DocIntel: LLM-based Document Analysis System - Visual Workflow

Phase 1	Document Upload & Text Extraction	Extract text using PyMuPDF or pdfplumber from PDF/DOCX files.
Phase 2	Text Chunking & Embedding Generation	Split extracted text and generate embeddings using Sentence Transformers (MiniLM).
Phase 3	Vector Storage (Chroma / FAISS)	Store embeddings locally for semantic search and retrieval.
Phase 4	LLM Integration (Ollama)	Use lightweight local model (Phi-3, Mistral, Llama-3) to summarize or answer queries.
Phase 5	Retrieval-Augmented Generation (RAG)	Retrieve relevant document chunks, combine with query, and generate a contextual answer.
Phase 6	Metadata Extraction (NER + Classification)	Use spaCy or BERT to extract names, dates, amounts, and classify document types.
Phase 7	Backend (FastAPI)	Expose APIs for document upload, query, and summary retrieval.
Phase 8	Frontend / Interface (React or Gradio)	Provide a user-friendly dashboard for document interaction and visualization.
Phase 9	Deployment / Demo	Deploy on Render or Hugging Face Spaces (or run locally for demo).

High-Level Architecture:

1. User uploads document → Extract text
2. Text is chunked → Converted to embeddings → Stored in Chroma
3. User asks query → Relevant chunks retrieved
4. RAG prompt constructed → Sent to LLM (Phi-3 / Llama-3)
5. LLM generates contextual answer / summary
6. FastAPI serves responses → Frontend displays result