# **PDF Editor Project Overview**

## **1. Scope**

This project focuses on creating a PDF editor that allows users to load, fill out, and save PDF forms using pdf-lib. The main goal is to make sure that users can easily interact with PDF forms, fill them out, and save their work without any issues.

**2. Tools and Libraries Used**

* **React:** Used to build the user interface.
* **NestJS:** Handles the backend services and API interactions.
* **PDF-lib:** A JavaScript library that helps us create, modify, and work with PDF files.

## **3. Implementation Details**

### **3.1 Loading the PDF**

1. **Frontend (React):**
   * Used pdf-lib to load the PDF form into a viewer on the front-end.
   * The PDF can be fetched from the server.
2. **Backend (NestJS):**
   * The backend provides an API that sends the PDF to the frontend for rendering.

### **3.2 Filling Out the PDF Form**

1. **PDF-lib:**
   * pdf-lib was used to set the values in the form fields programmatically. This includes handling text fields, checkboxes, and radio buttons.

### **3.3 Saving the Filled PDF Form**

1. **Frontend (React):**
   * After the form is filled out, users can trigger a save action to send the filled PDF back to the server.
2. **Backend (NestJS):**
   * The backend receives the filled PDF, processes it, saves it to the server.
3. **PDF-lib:**
   * The save() function in pdf-lib was used to save the filled PDF. However, we ran into some issues here: the filled data wasn’t saved or displayed correctly in the final PDF.

### **3.4 Challenges Encountered**

* **Saving Issues:** The biggest problem was saving the filled PDF data. Although I could fill the form fields using pdf-lib, the data didn’t always save or display properly in the final document.
  + **Possible Reasons:**
    - The way we used the save() function in pdf-lib might have caused some issues.
    - There were challenges with how certain form fields, like checkboxes and radio buttons, were handled.
  + **What We Tried:**
    - I spent time debugging the save process to make sure data was correctly applied to the PDF.
    - I also looked into other methods within pdf-lib to see if they could help resolve these issues.

## **4. Next Steps**

* I need to dig deeper into how pdf-lib handles the save function to figure out what’s going wrong.

### **Steps for Cloning, Installing Dependencies, and Running the Servers**

#### **1. Clone the Repositories**

git clone https://github.com/neetasawant/pdf-editor.git

cd pdf-editor

#### **2. Install Dependencies**

**For Frontend**:

Navigate to the frontend directory:  
cd frontend

Install dependencies:  
npm install

**For Backend**:

Navigate to the backend directory:  
cd backend

Install dependencies:  
npm install

#### **3. Start the Servers**

npm start.

#### **4. Verify Everything is Running**

* **Frontend**: Open your browser and navigate to <http://localhost:3000>
* **Backend**: Ensure the backend server is running and can be accessed at http://localhost:5000