Introduction

📄 Document Signature App — Java Tech Stack

🧠 Introduction

The **Document Signature App** is a secure, enterprise-grade web application that enables users to **upload documents, place digital signatures, share signing links, and generate legally traceable signed PDFs**.

Built using the **Java ecosystem**, this project mirrors how **real-world enterprise document management systems** (like DocuSign, Adobe Sign, Zoho Sign) are developed — with a strong focus on **security, scalability, auditability, and compliance**.

This application demonstrates how Java is used to build **mission-critical backend systems** handling sensitive documents and workflows.

🎯 Aim of the Project

The aim of this project is to:

- Build a **DocuSign-like digital signature platform** using Java
- Learn **enterprise backend development with Spring Boot**
- Implement **secure authentication and authorization**
- Handle **PDF processing and digital signatures**
- Design **audit logs and document lifecycle workflows**
- Create a **job-ready enterprise portfolio project**

This project focuses on **real business logic**, not just CRUD operations.

🧩 Problems This Application Solves

- Manual document signing is slow and inefficient
- Physical documents lack traceability and security
- No audit trail for compliance verification
- Risk of document tampering after approval
- No centralized document status tracking

The app solves these issues through **digital signatures, immutable PDFs, access control, and audit logging**.

🤵 Real-World Use Cases

🧑‍💼 Real-World Use Cases

1️⃣ Corporate & Enterprise

- Vendor agreements
- Employee contracts
- Procurement approvals

2️⃣ HR & Recruitment

- Offer letters
- NDAs
- Policy acceptance

3️⃣ Legal & Compliance

- Contract approvals
- Regulatory documents
- Client consent forms

4️⃣ Finance & Banking

- Loan agreements
- KYC authorization
- Compliance documents

5️⃣ Education & Government

- Admission forms

- Certificates
- Official approvals

🏢 Industry Value of This Project

🏢 Industry Value of This Project

Why Companies Love This Project

This project reflects **real enterprise systems** used in:

- Banking & FinTech
- LegalTech platforms
- Government portals
- HR management systems
- Compliance-heavy industries

Skills Demonstrated

- Spring Boot expertise
- Secure API design
- PDF processing in Java
- JWT & Spring Security
- Audit & compliance workflows
- SaaS architecture understanding

🏆 Why This Project Is Resume-Strong

Java-based DocuSign clones demonstrate enterprise backend maturity

This project proves you can:

- Handle sensitive data

- Build secure systems
- Design scalable APIs
- Implement real business workflows

It’s **far more impressive** than typical CRUD projects.

🏗️ Tech Stack (Java-Based)

🏗️ Tech Stack (Java-Based)

🔹 Backend

- **Java 17+**
- **Spring Boot**
- **Spring Security**
- **JWT (jjwt)**
- **Spring Data JPA (Hibernate)**
- **PostgreSQL / MySQL**
- **Maven**

🔹 PDF & File Handling

- **Apache PDFBox** (PDF reading & writing)
- **iText (optional)** for advanced signing
- **MultipartFile** (file uploads)

🔹 Frontend

- **React**
- **Tailwind CSS**
- **react-pdf**
- **Drag & Drop (dnd-kit)**

🔹 Other Tools

- **Java Mail Sender** (email notifications)

- **Docker** (optional)
- **Cloud Storage** (AWS S3 / Azure Blob or supabase)

📆 2-Week Build Plan (Java Stack)

📆 2-Week Build Plan (Java Stack)

✅ Week 1: Backend Foundation & Core Features

📆 Day 1: Project Setup

- Initialize Spring Boot project
- Configure Maven dependencies
- Setup PostgreSQL / MySQL
- Setup frontend (React + Tailwind)

Skills: Spring Boot architecture, project scaffolding

📆 Day 2: Authentication & Authorization

- User entity (id, name, email, password)
- Password hashing using BCrypt
- JWT-based login & registration
- Secure APIs with Spring Security

