

# FULL STACK PROJECT: AUTOMATED FORM FILLING SYSTEM

TECH STACK: Spring Boot (Backend), React + Tailwind (Frontend), MySQL/PostgreSQL, AI/OCR Service

## MODULE 1: REQUIREMENT ANALYSIS & SYSTEM ARCHITECTURE

- Define complete feature list.
- Write functional and non-functional requirements.
- Draw high-level architecture(React→ Spring Boot→ DB→ File Storage->AI Service)
- Finalize data models, APIs, and user flow.

## MODULE 2: BACKEND SETUP (SPRING BOOT)

- Create Spring Boot project with Web, JPA, Security.
- Connect backend with MySQL/PostgreSQL.
- Create standard folder structure (controllers, services, repositories, entities, DTOs).
- Create basic test endpoints.

## MODULE 3: FRONTEND SETUP (REACT + TAILWIND)

- Initialize React (Vite or CRA).
- Install Tailwind CSS.
- Create Navbar, Footer, Layout components.
- Set up React Router and API helper (Axios).

## MODULE 4: USER AUTHENTICATION

- Backend: User model, JWT authentication, login/signup APIs.
- Frontend: Login page, signup page, store JWT, create Protected Routes.

## MODULE 5: SELECT OR UPLOAD FORM

- Frontend: Page to choose existing government form.
- Frontend: Page to upload custom form (PDF).
- Backend: API to upload and store form PDF.
- Validate PDF type and size.

## MODULE 6: AI FORM PARSING

- Backend: Use OCR/PDF extraction to detect form fields.
- Convert extracted fields into structured JSON.
- Frontend: Show loading UI and display extracted form fields.

## MODULE 7: DOCUMENT REQUEST MODULE

- Backend: Based on form fields, generate required document list.
- Frontend: Display list of required documents for the user to upload.

## **MODULE 8: USER DOCUMENT UPLOAD**

- Backend: Store user-uploaded documents (Aadhaar, photo, signature, etc.).
- Frontend: Upload UI + preview of uploaded files.

## **MODULE 9: AUTO-FILL ENGINE**

- Backend: Extract data from documents using OCR/AI.
- Map extracted values to form fields.
- Create validation logic.
- Frontend: Show auto-filled form and allow manual editing.

## **MODULE 10: FINAL FORM GENERATION**

- Backend: Fill the original PDF using PDF library (PDFBox/iText).
- Generate final filled PDF.
- Frontend: Button to download final form and preview.

## **MODULE 11: USER DASHBOARD & HISTORY**

- Frontend: Dashboard showing past forms, status, and download options.
- Backend: Store each form submission with timestamp and metadata.

## **MODULE 12: DEPLOYMENT & SCALING**

- Deploy Spring Boot backend to Render/Railway/AWS.
- Deploy React frontend to Netlify/Vercel.
- Use cloud database (Railway/PlanetScale).
- Use S3/local storage for files.
- Add logs and monitoring.