Take Home Assignment

Senior Software Engineer (AI Team)

Overview

We're excited to have you in the selection process for the Senior Software Engineer (AI) role. This assignment is designed to showcase your expertise in API development, specifically in creating RESTful services in Python. You'll be building an application that integrates file handling, simulates Optical Character Recognition (OCR), and employs Retrieval-Augmented Generation (RAG) for attribute extraction. This exercise is not just about coding but also about how you approach problem-solving, adhere to coding standards, and document your work.

Don't worry if you're unfamiliar with some aspects of the assignment; we encourage you to explore ChatGPT, open-source projects, and technical blogs. A few resources are listed below:
1. https://www.pinecone.io/learn/retrieval-augmented-generation/
2. RAG implementation
3. Embeddings

Assignment Outline

Develop a Python-based backend API using FastAPI. The API will feature endpoints for file uploads, a mock OCR function, and an attribute extraction mechanism powered by RAG.

Endpoints to Implement:

File Upload Endpoint

Path: `/upload`
Method: **POST**
Functionality:

- Accepts one or more file uploads (limited to `pdf`, `tiff`, `png, jpeg` formats).
- Saves the processed file to a cloud storage solution, returning one or more unique file identifiers or signed URLs for the upload. You can use tools like `Minio` to mimic blob storage.
- Food for thought: Is there a better way of handling file uploads?

Mock OCR Endpoint

Path: `/ocr`
Method: **POST**

Functionality:

- Simulates running an OCR service on a file for a given a signed url.
- Process OCR results with OpenAI's embedding models, then upload the embeddings to a vector database (e.g, Pinecone) for future searches.
- Food for thought: How will you do error handling?

Attribute Extraction Endpoint

Path: `/extract`
Method: **POST**
Functionality:

- Takes a query text and file_id as input, performs a vector search and returns matching attributes based on the embeddings.
- The vector search will help in identifying the relevant part(s) of the file and you may need to call openAI chat completion to generate the answer from the search result.
- Food for thought: Will the language of text matter in attribute extraction?

Technical Requirements

1. Utilize FastAPI for the backend service.
2. We recommend using `OpenAI` for embeddings and chat completions while `Pinecone` for the vector database, but feel free to use any other service.
3. For the OCR simulation, we will provide sample files and corresponding OCR jsons.
4. Process the OCR data with embedding models, uploading the output embeddings to a vector database.
5. The attribute extraction process should search within the vector database using the generated embeddings of the query to find and return relevant chunks of text.

Instructions for Completion

- Your code should be clean, well-documented, and include comprehensive logging and error handling.
- Provide a README file detailing setup and usage instructions, alongside endpoint usage examples.
- We recommend using Docker to streamline environment setup and dependency management.
- Incorporate tests to validate your implementation.
- Implement a GitHub CI workflow for running tests and lint checks.
- Set up a GitHub CD pipeline for automating Docker image creation.

Evaluation Criteria

Submissions will be assessed based on:

1. Functionality: All endpoints must operate as described.
2. Code Quality: Your code should follow PEP8 standards, include meaningful docstrings, and maintain a high level of readability.
3. Documentation: The README should clearly instruct on how to set up and interact with your API.
4. Error Handling: Your application should manage and report errors effectively.

Submission Guidelines

Please submit your completed assignment through a GitHub repository link. Make sure the repository is accessible publicly or shared directly with our team. Your submission should include all source code, the README documentation, and any additional materials necessary for evaluation.

We're looking forward to seeing your innovative approaches and solutions to this challenge. Best of luck!

Deadline

Your completed assignment should be submitted within two weeks from the date of receipt. If you would like to extend the deadline, that is completely fine however please let us know so we can plan accordingly.

Accounts and Sample Documents

Please create your OpenAI account and use your personal key for the assignment. If you are using a managed vector database like Pinecone, use your personal api keys. We can support usage costs of up to 5,000 yen

We will be providing 2 documents regarding Japanese building regulations and the corresponding OCR results in json format.