Intelligent SQL Querying with LLMs Using Gemini Pro

SmartBridge Virtual Internship Project

Team Details

Team ID: LTVIP2026TMIDS46423

Team Size: 3

Team Leader:
Aakisetti Karthik

Team Members:
Bhimineni Harsha Vardan Babu
Shak Adil Basha

---

1. INTRODUCTION

1.1 Project Overview

This project, titled “Intelligent SQL Querying with LLMs Using Gemini Pro”, is developed as part of the SmartBridge Virtual Internship Program. The objective of this project is to build an AI-powered system that converts natural language queries into structured SQL queries using Google’s Gemini Pro Large Language Model (LLM).

In traditional database systems, users must have knowledge of SQL syntax to retrieve information from databases. However, many non-technical users struggle with writing SQL queries. This project provides an intelligent interface where users can ask questions in plain English, and the system automatically generates optimized SQL queries and executes them on a database.

The system integrates Gemini Pro with a backend database engine, allowing users to interact with databases in a conversational manner. This demonstrates the real-world application of Generative AI in database management systems.

1.2 Purpose

The main purpose of this project is to apply theoretical knowledge of Artificial Intelligence and Python programming to build a practical real-world application.

This project helps in understanding:

- API integration
- Web application development
- Prompt engineering
- AI model interaction

It also improves technical and problem-solving skills.

---

2. IDEATION PHASE

2.1 Problem Statement

Many organizations rely on databases to store large amounts of structured data. However writing SQL queries requires technical expertise, Non-technical users cannot easily retrieve information, Incorrect queries may cause errors or security risks. Query development takes time.

There is a need for a system that:

- Converts natural language to SQL automatically
- Generates accurate and optimized queries
- Is easy to use
- Prevents malicious SQL injection

This project solves the problem using Gemini Pro LLM.

2.2 Empathy Map Canvas

What the User Thinks

- “I don’t know SQL syntax.”

- "I just want the data quickly."

What the User Feels

- Confused when writing complex queries
- Frustrated with syntax errors

What the User Says

- "Show me total sales for 2024."
- "List all employees in IT department."

What the User Does

- Attempts SQL queries
- Searches online for query examples

User Pain Points

- SQL complexity
- Time-consuming query writing
- Fear of database errors

User Needs

- Simple interface
- Accurate results
- Secure query execution

2.3 Brainstorming

Different approaches were considered:

- Manual SQL query writing
- Predefined query system
- AI-based SQL generation using Gemini Pro

AI-based generation was selected because it provides better scalability and faster results.

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User workflow:

1. User enters natural language query
2. System sends prompt to Gemini Pro
3. Gemini generates SQL query
4. Backend validates SQL query
5. Query executed on database
6. Results displayed to user

3.2 Solution Requirements

Functional Requirements

- Accept natural language input
- Generate SQL using Gemini Pro
- Connect to database
- Execute generated SQL
- Display query results
- Handle invalid queries

Non-Functional Requirements

- Fast response time
- Secure query validation
- Reliable API handling
- Scalable architecture

- User-friendly interface

3.3 Data Flow Diagram

User → Web Interface → Backend Server → Gemini Pro API → Generated SQL → Database → Query Result → Output Display

3.4 Technology Stack

Programming Language:

- Python

Framework:

- Streamlit

Libraries:

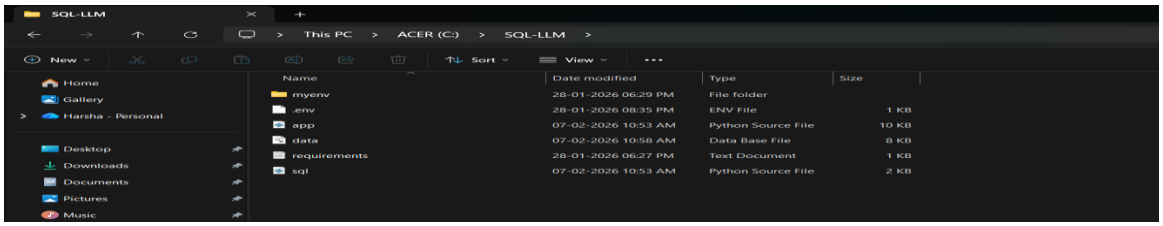
- Google GenAI
- Mysql-connector / sqlite3
- Python-dotenv
- Pandas

