📌 Project Introduction

📌 Project Introduction

The **Document Signature App** is a secure, full-stack web application that enables users to **upload documents, place digital signatures, share signing links, and generate legally traceable signed PDFs** — similar to platforms like **DocuSign** and **Adobe Sign**.

The app eliminates the need for physical paperwork by allowing documents to be signed electronically with full **audit trails, signer identity verification, and document integrity**. It is designed with real-world enterprise workflows in mind, including authentication, document ownership, status tracking, and signature history.

This project demonstrates how modern SaaS products handle **file security, digital trust, and collaborative workflows** at scale.

🎯 Aim of the Project

🎯 Aim of the Project

The primary aim of this project is to:

- Build a **production-ready digital signature system**
- Understand **end-to-end document lifecycle management**
- Learn how **secure SaaS platforms** are architected
- Simulate real **enterprise-grade backend and frontend workflows**
- Gain hands-on experience with **PDF processing, audit logging, and token-based access**

This project is not just a CRUD app — it models **real business logic** used in legal, HR, finance, and enterprise software.

🧩 Core Problems This App Solves

🧩 Core Problems This App Solves

- Manual document signing is **slow and error-prone**
- Physical paperwork is **hard to track and store**
- Emailing PDFs back and forth has **no auditability**
- No visibility into **who signed, when, and from where**
- Risk of document tampering after signatures

This app addresses all of these with **secure uploads, signature coordinates, immutable signed PDFs, and audit trails**.

🤵 Real-World Use Cases

🧑‍💼 Real-World Use Cases

1️⃣ Business Contracts & Agreements

- Vendors sign contracts remotely
- Status tracked as *Pending / Signed / Rejected*
- Signed documents stored securely

2️⃣ HR & Onboarding

- Offer letters
- NDA signing
- Policy acknowledgements

3️⃣ Freelancers & Agencies

- Client agreements
- Proposal approvals
- Payment authorization documents

4️⃣ Legal & Compliance Teams

- Legal documents with traceability
- Audit trails for disputes
- IP-based signer tracking

5️⃣ Education & Institutions

- Consent forms

- Admission documents
- Certificates and approvals

🔐 Key Features That Mimic Industry Standards

🔐 Key Features That Mimic Industry Standards

- JWT-based authentication
- Secure PDF upload & access control
- Drag-and-drop signature placement
- Server-side PDF modification (PDF-Lib)
- Tokenized public signature links
- Audit logs with timestamps & IP
- Status lifecycle: Pending → Signed → Rejected
- Deployment-ready architecture

These are the **exact patterns used in enterprise SaaS products**.

🏢 Industry Value of This Project

🏢 Industry Value of This Project

💼 Why Companies Care About This Project

This project showcases **skills companies actively hire for**:

✅ SaaS Architecture

- Multi-user authentication
- Role-based document ownership
- Tokenized external access

✅ Security & Compliance Thinking

- JWT authentication
- Audit trail implementation
- Protected document access
- Immutable signed outputs

✅ Backend Engineering Depth

- File handling at scale
- PDF manipulation
- Middleware logging
- Status workflows

✅ Frontend Product Thinking

- Drag-and-drop UX
- PDF rendering

- Responsive dashboards
- Status-based UI filtering

✅ Real Business Logic (Not Tutorials)

- Document lifecycle management
- Signature coordinates mapping
- Final PDF generation
- Email/share workflows

🧠 What This Project Signals on Your Resume

🧠 What This Project Signals on Your Resume

✔ “This developer can build **enterprise-grade SaaS features**”
✔ “Understands **security, audit, and compliance basics**”
✔ “Can work with **files, PDFs, and real data flows**”
✔ “Not just a frontend/backend dev — but a **product-aware engineer**”

This project sits in the same category as:

- CRM systems
- HR platforms
- LegalTech software
- FinTech document systems

🏆 Why This Is a Strong Portfolio Project

DocuSign-style apps prove you can handle real-world complexity beyond CRUD apps

Compared to todo apps or basic dashboards, this project demonstrates:

- **Stateful workflows**
- **Security-first design**
- **Cross-user interactions**
- **Document integrity guarantees**

It’s the kind of project that **stands out immediately** in interviews.

