

Assignment: "Interactive Resume Builder with Advanced React Features"

Objective:

Develop an **Interactive Resume Builder** application that:

1. Lets users input their details (e.g., Name, Education, Work Experience, Skills, and Projects).
2. Utilizes **state management** (React Context, Redux, or Zustand) to manage user data globally.
3. Allows users to **preview their resume** dynamically with different layouts.
4. Includes **theme switching** (Light and Dark mode).
5. Lets users **save the resume as a PDF**.
6. Includes **technical documentation** explaining the architecture and tech stack.

Guidelines:

1. **Originality:**
 - Do **not copy-paste code** from online sources or existing repositories.
 - Your submission will be checked using our **AI-powered Copy detection tool**, and copied code will result in immediate disqualification.
 2. **Documentation:**
 - Along with the code, submit a technical document (PDF or Markdown) explaining:
 - **Overview** of your project.
 - **Tech Stack** used.
 - Explanation of key **React concepts** applied (e.g., State Management, Hooks).
 - Challenges you faced and how you solved them.
 3. **Deployment:**
 - Deploy the application to a hosting platform like **Vercel, Netlify, or GitHub Pages**.
 - Submit the live application URL and a link to the GitHub repository containing your code.
 4. **Clean Code:**
 - Ensure modular, clean, and well-documented code.
 - Use proper folder structures and include comments where necessary.
 5. **Time Management:**
 - This assignment is designed to be completed in **one day** with focused effort.
-

Assignment Requirements:

1. Features:

- **User Inputs:**
 - Form fields for:
 - **Personal Info:** Name, Email, Phone, LinkedIn, GitHub.
 - **Education:** Institution, Degree, Year of Passing.
 - **Work Experience:** Company, Role, Duration, Achievements.
 - **Skills:** Multi-select dropdown.
 - **Projects:** Title, Description, Technologies Used.
- **State Management:**
 - Use a global state management tool (React Context, Redux etc) to manage user data.
- **Dynamic Resume Preview:**
 - Display a real-time preview of the resume as the user fills out the form.
 - Provide at least **two layout options** for the resume.
- **Theme Switching:**
 - Include Light and Dark mode themes.
 - Persist the user's choice using **localStorage**.
- **Save as PDF:**
 - Allow users to export the resume as a PDF using libraries like react-to-print or jspdf.

2. Deployment:

- The application must be deployed on a platform such as **Vercel**, **Netlify**, or **GitHub Pages**.
- Share the **live application URL** and **GitHub repository link**.

Evaluation Criteria:

1. **Originality:** Any detected plagiarism will result in disqualification.
2. **Functionality:** All features should work as described.
3. **React Concepts:** Proper usage of state management, Hooks, and reusable components.
4. **UI/UX:** Aesthetic, user-friendly, and responsive design.
5. **Documentation:** Clear explanation of the project in the technical document.
6. **Code Quality:** Clean, modular, and readable code.

Consequences of Copying Code:

- We use advanced AI-based tools to detect originality in both code and documentation.
- If any portion of your submission is found to be copied:
 - Your application will be **disqualified immediately**.
 - This may also affect your future opportunities with our organization.