**Design Document - INI8 Labs Full Stack Developer Assignment**

**🏗️ Project Overview**

This project is a Medical Document Management System allowing secure upload, listing, download, and deletion of patient-related PDF files.

**🧱 Architecture Overview**

[ React Frontend ] ←→ [ Flask REST API Backend ] ←→ [ SQLite Database ]

↑

Docker

* **Frontend**: React.js (with Material UI) serves a user-friendly interface for document operations.
* **Backend**: Flask API with JWT-based authentication, secure file upload handling, SQLite for metadata.
* **Containerization**: Docker + Docker Compose used to orchestrate backend and frontend services.
* **File Storage**: Uploaded PDFs are stored on the backend server’s filesystem.

⚙️ **Technology Stack**

| **Layer** | **Technology Used** |
| --- | --- |
| Frontend | React.js, Material UI |
| Backend | Python, Flask, SQLite |
| Auth | JWT (JSON Web Token) |
| Container | Docker, Docker Compose |
| File Handling | Secure Upload via Flask |
|  |  |

🛠️ **Major Features**

| Feature | Description |
| --- | --- |
| JWT Login | Generates token for user to access protected endpoints |
| PDF Upload | Allows uploading patient documents with metadata |
| List Documents | Displays all uploaded documents with metadata |
| Download Documents | Authenticated download using secure endpoint |
| Delete Documents | Authenticated delete from filesystem and DB |
| SQLite Integration | Metadata is stored using lightweight relational DB |

**🔒 Security Measures**

* Allowed file extensions: PDF only (.pdf)
* Max upload size: 10 MB
* Filenames are sanitized and uniquely stored
* JWT authentication on all endpoints
* CORS enabled

🔁 **API Endpoints Summary**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /login | Get JWT token |
| GET | /documents | List all documents (auth required) |
| POST | /documents/upload | Upload PDF (auth required) |
| GET | /documents/<doc\_id>/download | Download document (auth required) |
| DELETE | /documents/<doc\_id> | Delete document (auth required) |

**⚙️Docker Setup Overview**

* frontend and backend each have separate Dockerfiles.
* nginx used for frontend routing.
* docker-compose.yml manages orchestration.
* Healthcheck included for backend readiness.

📦 Folder Structure

