Computer Hub Assignment

Approach:

Ollama is an open-source project that is a powerful and user-friendly platform for running LLMs on your local machine. My approach towards this assignment is very open source, I have not used any paid APIs such as OpenAI, Replicate, etc.
Every development in this project follows the developer's workflow. I have developed a Flask API that has two endpoints, one endpoint is to upload the PDF(Context) and the other endpoint acts as a chatbot that is used to query the PDF and give answers based on the PDF. I developed this API using Retrieval Augmented Generation (RAG) architecture.

Frameworks, Libraries, Tools:

Python solution with llama3, LangChain, Ollama, and ChromaDB in a Flask API-based solution. API testing can be done with the help of Postman, Insomnia, etc.

Problem Faced while development:

Whenever I used to find example solutions of RAG implementation, most of the solutions are deployed on Google Colab, which restricts us from working on our local environment. Even though it provides us with GPUs, we cannot develop a production-level application out of the implementation on Google Colab. There were lots of APIs involved which were paid. Working on Google Colab notebooks does not let us control the latency and throughput. Even though there are easy-to-use, paid API options available it acts as a disadvantage to the openness of the software. As of complete open-source application, there is a 5-6 seconds of response time.

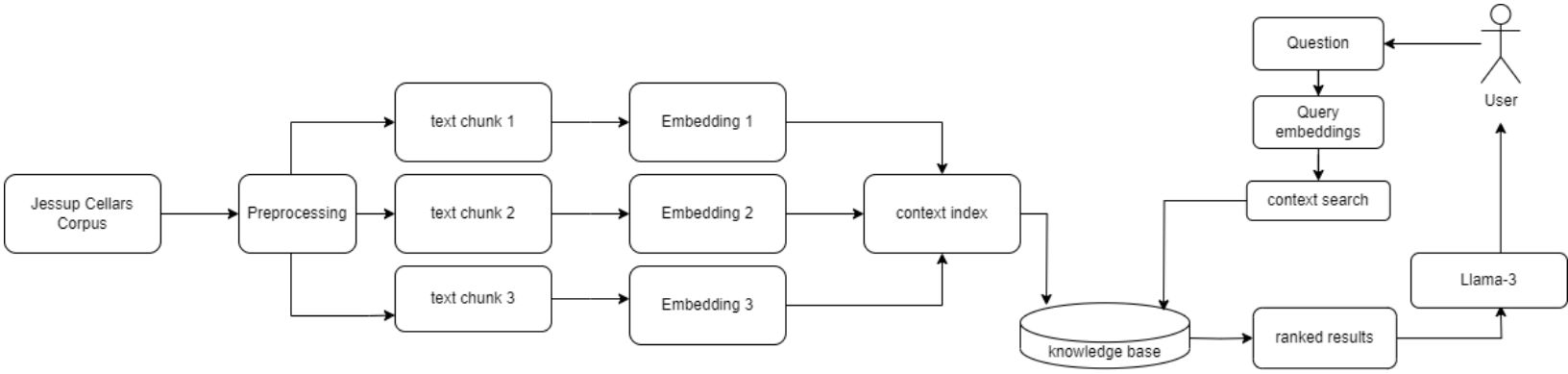
Solution to the problems:

Avoided Google Colab usage by developing very basic FlaskAPI. Used all open-source libraries and tools so that we are not reliant upon any kind of third party.

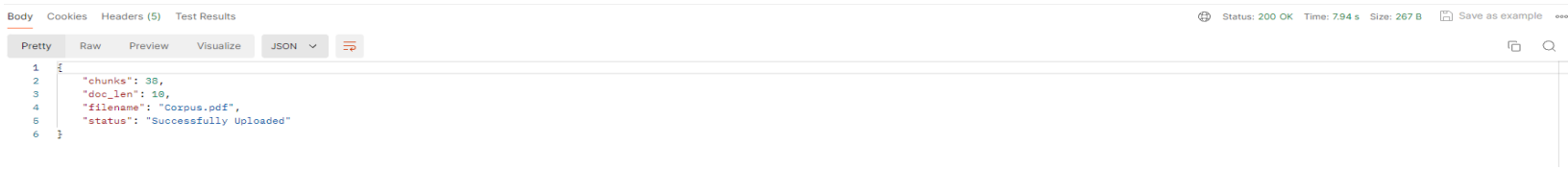
Future Scope:

One of the future scopes that might be very fascinating is adding a speech-to-text (and vice versa feature) which will be even more interactive and easy to use for the end user. One can interact with the chatbot as if they are having normal verbal communication. We can use, GCP and AWS cloud platforms and their ML as a Service to boost our scalability and userbase. We can smartly filter user queries and pass them to different AI Agents and make a chain out of each Agent. For example, one Agent is focused on Mathematical Queries, and the other Agent is focused on Content Generation, etc.

Architecture



Uploading the PDF



Context-Related Input

Incorrect Input