

EduMedBot - RAG-Based AI Chatbot (README)

Project Title

EduMedBot - RAG-Based AI Chatbot for Understanding PDFs

Description

EduMedBot is an AI-powered chatbot that allows users to ask questions based on content from research papers, class notes, and medical reports. It uses a RAG (Retrieval-Augmented Generation) approach with FAISS for vector search and Groq/Cohere APIs for responses.

Live Demo

Streamlit App: <https://your-streamlit-link-here>

Features

- Upload any PDF
- RAG-based response using vector search
- Two modes: AI-explained and direct PDF match
- Chat interface with memory
- Expandable source chunks

Tech Stack

- LangChain
- FAISS
- Groq API
- Cohere API
- Streamlit
- pdfplumber
- python-dotenv

Project Structure

```
pdf-chatbot/  
app.py          # Streamlit UI  
rag_pipeline.py # Embedding, retrieval, RAG query logic  
embedding_api.py # Embedding via Cohere  
groq_llm.py     # LLM wrapper (Groq API)  
preprocess.py   # Chunking and PDF text extraction  
vector_store.py # FAISS-based vector store  
requirements.txt # Python dependencies  
.env or secrets.toml # API keys for local/Streamlit Cloud
```

API Keys Required

- COHERE_API_KEY from Cohere dashboard
- GROQ_API_KEY from Groq API

EduMedBot - RAG-Based AI Chatbot (README)

Set them in ``.env`` (local) or ``.secrets.toml`` (Streamlit Cloud)

How to Run Locally

1. Clone the repo: `git clone https://github.com/NaveenReddy2004/pdf-chatbot.git`
2. Install dependencies: `pip install -r requirements.txt`
3. Create a ``.env`` file with your API keys
4. Run: `streamlit run app.py`

Streamlit Deployment

1. Push code to GitHub
2. Go to <https://share.streamlit.io>
3. Deploy your repo and set secrets in the dashboard

Use Cases

- Academic research assistant
- Class notes explainer
- Medical report chatbot

Author

Naveen Kumar Reddy

GitHub: <https://github.com/NaveenReddy2004>

LinkedIn: <https://linkedin.com/in/naveenreddy2004>

License

This project is open source under the MIT License.