**Gen AI Application to Query Over College Alumni Data**

# TEAM NAME : TEAM CATALYSTS

**TEAM MEMBERS:**

* **SOWBHARANIKA JANANI JS**
* **SUBHADEVI K**
* **RAHUL K**

# Project Overview

This documentation will describe how the design and architecture of the Alumni Data Query Chatbot was exactly made, how it queries and retrieves data from an alumni database, based on natural language queries in dynamically generated SQL queries by Google's Gemini AI that query a database on behalf of the user to create answers to questions based on the retrieved data.

# Key Decisions

## Architecture

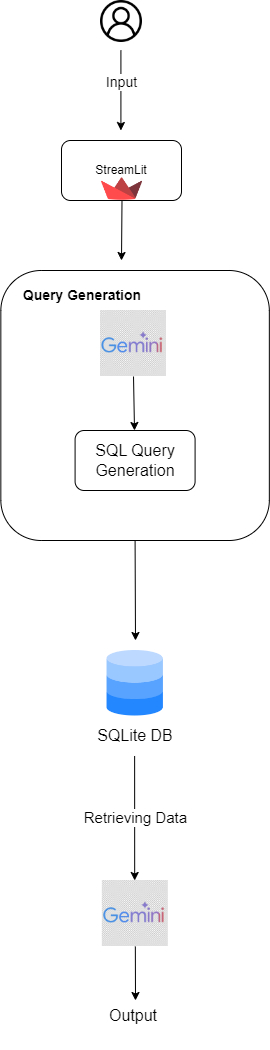
**User Interface (UI):** Streamlit is integrated to create the Simple and Intuitive Chat Interface. The user can ask questions about the alumni data, and the chatbot responds accordingly.

**Database:** Student data are stored in an in-memory SQLite database. All the data are loaded from a CSV file; for this purpose, Pandas was used to transform it into a DataFrame before storing data in SQLite for querying purposes.

**Backend Services:** The actual backend functionality is done by Google's Gemini AI (Generative AI), which ingests all user inputs, generates SQL queries, and formulates responses based on the returned query results.

## Dataflow

1. **Input:** Users interact with the chatbot by typing natural language queries.  
2. **Query Generation:** Gemini AI interprets the input and generates an appropriate SQL query to retrieve data from the SQLite database.  
3. **Data Retrieval:** The SQL query is executed, and relevant data is fetched.  
4. **Response Generation:** The result is passed to Gemini AI, which formulates a natural language response.  
5. **Output:** The chatbot displays the response, providing users with relevant alumni data.

****

## Vector Database / Semantic Search

Not Applicable: Since this project primarily uses structured SQL queries for data retrieval based on specific fields, a vector database or semantic search engine is not required. The queries directly target the SQLite database using SQL.

## Agents

Not Applicable: This project does not use autonomous agents or task automation frameworks like LangChain or Marvin.

## LLM Choice

**Large Language Model (LLM)**: Google Gemini 1.5 has generated the SQL queries and responses from the chatbot. This is chosen because Gemini has understood the meaning of the user query and returned an appropriate piece of information by performing the process of natural language processing.

## Chat Interface

**Interface Tool:** Streamlit provides a web-based interface for interacting with the chatbot. The chat input field allows users to ask questions, while the backend processes the query and displays the results.  
**User Interaction:** Users type queries in a conversational format, such as 'List students placed in Wipro in 2020,' and the chatbot responds accordingly. The chat interface supports ongoing conversation with session management.

# Technical Components

1. **Pandas for Data Processing:** Load CSV data and pre-process before uploading into SQLite with it, such involves using pandas to rename columns then save to Excel.  
2. **SQLite for Data Storage:** For this chatbot, data will be stored in an in-memory SQLite database. There exists a database design where the chatbot will query the data fast enough to answer real-time questions.  
3. **Google Gemini AI for SQL Generation:** Google Gemini AI dynamically generates SQL queries to the alumni data given the input of the user. The response that comes after execution of the queries is also taken care by AI models.

4. **Streamlit for UI:** Streamlit offers extremely easy, chat-based UI wherein users can directly work with their alumni data system.

# Conclusion

Alumni Data Chatbot shall be a functionality-enabled, AI-powered querying system that will make interactions with alumni data incredibly hassle-free. It will use Google Gemini for natural language processing, SQL query generation, and response formulation in the form of a user-friendly Streamlit interface. Future versions of the product can really focus on improving data security, scaling up the database, and enriching user interactions.