Home Assignment for Insait's Backend internship

Task Overview

Create a simple Flask server that exposes an endpoint to ask a question. The server sends the question to an OpenAI API, receives the answer, and saves both the question and the answer in a PostgreSQL database. The server and the database should be dockerized and run with Docker Compose. Implement one test using pytest.

Requirements

1. **Flask Server**: Set up a Flask server with an endpoint to handle questions.
2. **OpenAI API Integration**: Integrate the OpenAI API to get answers for the questions.
3. **Database**: Use PostgreSQL to store questions and answers.
4. **Alembic**: Use Alembic for database migrations.
5. **Docker**: Dockerize the Flask server and the PostgreSQL database.
6. **Docker Compose**: Use Docker Compose to manage and run the containers.
7. **Testing**: Implement at least one test using pytest.
8. **GitHub Repository**: Share the code via a GitHub repository with a clean structure and meaningful commit history.
9. **Best Practices for DAL**: Implement best practices for Data Access Layer.

Instructions

1. **Flask Server**:
    - Create a Flask application.
    - Create an endpoint `/ask` that accepts a POST request with a JSON payload containing the question.
2. **OpenAI API**:
    - Use the OpenAI API to get an answer for the question.
    - You can use the OpenAI Python client library for this. (You can create an account and use the free tier)
3. **Database**:
    - Set up a PostgreSQL database.
    - Create the schema.
    - Use SQLAlchemy for ORM. (or any other ORM)
    - Use Alembic for database migrations. (or any other migration tool)
4. **Dockerization**:
    - Dockerize the Flask application.
    - Dockerize the PostgreSQL database.

    - Use Docker Compose to run both containers.
5. **Testing**:
    - Implement one test using pytest to ensure the endpoint works as expected.
6. **GitHub Repository**:
    - Maintain a clean directory structure.
    - Ensure meaningful commit messages.
    - Use branches if necessary and document your work in the README file.

Deliverables

- A link to the GitHub repository with the code
    - The repo should include:
        - Source code for the Flask server.
        - Dockerfile for the Flask application.
        - Dockerfile for the PostgreSQL database (or use an official PostgreSQL image).
        - docker-compose.yml to manage the services.
        - Alembic migration scripts.
        - A pytest test file.