**JUNIOR AI ENGINEER ASSIGNMENT**

**Fine-Tuned RAG Chatbot with Streaming Responses – Amlgo Labs**

**NAME – NILANSH KUMAR SINGH**

**MAIL TO –** [**s.nilansh07@gmail.com**](mailto:s.nilansh07@gmail.com)

**Let’s First Understand the Task :-**

Before Starting first let’s understand what we have to do so that we can understand more and proceeded forward for better result.

Lests understand this step by step: -

Step: 01 🡪 **Document Preprocessing & Chunking**

Given a long document (more than 10000 words)

Task: - 1. Clean it (Remove headers/footers, HTML tags)

2. Break it into small pieces (chunks).

Save them 🡪 { /chunks }

Step: 02 🡪 **Generate Embeddings & Save in Vector Database**

Use an embedding model like

1. all-MiniLM
2. bge-small-en

These models convert each chunk into numerical vector

Task: - 1. first store all those vector in a vector database like faiss chroma or quadrant

* It helps in fast searching when user ask a question
* save the vector DB in { /Vector DB }

**Step 3: 🡪 RAG Pipeline (Retriever + Generator)**

Build a system where:

1. Retriever:

* Takes the user’s query.
* Searches for the most relevant chunks using the vector database.

1. Generator (LLM):

* Combines the retrieved chunks + the user question into a custom prompt.
* Sends it to an LLM like mistral-7b-instruct, zephyr-7b, or llama-3.
* The model generates a response, based only on the document (grounded response).

**Step 4: Add Prompt Template**

* Create a smart prompt template

Eg:

Question Reterived\_Chunks Answer

(User Query)

It will make LLM to provide Response based on the Question

**Step 5: Streamlit Interface (Frontend Chatbot)**

* **Streamlit app** (app.py)
* A **chat box** where users can type their questions.
* The model replies **token-by-token or sentence-by-sentence** (real-time feel).
* It also shows:
  + The **source chunks** used in the answer.
  + In the sidebar or footer:
    - Model being used.
    - Number of chunks/documents indexed.
* Also add a **Reset/Clear chat** button.

**Approach: -**

1. Split the file into small pieces (chunks) and save them.
2. Convert chunks into number (Called Vector), So that AI can understand.
3. Store chunks + their Vector into special searchable library called vector database like FAISS.
4. Now when someone ask question we convert the question to number and then match it with relevant chunks in our data.
5. Now after this our first task is done
   1. we have
      1. Chunks in vector form
      2. Question in vector form
      3. Result we got like relevant vector with respect to the question asked.

So, we send these relevant chunks and questions which is in vector form to smart brain called LLM large language model.

* Now we tell our LLM with prompt that “Please answer the question but only with the chunks I gave you”.
* AI write the answer based on the document info.

1. Make a chat app (Streamlit)
   1. Chatbot interface - chat box
   2. It replies live word by word
   3. Also give the reference

**🡪 Full pipeline: -**

1. Chunking = Break document into small parts.
2. Embedding = Convert each part into number.
3. Retrieval = Find most relevant part of the question.
4. Generation = Ask LLM to answer based on the info given.
5. Streamlit = Show all this on chat window.