ALEMENO- Assignment for Internship – AI/ML

B V Soma Adithya

B Tech 4th Year (AI Branch)

Mahindra University

bvsomaadithya@gmail.com

Tech Stack used:

Langchain (For developing of the RAG)

ChromaDB (Storing of the document embeddings)

Streamlit (Used for developing a simple interface)

Ollama (Used for running the model locally)

LLM Model: llama3.2:3b-instruct-q5_K_S (from Ollama)

I chose the Llama 3.2:3b-instruct-q5_K_S model from Ollama because I have experience with Llama 3, and I have also tried other models. This one provided decent results and is relatively the latest. Instruct models are better suited for chatbots and query answering, and the 5-bit quantization is the highest bit model that can run on my laptop. I also tried the 6-bit quantized version but encountered a memory error.

Embedding Model: nomic-embed-text (from Ollama)

I used the *nomic-embed-text* model from Ollama, which is one of the most widely used and highly regarded embedding models. It offers 16-bit quantization, providing a good balance of performance and efficiency.

```
C:\Users\somaa>ollama list
NAME                            ID              SIZE      MODIFIED
nomic-embed-text:latest         0a109f422b47    274 MB    12 hours ago
llama3.2:3b-instruct-q5_K_S     97ef2f873c2c    2.3 GB    20 hours ago
llama2:latest                   78e26419b446    3.8 GB    27 hours ago
phi3:latest                     4f2222927938    2.2 GB    4 days ago
```

Local_Embeddings Notebook :

In this Python notebook, I created the embeddings and stored them locally on my computer.
I used semantic chunker from langchain which chunks sentence wise and add sentences if they are semantically similar. It took around 5 hours to convert the 3 pdfs into vector embeddings.

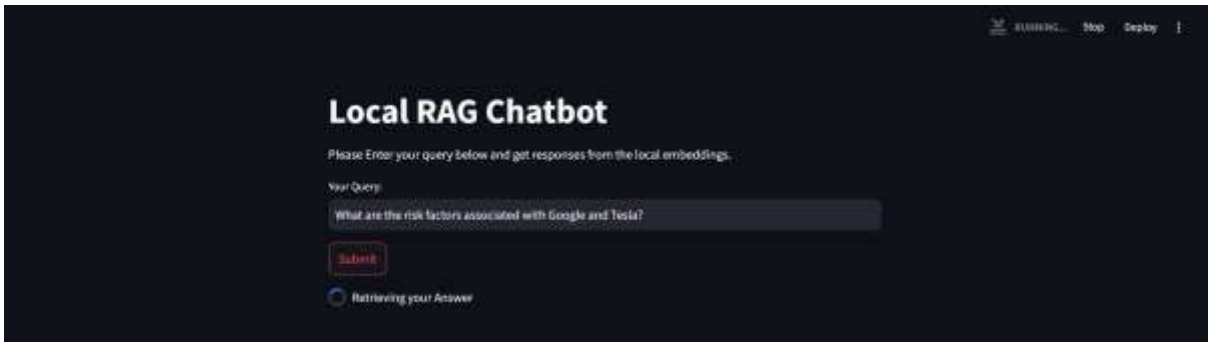
Query_Handling Notebook:

