



Project
on
Resume Builder for Freshers

Final Report
Submitted to
Ms. HEMANGINI MEHTA
Assistant Professor
P P Savani University,
Surat

Contributors:

MO FAHIM RAI(23SS02IT161)
PARTH PUROHIT(23SS02IT157)
BHAVYA RAMANI(23SS02IT168)
SUCHIT PATEL(23SS02IT139)

BACHELOR OF SCIENCES IN INFORMATION
TECHNOLOGY(B.SC.-IT)

Institute of Computer Science and Applications

P P Savani University

November, 2025

ABSTRACT

The Resume Builder project is a full-stack web application that makes creating professional resumes easier. In today's competitive job market, a resume is often the first contact point between job seekers and employers. This makes it an important tool for career advancement. However, many job seekers, especially students and recent grads, find it hard to format and design resumes that are both professional and impactful. This project aims to fill that gap by offering a simple, interactive, and customizable platform. It allows users to enter their information, choose from pre-designed templates, and generate a professional resume in PDF format.

The application uses modern web technologies. React.js powers the frontend, while Redux Toolkit manages the state. Node.js and Express.js handle backend services, and MongoDB manages the database. Tailwind CSS creates a responsive and visually appealing user interface. Libraries like jsPDF and react-pdf enable the downloading and exporting functions. The combination of these technologies helps ensure the application is scalable, maintainable, and performs well across devices.

The project used an iterative development approach. It focused on multiple milestones, including frontend design, backend integration, error handling, state management, and export features. Each phase improved both usability and stability. The outcome is a fully functional application that meets its initial goals and sets the stage for future innovations. Possible upgrades, such as AI-driven content suggestions, cloud synchronization, and support for multiple languages, could help the Resume Builder grow into a complete career-building tool.

INDEX

1. INTRODUCTION

- 1.1 Introduction
- 1.2 Purpose
- 1.3 Scope
- 1.4 Overview
- 1.5 Technology Used
- 1.6 Project Requirements
- 1.7 Problem

2. FEASIBILITY STUDY

- 2.1 Technical Feasibility
- 2.2 Economical Feasibility
- 2.3 Operational Feasibility
- 2.4 Schedule Feasibility

3. SYSTEM ANALYSIS

- 3.1 Existing/Present System
- 3.2 Proposed System
- 3.3 Architecture Design
- 3.4 Data Flow Diagram
- 3.5 UseCase Diagram
- 3.6 Class Diagram
- 3.7 ER Diagram

4. SYSTEM DESIGN

- 4.1 System Design
- 4.2 System Architecture
- 4.3 Database Design and Workflow

5. IMPLEMENTATION AND TESTING

6. CHALLENGES FACED AND SOME SNAPSHOTs

7. CONCLUSION AND FUTURE SCOPE

REFERENCES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Resume is the first meeting between you and a prospective employer more often now than ever. So, how do you want to be remembered ? Wrinkled and unorganized. Neat and structured. Long and boring. Precise and interesting. Companies do not have the time to interview every applicant that is interested in the job. If they did, there would not be a company to work for. They use an eliminating process. That's right - resumes. When a job seeker wants to apply for a job online then generally he/she needs to attach his/her resume with the email. Online Resume Builder System provides the users the popular resume formats & a better way to show their resumes to the employers. A job seeker does not need to attach a resume with every email, he/she just has to include the URL of his/her resume and the employer can view the resume online by clicking on the link and can download as well.

1.2 Purpose

The purpose of Online Resume Builder is to provide a way to the customers to design their resumes according to their requirements.

The core functionalities of this system are designed to offer a comprehensive and user-friendly experience:

- **Creating resumes online :** The platform provides a streamlined process for users to build their resumes from scratch or by utilizing pre-designed templates. This includes guided input for personal details, work experience, education, skills, and other relevant sections, ensuring a complete and professional document.
- **Customizing the look and details:** Beyond basic data entry, the Online Resume Builder offers extensive customization options. Users can control the visual aesthetics of their resumes, including font styles, colors, layouts, and thematic designs.
- **Keeping track of the customers and their resumes:** A critical aspect of the system is its ability to efficiently manage both customer profiles and their associated resumes. This includes secure user accounts, enabling individuals to save, edit, and access multiple versions of their resumes at any time.