Skills: JWT, Spring Security, password encryption

📆 Day 3: File Upload API

- Upload PDFs using MultipartFile
- Save file metadata to DB

- Secure upload endpoint

Skills: File handling, REST APIs

📅 Day 4: Document Listing & Preview

- Fetch documents for logged-in user
- View document details
- Frontend PDF preview using react-pdf

Skills: RESTful design, frontend-backend integration

📅 Day 5: Signature Schema & Logic

- Signature entity (docId, signerId, x, y, page, status)
- Save signature coordinates
- Display signature placeholders

Skills: Entity relationships, coordinate systems

📅 Day 6: Drag-and-Drop Signature UI

- Frontend overlay on PDF
- Capture relative coordinates
- Save via REST API

Skills: UX logic, spatial data handling

📅 Day 7: Buffer & Testing

- API testing (Postman)
- Fix integration bugs

✅ Week 2: Enterprise Features & Deployment

📅 Day 8: Generate Signed PDF

- Embed signature text/image using PDFBox
- Generate final signed PDF
- Prevent document modification

Skills: PDF processing, file immutability

📅 Day 9: Email & Public Signature Links

- Generate token-based public signing URLs
- Send email notifications

Skills: Tokenized access, email integration

📅 Day 10: Audit Trail

- Log actions (upload, view, sign)
- Capture timestamp & IP
- Fetch audit history

Skills: Middleware, compliance logging

📅 Day 11: Signature Status Flow

- Pending / Signed / Rejected
- Rejection reason handling

Skills: Workflow management

📅 Day 12: Dashboard UI Polish

- Filter documents by status
- Responsive Tailwind UI

📅 Day 13: Deployment

- Backend: AWS EC2 / Render / Railway
- Frontend: Vercel / Netlify
- DB: RDS / Supabase

📅 Day 14: Final Testing & Demo

- README documentation
- Demo video walkthrough
- Sample test accounts

📁 Java Project Folder Structure

📁 Java Project Folder Structure

```
/signature-app
|── /backend
|    ├── controller
|    ├── service
|    ├── repository
|    ├── model
|    ├── security
|    ├── dto
|    └── SignatureAppApplication.java
|
|── /frontend
|    ├── /src
|    |    ├── components
|    |    ├── pages
|    |    └── utils
|
|── application.yml
|── README.md
```

🔎 API Endpoints

🔐 Auth APIs

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

📄 Document APIs

```
POST /api/docs/upload
GET  /api/docs
GET  /api/docs/{id}
```

✍️ Signature APIs

```
POST /api/signatures
```

```
GET  /api/signatures/{docId}
POST /api/signatures/finalize
```

🕵️ Audit APIs

```
GET /api/audit/{docId}
```

📚 Learning Resources – Java DocuSign Clone

📚 Learning Resources –

🎯 How to Use These Resources

- Don’t watch everything end-to-end
- Use them **day-wise**, only when that feature comes up
- Build → get stuck → watch → implement

🔐 1. Java + Spring Boot Fundamentals

📺 YouTube

Spring Boot Full Course (Beginner to Intermediate)
👉 https://www.youtube.com/watch?v=vtPkZShrvXQ
✅ Covers:

- Controllers
- REST APIs
- Dependency Injection
- Project structure

Spring Boot REST API Tutorial (Practical)
👉 https://www.youtube.com/watch?v=9SGDpanrc8U
✅ Perfect for:

- `/api/docs`
- `/api/signatures`
- `/api/audit`

📄 Documentation

- Spring Boot Official Docs
  👉 https://spring.io/projects/spring-boot

🔑 2. Authentication & JWT (VERY IMPORTANT)

📺 YouTube

Spring Boot JWT Authentication (BEST ONE)
👉 https://www.youtube.com/watch?v=X80nJ5T7YpE
✅ Covers:

- Register & Login
- JWT creation
- JWT filter
- Securing routes

Spring Security + JWT Full Tutorial
👉 https://www.youtube.com/watch?v=KxqlJblhzfI
✅ Enterprise-style security setup

