Project Report

LLM-SQL Query Generator

GitHub Repository: https://github.com/MyGitHubHere

1 Objective

The objective of this project was to build a robust system that translates natural language queries into SQL commands using Google Gemini API. The application integrates a user-friendly frontend with a reliable backend for database interactions, enabling efficient and schema-aware SQL query generation.

2 Methodology

- **LLM Integration:** Used Google Gemini API to generate SQL queries from natural language questions.
- **Frontend Development:** Designed an interactive interface in **Streamlit** with custom CSS styling for clarity and user experience.
- **Database Connectivity:** Executed generated SQL queries on **PostgreSQL**, ensuring compatibility with relational databases.
- **System Enhancements:** Added accuracy controls, query sanitation, vague question detection, and dynamic prompt engineering for improved reliability.

3 Key Features

1. Natural language to SQL translation with schema-awareness.
2. Adjustable accuracy settings (Precise, Balanced, Creative).
3. Query sanitation and error handling to prevent unsafe SQL operations.
4. User-friendly frontend with Streamlit and CSS customization.

4 Results

- Successfully generated reliable SQL queries from English questions.
- Ensured robustness with query sanitation and vague input detection.
- Delivered a working end-to-end system connecting LLM, Streamlit UI, and PostgreSQL database.

5 Conclusion

This project demonstrates the application of Large Language Models for natural language database querying. By combining Google Gemini with Streamlit and PostgreSQL, the system bridges the gap between non-technical users and structured data retrieval.