AI-Powered Legal Document Assistant - PRD

Purpose

The goal is to build an AI-powered assistant that helps users upload, understand, and analyze legal documents. This assistant will use Retrieval-Augmented Generation (RAG), Agentic AI, and Modular Code Packaging (MCP) with PostgreSQL for storage and search.

Objectives

- Enable users to upload legal documents (contracts, policies, agreements).
- Extract and store structured + unstructured data in PostgreSQL.
- Use vector embeddings and RAG to find relevant clauses.
- Apply Agentic AI for intelligent task chaining (e.g., risk analysis, clause extraction).
- Modularize components using MCP for reusability and scaling.

Target Users

- Legal professionals (lawyers, paralegals)
- Contract managers
- Businesses reviewing third-party agreements
- Tech-savvy individuals reviewing policies or leases

Key Features

Document Management

- Upload DOCX, PDF, or TXT files
- Store full text + metadata in PostgreSQL
- Extract named entities (e.g., parties, dates, clauses)

Semantic Search + RAG

- Create and store vector embeddings
- Use similarity search

Agentic AI Tasks

AI-Powered Legal Document Assistant - PRD

- Summarize documents or sections
- Detect risky clauses
- Compare documents
- Answer questions in plain English

Modular Code Packaging (MCP)

- Each core function as a reusable module
- Orchestrate using task runners

System Architecture Overview

[Frontend] -> [API Gateway] -> [Agentic Orchestrator]

<- Ingestion | Retriever (RAG) -> Analyzer Modules

| PostgreSQL | Vector DB -> LLM API

Tech Stack

- Backend API: Python + FastAPI
- Database: PostgreSQL + pgvector
- Vector Search: pgvector / FAISS
- LLM: OpenAI / HuggingFace
- Frontend: React / Streamlit / Gradio
- Code Packaging: MCP-style modules

Milestones (2-Week Roadmap Preview)

Day 1: Project structure + PostgreSQL setup

Day 2-3: Upload & parsing module

Day 4-5: Ingestion pipeline

Day 6-7: RAG retrieval

Day 8: Clause summarization

Day 9: Risk detection

Day 10: Contextual Q&A

AI-Powered Legal Document Assistant - PRD

Day 11: Document comparison

Day 12: MCP integration

Day 13: UI setup

Day 14: Testing & Documentation

Success Criteria

- Upload and parse at least 5 legal documents
- Retrieve relevant clauses via vector search
- Agent can summarize and detect risks with >80% accuracy
- MCP modules reusable for other types