

# RAG-Based PDF Question Answering System

## (Mistral + Ollama + FAISS)

### Abstract:

A lightweight, fully local Retrieval-Augmented Generation (RAG) system built using **Mistral** (**Ollama**) for embeddings and text generation, **FAISS** for vector search, and **Python** for PDF processing.

This project allows users to upload a PDF, generate embeddings, store them in a vector database, and then ask natural language questions related to the document.

---

### What is RAG?

**Retrieval-Augmented Generation (RAG)** combines a Language Model (LLM) with an external knowledge source.

Instead of the LLM trying to “remember” everything, it **retrieves relevant chunks** from the PDF and uses them to generate precise, contextual answers.

In simple words:

 *The LLM becomes smarter by reading your PDF before answering.*

---

### Tech Stack

Component	Technology
Embeddings	<b>Mistral via Ollama</b>
LLM for answering	<b>Mistral via Ollama (chat mode)</b>
Vector Database	<b>FAISS (IndexFlatL2)</b>
PDF Extraction	<b>PyPDF</b>
Storage	vectors.index + chunks.pkl
Programming Language	<b>Python</b>

## How the System Works

### **1 PDF → Text**

The system extracts text from every page using **PyPDF**.

### **2 Text → Chunks**

The text is split into **500-character chunks**.

This makes embedding + search efficient.

### **3 Chunk Embeddings (via Ollama Mistral)**

Each chunk is converted into a vector using:

```
ollama.embeddings(model="mistral", prompt=chunk)
```

All vectors are stored in **FAISS** and saved as:

- `vectors.index` → FAISS index
- `chunks.pkl` → text chunks + metadata + total pages
- 

### **4 Asking a Question**

When the user asks a question:

1. The question is embedded (same Mistral embedding model)
2. FAISS retrieves the **top 3** most relevant chunks
3. These chunks are combined into a context
4. Mistral (chat mode) generates the final answer using:

```
ollama.chat(model="mistral", messages=[...])
```

### **5 Final Answer**

The user gets an answer **grounded in the PDF**, not hallucinated.

## File Structure

```
rag/
|— pdf-vector.py      # Convert PDF → chunks → embeddings → FAISS
|— question-vector.py # RAG querying system
|— vectors.index      # FAISS index file
|— chunks.pkl         # Chunk text + metadata
|— ragmist/           # Python virtual environment
```

---

## Step-by-Step Usage

### 1. Generate Vector Database from PDF

Set your PDF name inside pdf-vector.py, then run:

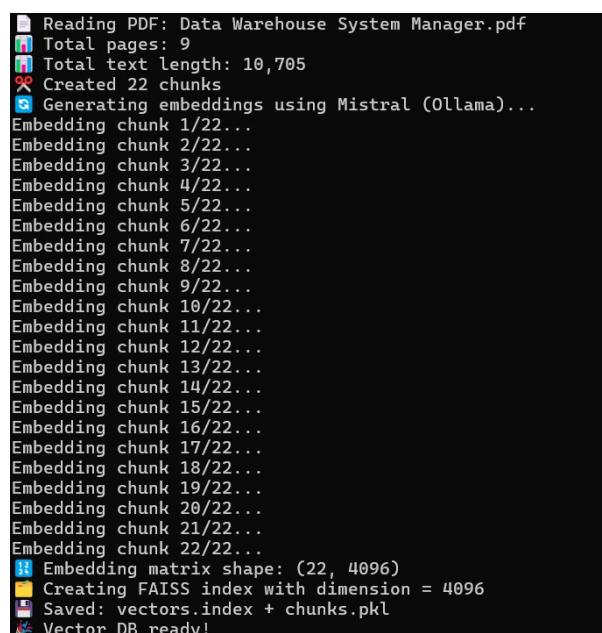
```
python pdf-vector.py
```

This creates:

- vectors.index
- chunks.pkl

[ create the embedding for the pdf by splitting into chunks and store the metadata in the chunk.pkl, and store the vector embedding in the FAISS database. ]

### Snapshot for the embedding:



A terminal window showing the execution of the `pdf-vector.py` script. The output indicates the script is reading a PDF, determining it has 9 pages, and creating 22 chunks. It then generates embeddings using the Mistral (Ollama) model for each chunk, reporting progress from 1/22 to 22/22. Finally, it creates a FAISS index with a dimension of 4096 and saves the vectors and chunks.

```
█ Reading PDF: Data Warehouse System Manager.pdf
█ Total pages: 9
█ Total text length: 10,705
█ Created 22 chunks
█ Generating embeddings using Mistral (Ollama)...
Embedding chunk 1/22...
Embedding chunk 2/22...
Embedding chunk 3/22...
Embedding chunk 4/22...
Embedding chunk 5/22...
Embedding chunk 6/22...
Embedding chunk 7/22...
Embedding chunk 8/22...
Embedding chunk 9/22...
Embedding chunk 10/22...
Embedding chunk 11/22...
Embedding chunk 12/22...
Embedding chunk 13/22...
Embedding chunk 14/22...
Embedding chunk 15/22...
Embedding chunk 16/22...
Embedding chunk 17/22...
Embedding chunk 18/22...
Embedding chunk 19/22...
Embedding chunk 20/22...
Embedding chunk 21/22...
Embedding chunk 22/22...
█ Embedding matrix shape: (22, 4096)
█ Creating FAISS index with dimension = 4096
█ Saved: vectors.index + chunks.pkl
█ Vector DB ready!
```

