AI-Powered Document Consultant for a Legal Firm

Scenario:

You have been hired by a legal consulting firm to build an **AI-powered document consultant**. The firm wants its staff to upload legal documents (contracts, agreements, case studies) and ask questions based on the content. The system should retrieve relevant document , provide detailed answers, and allow multi-turn conversations with contextual memory.

Your Mission:

Develop an **interactive AI document consultant** using the skills from the curriculum to help the legal team quickly find answers and insights from uploaded legal documents.

Key User Stories:

1. **Upload & Ingest Documents:**
   - **User Story:** As a legal assistant, I want to upload legal contracts so that they are stored for future reference. Show only accept legal doc.
   - **Requirement:** Upload PDFs or text files, extract content, and generate embeddings.
2. **Search for Relevant Information:**
   - **User Story:** As a lawyer, I want to ask questions about specific sections of a contract and get the most relevant excerpt.
   - **Requirement:** Use Pinecone to retrieve top-k relevant sections of the document.
3. **Contextual Conversations:**
   - **User Story:** As a paralegal, I want to ask follow-up questions without rephrasing everything.
   - **Requirement:** Implement LangChain memory to maintain conversation history and context across multiple turns.
4. **Multi-Document Search:**
   - **User Story:** As a legal researcher, I want to search across multiple documents and get consolidated answers.
   - **Requirement:** Aggregate embeddings for all documents and retrieve answers from the entire knowledge base.
5. **Real-Time Interaction:**
   - **User Story:** As a user, I want real-time responses in a chat interface with loading indicators.
   - **Requirement:** Build a chat UI using Next.js with WebSockets or Server-Sent Events for seamless conversation.

6. **Prompt Customization:**
   - **User Story:** As a manager, I want to get both summaries and detailed explanations for legal questions.
   - **Requirement:** Create a prompt library for summarization and Q&A prompts to provide both detailed and concise responses.

Deliverables:

- **Document Upload Functionality:** Users can upload, process, and store legal documents.
- **Search & Retrieval:** Users can search for content based on natural language questions.
- **Conversational AI:** The system retains context across multi-turn conversations.
- **User Interface:** Real-time chat with seamless interaction and error handling.
- **Prompt Customization:** Users can toggle between summary and detailed response modes.

Tools & Concepts to Use:

- **Next.js** for UI and API routes
- **LangChain** for memory, document loaders, and chains
- **Pinecone** for vector embeddings and search
- **OpenAI** for text generation and summarization
- **WebSockets/SSE** for real-time chat
- **Prompt Engineering** for varied response styles