Legal Document QnA Project

Project Name: Legal Document QnA System **Version:** 1.0.0 **Technology Stack:** Python, FastAPI, PyMuPDF, Transformers (FLAN-T5), SentenceTransformers, NumPy, Weaviate (vector store), Frontend (React/Next.js)

Project Summary:

The Legal Document QnA System is an AI-powered question-answering platform designed to help users interact with large legal documents. Users can upload PDF files, which are processed and stored in a vector database. Using Retrieval-Augmented Generation (RAG), the system provides concise and accurate answers based on uploaded legal content.

Key Features:

1. **PDF Upload & Validation**
2. Upload multiple PDF files.
3. Validate file type and ensure content extraction.
4. Unique file identification and safe storage.
5. **Text Extraction & Chunking**
6. Extract text from each PDF page using PyMuPDF.
7. Split large pages into smaller chunks with configurable overlap.
8. Ensure high-quality content for embedding and search.
9. **ScaleDown Integration**
10. Compress text chunks via ScaleDown API for reduced token size.
11. Estimate token counts pre- and post-compression.
12. Fallback to original text if compression fails.
13. **Vector Embeddings & Storage**
14. Generate embeddings using `all-MiniLM-L6-v2`.
15. Store compressed chunks in Weaviate vector store.
16. Retrieve top relevant chunks using cosine similarity.
17. **LLM-Based Q&A**
18. Generate answers with FLAN-T5 model.
19. Context built from top relevant chunks for accurate responses.

20. Robust handling for low-confidence cases and fallback to context.

21. **Frontend Integration**

22. Fully interactive frontend with upload, chat, and document list.
23. Display citations and precedents for answers.

24. Health and stats monitoring endpoints.

25. **API Endpoints**

26. `/api/documents/upload` - Upload and process PDFs.
27. `/api/chat` - Ask questions and get answers.
28. `/api/documents/list` - List uploaded documents and stats.
29. `/api/documents/{file_name}` - Delete a document (placeholder).
30. `/api/stats` - System statistics.

31. `/api/debug/vector-store` - Debug vector store connection.

32. **Startup & Health Checks**

33. Startup/shutdown events for logging.

34. Health endpoint to check model loading, upload directory, and vector store.

35. **Error Handling & Logging**

36. Graceful error handling across all stages.
37. Detailed logs for ingestion, compression, and LLM generation.

Outcome:

The system enables users to ask natural language questions about legal documents and receive relevant, concise answers, with citations and previewed precedents. This fully covers the project description requirements, integrating document processing, vector search, and LLM-based answer generation.

**End of Project Summary.