Product Requirements Document: Local RAG PDF Assistant

1. Executive Summary

The Local RAG PDF Assistant is a desktop-based application that allows users to upload PDF documents and query their content using Retrieval-Augmented Generation (RAG). Unlike cloud-based AI, this project focuses on **privacy, zero cost, and offline accessibility** by running all computations locally via Ollama.

2. Target Audience

- **Privacy-Conscious Professionals:** Users handling sensitive data (legal, medical, or corporate) that cannot be uploaded to the cloud.
- **Students/Researchers:** Users who need to query long research papers without subscription fees.
- **Developers:** Those looking for a template to build local-first AI applications.

3. Functional Requirements

3.1 Document Ingestion & Processing

- **PDF Support:** The system must extract text from uploaded PDF files using PyPDF2.
- **Smart Chunking:** Text must be broken into segments (default 1,000 characters) with a 200-character overlap to preserve context.
- **Vectorization:** Text chunks must be converted into numerical embeddings using the all-MiniLM-L6-v2 model.

3.2 Information Retrieval (The "R" in RAG)

- **Semantic Search:** Use **FAISS** to perform similarity searches between user queries and document chunks.
- **Adjustable Retrieval:** Users must be able to define the number of context chunks (top_k) sent to the LLM.

3.3 AI Generation

- **Local LLM Integration:** Connect to the **Ollama** API (running at localhost:11434) for text generation.
- **Grounded Responses:** The system must use a "system prompt" to ensure the AI only answers based on the provided context to prevent hallucinations.
- **Model Selection:** Users should be able to choose between installed models (e.g., Llama3, Mistral, Phi).

4. Technical Stack

- **Frontend:** Streamlit.
- **Backend:** Python 3.x.
- **Orchestration:** Ollama.
- **Vector Database:** FAISS (CPU version).
- **Embeddings:** Sentence-Transformers.

5. User Interface (UI) Requirements

- **Status Dashboard:** Real-time indicator showing if the Ollama server is active.
- **Two-Column Layout:** Left side for file management and settings; right side for the chat interface.
- **Source Transparency:** A "View Sources" section to show users which parts of the PDF were used to generate the answer.