AI SQL Query Assistant

Internship Project – May 2025

Submitted by: [Your Name]

Internship at: AXP TECHNOLOGIES PRIVATE LIMITED

# Declaration

I hereby declare that the project titled "AI SQL Query Assistant" is a part of my internship at AXP TECHNOLOGIES PRIVATE LIMITED, under the guidance of Randhir Panda and Sachin Rana. The project involves building a smart assistant to convert natural language into SQL queries using Google Gemini, and execute those queries on a MySQL database. This work is the result of my own effort and the learnings during the internship.

# Acknowledgement

I would like to express my sincere gratitude to my mentors Randhir Panda and Sachin Rana for their continuous guidance and support. I also extend thanks to the AXP Technologies team for providing a collaborative environment and to my peers and family for their constant motivation.

# Abstract

The AI SQL Query Assistant is a web-based AI-driven tool that allows users to query SQL databases using plain English. The assistant leverages LangChain, Google Gemini, and Streamlit to bridge the gap between technical complexity and user accessibility. It translates user queries into optimized SQL commands, runs them on a MySQL database (axp\_demo), and returns human-readable results.

Key Features:

* - Natural Language to SQL conversion via Gemini
* - Schema-aware SQL generation using LangChain
* - Secure MySQL database integration
* - Conversational chat interface using Streamlit
* - Chat history persistence for user context
* - Example queries for improved usability
* - Modular and scalable architecture

# 1. Introduction

This project aims to enable users—especially those from non-technical backgrounds—to query a relational SQL database using natural human language. It acts as a bridge between technical systems and everyday users by combining state-of-the-art language models, prompt engineering, and database access layers.

Motivations for the project:

* - Eliminate dependency on technical teams for simple data access.
* - Accelerate business decision-making with immediate data insights.
* - Improve database accessibility through conversational interfaces.
* - Provide AI-powered explanations to improve transparency.

# 2. System Architecture

The system is composed of several interconnected layers:

* - Frontend: Streamlit-based chatbot UI
* - Backend: Python code interfacing with LangChain and Gemini
* - NLP Layer: Gemini LLM converting text to SQL and interpreting results
* - Database: MySQL database with invoice and supplier data

Workflow: User → Streamlit → LangChain → Gemini → SQL Query → MySQL → Result → Explanation → User

# 3. Implementation Details

The app starts with a login interface for demo authentication. Upon successful login, users can connect to the axp\_demo database using credentials. They may ask questions like 'Show invoices from last month'. The system retrieves schema, creates a prompt, sends it to Gemini, and runs the query. The results are explained and shown in a human-readable form.

Unique Implementation Aspects:

* - Uses LangChain’s SQLDatabase utility to load and explore schema.
* - Prompt engineering includes chat history for contextual awareness.
* - Stores last executed SQL for debugging or reuse.
* - Includes a list of example questions to assist user discovery.
* - Designed with modular functions for maintainability.

# 4. Technologies Used

The following technologies were employed:

|  |  |  |
| --- | --- | --- |
| Category | Technology | Purpose |
| Frontend | Streamlit | UI & chat interaction |
| Backend | Python, LangChain | NLP and query handling |
| LLM | Google Gemini | Natural language to SQL transformation |
| Database | MySQL (axp\_demo) | Stores invoice/supplier data |
| Tools | dotenv, pymysql | Configuration and DB connectors |

# 5. Database Schema

The `axp\_demo` database consists of three main tables:

* - `invoice\_detail`: Contains invoice line-item details.
* - `invoice\_header`: Holds metadata like supplier ID, invoice date, etc.
* - `supplier\_master`: Supplier contact and status information.

This schema supports diverse queries like recent invoices, total invoice amounts per supplier, and active suppliers by city.

# 6. Sample Results

Example queries and results handled by the system include:

* - "List invoices created in the last 30 days."
* - "What is the total invoice amount for each supplier?"
* - "Show all suppliers from the city 'LIVONIA'."

The assistant returns both the SQL used and a human-readable response. Screenshots of the app showing inputs and outputs can be inserted here.

# 7. Conclusion & Future Scope

This AI-powered assistant was successfully developed to help users query SQL databases without writing a single line of code.

Planned Improvements:

* - Add voice input/output for multimodal interaction.
* - Integrate visualizations (charts/graphs) for numerical data.
* - Add support for additional databases like PostgreSQL, MSSQL.
* - Build a user dashboard for saved queries and usage logs.
* - Enable user-based query restrictions for security.

# 8. References

- Google Gemini API Docs

- LangChain SQLDatabase Docs

- Streamlit Documentation

- Internship Mentors: randhir.panda@aristagroup.net, sachin.rana@aristagroup.net