Create a Streamlit Application for Document Loading and Retrieval-Augmented Generation (RAG)

This application has a user-friendly interface designed to load documents from a pre-defined folder called “data”, process these documents to extract their content, including text contained within images within documents, and utilize a local LLM to create a Retrieval-Augmented Generation (RAG) solution. The main functionalities of the application include:

Document Loading:

At launch, the system will check to see if any files in the data folder are new or have changed. If there are files that are new or have changed, they are loaded into the persistent vector DB. Otherwise, the application loads normally. Documents in the data folder include PDF, PDF with images, TXT, CSV, DOC, DOCX, PPT, PPTX, HTM, and HTML.

When extracting content from files, text is split into manageable chunks and embedded using the nomic-embed-text library. The embeddings are stored in a FAISS vector store, which allows for efficient similarity searches.

The vector store is persisted to a file (FAISS.pkl) to avoid reprocessing documents every time the application is restarted. While running, the application also checks for new documents in the folder and updates the vector store hourly.

Query Processing:

The application integrates use ollama to handle user queries. Users can input their queries into the application, and the local LLM generates responses based on the document content stored in the vector store.

Results should also include citation to the documents used to generate the response.

The LLM interacts with the vector store to retrieve relevant document chunks and generate coherent, contextually accurate responses.

User Interface:

The interface is built with Streamlit, providing a simple and interactive way for users to upload additional documents, and to ask questions.

Users can see the status of document uploads, well as during loading into the vector DB.

Lastly, all application packages should be compatible with Python 3.12.4, and dependencies installed using conda forge.