📄 Documentation

- Spring Security Docs
  👉 https://spring.io/projects/spring-security
- JWT (jjwt library)
  👉 https://github.com/jwtk/jjwt

📄 3. File Upload (PDF Upload API)

📺 YouTube

Spring Boot File Upload (MultipartFile)
👉 https://www.youtube.com/watch?v=YijgGq3Fqys
✅ Exactly for:

- `/api/docs/upload`
- PDF uploads
- Saving files on server

📄 Documentation

- Multipart File Upload (Spring)
  👉 https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#mvc-multipart

---

📕 4. PDF Reading & Writing (SIGNATURE CORE)

📺 YouTube

Apache PDFBox Full Tutorial
👉 https://www.youtube.com/watch?v=4R5S3uP8b1Q
✅ Covers:

- Reading PDFs
- Writing text
- Adding images
- Editing pages

Add Image/Text to PDF using PDFBox
👉 https://www.youtube.com/watch?v=6RdbzXzqQyw
✅ Perfect for embedding signatures

📄 Documentation

- Apache PDFBox Official Docs
  👉 https://pdfbox.apache.org/

✍️ 5. Digital Signature Logic (Coordinates & Embedding)

📺 YouTube

How Digital Signatures Work (Conceptual)
👉 https://www.youtube.com/watch?v=GSIDS_lvRv4
✅ Understand:

- Why coordinates matter
- How signatures are rendered

PDFBox Positioning Text & Images
👉 https://www.youtube.com/watch?v=E9nGJ3Z9N8Q
✅ Needed for:

- x, y positioning
- Page number logic

🕵️ 6. Audit Logs & Middleware

📺 YouTube

Spring Boot Filters & Interceptors
👉 https://www.youtube.com/watch?v=QxHk9xYz5vw
✅ Used for:

- Logging IP
- Logging timestamp
- Audit trails

📄 Documentation

- Spring Interceptors
  👉
  https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#mvc-handlermapping-interceptor

---

📬 7. Email & Public Signing Links

📺 YouTube

Send Email Using Spring Boot (JavaMailSender)
👉 https://www.youtube.com/watch?v=QqP_0v5cB8Q
✅ Used for:

- Sending signature links
- Email notifications

📄 Documentation

- Spring Email Docs
  👉 https://www.baeldung.com/spring-email

🧪 8. Testing APIs (Postman)

📺 YouTube

Postman API Testing for Beginners
👉 https://www.youtube.com/watch?v=VywxIQ2ZXw4
✅ Use for:

- Auth testing
- File upload testing
- Signature APIs

🎨 9. Frontend (React + PDF + Drag & Drop)

📺 YouTube

React PDF Viewer Tutorial
👉 https://www.youtube.com/watch?v=Uu4W8r9sKxQ
✅ Used for:

- Viewing PDFs
- Overlaying signature boxes

Drag & Drop in React (dnd-kit)
👉 https://www.youtube.com/watch?v=FZt2N6R0YkY
✅ For:

- Signature placement

☁️ 10. Deployment (Optional but Valuable)

📺 YouTube

Deploy Spring Boot App (Render / Railway)
👉 https://www.youtube.com/watch?v=1h4XxW6XnGQ

Deploy React App on Vercel
👉 https://www.youtube.com/watch?v=2h6n5Dgk0oI

📌 Recommended Learning Order (VERY IMPORTANT)

Step	Learn
1	Spring Boot REST APIs
2	JWT + Spring Security
3	File Upload
4	PDFBox
5	Signature Coordinates
6	Audit Logs
7	Email
8	UI Polish
9	Deployment

🏆 Why These Resources Are Enough

These resources cover 100% of what real enterprise Java apps use

If you complete this project using these links:

- You are **job-ready for Java backend roles**
- You can confidently explain **security, files, PDFs, and workflows**
- Your portfolio will stand out massively