Project Report: Sales Copilot

Objective:

The Sales Copilot project aims to create an intelligent system that processes user input in the form of text queries and provides relevant, actionable responses based on available data. The system leverages tools for structured and unstructured data retrieval to generate dynamic and contextually accurate answers.

Work Done Till Now:

1. **Agent Implementation with LangChain:**
   An agent has been implemented using LangChain, serving as the core decision-maker for selecting the appropriate tool based on user input. The agent processes the input text and determines whether to use the **SQL Tool** or the **Unstructured Data Tool** to retrieve the most relevant information.
2. **SQL Tool Implementation:**
   - The **SQL Tool** is responsible for querying structured data stored in a PostgreSQL database.
   - **Prompt Engineering:** A prompt has been defined that includes a clear schema of the tables and columns in the database, providing context for understanding the data structure.
   - **Keyword Matching:** Relevant keywords from the user input are identified to determine the corresponding tables in the database schema.
   - **SQL Query Generation:** The Gemini model is used to generate appropriate SQL queries based on the identified tables and the user query.
   - **Data Retrieval:** Once the query is executed, the PostgreSQL database returns the required data for generating the response.
3. **Unstructured Data Tool Implementation (RAG):**
   - For unstructured data, a **Retrieval-Augmented Generation (RAG)** approach is utilized, combining retrieval mechanisms with language generation.
   - **Data Retrieval:** Relevant documents or chunks of information are retrieved based on the user query.

    - **Response Generation:** Retrieved data is processed to generate an appropriate response addressing the user query.
    - **Tabular Data in Vector Database:** Tabular data has been stored in a **vector database** to improve the system's ability to handle reasoning-based queries. The unstructured data tool retrieves relevant information from the vector database when such queries are detected.
4. **Handling Larger Databases:**
    - The system manages a database containing **3 tables with 20,000 to 30,000 rows** of data effectively.
    - Most queries return responses in **under 10 seconds**, ensuring efficiency.
    - **Performance:** A slight slowdown occurs after processing **2-3 consecutive inputs**, and optimizations are underway to address this issue.
5. **Final Response Generation:**
    - After retrieving data from the SQL or Unstructured Data Tool, the agent generates a final response by integrating the findings.
    - The response is designed to be contextually accurate and aligned with the user's query.

Challenges and Current Work:

- **Issue:** Certain user queries do not return the expected responses, despite using both tools effectively.
- **Work in Progress:** Efforts are being made to improve the system's ability to choose the right tool based on a wider range of queries. Enhancements to the retrieval logic are also being explored to ensure more relevant data is retrieved for complex queries.
- **Future Steps:**
    - Refining the tool selection process by improving keyword extraction and relevance matching.
    - Enhancing the retrieval mechanism for unstructured data to ensure more accurate document matching.
    - Optimizing the agent workflow for faster and more efficient responses, especially for consecutive queries.

Technologies Used:

- **LangChain:** For building the agent that selects the appropriate tool.
- **PostgreSQL:** For structured data storage and retrieval.
- **Gemini:** For generating SQL queries based on a defined prompt.
- **Vector Database:** For storing tabular data to handle reasoning-based queries effectively.
- **RAG:** For unstructured data retrieval and response generation.

Below are the Output examples

Dynamic Document and Query Manager

Ask a question:

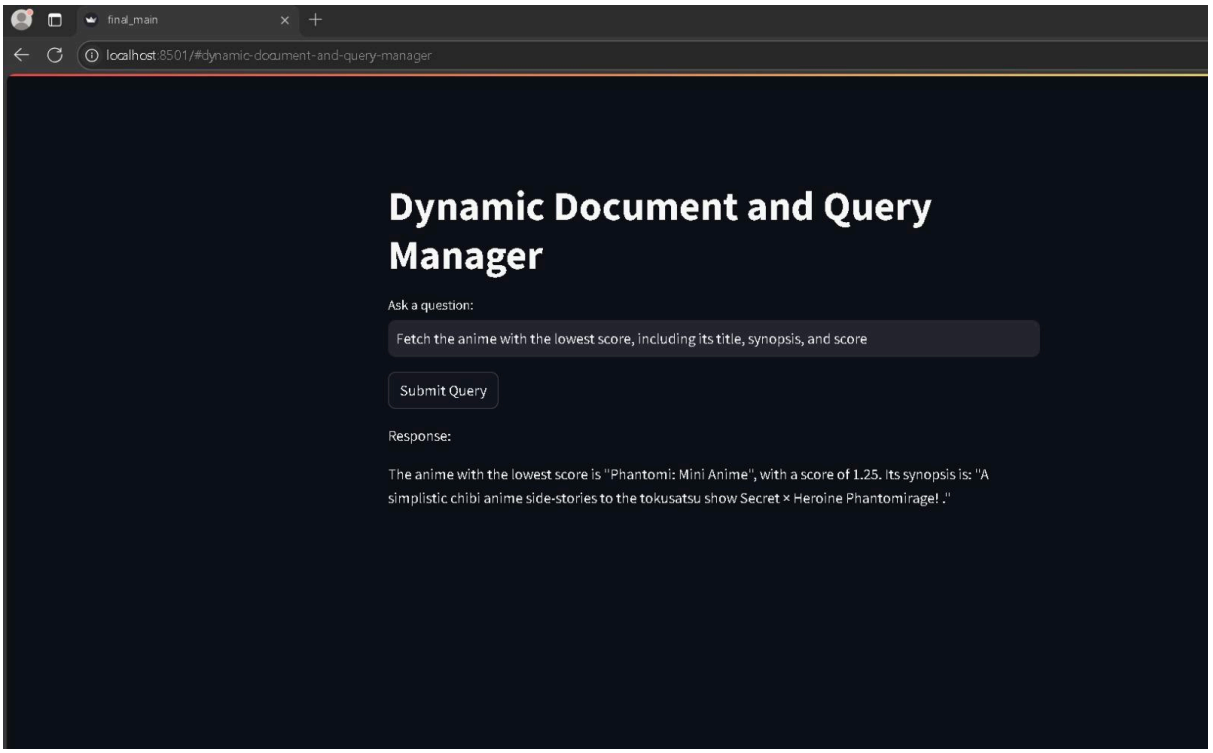
Which are the top 5 animes with the highest number of members associated with them?