1.3 Scope

Online Resume Builder can be used in accordance with the requirements of the customers. Customers can customize their resumes with their choice of themes &

details. The services are hard to be defeated by the competitors as the system is providing the customers exactly what they want.

1.4 Overview

This project maintains 3 types of users.

- Administrator User
- Users(Customers)
- Viewers

Facilities provided by this projects are as follows:

- Details of customers are recorded.
- Updating data is easy.
- Flow of information is fast and easy.
- Customers can login to their accounts and view & update their data.
- Notifications about resume views & downloads

1.5 Technology Used

Frontend: React.js, Tailwind CSS

Backend: Node.js, Express.js

Database: MongoDB (with Mongoose ODM)

Authentication: JSON Web Tokens (JWT)

PDF Generation: A library such as **html2pdf.js** or a server-side solution.

Deployment: Vercel (Frontend), Render (Backend), MongoDB Atlas (Database).

1.6 Project Requirements

Software Requirements

Category	Details
Operating System	Win-98, Win-XP, Linux, or any other higher version
Web Browser	Internet Explorer, Mozilla Firefox, or any web browser

Hardware Requirements

Category	Details
Processor	Pentium II, Pentium III, Pentium IV or higher
RAM	64MB or Higher

1.7 Problem

Fresh graduates often face significant hurdles when attempting to build their first resume. The most common challenges include:

- **Lack of Content Knowledge:** Uncertainty about what sections to include and how to articulate their skills, projects, and internships effectively.
- **Formatting and Design Issues:** Difficulty in creating a layout that is both visually appealing and compliant with Applicant Tracking Systems (ATS), which are widely used by employers to screen candidates.
- **Generic and Ineffective Language:** Using weak or passive language instead of powerful action verbs and industry-specific keywords that capture an employer's attention.
- **One-Size-Fits-All Approach:** Creating a single, generic resume and using it for all job applications, which fails to target the specific requirements of different roles.

Existing tools like standard word processors are often too manual and lack the guidance required, while some online builders are overly complex or expensive. This creates a clear need for a specialized, user-friendly tool tailored for freshers.

CHAPTER 2

FEASIBILITY STUDY

Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. FEASIBILITY STUDY is a test of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. It focuses on these major questions: What are the users demonstrable needs and how does a system meet them? What resources are available for a given system? What are the likely impacts of the system on the organization? Is it worth solving the problem? During feasibility analysis for this project, following primary areas of interest are to be considered. Investigation and generating ideas about a new system does this.

2.1 Technical Feasibility

Technical feasibility is the study of resource availability that may affect the ability to achieve an acceptable system. This evaluation determines whether the technology needed for the proposed system is available or not. Can the work for the project be done with current equipment, existing software technology & available personnel? Can the system be upgraded if developed? If new technology is needed then what can be developed?

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may include:

Front-end and back-end selection:

An important issue for the development of a project is the selection of suitable front-end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the academy as well as helps in development of the project. The aspects of our study included the following factors.

Front-end selection:

- It must have a graphical user interface that assists users that are not an advanced user of the computer.
- Scalability and extensibility.
- Flexibility.
- Robustness.

According to the organization requirement and the culture. Must provide excellent reporting features with good printing support. Platform independent. Easy to debug and maintain. Event driven programming facility.

Front-end must support some popular back end like Ms Access. According to the above stated features we selected Web Browser as the front-end for developing our project.

Back-end Selection:

- Multiple user support.
- Efficient data handling.
- Provide inherent features for security.
- Efficient data retrieval and maintenance.
- Stored procedures. Popularity.
- Operating System compatible.
- Easy to install.
- Various drivers must be available.
- Easy to implant with the Front-end.

According to the above stated features we selected MY SQL as the backend. Technical feasibility is frequently the most difficult area encountered at this stage.

It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed system.

2.2 Economical Feasibility

The development costs are minimal as the project relies entirely on open-source technologies. Hosting costs can be kept low using platforms like Vercel for the frontend and Render or MongoDB Atlas for the backend and database.

2.3 Operational Feasibility

It is mainly related to human organizations and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organization structures are disturbed?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

The system is operationally feasible as it is very easy for the End users to operate it.

2.4 Schedule Feasibility

Time evaluation is the most important consideration in the development of a project. The time schedule required for the development of this project is very important since more development time affects machine time, cost and causes delay in the development of other systems. Online Resume Builder can be developed in a considerable amount of time.

