

1. INTRODUCTION

1.1. Project Overview

IntelliSQL: Intelligent SQL Querying with LLMs using Gemini Pro is an intelligent system designed to simplify database interaction by allowing users to query databases using natural language. Many users find it difficult to write complex SQL queries because it requires knowledge of database syntax and structure. This project solves that problem by using Large Language Models (LLMs) to automatically convert human language into accurate SQL queries. The system allows archivists and administrators to upload, update, and maintain artifact records, while researchers, students, and educators can easily search and access artifact information through a user-friendly interface.

Overall, the project demonstrates how modern AI technologies like Gemini Pro can enhance traditional database systems, making them more accessible, intelligent, and efficient for a wide range of users. The system uses Gemini Pro to understand user questions, analyze the database schema, and generate optimized SQL queries to give and best understandings queries.

1.2. Objectives

The main objectives of the intelligent sql querying with llms using gemini pro are:

- To develop an intelligent system that converts natural language queries into SQL queries.
- To integrate LLMS(Gemini Pro) for understanding user questions and database structure.
- To simplify database access for non-technical users who do not know SQL
- To generate accurate and optimized SQL queries automatically
- To reduce the time and effort required to write complex database queries

2. IDEATION PHASE

2.1. Problem Statement

The problem is to develop a system that allows users to interact with databases using simple natural language instead of manually writing SQL queries. The system should be able to understand the user's request, convert it into an accurate SQL query, execute it on the database, and return meaningful results. automatically translate natural language questions into SQL queries, making database access easier, faster, and more efficient for all types of users

Example:

Customer Problem Statement Template				
I am	I'm trying to	But	Because	Which makes me feel
A vehicle buyer (two-wheeler or four-wheeler)	Get complete details about a vehicle such as brand, model, mileage, features, and price	Vehicle information is spread across many websites and takes a lot of time to search	Most platforms show generic or incomplete details and do not give instant insights from images	Confused and unsure about making the right vehicle choice
A student or common user interested in vehicles	Understand vehicle specifications quickly using images	Reading long articles and specifications is boring and time-consuming	There is no simple AI-based tool that explains vehicle details in an easy format	Frustrated and less confident in my understanding
An eco-conscious user	Find fuel-efficient or eco-friendly vehicles	Information about mileage, resale value, and efficiency is not clearly available	Most platforms do not summarize vehicle details in one place	Unsatisfied and overwhelmed while choosing a vehicle

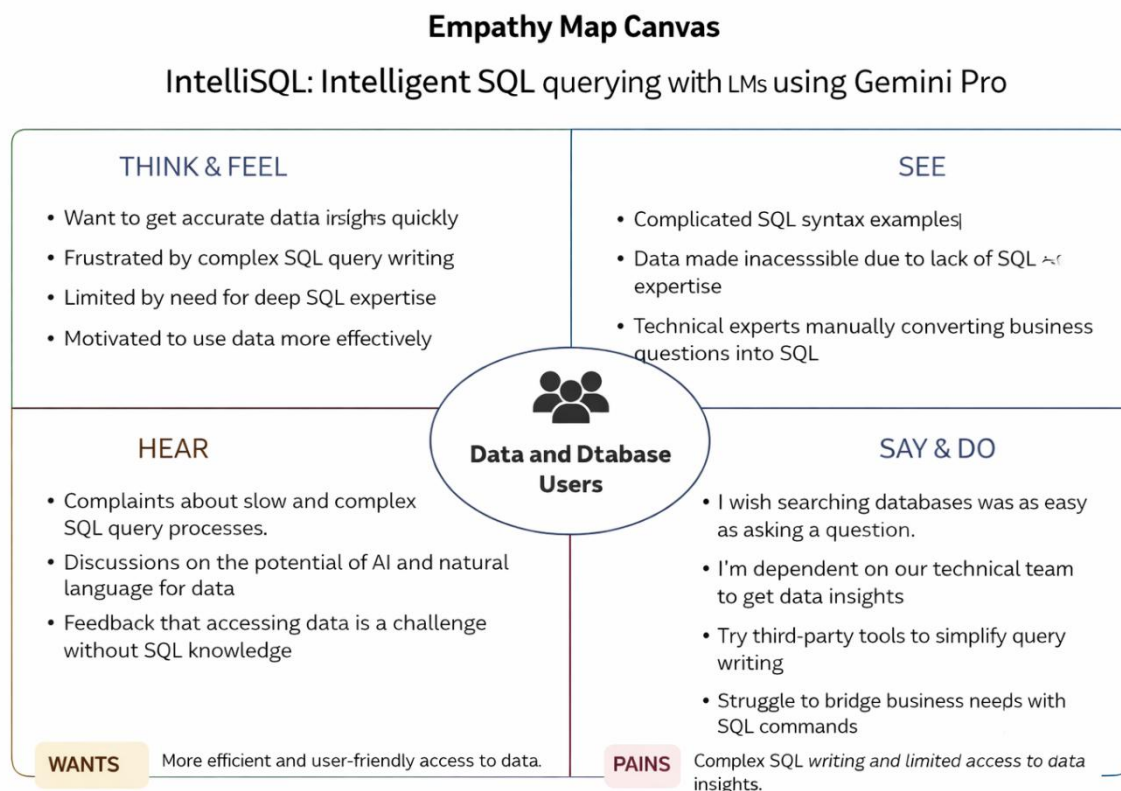
I am	I'm trying to	But	Because	Which makes me feel
A Data Analyst	Query database using natural language	Writing complex SQL queries takes time and requires strong SQL	Traditional database systems only understand structured SQL commands	Frustrated, slow in analysis, and dependent on technical experts
A Developer	Build smarter data applications	Integrating natural language with SQL database is complex	Traditional systems lack AI-powered query understanding	Challenged to provide user-friendly

