Project 1 - RAG System

1. **Platform**: Lang Graph, Weaviate, Docker, Ollama, Python
   Due to the simplistic nature of the requirements, Lang Chain and Lang Graph were able to provide adequate APIs to implement seamlessly without the need to climb a steep learning curve on other agentic frameworks.

   Weaviate was chosen for the RAG implementation because of its integrated implementation of Search and Storage of documents in one platform.
   Ollama enables local hosting of several generative and embedding models, preventing the need for API calls across the network that can lead to performance and cost issues. Hence, a quick choice was made for speedy experimentation and development.

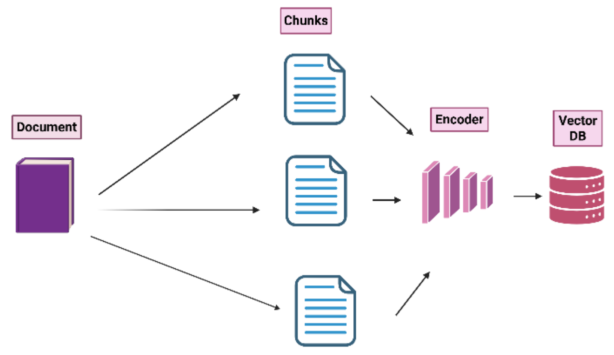

Figure 1: Document chunking, Embedding, and Storage in DB

2. **Chunking**
   As per some quick statistics, each document is sized around 500 words, and due to the diversity of content of news items containing information about events across time, splitting the document simply based on size may work without spending much effort. A small document size of 100 words was chosen to achieve high precision, with the idea that recall can be tuned by controlling the number of documents to be used in context. The modular flow is depicted in Figure 1.

3. **Models**
   Models : all-minilm, llama3.2
   The MiniLM BERT embedding model was chosen for its size and speed on large datasets on a personal laptop, as I did not procure larger infrastructure until Project 2. A smaller model, llama3.2, was used on limited desktop RAM.

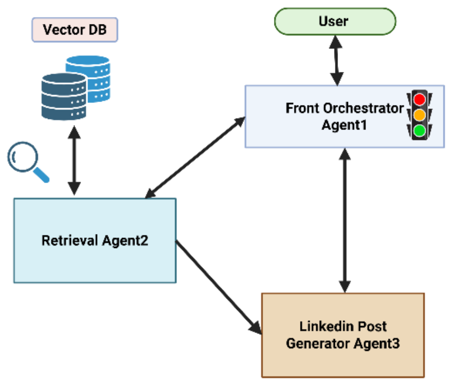

Figure 2: Agentic Architecture

4. **Orchestration**
   Agent Orchestration is highly basic and limited, and is based on the presence of keywords in the user input. It is depicted in Figure 2.

5. **Scopes of Improvements**
   A larger model with advanced prompt templates, along with a few-shot examples, may help achieve dynamic routing. More Chunking strategies and larger embeddings need to be tried for the RAG system. RAG needs to be tested with Benchmarks for Information Extraction and Retrieval **(**BIER**)** for precision and recall.