Second Delivery Guidelines **(until 1/12)**

Data Sources and Database Design Guidelines

Grading Breakdown

2 points (10% of the final grade)

Objective

Groups are expected to curate and present their data sources, which may include a mix of real-world and synthetic data. Additionally, a thorough explanation of their relational database design is required, illustrating how the chatbot will interact with and utilize this data. A SQL schema detailing the database's structure must accompany the submission to reinforce understanding.

Deliverables

1. **Data Sources**

- For real data, provide accurate **citations or links** to the data sources.
- For synthetic data, include a detailed **script** with the model name, as well as all hyperparameters and prompts used during data creation.

2. **Database Design**

- **SQLite Schema**: Present a detailed SQL schema detailing the database's structure. This should include table definitions, attributes, relationships, and any relevant constraints such as primary and foreign keys. As well as some text describing why you created each table for your particular use case.

Requirements

Database Structure

- Every group project must include a 'clients' table with a minimum of 10 entries. Each client must have a **'username'** and **'password'** to facilitate a login system, details of which will be elaborated in upcoming classes.
- The database should comprise at least **4 tables.** The final number may vary based on the project's specific use case but should adequately represent the project scope and requirements.
- Each table must contain a minimum of **50 rows** to ensure sufficient data for chatbot development and testing.

PDF Files

- Submit at least **3 PDF documents** or a minimum total of **15 pages** containing relevant information about the company or product. These documents will be used in subsequent classes to demonstrate data retrieval and utilization in chatbot conversations.

Chatbot Architecture Guidelines

Grading Breakdown

3 points (15% of the final grade)

Objective

Groups are tasked with detailing the backend architecture for their chatbots, providing comprehensive explanations and visual diagrams to illustrate how each chatbot feature is executed. The goal is to elucidate the data flow from user input to final response. This involves defining the complete data processing path for every chatbot functionality, thereby enabling precise implementation and troubleshooting.

Deliverables

User Stories for Each Feature

- Each feature of the chatbot should be presented as a user story. A user story should define who the user is, what they want to achieve, and why this feature is important.
- Example format for a user story:

  As a [user type],

  I want [functionality or action],

  So that [benefit or reason].

- Include user stories for all chatbot features, such as responding to company queries, database modification, RAG integration, and handling user intents.

Architecture Diagrams

- **Data Flow Diagrams for Core Functionalities**: Visualize the data flow for various user queries, highlighting different pathways based on user intention. Functionalities like:
  - CRUD Operations: Clearly illustrate the process and data pathways for executing SQL queries within the SQLite database.
  - Agents and Tools: Demonstrate the coordination and function of agents and tools.
  - Chains: Detail the execution flow of the chains.
  - PDF Querying and RAG: Map out the process for extracting data from PDFs and utilizing RAG capabilities.
  - Other APIs: If applicable, showcase the interaction with any additional APIs.

Requirements

Company Information Response Capability

- Groups must ensure that the chatbot can accurately respond to inquiries about the company by utilizing preloaded information.

Database Modification Capabilities

- The chatbot must support **CRUD (Create, Read, Update, Delete) operations** on at least one table within the SQLite database. This includes managing client information, with authorized modifications being logged for accountability.

Prompt Injection Tolerance

- Implement robust input validation to mitigate Prompt Injection risks. This includes establishing filters and input limits to ensure system security and integrity.

Data Extraction and Querying

- **SQLite Database**: The chatbot must be capable of accessing and utilizing client information and company details stored within the database.
- **PDF Documents**: Employ libraries such as pdfplumber or PyMuPDF to extract content from PDFs. Store this data as embeddings in Pinecone for Retrieval-Augmented Generation (RAG).

User Intentions and Functionality

- Implement at least **8 distinct user intentions**, similar to those discussed in class, including at least three functionalities unique to the chatbot's use case.
- Include at least **1 agent** and develop a minimum of **8 chains/runnables** and **3 custom tools** or use Langchain's built-in tools.

Memory Implementation

- Integrate memory functionality to allow the chatbot to retain context over multiple interactions, enhancing user experience by maintaining continuity and improving response relevance.

Retrieval-Augmented Generation (RAG) with Pinecone

- Use Pinecone to store embeddings of documents and relevant data, enabling the chatbot to pull contextual information for generating responses via a language model. Ensure that **at least one intention** can leverage RAG for its response.