CHAPTER 3

SYSTEM ANALYSIS

System analysis uses a combination of text and diagrammatic forms to depict requirements for data, function and behavior in a way that is relatively easy to understand, and more important, straightforward to review for correctness, completeness and consistency

3.1 Existing/Present System

In the existing system, most users rely on manual methods or basic online templates to create their resumes. These traditional approaches often lack flexibility, automation, and customization options. Users typically use tools such as Microsoft Word, Google Docs, or generic resume template websites that offer limited personalization and no data management capabilities.

The existing system has several limitations:

1. Manual Data Entry:

Users must repeatedly enter their personal, educational, and professional details every time they create or update a resume.

2. Limited Customization:

Predefined templates provide restricted design options and do not allow users to dynamically add or remove sections like multiple projects or certifications.

3. No Centralized Data Storage:

There is no mechanism to securely store user data or retrieve previous versions of resumes, leading to loss of information and redundancy.

4. Lack of Automation:

The process of generating and formatting resumes is fully manual, making it time-consuming and prone to errors.

5. Inconsistent Formatting:

Different devices or software versions may display the same resume differently, causing formatting inconsistencies.

6. No Integration with Modern Technologies:

Existing systems generally lack features such as real-time preview, data synchronization, or export in multiple formats (PDF, DOCX, etc.).

3.2 Proposed System

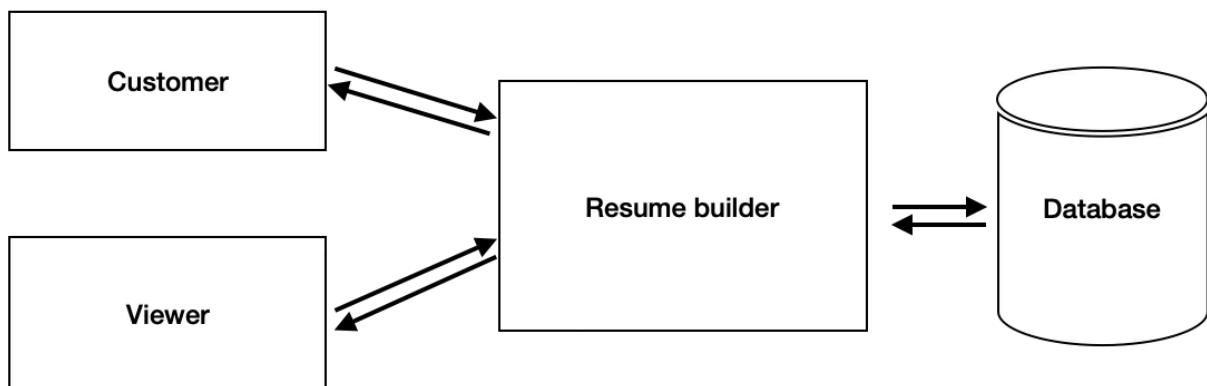
The proposed system is a dynamic, MERN-stack-based web application specifically designed for freshers. It addresses the shortcomings of existing systems by offering:

- **A Guided User Experience:** The system breaks down the resume into logical sections (Education, Projects, Skills, etc.) and uses an intuitive form-based interface to collect information.
- **Professionally Designed Templates:** A curated selection of modern, ATS-friendly templates that emphasize skills and projects.
- **Real-Time Preview:** A live preview panel that renders the resume as the user enters data, providing immediate visual feedback.
- **Centralized Management:** A user dashboard where multiple versions of a resume can be created, stored, edited, and downloaded in PDF format.

