Advanced RAG System for “Chat-with-PDF”

ChatPDF is a document-based conversational AI system that allows users to upload PDFs and ask questions about their content. The system maintains conversation context across interactions and provides source references. This design document outlines the architecture and key decisions.

Data Flow

1. Document Upload → Text Extraction → Chunking → Vector Embedding → Storage
2. User Question → Context Retrieval → LLM Processing → Response Generation
3. Conversation Pair → Vector Embedding → History Storage → Future Context

Core Components:

1. **Document Processing Pipeline**
    - Extracts text from PDFs while preserving file/page metadata
    - Splits text into chunks (1000 tokens) with overlap for context
    - Generates embeddings (Google Generative AI) and stores in FAISS vector DB
2. **Conversation Management**
    - **SessionManager**: Tracks active sessions, chat history, and system prompts in-memory
    - **ChatHistoryManager**: Stores conversation history in a separate FAISS index for semantic retrieval
3. **Question Answering Engine**
    - Retrieves relevant document chunks and prior chat history
    - Augments query with context and sends to Groq’s LLM (e.g., Llama3-70B)
    - Returns answer with source attribution (file + page)

Key Decisions:
Hybrid Context: Combines document and conversation vectors for accurate, continuous dialogue
Metadata Preservation: Tracks sources (file + page) for transparency
Modular Storage: FAISS indexes per session (documents + chat history)
Structured Prompts: Explicit sections for document context, chat history, and query

Scalability & Security:
Stateless API (FastAPI) + session isolation via UUIDs
Secure API key handling, CORS restrictions, and temp file cleanup

Tech Stack:

- **Backend**: FastAPI (Python)
- **Vector DB**: FAISS

- **Embeddings**: Google Generative AI
- **LLM**: Groq (Llama3, Mixtral)

Outcome:
A scalable, context-aware PDF chatbot with precise source attribution and seamless conversation flow.