2.2. Empathy Map Canvas

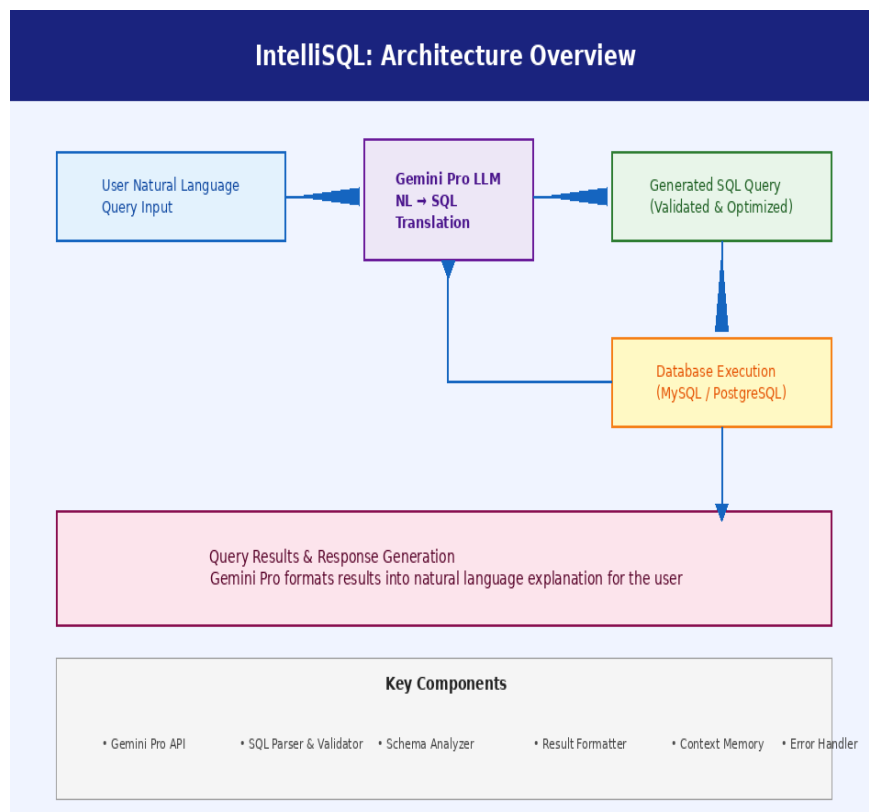
Empathy Map Canvas:

The IntelliSQL Empathy Map focuses on understanding the needs, challenges, and expectations of users who interact with databases but may not have strong knowledge of SQL. These users include students, data analysts, researchers, business professionals, and beginners in database management. Many users need information from databases but find it difficult to write correct SQL queries or understand complex database structures. They often look for a simpler way to access data without learning technical query languages.