📆 2-Week Build Plan

📆 2-Week Build Plan

✅ Week 1: Core Features, Backend & Frontend Setup

Day 1: Project Setup & Repo Initialization

- Create GitHub repo with MERN folder structure
- Initialize React app with Tailwind CSS
- Setup Node.js + Express + MongoDB (Mongoose)
- Install necessary libraries (Multer, PDF-Lib, bcrypt, JWT)
  Skills: Project scaffolding, Tailwind setup, server setup

Day 2: Auth System (JWT)

- User model: name, email, password
- Routes: /register, /login
- JWT auth + bcrypt password hashing
- Auth middleware for protected routes
  Skills: JWT, secure user auth, API routes

Day 3: File Upload API

- Setup Multer for PDF uploads
- Store file path & metadata in MongoDB
- Route: /api/docs/upload
  Skills: File handling, middleware, API design

Day 4: View & List Documents

- API to fetch user's uploaded files
- Display files in a dashboard
- Add preview functionality using `react-pdf`
  Skills: RESTful APIs, PDF rendering in React

Day 5: Signature Schema & Logic

- Signature Model: fileId, coordinates, signer, status
- Route to save signature positions (x, y)
- Display signature placeholder on PDF
  Skills: Schema relations, position-based rendering

Day 6: PDF Editor Integration

- Add drag-and-drop signature field (HTML overlay)
- Save coordinates relative to page
- Route: POST /api/signatures
  Skills: Coordinate systems, frontend UX logic

Day 7: Buffer / Testing

- Debug UI and backend integration
- Write Postman tests for API

✅ Week 2: Signature Rendering, Sharing & Polish

Day 8: Generate Final Signed PDF

- Use `PDF-Lib` to embed signature text/image
- Export signed version to disk/cloud
  Skills: PDF processing, server-side document generation

Day 9: Email + Public Signature Links

- Generate tokenized URL for external signature
- Email link to signer (use nodemailer or mock)
  Skills: Token auth, email logic

Day 10: Audit Trail

- Log who signed, when, and IP (use middleware)
- Route: GET /api/audit/:fileId
  Skills: Middleware logging, basic audit management

Day 11: Signature Status Updates

- Status: Pending, Signed, Rejected
- Allow signer to accept/reject with reason
  Skills: Status flows, conditional logic

Day 12: Dashboard UI Polish

- Filter by signature status
- Responsive layout with Tailwind
  Skills: Tailwind UI polish, component reuse

Day 13: Deployment

- Backend: Render / Railway
- Frontend: Vercel / Netlify
- MongoDB Atlas for DB

Day 14: Final Testing + Demo

- Prepare a GitHub README
- Record a 2-min demo walkthrough

🌐 Web Tech Stack for Document Signature App

🌐 Web Tech Stack for Document Signature App

🧱 Recommended Tech Stack (Modern & Industry-Ready)

🎨 Frontend

- **React (Vite or Next.js)**
- **TypeScript** (highly recommended)
- **Tailwind CSS**
- **react-pdf**
- **dnd-kit** (drag & drop)
- **Axios**
- **React Hook Form**
- **Zod** (validation)

⚙️ Backend (API Layer)

- **Node.js + Express**

🔐 Auth

- **JWT (access + refresh tokens)**
- **Protected routes**

☁️ Storage

- **Local storage (dev)**
- **Supabase Storage / AWS S3 (prod)**

🚀 Deployment

- **Frontend:** Vercel / Netlify
- **Backend:** Render / Railway
- **DB:** Supabase / MongoDB Atlas

🔎 API Endpoints & Mock Data

🔎 API Endpoints & Mock Data

Auth APIs

POST /api/auth/register
POST /api/auth/login

Document APIs

POST /api/docs/upload // Upload PDF
GET /api/docs/ // List user PDFs
GET /api/docs/:id // View specific doc

Signature APIs

POST /api/signatures // Save signature position
GET /api/signatures/:id // Get signatures for document
POST /api/signatures/finalize // Embed signature into PDF

Audit APIs

GET /api/audit/:docId

Mock Data: `User`

```
{
  "name": "Alice",
  "email": "alice@example.com",
  "password": "hashedpassword"
}
```

Mock Data: `Signature`

```
{
  "documentId": "abc123",
  "userId": "xyz456",
  "x": 120,
  "y": 330,
  "page": 1,
  "status": "signed"
}
```

