Create a web-based application for a RAG system. The application should support multiple users and only accessible via login and password.

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 cloud based 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 in a vector store, which allows for efficient similarity searches.

The vector store is persisted to a file or database to avoid reprocessing documents every time the application is restarted. The application also checks the data store for new documents in the folder and updates the vector store every 15 minutes.

Query Processing:

The application should be configurable to choose which LLM service to use (eg: OpenAPI, Anthropic or Google) to fulfill user queries.

Users should be able to prompt the LLM to ask questions about the data in the data store.

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

User Interface:

The interface should be built using a modern and robust web framework, providing a simple and interactive way for users to upload additional documents, and to ask questions.

When uploading documents, users shoud be able to see the status of document uploads.

Lastly, all application packages should be compatible with Python 3.11.