Example:



Example: intelligent sql querying with llms using gemini pro



2.3 Brainstorming

Brainstorm & Idea Prioritization:

Brainstorming for “Gemini historical artifact description”:

Brainstorming for the project “IntelliSQL: Intelligent SQL Querying with LLMs using Gemini Pro” focuses on exploring how large language models (LLMs) can be used to translate natural language questions into accurate SQL queries. The main goal of this brainstorming session is to generate ideas on how IntelliSQL can assist students, developers, analysts, and businesses in querying databases without needing expertise in SQL syntax.

IntelliSQL can solve this problem by leveraging Gemini Pro’s advanced natural language understanding to interpret user queries and generate optimized SQL statements. The system will bridge the gap between non-technical users and relational databases by providing accurate, context-aware SQL generation along with query explanations and result summaries.

Step-1: Team Gathering, Collaboration and Select the Problem Statement:

Brainstorm & idea prioritization

Use this template to run your own brainstorming sessions or your team can explore IntellisQL ideas even if you're not sitting in the same room.

- 10 minutes to prepare
- 1 hour to collaborate
- 2-8 people recommended

1 Before you collaborate

A little preparation goes a long way with this session. Here's what you need to do to get going.

8-10 minutes

- Team gathering**
Who should participate in the session and send an invite. Brainstorm information or pre-work on.
- Set the goal**
Think about the problem you'll be focusing on solving in this brainstorming session.
- Learn how to use the tools**
Use IntellisQL Supapowers to run a happy and productive session.

[Open article 1](#)

2 Define your problem statement

What problem are you trying to solve? Frame your problem as a low-ambiguity statement. This will be the focus of your brainstorm.

8-15 minutes

IntellisQL: Intelligent SQL Querying with LLMs using Gemini Pro

IntellisQL is an AI-powered system that enables users to query relational databases using plain English. Powered by Gemini Pro LLM, it translates natural language into accurate SQL queries, making databases accessible to all.

Team gathering

Brainstorm ideas

Group & prioritize

Select best idea

Step-2: Brainstorm, Idea Listing and grouping:

3 Brainstorm – IntellisQL SQL Querying System

Write down any ideas that come to mind that address the IntellisQL problem statement.

8-10 minutes

Person 1
Developer

- Query DB with plain English
- Auto-generate SQL queries
- Explain SQL in simple terms

Person 2
Data Analyst

- Filter data by date/name
- Compare sales data easily
- Generate reports fast

Person 3
Student

- Learn SQL by seeing queries
- Ask questions about data
- Understand query results

Person 4
Business User

- Get sales insights fast
- No SQL skills needed
- Easy data exploration

Person 5
DB Admin

- Validate LLM SQL output
- Optimize generated SQL
- Secure query execution

TIP
Use sticky notes to write one idea per note. Focus on IntellisQL SQL querying.

4 Group Ideas – IntellisQL SQL Querying System

Take turns sharing ideas while clustering similar or related notes. Give each cluster a label.

8-20 minutes

Group 1: Natural Language Query Interface

- NL to SQL conversion
- Query autocomplete

Group 2: SQL Generation & Optimization

- Accurate SQL generation
- Query optimization
- Schema-aware generation
- Join & filter handling

Group 3: Result Explanation & Visualization

- Explain SQL results
- Summary reports
- Chart & graph output
- Data export

Group 4: Database Compatibility

- MySQL support
- SQLite connection
- PostgreSQL integration
- Cloud DB support

Group 5: Security & Performance

- Secure query execution
- Fast query responses
- Access control
- Error handling

Step-3: Idea Prioritization:

4 Prioritize – IntellisQL SQL Querying System

Your team should focus on the most important and feasible ideas for IntellisQL.

8-20 minutes

TIP
Provide ideas for discussing and placing them on this matrix. Focus on ideas with high importance and high feasibility. These will have the most impact.

5 After you collaborate

Select the most important and feasible solution.

