There was some ambiguity about whether the PDF should be pre-existing or uploaded by the user. I chose to implement user upload, as it’s more flexible and realistic for production use. Although slightly more complex, it required only 2–3 extra lines of code to implement pre-existing way.

Scope for Improvement

Chunk Storage Format
I used `.txt` files for simplicity, but it's not ideal. Switching to **JSON** or **SQLite** would enable better structure and easier integration.

Virtual Environment
I missed using a virtual environment, which led to unnecessary packages in `requirements.txt`. Using `venv` or `conda` would help manage dependencies cleanly.

Using LangChain Chains
I wanted to use LangChain’s `RetrievalQA` chain but didn’t due to time and file complexity. I plan to refactor and integrate this soon.

Model Speed
Local models via Ollama were slow. Paid APIs like OpenAI would offer better speed and reliability for production use.

Embedding Quality
Current embeddings work, but **OpenAI embeddings** provide better semantic understanding. I aim to switch for improved results.

1. Document Structure and Chunking Logic

The RAG Chatbot is designed to answer questions about the content of any PDF document you upload. When a PDF is uploaded, the system first processes the document by splitting it into smaller, manageable pieces called “chunks.” This is done using the following logic:
The PDF is loaded and converted into text.
The text is split into overlapping chunks, each containing up to 1000 characters, with a 200-character overlap between consecutive chunks.
Each chunk is saved as a separate text file for further processing.
This chunking approach ensures that each piece of information is small enough for efficient retrieval, while the overlap helps preserve context that might otherwise be lost at chunk boundaries.

2. Embedding Model and Vector Database

After chunking, each text chunk is converted into a numerical vector (embedding) using a pre-trained model:
Embedding Model:
The system uses the **all-MiniLM-L6-v2** model from HuggingFace. This model is efficient and provides high-quality semantic embeddings, meaning it can capture the meaning of text in a way that is useful for finding similar content.
Vector Database:

All embeddings are stored in a **FAISS (Facebook AI Similarity Search)** vector database. FAISS allows for fast and accurate similarity searches, making it possible to quickly find the most relevant chunks for any user query.

3. Prompt Format and Generation Logic

When a user asks a question, the following steps occur:

The system searches the FAISS database to find the top 10 most relevant chunks based on the user's query.

These chunks are combined and used as the "context" for the language model.

The prompt given to the language model (Llama 3.1 via Ollama) is structured as follows:

You are a helpful assistant.
Answer ONLY from the provided context.
If the context is insufficient, just say you don't know.

The model generates an answer using only the information from the retrieved chunks, which helps reduce the risk of hallucination.

4.Example queries with responsesotes

I)explain about aiver of Jury Trial. success

According to Section 2 of the User Agreement, you and eBay have waived your constitutional and statutory rights to sue in court and have a trial in front of a judge or jury. This means that all disputes shall be resolved by arbitration under this Agreement to Arbitrate, instead of going through a traditional court process with a judge or jury. This waiver is stated as follows:

"YOU AND EBAY HEREBY WAIVE ANY CONSTITUTIONAL AND STATUTORY RIGHTS TO SUE IN COURT AND HAVE A TRIAL IN FRONT OF A JUDGE OR A JURY. You and eBay are instead electing that all Disputes shall be resolved by arbitration under this Agreement to Arbitrate, except as specified in Section 19.B ("Agreement to Arbitrate")."

2) Tell me about ebay.success

According to the provided context, here's what I can tell you about eBay: eBay is a marketplace that allows users to offer, sell, and buy goods and services in various geographic locations using a variety of pricing formats. eBay is not a party to contracts for sale between third-party sellers and buyers, nor is it a traditional auctioneer. eBay provides guidance as part of its Services, such as pricing, shipping, listing, and sourcing, but this is solely informational and users may decide to follow it or not. eBay uses artificial intelligence or AI-powered tools and products to provide and improve its Services. eBay has various offices around the world, depending on the user's location (e.g., eBay GmbH in Germany for European Union residents). Users agree that any reliance on vehicle information on eBay sites is at their own risk. That's a summary of what I can find about eBay from the provided context. Let me know if you have specific questions or if there's anything else I can help with!

3) Please explain Fees and Taxes success

According to Section 6 of the User Agreement, eBay charges sellers for the use of their Services. The fees they charge are listed on the "Selling fees" pages. They may change these fees from time to time by posting the changes on the eBay site 14 days in advance, but with no advance notice required for temporary promotions or any changes that result in a reduction of fees. Additionally, if eBay issues a refund to a buyer on behalf of a seller, they may charge the seller for the amount of the refund. They may also charge sellers for the cost of return shipping labels and/or other reasonable fees when certain conditions are met.

5. Notes on Hallucinations, Model Limitations, and Response Speed

Hallucinations:

While the system is designed to answer only from the provided context, language models can sometimes generate information that is not present in the source document. The prompt explicitly instructs the model to avoid this, but it is not foolproof.

Model Limitations:

The quality of answers depends on the quality and relevance of the retrieved chunks. If the answer is not present in the document, the model should say “I don’t know.”

The system currently processes one PDF at a time and does not support multi-document retrieval.

The model may struggle with very large or highly technical documents if relevant information is split across many chunks.

Response Speed:

The first query after uploading a PDF may take longer, as the system needs to process the document, create embeddings, and build the vector database.

Subsequent queries are faster, as only retrieval and answer generation are required.

The app shows a loading spinner while processing, but does not display a word-by-word streaming response.

This report summarizes the core logic and limitations of the RAG Chatbot system. For further details, please refer to the project’s README or source code.