📅 Skills You Will Learn by Stage

📅 Skills You Will Learn by Stage

Week	Skill	Description
1	Auth & API	JWT, bcrypt, REST APIs
1	File Upload	Multer, server-side storage
1-2	Frontend UX	Drag-and-drop UI, react-pdf
2	PDF Generation	`pdf-lib` for embedding data
2	Security	Tokenized links, protected routes
2	Audit Logging	Middleware, action tracking
2	Deployment	Hosting full-stack app

Resources

📚 Learning Resources

1. **PDF-Lib JS Docs:** https://pdf-lib.js.org/docs
2. **MERN Auth (JWT):** https://www.youtube.com/watch?v=2jqok-WgelI (PedroTech)
3. **Multer File Upload:** https://www.youtube.com/watch?v=9Qzmri1WaaE
4. **PDF Signature with Node:** Signing a PDF with Javascript - Going though the code
5. **React PDF Viewer:** How to View PDFs in React JS with React PDF Viewer
6. **React Drag and Drop Signatures:**
   React Drag And Drop (dnd-kit) | Beginners Tutorial

📚 Web Tech Stack Resources (Step-by-Step)

⚛️ 1. React (Core)

📺 YouTube

React Full Course (Beginner → Advanced)

👉 https://www.youtube.com/watch?v=bMknfKXIFA8

✅ Covers:

- Hooks
- State
- Components
- API calls

📄 **Docs**

- **React Official Docs**
  👉 https://react.dev/

⚡ 2. Vite / Next.js Setup

📺 **YouTube**

Vite + React Setup
👉 **https://www.youtube.com/watch?v=GgMbL0z_h8A**

Next.js App Router (Optional)
👉 https://www.youtube.com/watch?v=wm5gMKuwSYk

🎨 3. Tailwind CSS (UI Polish)

📺 **YouTube**

Tailwind CSS Crash Course
👉 https://www.youtube.com/watch?v=dFgzHOX84xQ
✅ **Perfect for dashboards**

📄 **Docs**

- **Tailwind Docs**
  👉 https://tailwindcss.com/docs

📄 4. PDF Rendering (CRITICAL)

📺 **YouTube**

View PDF in React using react-pdf
👉 **https://www.youtube.com/watch?v=Uu4W8r9sKxQ**

📄 **Docs**

- **react-pdf Docs**
  👉 https://github.com/wojtekmaj/react-pdf

✍️ 5. Drag-and-Drop Signatures (CORE FEATURE)

📺 **YouTube**

Drag & Drop using dnd-kit
👉 **https://www.youtube.com/watch?v=FZt2N6R0YkY**

📄 **Docs**

- dnd-kit Docs
  👉 https://docs.dndkit.com/

🔄 6. API Integration (Axios + JWT)

📺 YouTube

Axios Crash Course
👉 https://www.youtube.com/watch?v=6LyagkoRWYA

JWT Auth in React
👉 https://www.youtube.com/watch?v=nDGA3km5He4

📄 Docs

- Axios Docs
  👉 https://axios-http.com/

🔐 7. Auth & Protected Routes

📺 YouTube

Protected Routes in React
👉 https://www.youtube.com/watch?v=2k8NleFjG7I

🧪 8. Forms & Validation

📺 YouTube

React Hook Form + Zod
👉 https://www.youtube.com/watch?v=Z1Nw5FfKqC8

📄 Docs

- **React Hook Form**
  👉 https://react-hook-form.com/
- **Zod**
  👉 https://zod.dev/

🧠 9. Coordinate System (IMPORTANT)

Concept to Learn

- Convert absolute → relative coordinates
- Save `(x %, y %)`
- Re-render correctly on different screens

📺 Video

Mouse Position & Bounding Box in React
👉 https://www.youtube.com/watch?v=qd5y8T7Zx9U

☁️ 10. Deployment

📺 YouTube

Deploy React App on Vercel
👉 https://www.youtube.com/watch?v=2h6n5Dgk0ol

🏆 Recommended Learning Order

Step	Learn
1	React basics
2	Tailwind
3	react-pdf
4	Drag & Drop
5	Axios + JWT
6	Protected routes

7	Coordinate logic
8	UI polish
9	Deployment

💡 Why This Web Stack Is Perfect

This tech stack matches real SaaS dashboards used in industry

You'll be learning:

- **Component-driven UI**
- **Drag-and-drop UX**
- **Secure frontend auth**
- **PDF rendering workflows**
- **Responsive enterprise UI**