**🧑‍⚖️ User Flow Document**

**🔐 1. User Logs In**

* **Frontend Route**: /login
* **Component**: LoginForm.jsx
* **Backend Endpoint**: POST /api/auth/login
* **DB**: users
* **Response**: JWT token

**🧾 2. Create a New Legal Case**

* **Frontend Route**: /cases/new
* **Component**: NewCaseForm.jsx
* **Backend Endpoint**: POST /api/cases
* **DB Schema**:

CREATE TABLE cases (

id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES users(id),

title TEXT NOT NULL,

description TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**📁 3. Upload Document to Case**

* **Frontend Route**: /cases/:caseId/upload
* **Component**: DocumentUpload.jsx
* **Backend Endpoint**: POST /api/documents/upload
* **ML Endpoint**: POST /analyze (FastAPI)
* **DB Schemas**:

CREATE TABLE documents (

id SERIAL PRIMARY KEY,

case\_id INTEGER REFERENCES cases(id),

filename TEXT,

upload\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE clauses (

id SERIAL PRIMARY KEY,

document\_id INTEGER REFERENCES documents(id),

clause\_text TEXT,

risk\_level TEXT CHECK (risk\_level IN ('low', 'medium', 'high')),

summary TEXT

);

**🔍 4. View Annotated Document**

* **Frontend Route**: /cases/:caseId/documents/:docId
* **Component**: ClauseViewer.jsx
* **Features**: Color-coded clause list, risk filter

**🤖 5. Ask Questions About the Case**

* **Frontend Route**: /cases/:caseId/ask
* **Component**: CaseChatbot.jsx
* **Backend Endpoint**: POST /api/cases/:id/ask
* **ML Endpoint**: POST /ask
* **DB Schema**:

CREATE TABLE queries (

id SERIAL PRIMARY KEY,

case\_id INTEGER REFERENCES cases(id),

question TEXT,

answer TEXT,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**📝 6. Add Notes or Flags**

* **Component**: ClauseNote.jsx
* **Backend Endpoint**: POST /api/clauses/:id/notes
* **DB Schema**:

CREATE TABLE notes (

id SERIAL PRIMARY KEY,

clause\_id INTEGER REFERENCES clauses(id),

user\_id INTEGER REFERENCES users(id),

content TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**📤 7. Download Summary Report**

* **Frontend Route**: /cases/:caseId/report
* **Component**: DownloadReportButton.jsx
* **Backend Endpoint**: GET /api/cases/:id/report
* **Feature**: Generates downloadable PDF with:
  + Risk clauses
  + Summaries
  + AI responses
  + User notes

**⚛️ Frontend Routes and Components Map**

| **Route** | **Component** | **Purpose** |
| --- | --- | --- |
| /login | LoginForm.jsx | User authentication |
| /dashboard | Dashboard.jsx | List of user cases |
| /cases/new | NewCaseForm.jsx | Create new case |
| /cases/:id | CaseDetails.jsx | View documents & chat |
| /cases/:id/upload | DocumentUpload.jsx | Upload legal docs |
| /cases/:id/documents/:docId | ClauseViewer.jsx | View clause analysis |
| /cases/:id/ask | CaseChatbot.jsx | Ask questions to AI |
| /cases/:id/report | DownloadReportButton.jsx | Export case summary |

**🐳 Docker Compose Excerpt**

version: '3.8'

services:

backend:

build: ./backend

ports:

- "8080:8080"

frontend:

build: ./frontend

ports:

- "3000:3000"

ml-service:

build: ./ml-service

ports:

- "5000:5000"

db:

image: postgres:13

environment:

POSTGRES\_DB: lexiguard

POSTGRES\_USER: user

POSTGRES\_PASSWORD: pass

ports:

- "5432:5432"

Let me know if you want wireframes or a GitHub repo template next!