Heavy AI Project: Legal Document Analyzer

Problem It Solves:

Many people (non-lawyers, freelancers, startups) struggle to understand legal documents. They need summaries, simple explanations, red-flag detection, and a chat interface to ask questions about the document.

AI Agent Features:

- Upload & parse PDF
- RAG-based Q&A (Retrieval-Augmented Generation)
- Clause-wise summarization
- Red flag detection (non-compete, termination clauses)
- Multilingual summary
- Interactive chat with document

Tech Stack:

- Frontend: Next.js + Tailwind + shadcn/ui
- Backend: Flask or FastAPI
- LLM Router: LiteLLM (OpenRouter, Gemini, Claude)
- PDF Parsing: PyMuPDF or pdfminer
- Vector Store (for RAG): Chroma or FAISS
- Deployment: Vercel (Frontend) + Railway (Backend)

Deployment:

- Deploy UI on Vercel
- Backend on Railway or Render

Intermediate Project: AI Resume Analyzer

Problem It Solves:

Helps job seekers or recruiters get insights from resumes - skills summary, role match %, missing keywords, and improvement tips.

Tech Stack:

- React or Next.js frontend with Tailwind
- File upload (resume in PDF)
- Python backend (Flask or FastAPI)
- NLP using spaCy or OpenAI GPT-4
- Suggestions: Soft skills, formatting tips, ATS optimization
- Optional: Role matching using OpenRouter with job title input

Skills You Will Build:

- Build file upload UI in Next.js
- Call AI models from backend
- Extract & analyze text from PDF
- Display AI output and feedback
- Deploy using Vercel and test APIs