Project Explanation

Automated PDF Resume Updater – Layout Preserving Tool

Developed Using: React + Spring Boot

Project Overview

This project is a web-based resume editing tool built using **React (Frontend)** and **Spring Boot (Backend)**.

Although the assignment required a script, I implemented a scalable web application to make the resume updating process more dynamic, reusable, and user-friendly.

The system allows users to:

- Upload a PDF resume
- Apply required edits (add experience, modify skill, add certification)
- Automatically generate an updated PDF
- Preserve the original layout of the resume

Backend Architecture (Spring Boot)

The backend is developed using Spring Boot.

PDF Dependency

I used **Apache PDFBox** for:

- Reading PDF content
- Writing updated text
- Preserving layout positioning

Structure

- **Controller Layer**
    - Handles REST APIs
    - Accepts PDF file using MultipartFile
    - Calls service layer
- **Service Layer**
    - Contains complete PDF editing logic

- Extracts text from PDF
- Identifies specific sections (Experience, Skills, Certifications)
- Applies modifications
- Writes updated content at the same coordinates
- Generates new updated PDF

Frontend Architecture (React)

The frontend is built using React.

Features

- File upload interface
- Input fields to modify:
  - Add 5 experience lines
  - Modify one skill
  - Add certification
- API call to backend
- Automatic updated PDF download

Working Flow

1. User uploads resume
2. User enters changes
3. React sends data to backend API
4. Backend processes PDF
5. Updated resume is returned
6. User downloads updated version

All changes made in frontend are dynamically reflected in the generated resume.

How All Components Work Together

Frontend and backend are connected via REST API.

- Both applications are run separately.
- Backend runs on port 8080.
- Frontend communicates using HTTP requests.
- On modification in frontend, backend automatically updates the resume.

Key Technical Highlights

Layout preserved
Unicode-safe font handling
Clean separation of concerns
Reusable service logic
Dynamic editing
Multiple resumes supported

Challenges Faced

- Handling Unicode characters in PDF
- Maintaining exact layout positioning
- Preventing font encoding errors
- Ensuring formatting consistency across resumes

These were solved using proper font embedding and coordinate-based text rendering.

mvn clean install
mvn spring-boot:run
cd frontend
npm install
npm run dev