Programming Assessment Project
for Artificial Intelligence

Deadline:

MeetingInsights

Overview

MeetingInsights is an AI-powered service that **accepts meeting transcripts** and processes them via an LLM-driven analysis engine (You can produce the meeting text yourself). The tool is designed to generate concise **summaries and extract actionable items from the transcripts**. The candidate is expected to build a **streamlined API service using Django Ninja**, **design a thoughtful PostgreSQL schema**, **manage asynchronous transcript processing**, and **containerize the solution using Docker and Docker Compose**.

Core Requirements

1. **Meeting Management API**

- **CRUD Operations:** Use Django Ninja to expose endpoints that allow for creating, updating, retrieving, and deleting meetings.
- **Metadata Flexibility:** Capture standard meeting details (e.g., title, date, participants) with the option to include additional metadata via JSON fields, balancing schema complexity with flexibility.

2. **Transcript Submission API**

- **Transcript Ingestion**: Implement endpoints that accept and associate meeting transcripts with their corresponding meetings.
- **Data Validation:** Ensure inputs (text or file uploads) are properly validated and parsed.

3. **AI Processing Workflow**

- **Asynchronous Task Offloading:** Upon transcript submission, trigger an asynchronous task that simulates or integrates with an LLM to:
  - Generate a meeting summary.
  - Extract and return actionable items.
- **Status Endpoint:** Provide an API endpoint to query the processing status (e.g., pending, processing, completed, error).
- **Error Resiliency:** Strategically manage timeouts, retries, and error logging within the asynchronous workflow.

4. **Data Retrieval API**

- **Results Access:** Allow clients to fetch analysis results (summaries and actionable items) via clearly designed endpoints.
- **Filtering and Pagination:** Include mechanisms to filter and paginate data where it makes sense.

5. **Technical Framework & Design**

- **Backend Framework:** Develop all API endpoints using **Django Ninja** to ensure a fast, type-hinted, and modern API design.
- **Database Design:** Architect a PostgreSQL-based data model with:
  - Normalized tables for meetings, transcripts, and analysis results.
  - Strategic use of JSON fields where dynamic or less-structured data is needed.
  - Appropriate indexing for optimal query performance.
- **Documentation:** provide a comprehensive README that explains:
  - Setup instructions.
  - Architectural decisions and trade-offs.

- How to run and test the system locally.

Bonus Enhancements (Optional for Extra Points)

- **Containerization:** Create a `docker-compose.yml` file to orchestrate the environment, including:
    - - A container for the Django Ninja app.
    - - A container for PostgreSQL.

  (Optionally) A separate container for background processing if using Celery or another task queue.
- **Authentication & Security:** Implement simple token-based authentication (e.g., JWT) to secure your endpoints.
- **Enhanced Documentation Interface:** While Django Ninja automatically provides interactive API documentation, consider adding notes or custom tweaks that improve developer experience.
- **Improved Logging & Monitoring:** Integrate basic logging for asynchronous tasks and consider lightweight monitoring strategies to ensure smooth operations under load.

Tools You Can Use

- **For User Interface:** You can use **STREAMLIT** to create a user interface. This tool allows you to easily build an interactive interface with minimal code.
- **For LLM Model: GEMINI** is a free API that you can use for text processing and building natural language models. If you are looking for other free models for text processing, I recommend you to see this link https://github.com/cheahjs/free-llm-api-resources?tab=readme-ov-file#groq , which is a list of free models that you can use for speech-to-text conversion or LLM models.

Sample Meeting Transcript Between Employer and Employee

Here’s a sample hypothetical conversation from a meeting between an employer and an employee where tasks have been assigned to the employee and actions to be taken are mentioned:

کارفرما: سلام، وقت بخیر آقای جمالی. ما امروز می‌خواهیم در مورد پروژه جدید صحبت کنیم. در ابتدا، لطفاً گزارشی از پیشرفت پروژه‌های قبلی بدهید.

کارمند: سلام ممنونم وقت شما هم بخیر جناب مهندس کریمی. بله، در حال حاضر پروژه‌ها در حال پیشرفت هستند. اما نیاز به برخی تغییرات جزئی در طراحی داریم.

کارفرما: خوب است. پس لطفاً تغییرات پیشنهادی را تا پایان این هفته ارسال کن. همچنین، لطفاً وضعیت تامین منابع را بررسی کن و گزارش آن را تا روز پنجشنبه ارسال کن.

کارمند: حتماً، من گزارش تغییرات را آماده می‌کنم و منابع را نیز بررسی می‌کنم.

کارفرما: خیلی خوب. نکته بعدی این است که در پروژه جدید، تیم را برای انجام طراحی‌های اولیه آماده کن. می‌خواهم که تا دو هفته آینده طراحی‌های اولیه ارائه شود. آیا برای این کار آماده هستید؟

کارمند: بله، آماده‌ام و تیم را برای شروع آماده می‌کنم.

کارفرما: عالی است. پس لطفاً با تیم خود تماس بگیر و تا دو روز آینده یک جلسه برنامه‌ریزی برای طراحی داشته باشید. به علاوه، گزارش پیشرفت پروژه‌های قبلی را به من ارسال کن.

کارمند: بله، حتماً.

Data Tables Example: This is just an example and you can design these tables however you want.

Meeting table:

id	Meeting Transcript	Meeting summary	Participants	Meeting Date
12				

Task table:

meeting_id	Key Points	Tasks	Responsible	Deadline for Tasks
12				

Evaluation Criteria

1. API Design & Django Ninja Implementation
2. Database Architecture
3. Asynchronous Processing
4. Code Quality & Maintainability
5. Containerization & Deployment
6. Bonus Enhancements: Extra points will be awarded for integrating additional features (such as **Containerization ,** authentication) that add real-world value without over-complicating the core solution.

“The only way to do great work is to love what you do.” Steve Jobs