AI-Powered Knowledge-Base Search Engine (RAG System)

Objective:

Create a system that accepts text/PDF documents, processes them using embeddings and a vector database, and answers user queries using Retrieval-Augmented Generation (RAG).

Features:

1. Document Management:

- Upload and manage PDF/TXT documents

- Extract text automatically

- Chunking + preprocessing

- Embeddings generation

- Vector store insertion (FAISS/Chroma/Pinecone)

2. Intelligent RAG Search:

- Retrieve top relevant document chunks

- RAG prompt applied to LLM

- Synthesized clear answer with citations

3. Modern Frontend:

- Clean UI using React/Next.js + TailwindCSS

- Document upload interface

- Search input page

- Display answers, sources, citations

4. Backend API (FastAPI / Node.js Express):

POST /upload - Upload docs, extract text, create embeddings

GET /documents - List uploaded docs

POST /query - RAG search + LLM answer

DELETE /document/id - Delete document & embeddings

5. LLM Integration:

- Works with GPT-4, Llama-3, Groq, etc.

- RAG prompt template included

6. Technical Expectations:

- Embeddings model: sentence-transformers/all-MiniLM-L6-v2

- Vector DB: FAISS / ChromaDB / Pinecone

- Backend: clean modular structure

- Frontend: responsive & attractive

7. Deliverables:

- Full code (frontend + backend)

- Architecture diagram

- RAG sequence diagram

- API documentation

- README

- Deployment Guide

- Demo script

8. Project Structure:

backend/

routes/

services/

rag/

utils/

main.py

frontend/

components/

pages/

styles/

Evaluation Focus:

- Retrieval accuracy

- Synthesis quality

- Code structure

- Clean UI

- Proper LLM integration