LU Faculties Chatbot

The LU Faculties Chatbot is an intelligent conversational agent built using Google Gemini, a powerful Large Language Model (LLM), combined with the Retrieval-Augmented Generation (RAG) technique to enhance the accuracy and relevance of its responses. The project leverages various components from the LangChain framework to integrate information retrieval and response generation, offering users a chatbot that can effectively respond to queries related to the faculties at Lebanese University.

1 – **Google Gemini Chat Model:**

An LLM model used to generate responses to user queries.

Note: Gemini is based on the transformer architecture, that rely on self-attention mechanisms, which allow the model to process input sequences in parallel and capture relationships between words over long distances.

2 – **LangChain Framework:**

1- Chat Google Generative AI: Langchain component that allows user interaction with Google's generative Ai model 'Gemini' via API
   API key is injected into environment of the program to allow communication with Google Gemini's API.
2- Web Base Loader: Langchain component used to load content from the specified web pages. It retrieves HTML code, parses it to extract useful text (such as paragraphs, headings), and converts the content into Document objects containing the extracted text.
3- Text Splitters: Langchain component used to split large documents into smaller chunks based on size (1000 characters) to ensure that documents can be processed and embedded without overwhelming the model or vector store.
4- Embedding: Langchain method that converts chunks of documents into numerical vectors (embeddings). These embeddings can be stored in a vector database for semantic search.
   Note: Embedding method used is word embedding
5- Vector Store: Langchain component that stores embeddings of documents. Chroma is a tool used to create and manage the vector database that stores these embeddings. The Chroma client is used to interact with the vector store.
6- Retriever: Langchain component that fetches the most relevant document chunks from the vector store based on user queries. It retrieves documents based on their semantic similarity to the query.

7- RagChain: langchain method that combines doc retrieval, formatting them and passing them to LLM to generate response in a single chain.
8- Prompt Engineering: Langchain method that involves crafting or applying predefined prompts (from LangChain Hub) to guide the model in generating more accurate or relevant responses. It is part of the RAG chain to enhance the quality of responses.

```
You are an assistant trained to provide answers based on the context provided.
You are given several documents that may contain relevant information about the
user's query. Your task is to generate a detailed and accurate response using the
information from the documents.

Context:

{context}

Question:

{question}

Answer:
```

The model is specifically tasked with question-answering tasks.
It is instructed to use the provided context to answer the question. This means the model should not generate a response purely based on its internal knowledge, but instead should rely on external retrieved context (the documents pulled by the retriever).
If the model doesn't have enough information from the context, it is told to simply respond with "I don't know".
The response should be concise (no more than three sentences) and to the point.

3 – **Streamlit Framework:**

Streamlit is used to build the user interface (UI) that allows users to interact with the chatbot. It facilitates easy creation of web-based applications for real-time interaction with the model. It includes an input form for user queries and displays the chatbot's responses.

Streamlit's session state is used to store the chat history to keep track of the ongoing conversation between the user and the bot.