Design Document – Document Management System (FastAPI + PostgreSQL + Local Storage)

1. Tech Stack Choices

Q1. What frontend framework did you use and why?

Frontend: React (with Next.js or Vite-React)

- React is widely adopted, fast, and component-based.

- Excellent developer ecosystem.

- Easy API integration.

Q2. What backend framework did you choose and why?

Backend: FastAPI

- High performance

- Pydantic validation

- Auto OpenAPI docs

Q3. What database did you choose and why?

Database: PostgreSQL

- Production-grade, reliable, Docker-friendly.

Q4. Scaling to 1,000+ users:

- Move storage to S3

- Add Redis caching

- Use PgBouncer

- Horizontal scaling (Gunicorn workers)

2. Architecture Overview

Flow:

User → Frontend → Backend Routers → Services → PostgreSQL + Local Storage

3. API Specification

POST /documents/upload

Request: multipart/form-data (file=PDF)

Response: { id, filename, file_size, created_at }

GET /documents

Response: list of documents

GET /documents/:id

Response: PDF file stream

DELETE /documents/:id

Response: { message }

4. Data Flow

Upload:

1. User sends PDF

2. Validate PDF + size

3. Save file locally

4. Save metadata in DB

5. Return JSON

Download:

1. Fetch metadata

2. Locate file on disk

3. Return file stream

5. Assumptions

- Only PDFs allowed
- No authentication
- Local disk available
- Low-to-medium traffic
- DB already running