیمه‌کدنویسی‌شده به هم وصل کرد.

در این پروژه n8n برای ساخت **pipeline** استفاده شده است:

1. دریافت پیام از dashboard
2. اجرای منطق تحلیل
3. تولید نتیجه
4. ارسال نتیجه به dashboard

README پروژه هم می‌گوید که n8n نقش **agentic pipeline** را دارد و **workflow** از طریق **webhook** متن ایمیل یا سند را دریافت می‌کند و نتیجه ساختار یافته برمی‌گرداند.

مزیت n8n برای دانشجو این است که بعداً می‌تواند nodeهای جدید اضافه کند، مثلاً:

- اتصال به Gmail
- اتصال به Google Drive
- اتصال به Google Calendar
- اتصال به Notion یا Airtable
- اضافه کردن LLM برای تحلیل هوشمندتر

۱۳. این پروژه واقعاً «هوش مصنوعی» دارد یا فقط قانون ساده است؟

نسخه فعلی پروژه بیشتر یک نسخه پایه و قابل توضیح است. یعنی از قوانین ساده و قابل فهم استفاده می‌کند تا دانشجو بتواند دقیقاً توضیح دهد سیستم چگونه تصمیم می‌گیرد.

مثلاً اگر متن شامل **urgent** یا **deadline** باشد، سیستم می‌فهمد احتمالاً پیام مهم است. اگر متن شامل **newsletter** یا **FYI** **only** باشد، احتمالاً پیام کم‌اهمیت است.

این روش برای شروع خوب است چون استاد و دانشجو می‌توانند منطق سیستم را بفهمند. اما پروژه قابل توسعه است و در مراحل بعدی می‌توان یک مدل زبانی بزرگ یا LLM به آن اضافه کرد. **README** هم این را به عنوان **milestone** پنجم پیشنهاد می‌کند.

پس اگر استاد پرسید:

آیا این پروژه از **AI** واقعی استفاده می‌کند؟

می‌توانید جواب دهید:

نسخه اولیه از **rule-based intelligence** یا **heuristic reasoning** استفاده می‌کند تا تصمیم‌ها قابل توضیح باشند. اما معماری پروژه طوری طراحی شده که می‌توان در مرحله بعد یک **LLM node** به n8n اضافه کرد تا تحلیل معنایی پیشرفته‌تر انجام شود.

۱۴. قابلیت‌های اصلی پروژه

این پروژه می‌تواند:

قابلیت	توضیح ساده
خلاصه‌سازی	متن طولانی را به یک توضیح کوتاه تبدیل می‌کند.
تشخیص فوریت	بررسی می‌کند آیا پیام urgent یا مهم است.
استخراج deadline	زمان‌ها و مهلت‌ها را پیدا می‌کند.
استخراج task	کارهایی مثل submit ، review ، sign یا reply را تشخیص می‌دهد.
تشخیص پیام حساس	اگر متن درباره موضوعات مالی، حقوقی، پزشکی یا محرمانه باشد، آن را حساس می‌داند.
پیشنهاد اقدام	می‌گوید بهتر است کاربر چه کاری انجام دهد.

درخواست تأیید انسانی برای موارد حساس یا مهم، قبل از اقدام نهایی از کاربر تأیید می‌خواهد.

این قابلیت‌ها در README و منطق کد پروژه توضیح داده شده‌اند.

۱۵. سناریوی ارائه کوتاه برای استاد

اگر بخواهید خیلی خلاصه و رسمی پروژه را معرفی کنید، می‌توانید بگویید:

پروژه من یک عامل هوشمند برای **triage** یا دسته‌بندی اولیه ایمیل‌ها و اسناد است. این سیستم با استفاده از **n8n** یک **workflow** می‌سازد که متن ایمیل یا سند را از **dashboard** دریافت می‌کند، آن را تحلیل می‌کند، **deadline**ها و **task**ها را استخراج می‌کند، سطح اولویت را مشخص می‌کند و یک اقدام پیشنهادی ارائه می‌دهد. داشبورد پروژه ساده است و کاربر می‌تواند متن را وارد کند و نتیجه را ببیند. نکته مهم این است که سیستم برای کارهای حساس یا پرریسک خودش اقدام نمی‌کند و از کاربر تأیید می‌خواهد. نسخه اولیه از قوانین قابل توضیح استفاده می‌کند، اما در آینده می‌توان آن را با **LLM** پیشرفته‌تر کرد.

۱۶. معماری پروژه به زبان خیلی ساده

می‌توانید معماری را این‌طور توضیح دهید:



summary, priority, deadlines, tasks, suggested action, safety notes

داشبورد ورودی‌ها و خروجی‌ها را نمایش می‌دهد.

n8n workflow نیز node، webhook، تحلیل، و node پاسخ‌دهنده دارد.

