**Documentation: Retrieval-Augmented Generation (RAG) System**

**Introduction**  
This system extracts text from a PDF, generates embeddings, stores them in ChromaDB, and retrieves relevant chunks to answer user queries.Get started  
1. Setup  
Ensure you have Python installed along with the required dependencies. Install the necessary packages using:  
pip install chromadb cohere pdfplumber python-dotenv2. System flow

* **Start the script:** Run the script using:
* python Assign.py
* **Provide a PDF file path** when prompted.
* **Ask questions** based on the document’s content.
* **Receive AI-generated answers** using retrieved document chunks.

3. Example  
Enter the path to your PDF file: sample.pdf  
Total chunks created: 10  
Enter your question: What is the summary of chapter 1?  
Answer:answer based on contentTechnologies  
1. PDF Text Extraction

* **Library Used:** pdfplumber
* **Method:** extract\_text\_from\_pdf() reads and extracts text from all pages in a PDF.
* **Alternative:** PyPDF2.

2. Split text to chunks

* **Method:** split\_text\_pdf() splits the extracted text into chunks using paragraph breaks and sentence-ending punctuation. You can split text on the basis of no. of pages also.
* **Alternative:** LlamaIndex, LangChain

3. Embedding Generation

* **Model Used:** Cohere Embed English v3.0
* **Method:** get\_embeddings(chunks) generates dense vector representations for text chunks.
* **Alternative:** OpenAI’s text-embedding-ada-002, text-embedding-3-small

4. Vector Database (ChromaDB)

* **Method:** store\_embeddings(text) stores chunk embeddings into ChromaDB.
* **Alternative:** FAISS, Pinecone for scalable vector storage.

5. Retrieval

* **Method:** get\_chunks(query, top\_k=3) retrieves top-K similar chunks using vector similarity search.

6. Answer Generation

* **Model Used:** Cohere command-xlarge-nightly
* **Method:** get\_answer(context, query) uses a prompt to generate an AI-based response.
* **Alternative:** OpenAI’s GPT-4, GPT-4o mini.

Challenges & Resolutions

* **Chunks generation**: Chunk size and the no.of documents stored.
* **Unique id**: With uuid, the process happened to get slower than usual

**TODO:**

* Implement LlamaIndex
* Using Openai

Conclusion  
This RAG system allows users to query a PDF document and receive AI-generated responses based on retrieved chunks. This system is using Cohere’s embedding and LLM models, along with ChromaDB for vector storage