

Project Description: Text-to-SQL with LLM Integration and PostgreSQL

This project integrates a local Large Language model (LLM) to translate natural language queries into SQL statements, allowing users to interact with a PostgreSQL database using simple English queries. The solution is built using Flask, a lightweight Python web framework, to create a user interface where queries can be inputted and results displayed in real time. The key components of the project are as follows:

1. Natural Language to SQL Query Conversion:

- The project uses the Ollama LLM (version 3.1) to convert natural language inputs into SQL queries. Users can ask questions like "How many students are in section A?" or "List all orders and their customer names," and the LLM generates the appropriate SQL queries.
- The conversion is guided by an expertly designed prompt, including several SQL query examples to ensure the LLM understands the desired output format.

2. PostgreSQL Database Interaction:

- The project is integrated with a PostgreSQL database that holds the data to be queried. The database connection is managed with the psycopg2 library, and schema settings ensure correct data retrieval.
- SQL queries generated by the LLM are executed on the PostgreSQL database using a cursor with dictionary-like row access, enabling the system to return results in a flexible format.

3. Flask-Based Web Application:

- The application provides a simple interface where users can input natural language queries. Upon submission, the query is processed through the LLM, translated into SQL, executed on the database, and the results are displayed in an HTML table format.
- Error handling is included to manage invalid queries or database connection issues, ensuring a robust user experience.

4. Markdown-Based Output Formatting:

- The project includes functionality to format responses by converting Markdown-style bold text into HTML tags, providing a cleaner output for certain query results or descriptions.

5. Logging and Debugging:

- Comprehensive logging is enabled throughout the system to track database connections, query execution, and LLM interactions. This makes it easier to debug issues related to query generation or execution failures.

Key Technologies:

- **Flask:** Used to build the web interface for user interaction.
- **Ollama LLM:** Responsible for converting natural language to SQL.

- **PostgreSQL:** The relational database storing the data and executing the SQL queries.
- **psycopg2:** A PostgreSQL adapter for Python, facilitating database connectivity and query execution.
- **HTML and CSS:** Used for rendering query results in a table format on the front end.

Use Case:

This system is ideal for non-technical users who need to retrieve data from databases without writing SQL queries. By using natural language, users can ask questions and receive database information seamlessly, reducing the learning curve associated with database querying. It can be applied in scenarios like business intelligence, customer service analytics, and educational management systems where data retrieval is essential but technical knowledge of SQL is limited.

Future Enhancements:

- **Multi-database support:** Extend support to other database systems like MySQL or Oracle.
- **Improved LLM training:** Incorporate domain-specific training for better accuracy in SQL generation.
- **User Authentication:** Add role-based access control to ensure secure usage in multi-user environments.
- **Query Optimization:** Introduce query optimization techniques for more complex database structures and larger datasets.

This project demonstrates a powerful combination of AI, natural language processing, and database interaction, offering an accessible way to query databases using everyday language.