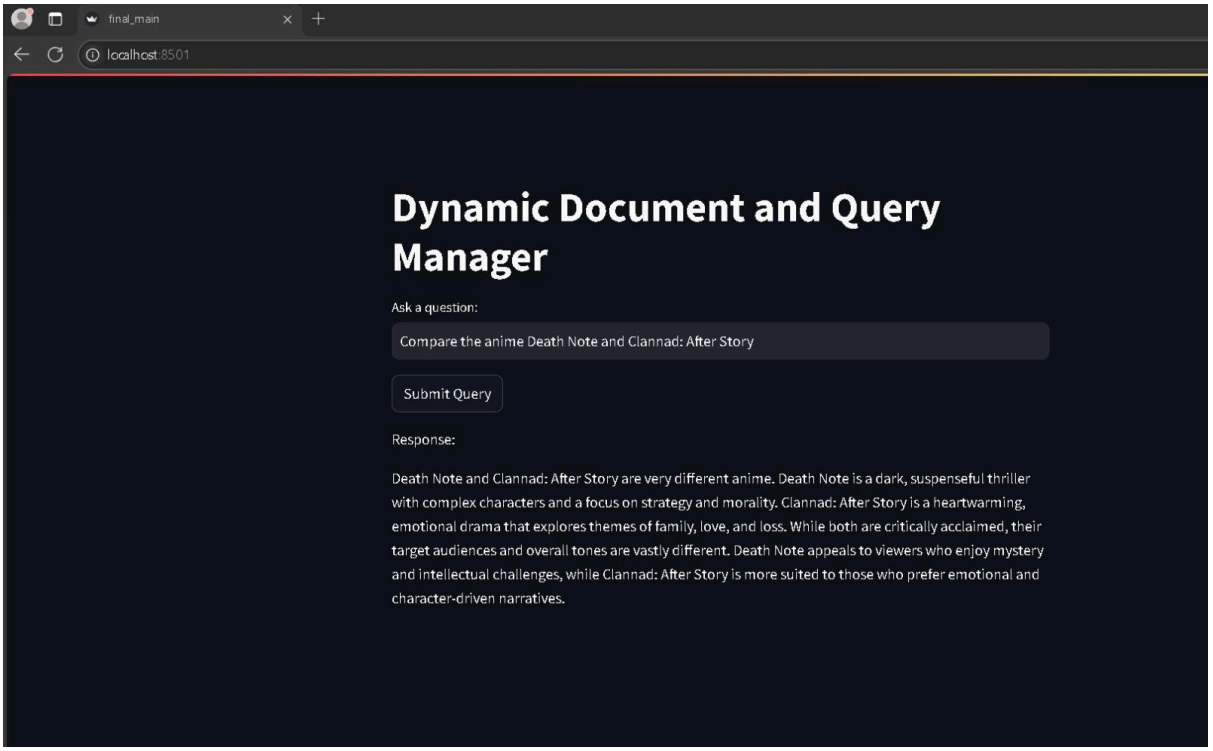
Submit Query

Response:

The top 5 animes with the highest number of members are:

1. Death Note (3416 members)
2. Steins;Gate (3116 members)
3. Kimi no Na wa. (2872 members)
4. Fullmetal Alchemist: Brotherhood (2548 members)
5. Clannad: After Story (2388 members)




final_main
localhost:8501
Dynamic Document and Query Manager
Ask a question:
Compare the anime Death Note and Clannad: After Story
Submit Query
Response:
Death Note and Clannad: After Story are very different anime. Death Note is a dark, suspenseful thriller with complex characters and a focus on strategy and morality. Clannad: After Story is a heartwarming, emotional drama that explores themes of family, love, and loss. While both are critically acclaimed, their target audiences and overall tones are vastly different. Death Note appeals to viewers who enjoy mystery and intellectual challenges, while Clannad: After Story is more suited to those who prefer emotional and character-driven narratives.