## 2. Ask Questions from the PDF

python question-vector.py

Example prompts:

"Explain page 4"

"What is data warehousing?"

"Define OLAP operations"

[ creates embedding for the question and retrieves 3 most relevant chunks from the vectorDB and send it to the Ollama mistral LLM to get the relevant data ]

### Snapshot of the output:

```
Big System Ready. Ask me questions about your PDF.
Type bye', quit, exit' or q' to exit
Type lme' to see database statistics
=====
? Your question: who is the author of the book?
Searching and generating answer...
Found 3 relevant chunks:
Chunk 1: Score 0.954, 501 (Page 1)
Chunk 2: Score 0.711, 386 (Page 4)
Chunk 3: Score 0.696, 344 (Page 12)

? Answer: The document provided does not contain information about a book, so it's impossible to determine the author from this context.

? Your question: what does the pdf cover?
Searching and generating answer...
Found 3 relevant chunks:
Chunk 1: Score 0.954, 501 (Page 6)
Chunk 2: Score 0.885, 496 (Page 18)
Chunk 3: Score 0.876, 531 (Page 1)

? Answer: The document does not provide information about the author. However, it appears to be related to a technical guide or manual about Data Warehouse System Management, as suggested by the title "WMT-5 Data Warehouse System Manager" on page 1.

? Your question: what does the pdf cover?
Searching and generating answer...
Found 3 relevant chunks:
Chunk 1: Score 0.952, 321 (Page 12)
Chunk 2: Score 0.952, 321 (Page 12)
Chunk 3: Score 0.952, 321 (Page 12)

? Answer: The PDF appears to cover various IT management solutions, including System Configuration Manager, System Scheduling, and Monitoring & Diagnostics. These topics are discussed in terms of their roles in ensuring optimal performance, maintaining downtime, enhancing security, managing large-scale data, handling complex queries, continuously monitoring the system for issues proactively detecting and resolving problems, maintaining scalability, and maintaining reliability. Additionally, the document emphasizes the importance of these solutions in meeting regulatory requirements for data retention and disaster recovery, as well as providing peace of mind by offering confidence that data can be retrieved even in worst-case scenarios. (References: Page 21, Pages 3-4)

? Your question: explain page 15
Searching and generating answer...
Found 3 relevant chunks:
Chunk 1: Score 0.775, 694 (Page 11)
Chunk 2: Score 0.656, 578 (Page 4)
Chunk 3: Score 0.668, 411 (Page 7)

? Answer: I'm sorry for any confusion, but there seems to be a mistake in the document reference as there is no Page 15 mentioned in the provided context. The context only includes Page 21, 4, and 7, related to System Configuration Manager and System Scheduling. Manager. If you have a separate document or additional information about Page 15, I'd be happy to help explain it based on that.

? Your question: explain page 21
Searching and generating answer...
Found 3 relevant chunks:
Chunk 1: Score 0.725, 369 (Page 21)
Chunk 2: Score 0.656, 541 (Page 4)
Chunk 3: Score 0.670, 461 (Page 7)

? Answer: Page 21 discusses the benefits of systems or services that support compliance by helping meet regulatory requirements for data retention and disaster recovery. This means that such a system will ensure that businesses can retain their data for the required period as dictated by regulations, while also having the ability to recover data in worst-case scenarios, thereby providing peace of mind.
```

## Challenges Faced

- Mistral embeddings occasionally fail for empty chunks
  - Folder paths must be accurate when loading vectors.index and chunks.pkl
  - Chunk/page estimation isn't perfect, but works well for general PDFs
  - Ollama must be running locally for embeddings + chat generation
- 

## Why Mistral + Ollama?

- Fully offline
  - Fast embedding generation
  - Great quality for answering
  - No API keys, no cloud billing
  - Perfect for student projects & local experimentation
- 

## Outcome

You now have a complete, local, efficient **PDF Question Answering System** using:

- **Mistral for embeddings & LLM answers**
- **FAISS for similarity search**
- **Python for preprocessing**

A simple but powerful RAG pipeline that works end-to-end.