IntellisQL: AI-Powered Natural Language to SQL Query System

Create system prototype (MVP)

- NL Query Input Interface**
Accept plain English questions and display results
- Gemini Pro SQL Generator**
Translate queries to SQL using Gemini Pro LLM
- Database Connector**
Connect to MySQL, PostgreSQL, SQLite databases

Future improvements

- Query history & favorites**
Save and reuse past queries
- Voice-to-SQL support**
Accept voice queries and convert to SQL

Empty state

Add ideas

Focus ideas on matrix

Final prioritization

3. Requirement Analysis

3.1. Solution Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Natural Language input	Accept natural language questions or queries from the user through a web-based interface and pass them to the Gemini Pro LLM
FR-2	SQL Query Generation	Automatically convert user's natural language input into syntactically correct
FR-3	Database Schema Integration	Allow users to upload or define their database schema so the LLM can generate context-aware SQL queries
FR-4	Query Execution, access, and maintain system security	Execute the generated SQL query against the connected relational database (MySQL / PostgreSQL / SQLite) and retrieve the results for display

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

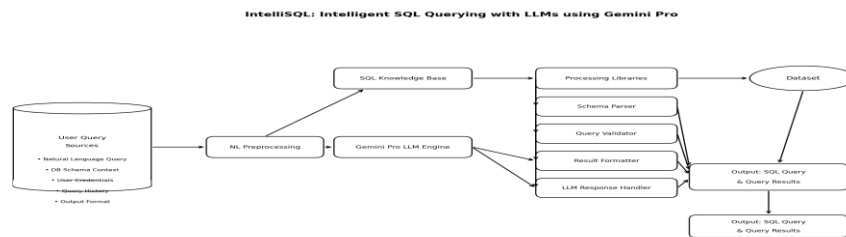
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system shall provide a clean, intuitive, and user-friendly web interface that requires no SQL knowledge, enabling non-technical users
NFR-2	Reliability	The system shall maintain 99.5% uptime and implement robust error handling to gracefully manage LLM API failures
NFR-3	Performance	The system shall generate and execute SQL
NFR-4	Availability	The system shall be accessible 24/7 during academic and production usage with minimal planned

3.2. Data Flow Diagrams

Data Flow Diagrams:

The IntelliSQL System is designed to enable intelligent, natural language-driven SQL querying using Large Language Models (LLMs) powered by Gemini Pro. The system begins with user query inputs, which include natural language questions, database schema context, user credentials, and query preferences. This data is processed by the LLM Query Engine, which serves as the central component

Flow Diagram:



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Data Analyst	Natural Language Query Input	USN-1	As a Data Analyst, I want to enter natural language queries so that the system	System allows artifact upload with name, mission, description, and images. Data is stored securely.	High	Sprint-1
Business User	SQL Query Generation and Access	USN-2	As a Business User, I want to search and retrieve database records using plain English	System provides accurate artifact search results. Artifact details and images are displayed clearly.	High	Sprint-1
Student / Public User	Query Results Viewing	USN-3	As a Student, I want to view query results and generated SQL so that I can learn how natural language maps to database queries.	Artifact information is displayed clearly. Images and mission details are accessible.	Medium	Sprint-2
System Administrator	Database Schema Management	USN-4	As an Administrator, I want to manage artifact records so that the system remains accurate and secure.	Admin can add, edit, and delete artifact records. System maintains data integrity.	High	Sprint-1
Database Admin	Query History	USN-5	As an Administrator, I want to manage artifact record	System allows preservation status updates.	High	Sprint-2

3.3. Technology Stack

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Critical element designed for users to input natural language queries and receive SQL output, ensuring an intuitive and informative experience	HTML, CSS, JavaScript
2.	Application Logic-1	Involves a robust backend system responsible for processing natural language queries, calling Gemini Pro API	Python
3.	Database	Involves the storage and management of structured data used to validate and execute.	SQLite / MySQL
4.	File Storage/ Data	Involves managing API calls to Gemini Pro for translating natural language to SQL, including prompt engineering and response parsing.	Google Gemini Pro API
5.	Frame Work	It is a crucial part of our program as it is responsible for connecting the frontend with the backend and routing API requests	Python Flask

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source frameworks accelerate development and ensure the reliability of IntelliSQL's backend pipeline	Python's Flask
2.	Scalability	System can handle increasing query loads and support multiple concurrent users.	REST APIs, load balancing
3.	Performance	Regular performance testing, monitoring, and optimization ensure accurate SQL generation with minimal latency.	Query accuracy metrics, response time benchmarks
4.	Availability	Application can be deployed on cloud or web servers, making it available at all times with minimal downtime.	High speed Linux based web servers.

