**Local RAG Chatbot**

**Project Description**

This is an end-to-end Retrieval-Augmented Generation (RAG) system built with an all-local, open-source stack. The system can ingest PDFs, answer questions with citations, and provide a user-friendly chat interface. The primary goal of this project was to build a robust, modular, and verifiable RAG pipeline from scratch.

**Key Deliverables**

* **Ingestion Pipeline:** The system can process multiple PDF files, extracting text and metadata (page numbers) from each.
* **Chunking + Embeddings:** Documents are broken into semantically coherent chunks, which are then converted into numerical vector embeddings.
* **Vector Index:** A local, persistent vector store is maintained, with unique files for each processed PDF, allowing the system to handle multiple documents without mixing up data.
* **Retriever:** A retrieval component identifies the most relevant text chunks from the vector store in response to a user query.
* **Generator:** A local, open-source Large Language Model (LLM) is used to generate a concise and accurate answer based on the retrieved context.
* **Citations & Evidence:** The LLM is prompted to include page numbers in its answers, and the raw source text is provided as supporting evidence.
* **Demo UI:** A Streamlit web application provides a visual and interactive chat interface for a better user experience.

**Technical Stack**

* **PDF Processing:** PyPDF2
* **Embeddings:** sentence-transformers
* **Vector Store:** NumPy (for vector storage) and JSON (for chunk metadata)
* **Local LLM:** Ollama with Llama 3
* **Web UI:** Streamlit

**Setup and How to Run**

This guide assumes you have a recent version of Python and Git installed on your system.

**Step 1: Clone the Repository**

Start by getting the project files.

git clone <repository-url>

cd rag\_project

**Step 2: Set Up the Python Environment**

Create and activate a virtual environment, then install all the necessary libraries.

python -m venv venv

.\venv\Scripts\Activate.ps1

pip install PyPDF2 sentence-transformers streamlit ollama numpy

**Step 3: Install and Run Ollama**

Download and run the Ollama application. You can find the installer on the [Ollama website](https://ollama.com/). Once installed, open a separate terminal and run the Llama 3 model to ensure the server is active.

* + **ollama run llama3**

**Step 4: Run the Demo Application**

With the Ollama server running, you can now launch the Streamlit UI. This command will open a new tab in your web browser.

* + **streamlit run app.py**

**How to Use the App**

1. **Upload a PDF:** In the sidebar, use the file uploader to select a PDF document. The app will process the file, create embeddings, and store them locally.
2. **Ask Questions:** Once the file is processed, a chat input box will appear. Type your questions about the document.
3. **Get Answers:** The chatbot will provide an answer, along with citations and the raw supporting evidence from the PDF.