Introduction

Students, developers, and professionals often depend on complex technical documents such as engineering manuals, research guides, API documentation, and college syllabi. Navigating and extracting answers from such large texts can be slow and frustrating. While tools like ChatGPT provide general assistance, they cannot access or reason over private documents. This project proposes "DocuBot AI", a Retrieval-Augmented Generation (RAG)-based chatbot that allows faculty and students to interact with custom documents via natural language questions, using a locally deployed language model and hybrid search techniques for high accuracy and privacy.

Problem definition

Existing solutions often fall short in several key areas. They struggle to comprehend domain-specific academic or technical language, cannot access private or proprietary documents, and pose privacy concerns—especially when handling sensitive information. Additionally, most rely heavily on constant internet and cloud usage, which may not always be feasible or secure. *DocuBot AI* addresses these limitations by leveraging locally hosted large language models (LLMs) such as Mistral via Ollama. It allows faculty members to upload documents and enables students to query them at their level of understanding. To enhance the accuracy and relevance of search results, it also integrates a hybrid retrieval approach using FAISS combined with keyword-based methods for superior semantic search.

Objective of the Project

- To build a role-based document chatbot system (Faculty uploads, Students query).

- To support smart Q&A over academic syllabi, manuals, and internal reports.

- To use semantic search + keyword match reranking for better accuracy.
- To deploy offline LLMs (like Mistral) using Ollama to preserve privacy.
- .To offer an intuitive interface using Streamlit or React + FastAPI.

Hardware and Software requirements:

Software Requirements

- Backend: Python, LangChain, FastAPI
- Database: FAISS for vector storage
- Frontend: Streamlit / React.js
- PDF Parsing: PyPDF2, Tesseract (OCR)
- LLM: Mistral 7B (via Ollama)
- Embeddings: SentenceTransformers (MiniLM)
- Git & GitHub for version control.

Hardware Requirements

- 8-16 GB RAM, i5 or higher CPU
- OS: Windows/Linux/macOS
- Internet required only during initial setup

Project Outcomes

- Fast answers to complex, document-based questions
- Fully offline chatbot with no cloud dependency
- Accurate semantic search using FAISS + keywords
- Scalable system: supports more files, more roles
- Smart assistant for academia and industries alikee

USN	NAME	SIGN
2JR22CS006	Aishwarya V Balaraddi	
2JR22CS013	Altaf Shekali	
2JR22CS027	Basavaraj Chippalakatti	
2JR22CS046	Monali kedar	

Signature

(Prof. Sukhada)