

Project Title: Document Signature App (DocuSign Clone)



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
- Routes: /api/docs/upload
Skills: File handling, middleware, API design

Admin Aditya

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: field, 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

GitHub Folder Structure

```
/signature-app
|-- /client
|   |-- /src
|       |-- /components
|       |-- /pages
|       |-- /utils
|-- /server
|   |-- /controllers
|   |-- /models
|   |-- /routes
|   |-- /middleware
|   |-- /utils
|-- .env
|-- README.md
```

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

| 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 | <code>pdf-lib</code> for embedding data |
| 2 | Security | Tokenized links, protected routes |
| 2 | Audit Logging | Middleware, action tracking |
| 2 | Deployment | Hosting full-stack app |



Learning Resources

1. **PDF-Lib JS Docs:** <https://pdf-lib.js.org/docs>
 2. **MERN Auth (JWT):** <https://www.youtube.com/watch?v=2jqok-Wgell> (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
-

Tech Stack

Frontend - React or Its Framework, Tailwind CSS

Backend - Node Js, Express Js

Database - Any DB which you know (Supabase, Postgress SQL, Mongo DB)



Final Deliverables

- Responsive full-stack app
- Upload/view/sign/share PDFs
- Secure signature handling with audit logs
- GitHub repo + README + demo video