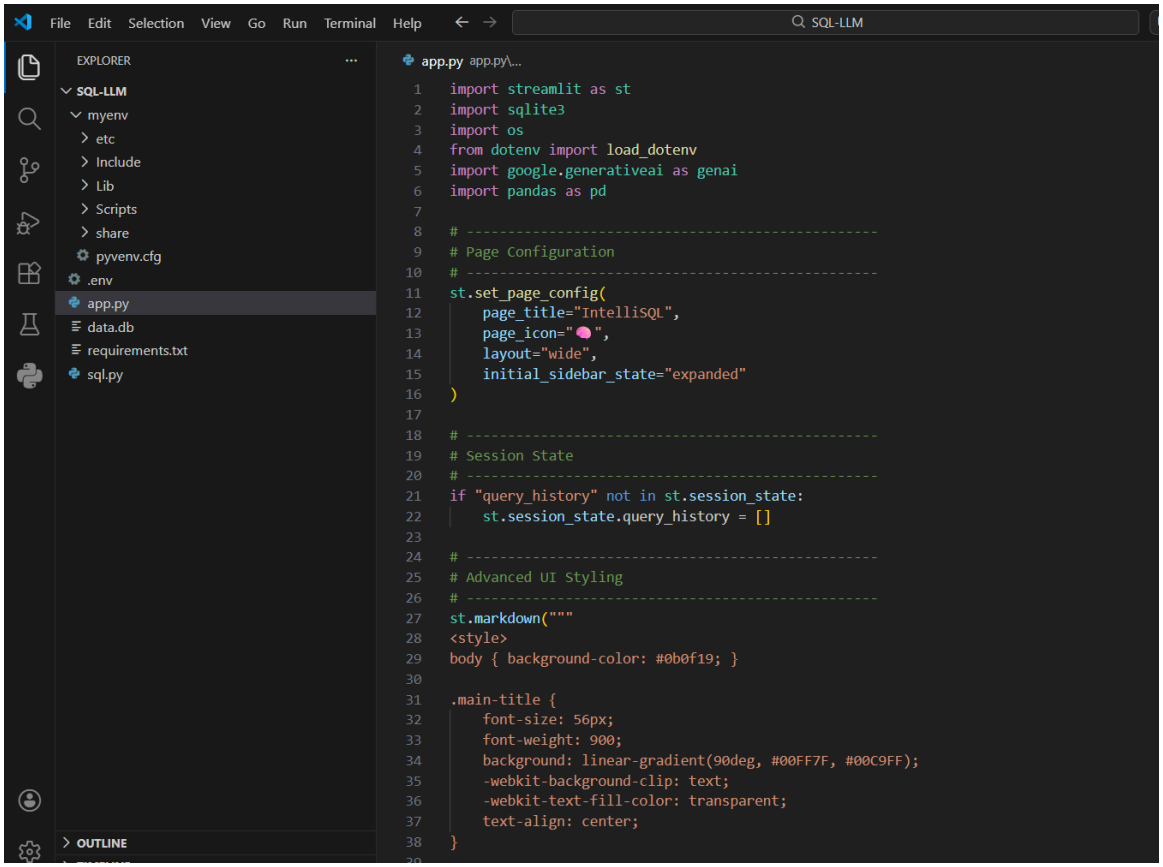
Tools:

- VS Code
- GitHub
- Postman

Platform:

- Windows OS





---

4. PROJECT DESIGN

4.1 Problem–Solution Fit

Problem: Writing SQL queries manually is difficult for many users.

Solution: An AI-powered system converts plain English into SQL, executes it automatically, and returns structured results reducing the need for SQL expertise.

