Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2026TMIDS41804
Project Name	IntelliSQL: Intelligent SQL Querying with LLMs Using Gemini Pro
Maximum Marks	4 Marks

Technical Architecture:

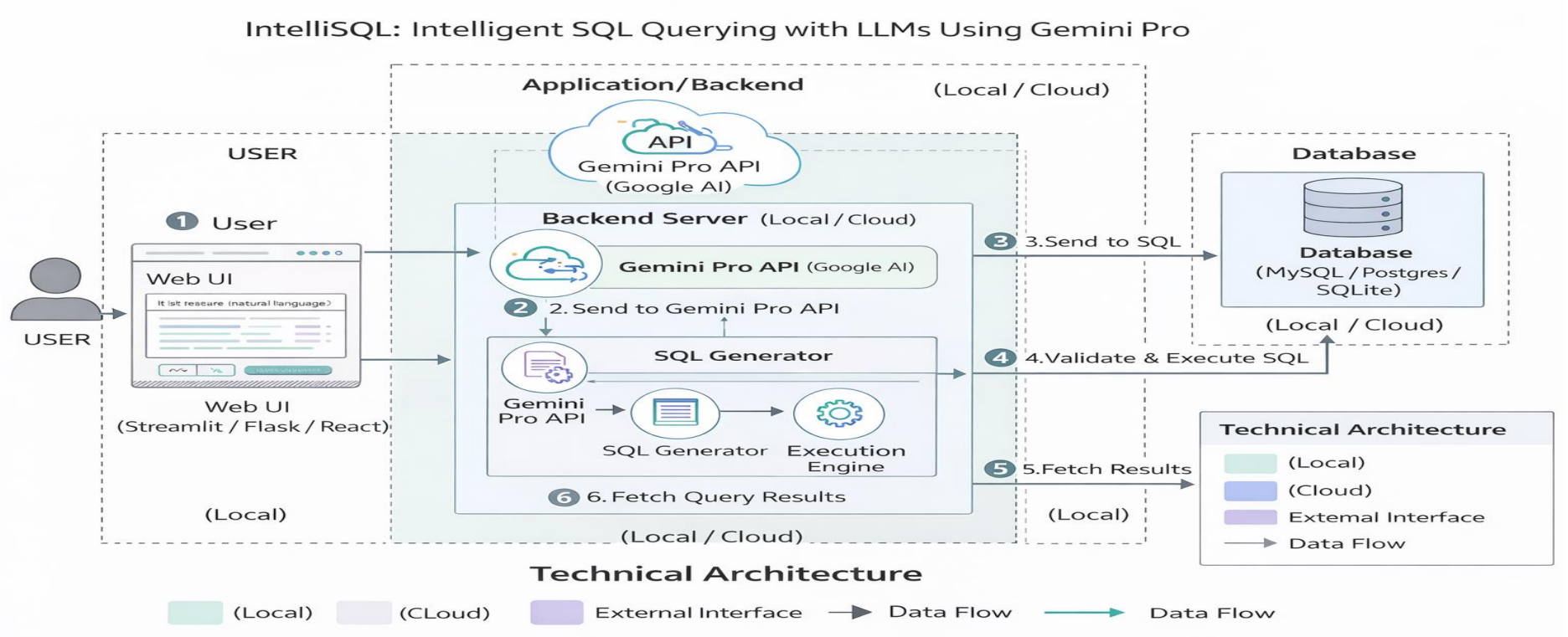

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web interface where user enters natural language queries and views results	Streamlit / Flask (HTML, CSS, JavaScript if applicable)
2.	Application Logic-1	Handles user input processing and request routing	Python (Backend Logic)
3.	Application Logic-2	Converts natural language query to SQL using LLM	Gemini Pro API (Google AI)
4.	Application Logic-3	SQL validation and execution engin	Python SQL Connector / SQLAlchemy
5.	Database	Stores structured data (tables, records)	MySQL / PostgreSQL / SQLite
6.	Cloud Database	Optional cloud-hosted database service	Google Cloud SQL / AWS RDS (if cloud deployed)
7.	File Storage	Stores logs, configuration files, query history	Local File System / Cloud Storage
8.	External API-1	Natural Language to SQL conversion	Gemini Pro API
9.	External API-2	Authentication or user management (if implemented)	Firebase Auth / OAuth (Optional)
10.	Machine Learning Model	Large Language Model used for SQL generation	Gemini Pro (LLM)
11.	Infrastructure (Server / Cloud)	Deployment of application	Local System / Google Cloud / AWS / Docker

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks used to build the system	Python, Streamlit / Flask

S.No	Characteristics	Description	Technology
2.	Security Implementations	Protect API keys and database access, secure communication	HTTPS, API Key Authentication, Environment Variables
3.	Scalable Architecture	Supports multiple users and large databases	Modular Backend Design, Cloud Deployment
4.	Availability	System accessible anytime with minimal downtime	Cloud Hosting / Local Server
5.	Performance	Fast NL-to-SQL conversion and quick database response	Optimized SQL Queries, Efficient API Calls

References:

- **https://ai.google.dev/**
(Official Google AI / Gemini documentation)
- **https://cloud.google.com/vertex-ai**
(Google Cloud AI & LLM services documentation)
- **https://cloud.google.com/architecture**
(Google Cloud Architecture best practices)
- **https://docs.streamlit.io/**
(Streamlit framework documentation)
- **https://flask.palletsprojects.com/**
(Flask backend documentation)
- **https://www.mysql.com/**
(MySQL database documentation)
- **https://www.postgresql.org/docs/**
(PostgreSQL official documentation)

- **https://aws.amazon.com/architecture/**
(Cloud architecture reference models)

- **https://martinfowler.com/architecture/**
(Software architecture principles)

- **https://medium.com/topic/system-design**
(Technical architecture & system design articles)