4. PROJECT DESIGN

4.1. Problem Solution Fit

Problem – Solution Fit Template:

IntelliSQL addresses the critical problem of complex SQL query writing, which remains a significant barrier for non-technical users seeking insights from relational databases. Business analysts, researchers, and domain experts often depend on data engineers to retrieve information, creating bottlenecks and delays

Purpose:

- ❑ To enable non-technical users to query relational databases using plain English without writing SQL.
- ❑ To leverage Gemini Pro LLM for accurate, context-aware natural language to SQL translation.
- ❑ To reduce dependency on data engineers and accelerate data-driven decision making.
- ❑ To generate optimized, schema-aware SQL queries with explanations

Template:



4.2. Proposed Solution

Proposed Solution Template:

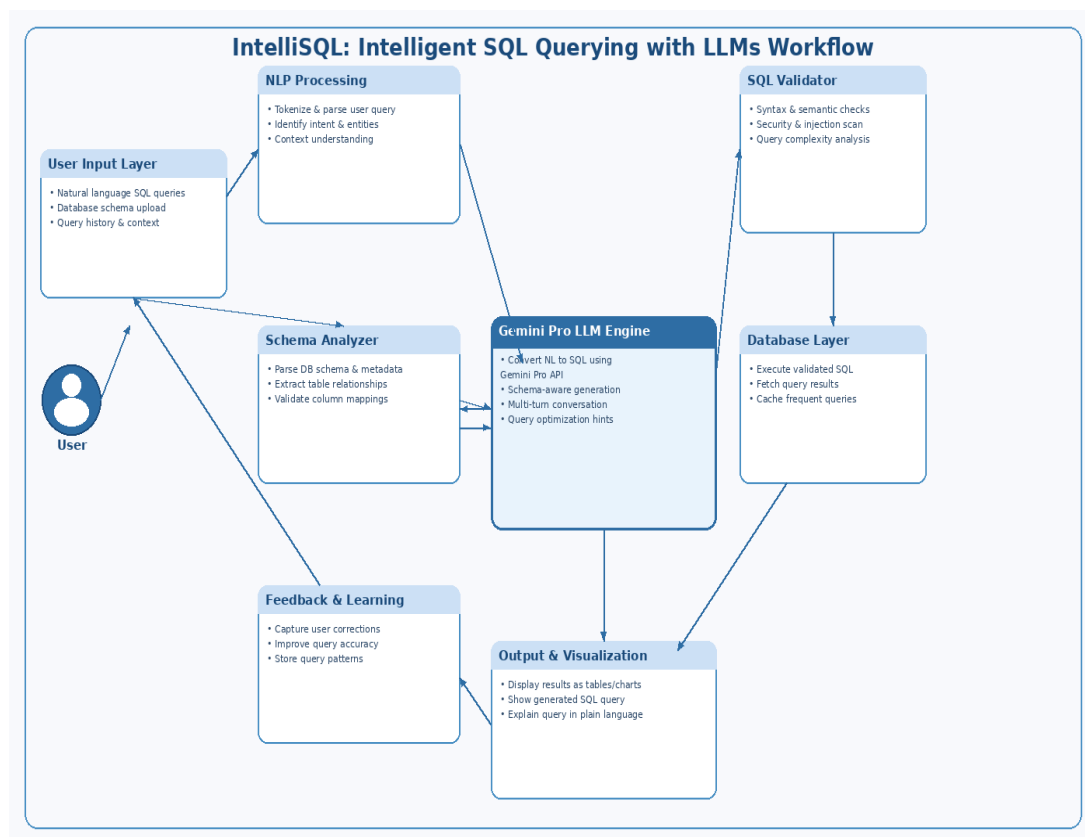
Project team shall fill the following information in the proposed solution template.

