Chat-Bot Project

Introduction

This project aims to create an interactive chatbot that utilizes natural language processing and database interactions to provide meaningful responses to user queries.

This documentation will cover the project's objectives, architecture, technologies used, instructions for running the application, and include screenshots to give you a visual understanding of the project.

Technologies Used:

Objectives

The primary objectives of the Chatbot Project are as follows:

- **User Interaction:** Develop a user-friendly interface where users can have interactive conversations with the chatbot.
- **Natural Language Processing:** Utilize the OpenAI API to process user messages and generate appropriate SQL queries for database interaction.

- **Database Interaction:** Connect to a MySQL database to retrieve table descriptions and execute SQL queries based on user requests.
- **Contextual Responses:** Generate responses that incorporate both the user's message and relevant query results to provide meaningful and contextual answers.

Architecture Overview

The Chatbot Project consists of a frontend component built with HTML and JavaScript, and a backend component implemented using the FastAPI framework in Python.

The project flow involves user interactions, OpenAI integration for query generation, MySQL database interactions, and contextual response generation.

Frontend (HTML and JavaScript)

The frontend component is responsible for the user interface and interaction. It includes the following elements:

- **Chat Interface:** An HTML layout that allows users to input messages and receive responses.
- **JavaScript Logic:** Interactive features implemented using JavaScript, including sending messages, displaying responses, and toggling the chat window.

Backend (FastAPI and OpenAI Integration)

The backend component handles the processing of user messages, interaction with the OpenAI API for query generation, execution of SQL queries on the MySQL database, and response generation. Key elements include:

- **FastAPI Endpoints:** The backend exposes two endpoints: / for rendering the chat interface and /chatbot for handling user messages and generating responses.
- **OpenAI Integration:** Utilizes the OpenAI API to convert user messages into SQL queries and generate responses based on query results.
- **Database Interaction**: Connects to a MySQL database to retrieve table descriptions and execute SQL queries.
- **Response Generation:** Creates responses that integrate user messages, query results, and relevant information to provide contextually accurate answers.

How to Run the Project

To run the Chatbot Project, follow these steps:

1. Install project dependencies using pip:

   ```
   pip install fastapi[all] openai mysql-connector-python
   ```

2. Obtain an OpenAI API key and update it in the openai.api_key field in the backend code (main.py).
3. Configure MySQL database settings (host, user, password, and database name) in the db_connection setup within the backend code.
4. Launch the FastAPI server using uvicorn:

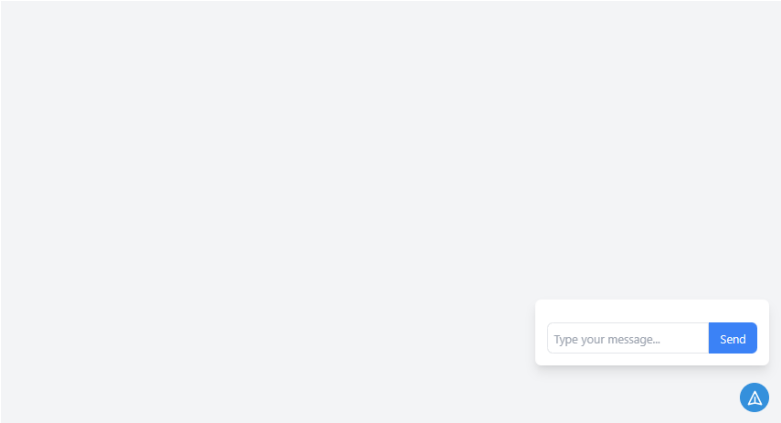
   ```
   uvicorn main:app --host 0.0.0.0 --port 8000
   ```

5. Access the chat interface by opening a web browser and navigating to http://localhost:8000.

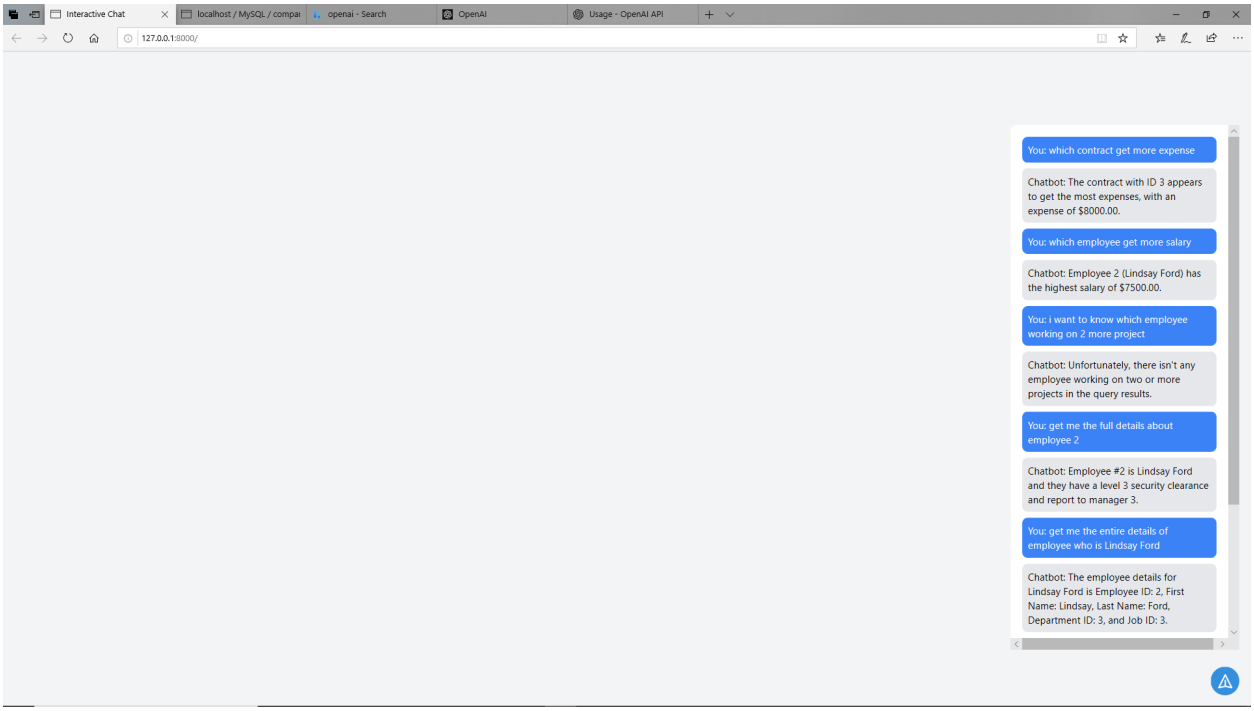
Screenshots

Here are a few screenshots of the Chatbot Project to provide a visual understanding:

Chat Interface:



Response Interface:



Conclusion

The Chatbot Project showcases the integration of frontend and backend technologies to create an interactive chatbot capable of natural language processing, database interactions, and contextual response generation. By following the instructions provided in this documentation, you can easily run the application and experience the capabilities of this innovative chatbot solution.