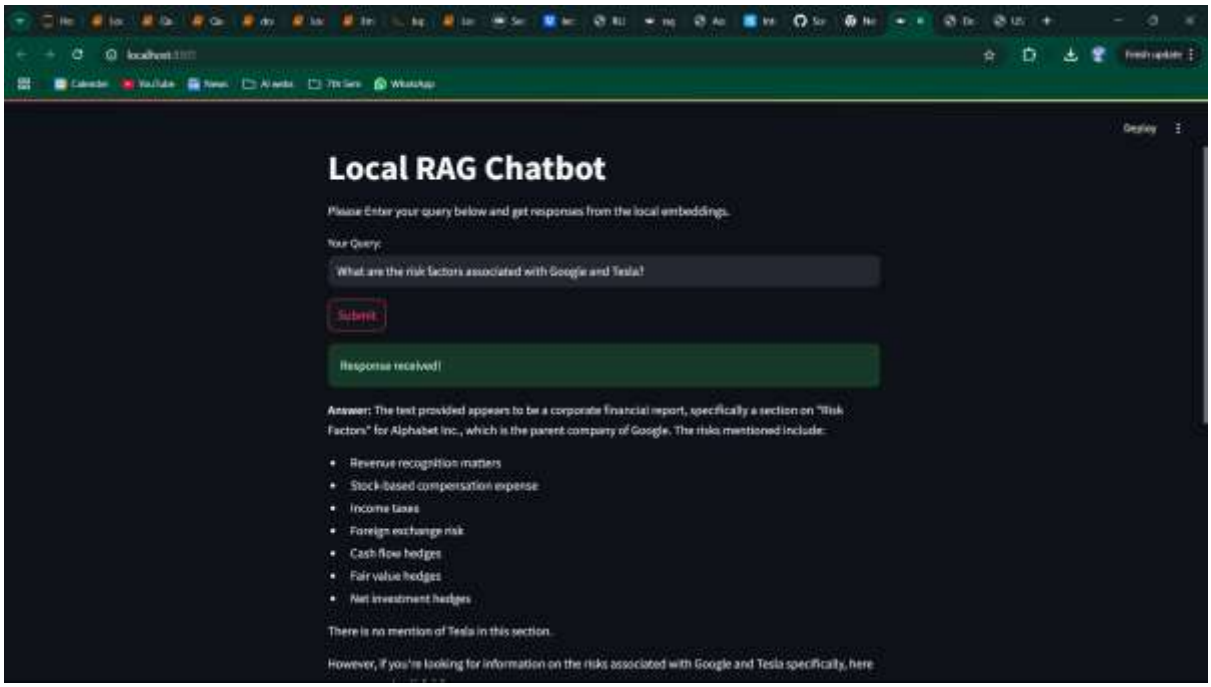
In this notebook, I took the query from the user and using retrieved the similar embeddings from local vectors. It is then passed onto the llama 3.2 and retrieved the answer. Below is an image of the response.

```
[10]: queries = [
          "What are the differences in the business of Tesla and Uber?",
      ]

[11]: for q in queries:
          print(f"Query: {q}")
          answer = query(q)
          print(f"Answer: {answer}")
          print("-" * 50)

Query: What are the differences in the business of Tesla and Uber?

llamaEmbeddings: 100%|██████████| 1/1 [00:02<00:00,  2.10 s/it]
Answer: A question about two well-known companies!

Tesla and Uber are both tech giants in their respective industries, but they have distinct business models and focuses. Here are some key differences:

**1. Products/Services:**
        * Tesla: Electric vehicles (EVs), clean energy solutions, solar power systems, and energy storage products. They aim to accelerate the world's transition to sustainable energy.
        * Uber: Ride-hailing and transportation network services (TNS). They connect passengers with drivers for on-demand rides, food delivery, and other transportation needs.

**2. Business Model:**
        * Tesla: Focuses on manufacturing and selling EVs, with a strategy of disrupting the automotive industry by offering high-performance, sustainable vehicles.
        * Uber: Operates as a platform that connects riders with drivers through an app, generating revenue primarily from commission-based fares.

**3. Revenue Streams:**
        * Tesla: Primarily generates revenue from vehicle sales, energy storage products, and solar panel installations.
        * Uber: Generates revenue mainly from ride-hailing services (fares) and food delivery fees.

**4. Scale and Reach:**
        * Tesla: Focuses on expanding its EV product lineup and building a strong brand presence globally, with plans to become the leading player in the automotive industry.
        * Uber: Has a vast network of drivers and riders worldwide, operating in over 10,000 cities across more than 69 countries.

**5. Market Size and Competition:**
        * Tesla: Enjoys significant market share in the EV market, but faces intense competition from established automakers and new entrants like Rivian and Lucid Motors.
        * Uber: Has become a dominant player in the TNS space, with a massive user base, but is facing increasing competition from rival ride-hailing companies like Lyft and Didi Chuxing.

**6. Financials and Valuations:**
        * Tesla: Listed on the NASDAQ stock exchange, with a market capitalization of over $1 trillion, making it one of the most valuable automakers in the world.
        * Uber: Listed on the New York Stock Exchange (NYSE), with a market capitalization of around $80 billion.

**7. Technology Focus:**
        * Tesla: Focused on developing advanced Autopilot technology and building a robust ecosystem for electric vehicles, including Supercharger networks and energy storage products.
        * Uber: Emphasizes leveraging AI, machine learning, and data analytics to improve the ride-hailing experience, enhance safety, and expand its services.

In summary, while both companies are tech-driven and innovative, Tesla focuses on disrupting the automotive industry with EVs and clean energy solutions, whereas Uber operates as a TNS platform connecting passengers with drivers.
--------------------------------------------------
```

rag_locally.py:

This file has the code for frontend part of the assignment. Using the Streamlit Library, I developed an simple user interface as below.





Future work:

Problems in my assignment is the high latency during response generation. The response time currently averages around 3 minutes, which needs to be addressed for improved performance.

Reasons for the Latency:

Hardware Limitations: My laptop, an HP Pavilion with an Intel i5 CPU and Intel Iris graphics card, is relatively slow for processing large models and handling heavy computational tasks.

I considered adding an option for users to upload PDFs directly through the interface. However, converting the PDFs into embeddings and storing them locally would take several hours. Given this significant processing time, I decided to provide users with the option to query pre-existing embeddings instead, ensuring a quicker and more efficient experience.