S.No	Parameter	Description
1.	Natural Language Input Module	Accept natural language questions from users via a web or CLI interface Parse and pre-process user queries before passing them to the LLM for SQL generation
2.	Gemini Pro LLM Integration	Send parsed natural language queries to the Gemini Pro API for SQL generation Receive SQL statements from the LLM and validate them for correctness and safety
3.	Database Schema Manager	Store and manage database schema metadata including table names, column types, and relationships Provide schema context to Gemini Pro for accurate
4.	SQL Query Execution Engine	Translate user questions into optimized SQL queries using prompt engineering Ensure generated queries are syntactically correct and logically accurate
5.	Results Display and Visualization	User interface to display generated SQL queries and corresponding result sets Show query output in tabular format
6.	User Authentication and Access Control	Manage users such as data analysts, developers, and administrators Control database access levels
7.	Query History and Performance Monitor	Track all previously executed queries and their execution times Help users refine queries and monitor system performance and LLM response

4.3 Solution Architecture

The solution architecture of IntelliSQL is designed to enable intelligent, natural language-driven SQL query generation using Google's Gemini Pro large language model. The system begins with the user providing a plain English query along with a database schema. This input is processed through an NLP layer that identifies intent and context, which is then forwarded to the Gemini Pro LLM engine for accurate SQL generation.

Solution Architecture:



5. PROJECT PLANNING & SCHEDULING

5.1. Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Environment Setup	USN-1	As a developer, I want to set up the development environment with Python	2	High	All Team Members
Sprint-1	API Configuration	USN-2	As a developer, I want to configure Google Gemini Pro API with secure	1	High	All Team Members
Sprint-2	Database Creation	USN-3	As a developer, I want to create SQLite database with sample data for testing	3	High	All Team Members
Sprint-2	Model Integration	USN-4	As a user, I want the system to translate my natural language questions into SQL .	3	High	All Team Members
Sprint-3	Query Processing	USN-5	As a user, I want to input natural language queries and receive accurate SQL	4	High	All Team Members
Sprint-3	Result Display	USN-6	As a user, I want to see query results in a clear, formatted	2	Medium	All Team Members

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Complete	Sprint Release Date (Actual)
Sprint 1	20	4 Days	28 January 2026	31 January 2026	20	31 January 2026
Sprint 1	20	4 Days	28 January 2026	31 January 2026	20	31 January 2026
Sprint 2	20	8 Days	02 February 2026	09 February 2026	20	09 February 2026
Sprint 2	20	8 Days	02 February 2026	09 February 2026	20	09 February 2026
Sprint 3	20	7 Days	12 February 2026	18 February 2026	20	18 February 2026
Sprint 3	20	7 Days	12 February 2026	18 February 2026	20	18 February 2026

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Test Scenarios & Results:

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Natural Language Query Input	Enter a valid natural language query and submit; also test empty input	Valid query accepted; error shown for empty input	As Expected	Pass
FT-02	SQL Query Generation via Gemini Pro	Enter a plain English database question and click "Generate SQL"	Syntactically correct SQL query generated and displayed	As Expected	Pass
FT-03	Gemini Pro API Connection Check	Trigger SQL generation with a valid Gemini Pro API key configured	API responds successfully and returns SQL	As Expected	Pass
FT-04	Query Result Display & Formatting	Generate SQL and verify output display in the results panel	SQL displayed with proper syntax highlighting	As Expected	Pass
FT-05	Schema-Aware Query Accuracy	Upload a DB schema and ask schema-specific questions	Correct table and column names used in generated SQL	As Expected	Pass
PT-01	SQL Response Time Test	Measure time from query submission to SQL output	SQL generated within 3–5 seconds	Within Limit	Pass
PT-02	Multiple Query Request Handling	Submit multiple SQL queries sequentially without reloading	Application handles all requests without crash or error	Stable	Pass
PT-03	Deployment Test	Access the deployed IntelliSQL app via browser and run a query	Application loads and SQL generation works correctly online Correctly give inputs Check the queries understanding	Working	Pass

7. RESULTS

7.1. Output Screenshots

The complete execution of IntelliSQL querying description is represented step by step in the following screenshots.

