CV Builder

Project design report

Table of contents

1. Introduction

This project is a web-based CV creation and management system designed for efficient CV creation, editing, and management. Key features include user registration, authentication, real-time CV editing, and PDF download capabilities. The system allows users to add and edit sections like Work Experience and Skills, while offering customizable templates. It is responsive across all devices, ensuring smooth usability on desktop, tablet, and mobile. The platform prioritizes ease of use, offering a seamless interface and functionality for both beginners and experienced users. The following report covers design and technical considerations for these features.

2. System Architecture

2.1. Front-End:

- Framework: React.js with TypeScript
- Styling: Bootstrap for responsive design and layout.
- State Management: React Context API for handling global states such as user authentication and CV data.
- API Interaction: Axios for HTTP requests to the back-end.

2.2. Back-End:

- Framework: Django for building the REST API.
- Authentication: JWT (JSON Web Token) for secure authentication and session management

2.3. Database:

- Database Management System:sqlite for storing user profiles, CV data, and related information.

2.4. PDF Generation:

- Service: A third-party library (react pdf maker) for generating downloadable PDF files from user CV data.

3. Functional Requirements

3.1. User Registration and Authentication

- **Overview:** Users will register and authenticate using an email and password. Once logged in, they can view and manage their account details.

- **Design Considerations:**
  - **Registration Form**: A simple form with input validation (e.g., strong password enforcement).
  - **Login Flow**: Token-based authentication using JWT to manage secure user sessions.
  - **Profile View**: Users can view and update their profile information after logging in.

- **UI Design**:
  - **Registration/Login**: A clean, intuitive form designed using Bootstrap's grid system, ensuring responsiveness.
  - **Profile Page**: Display user details in a neatly structured card layout.

3.2. Create CV

- **Overview**: Users can create CVs by filling out a user-friendly form, adding sections such as Work Experience, Education, Skills, and more.

- **Design Considerations:**
  - Form Design: A multi-step form to streamline the CV creation process. Users can save drafts and return later to complete their CVs.
  - Sections: Each section (e.g., Work Experience, Education) will be a collapsible form, allowing users to add multiple entries.
  - Real-Time Updates: CV data will auto-save as users enter it, preventing data loss.

- **UI Design**:
  - **Form Layout:** Use Bootstrap form components for input fields, with clear labels and placeholders.
  - **Preview:** Real-time preview on the right side, allowing users to see how their CV will appear.

3.3. Edit CV

- **Overview**: Users can edit CVs via an intuitive interface and reorder sections. The system provides various templates and styling options.

- **Design Considerations:**
  - **Reordering Sections**: Users can drag and drop sections to rearrange the layout.
  - **Template Selection**: Users can preview different template styles and apply them to their CV.

- **UI Design:**
  - **Search Bar**: A responsive search bar placed at the top of the CV management page.
  - **Filter Panel**: A sidebar or dropdown with filter options based on template styles.

3.4. Download CV

■ **Overview**: Users can download their CVs as PDFs and customize the filename.

■ **Design Considerations:**

- **PDF Download**: Provide a button that triggers the PDF generation and allows users to name their CV.

- **File Naming**: The user can provide a custom filename before downloading.

■ **UI Design:**

- **Download Button**: A prominent button labeled "Download as PDF," placed below the real-time preview area.

3.5. Responsive Design

- **Overview**: The application should work seamlessly across different screen sizes.

- **Design Considerations:**
  - **Grid System**: Utilize Bootstrap's responsive grid system to ensure the layout adjusts automatically on various devices.
  - **Touch Optimizations**: Ensure that buttons and forms are large enough for touch interactions on mobile devices.

- **UI Design:**
  - **Responsive Form Layout**: Inputs and preview panels should stack vertically on smaller screens and remain side by side on desktops.

3.6. User Profile

- **Overview**: Users can manage their profiles, view past CVs, and update personal details.

- **Design Considerations:**
  - **Profile Management:** The profile page will include options to update the user's name, email, password, and display the history of created CVs.
  - **CV History:** Users can view a list of their past CVs and open any for editing.
  -

- **UI Design:**
  - **My Cv Page**: Use card components to display embedded list of CV history.

3.7. Search and Filter CVs

- **Overview**: Users can search and filter their created CVs based on keywords .
- **Design Considerations:**
  - **Search Input: A search bar will allow users to filter CVs by name or content.**
- **UI Design:**
  - **Search Bar**: A responsive search bar placed at the top of the CV management page.

3.8. Responsive Design

- **Overview**: The application should work seamlessly across different screen sizes.
  - **Design Considerations:**
    - **Grid System**: Utilize Bootstrap's responsive grid system to ensure the layout adjusts automatically on various devices.
    - **Touch Optimizations**: Ensure that buttons and forms are large enough for touch interactions on mobile devices.
  - **UI Design:**
    - **Responsive Form Layout**: Inputs and preview panels should stack vertically on smaller screens and remain side by side on desktops.

4.   Technical Challenges and Solutions

4.1.   PDF Generation: Implementing a third-party library for PDF generation may involve layout challenges for different templates. To address this, we'll design a consistent structure that can adapt to all template formats.

4.2.   Real-Time Editing and Preview: Implementing real-time updates while maintaining performance is crucial. Debouncing techniques will be used to optimize input handling.

4.3.   Responsive Design: Ensuring all components look and behave properly across devices will require rigorous testing with Bootstrap's media queries and layout utilities.

5.   User Experience (UX) Considerations

5.1.   Simple and Intuitive UI: The overall design will focus on ease of use, ensuring that the flow from registration to CV download is seamless.

6.   Conclusion

The CV creation and management system provides users with a comprehensive yet easy-to-use tool for building, editing, and managing CVs. With a focus on responsive design, real-time functionality, and a user-friendly interface, this system ensures a smooth and engaging experience for users across all devices.