۱۷. نقاط قوت پروژه برای ارائه

برای اینکه پروژه قوی‌تر به نظر برسد، روی این نکات تأکید کنید:

1. کاربرد واقعی دارد
چون همه با ایمیل‌ها، deadlineها، فرم‌ها و اسناد سروکار دارند.
2. برای دانشجویها مفید است
دانشجوها معمولاً ایمیل‌های دانشگاه، استاد، مالی، ثبت‌نام و بورسیه دریافت می‌کنند.
3. امن طراحی شده است
سیستم فقط پیشنهاد می‌دهد و برای اقدامات حساس از کاربر تأیید می‌گیرد.
4. قابل توضیح است
تصمیم‌ها بر اساس نشانه‌های مشخص مثل urgency terms و deadline phrases گرفته می‌شوند.
5. قابل توسعه است
می‌توان در آینده Gmail، Google Calendar، Notion، یا LLM به آن اضافه کرد.
6. از n8n استفاده می‌کند
یعنی پروژه فقط یک صفحه وب ساده نیست، بلکه یک automation pipeline واقعی دارد.

۱۸. محدودیت‌های پروژه

برای اینکه ارائه شما واقع‌بینانه باشد، بهتر است محدودیت‌ها را هم بگویید:

1. نسخه فعلی هنوز به Gmail واقعی وصل نیست.
 2. استخراج deadlineها ساده است و ممکن است همه تاریخ‌ها را کامل تشخیص ندهد.
 3. نسخه اولیه بیشتر rule-based است، نه یک مدل زبانی کامل.
 4. برای استفاده واقعی باید authentication، ذخیره‌سازی امن، و مدیریت داده‌های حساس اضافه شود.
 5. اگر قرار باشد با PDF واقعی کار کند، باید مرحله استخراج متن از PDF هم اضافه شود.
- این محدودیت‌ها بد نیستند؛ اتفاقاً نشان می‌دهند شما پروژه را درست فهمیده‌اید و می‌دانید چه چیزهایی برای توسعه آینده لازم است.

۱۹. مسیر توسعه آینده

در آینده می‌توان این قابلیت‌ها را اضافه کرد:

1. اتصال به Gmail برای دریافت خودکار ایمیل‌ها
2. اتصال به Google Drive برای خواندن PDFها
3. استخراج متن از PDF
4. اضافه کردن OpenAI یا یک LLM دیگر برای تحلیل بهتر متن
5. ذخیره نتایج در دیتابیس
6. ساخت reminder در Google Calendar با تأیید کاربر
7. اضافه کردن login برای کاربران
8. داشبورد پیشرفته‌تر با history پیام‌ها
9. فیلتر بر اساس priority
10. گزارش روزانه از پیام‌های مهم

README پروژه نیز milestone‌هایی مثل بهبود استخراج deadline/task، اضافه کردن human approval loop، و افزودن LLM را پیشنهاد کرده است.

۲۰. جمع‌بندی نهایی برای ارائه

این پروژه یک سیستم ساده اما کاربردی برای مدیریت هوشمند ایمیل‌ها و اسناد است. کاربر متن یک ایمیل یا سند را وارد می‌کند، سیستم آن را از طریق n8n تحلیل می‌کند، موارد مهم مثل deadline و task را پیدا می‌کند، یک خلاصه کوتاه تولید می‌کند، اولویت را مشخص می‌کند، و یک اقدام پیشنهادی می‌دهد.

مهم‌ترین نکته پروژه این است که سیستم کمک‌کننده تصمیم‌گیری است، نه یک سیستم خودکار خطرناک. یعنی اگر موضوع حساس، فوری، یا کم‌اطمینان باشد، سیستم از انسان می‌خواهد تأیید کند. این باعث می‌شود پروژه از نظر کاربردی و اخلاقی قابل دفاع باشد.

متن خیلی کوتاه برای حفظ کردن

اگر بخواهید یک متن کوتاه برای توضیح سریع حفظ کنید:

این پروژه یک دستیار هوشمند برای دسته‌بندی ایمیل‌ها و اسناد است. کاربر متن ایمیل یا سند را در داشبورد وارد می‌کند. داشبورد داده را به n8n می‌فرستد. n8n متن را تحلیل می‌کند، فوریت، task، deadline، و موارد حساس را تشخیص می‌دهد و نتیجه‌ای شامل خلاصه، اولویت، مهلت‌ها و اقدام پیشنهادی برمی‌گرداند. هدف پروژه کاهش زمان خواندن و مدیریت پیام‌هاست. سیستم ایمن طراحی شده و کارهای حساس را بدون تأیید انسان انجام نمی‌دهد.