Step 1: To run the Streamlit Application we have to use the command `python streamlit run app.py` in the terminal in path where the `app.py` file is located.

```
PS C:\Users\pavan\Downloads\IntelliSQL-Intelligent-SQL-Querying-with-LLMs-Using-Gemini-Pro_pa1-main\SQL-LLM> streamlit run app.py

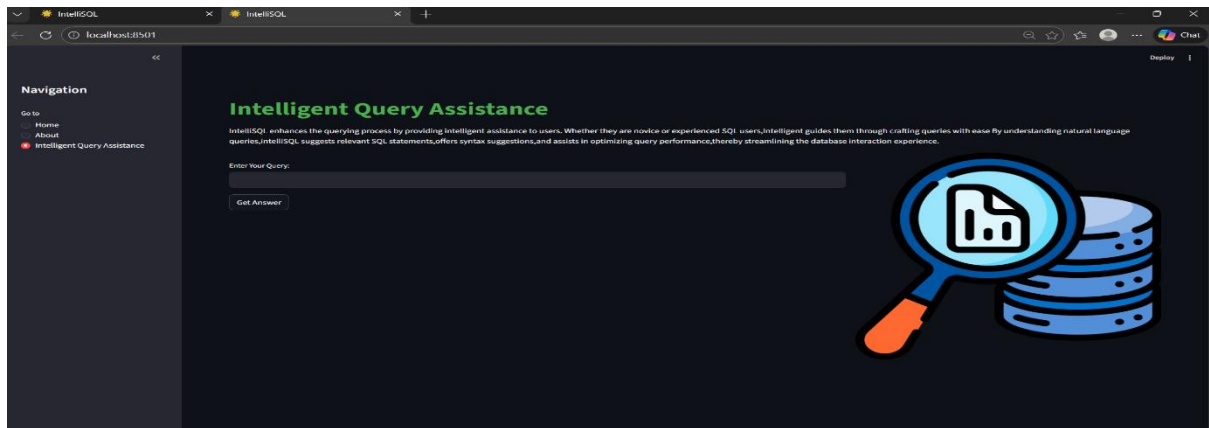
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://10.65.2.222:8501
```

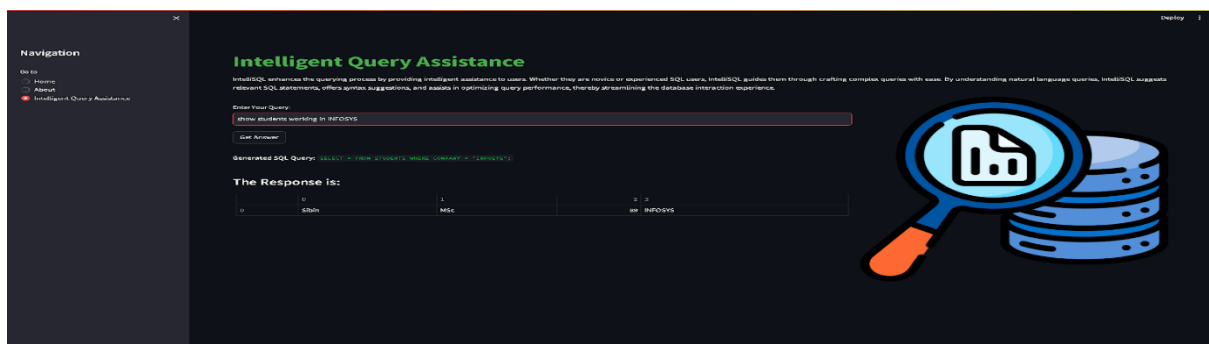
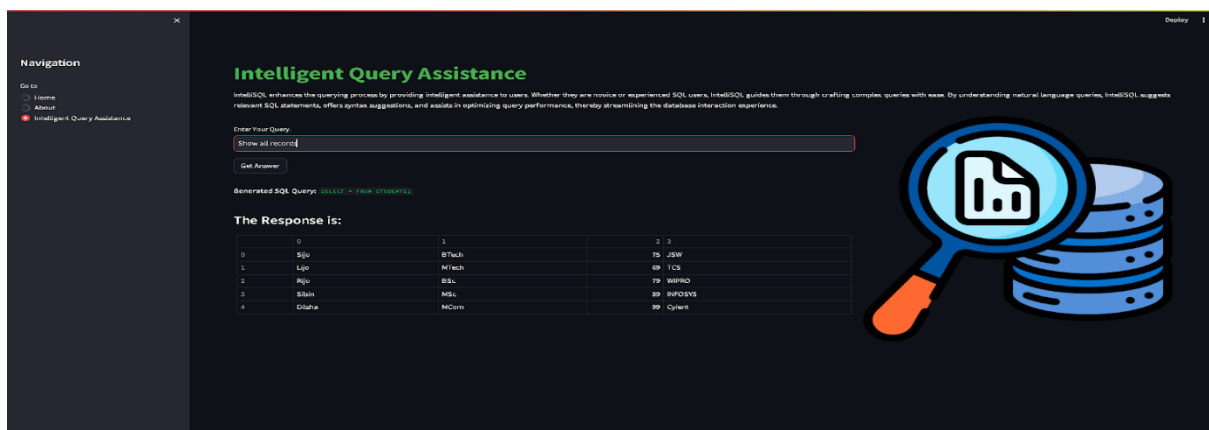
Step 2: After running the command in terminal, the code will get executed and the webpage will open directly. Another way to open webpage is that a localhost link will get generated in the terminal, we can access the webpage using that link.