4.2 Proposed Solution

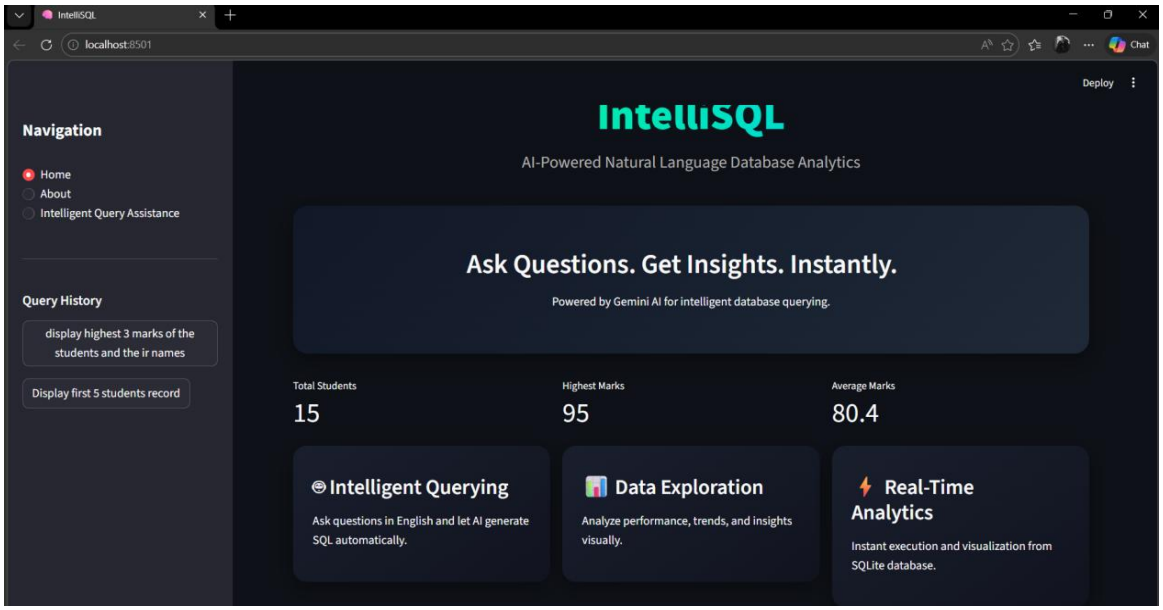
The system workflow:

- User enters natural language query
- Prompt engineered and sent to Gemini Pro
- Gemini generates SQL statement
- SQL query validated
- Executed on database
- Results formatted and displayed

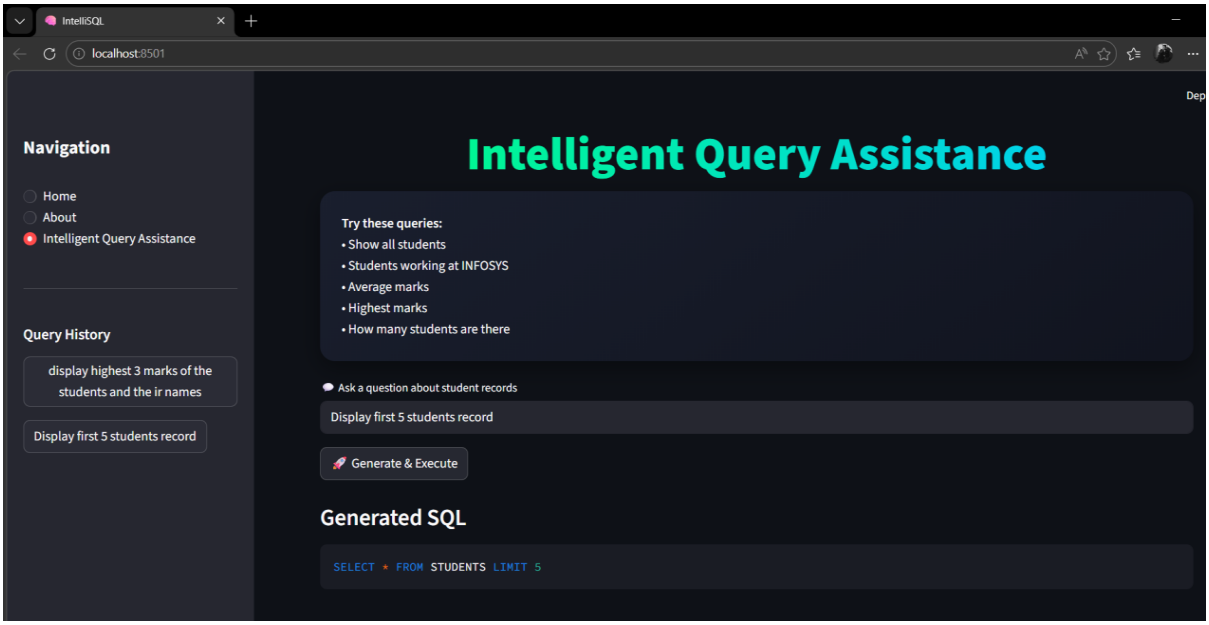
4.3 Solution Architecture

Components:

- User Interface (Streamlit / Web UI)
- API Integration Layer
- Gemini Pro LLM



- Output Display Module



---

5. PROJECT PLANNING AND SCHEDULING

Project Phases

1. Requirement Analysis
2. System Design
3. Development
4. Testing
5. Documentation

Schedule

Phase	Activity	Duration
Phase 1	Requirement Analysis	3 Days
Phase 2	Design	3 Days
Phase 3	Development	7 Days
Phase 4	Testing	4 Days
Phase 5	Documentation	3 Days

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Functional Testing

- Tested natural language input
- Verified SQL generation
- Verified database execution
- Tested error handling
- Validated result formatting

6.2 Performance Testing

- Checked response time
- Tested UI interaction
- Verified stability

---

7. RESULTS

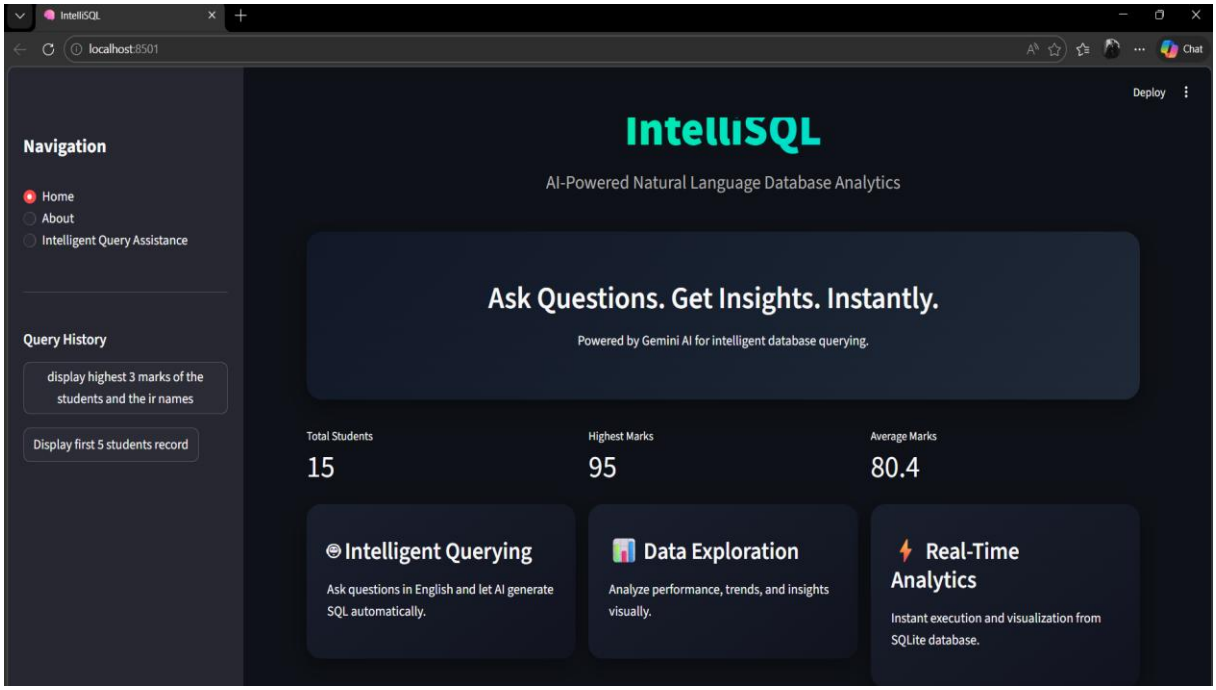
The application successfully:

- Converts natural language to SQL
- Executes generated SQL queries
- Displays structured results
- Handles invalid inputs
- Demonstrates real-world AI integration

7.2 Output Screenshots

Insert screenshots here:

Figure 1: Main Interface of the Application



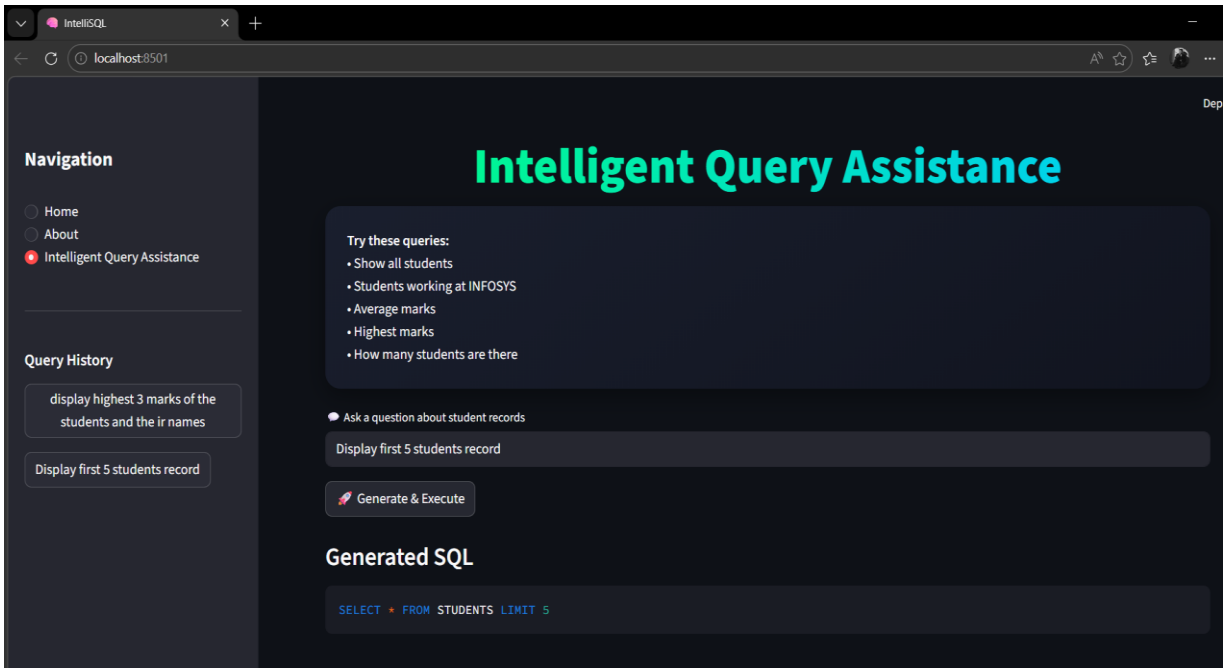


Figure 2: Sample SQL Query Generated

8. ADVANTAGES AND DISADVANTAGES

Advantages

- No SQL knowledge required

- Fast data retrieval
- AI-driven automation
- Scalable solution
- Improves productivity

Disadvantages

- Requires internet (API dependency)
- Possible incorrect SQL generation
- API usage limits

---

9. CONCLUSION

The "Intelligent SQL Querying with LLMs Using Gemini Pro" project demonstrates how Generative AI can simplify database interactions. By converting natural language into SQL queries, the system makes database management more accessible and efficient. This project highlights the powerful integration of Large Language Models with traditional database systems.

10. FUTURE SCOPE

- Download option
- Cloud deployment
- Multi-language support
- Mobile version

---

11. APPENDIX

11.1 Security Implementation

- API key stored in .env
- Not hardcoded

- .env ignored in Git

11.2 Project Demo

🔗 Demo Video:
https://drive.google.com/file/d/1dKx1-BbFcxTFMULD88edcdG3r5XKSwrn/view?usp=drive_link

The demo shows:

- Natural language query input
- SQL generation
- Database execution
- Result display

11.3 References

- Google Gemini API Documentation
- MySQL Documentation
- Python Documentation
- Streamlit Documentation
- NLP & LLM Research Papers