Step 3: The Streamlit webpage opens as shown in the figure given below. This is an automated webpage. No secondary HTML codes required to build this webpage. Python code itself consists the webpage building code.



Step 4: After opening this portal it displays the Intelli sql querying Description App, First we should type about any image description what ever we want contant in input prompt . It asks choose any image like wheather location etc. It generate based on the user inputs as shown in the following images



Navigation

Go to

Home

About

Intelligent Query Assistance

Intelligent Query Assistance

IntelliSQL enhances the querying process by providing intelligent assistance to users. Whether they are novice or experienced SQL users, IntelliSQL guides them through crafting complex queries with ease. By understanding natural language queries, IntelliSQL suggests relevant SQL statements, offers syntax suggestions, and assists in optimizing query performance, thereby streamlining the database interaction experience.

Enter Your Query

Calculate the average of marks in the database

Get Answer

Generated SQL Query

`SELECT AVG(MARKS) FROM STUDENTS;`

The Response is:

0	85.2000
---	---------

Display

Navigation

Go to

Home

About

Intelligent Query Assistance

Intelligent Query Assistance

IntelliSQL enhances the querying process by providing intelligent assistance to users. Whether they are novice or experienced SQL users, IntelliSQL guides them through crafting complex queries with ease. By understanding natural language queries, IntelliSQL suggests relevant SQL statements, offers syntax suggestions, and assists in optimizing query performance, thereby streamlining the database interaction experience.

Enter Your Query

Who got high marks?

Get Answer

Generated SQL Query

`SELECT NAME, MAX(Marks) AS Highest_Marks FROM STUDENTS;`

The Response is:

0	1
0	John

Display

8. ADVANTAGES AND DISADVANTAGES

Advantages

Automatically converts user questions into accurate SQL queries.

- Reduces the need for advanced SQL knowledge.
- Allows users to query databases using simple natural language.
- Saves time compared to manually writing complex queries.
- Improves productivity for analysts, students, and developers.
- Provides quick and meaningful insights from large datasets.
- Can be integrated into web applications, dashboards, and enterprise systems.

Disadvantages

- Performance depends on the quality of the LLM model.
- Requires internet connection or API access to Gemini Pro.
- May struggle with very complex database schemas.
- Security and data privacy must be handled carefully.
- Processing large queries may increase response time
- Generated SQL queries may sometimes be incorrect
- Natural language questions may be misunderstood sometimes
- Risk of generating inefficient SQL queries.

9. CONCLUSION

IntelliSQL demonstrates how Large Language Models such as Gemini Pro can simplify database interaction by allowing users to query data using natural language instead of writing complex SQL commands. The system interprets user intent, converts it into accurate SQL queries, executes them on the database, and returns meaningful results.

Github link:

[https://github.com/venkatapavan07/Intelligent SQL Querying Pavan.git](https://github.com/venkatapavan07/Intelligent_SQL_Querying_Pavan.git)

Demo link:

<https://drive.google.com/file/d/1Q2yQDSiW4oSckBg-dzW7AE-E5jkXzX8Q/view?usp=drivesdk>