3.3 Architecture Design

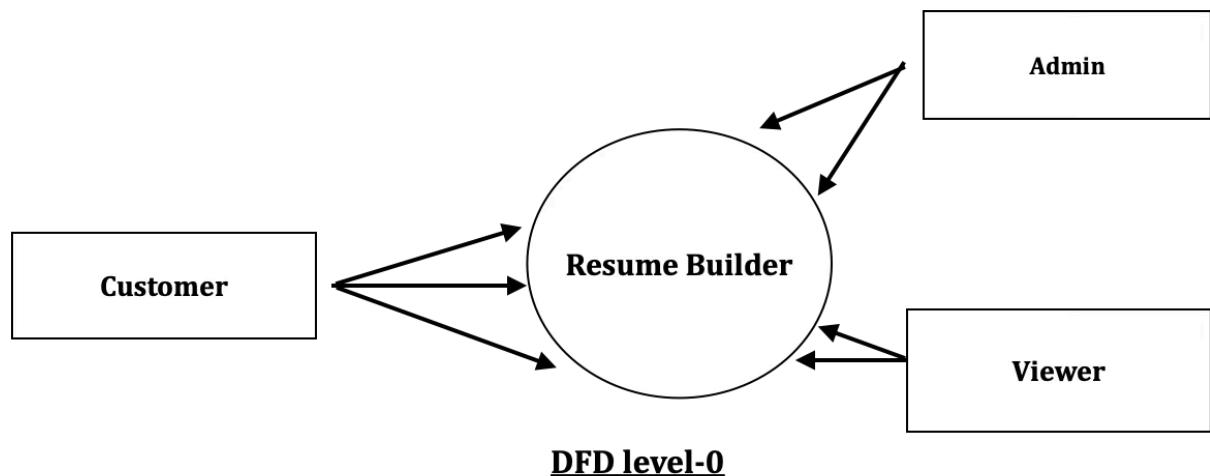
Architectural design represents the structure of data and program components that are required to build a computer-based system. It considers the architectural style that the system will take, the structure and properties of the components that constitute the system, and the interrelationships that occur among all architectural components of a system.

Architecture Design

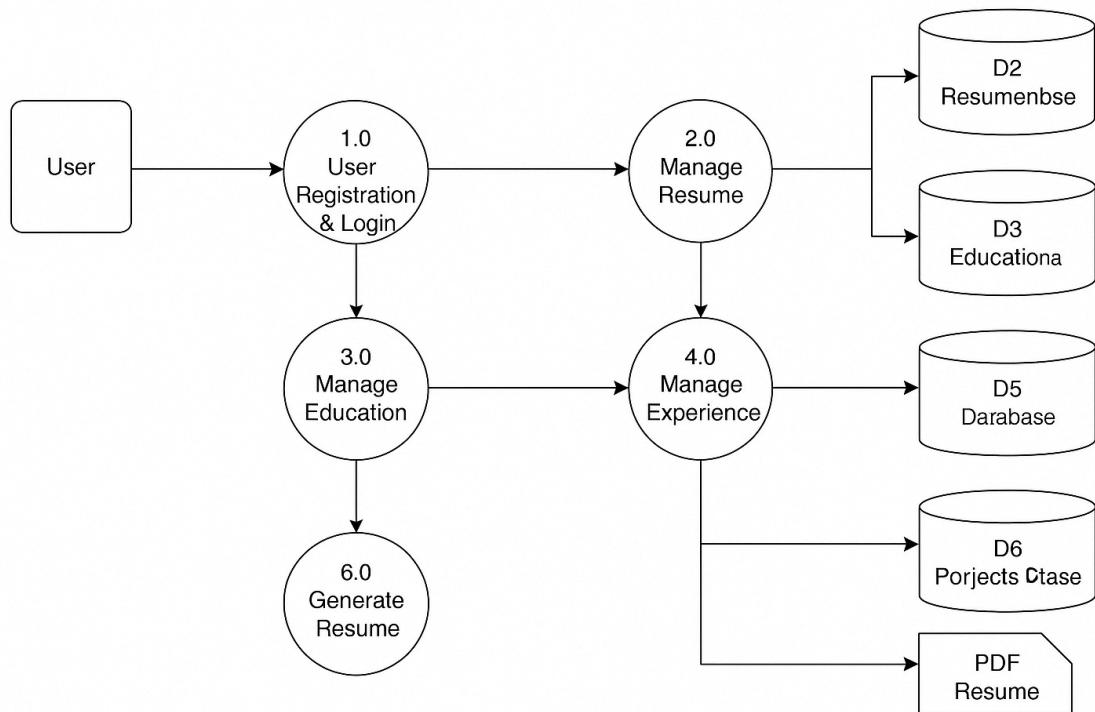


3.4 Data Flow Diagram

The data flow diagram enables the software engineer to develop models of the information domain and functional domain at the same time. As the DFD is refined into greater levels of detail, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in corresponding refinement of data as it moves through the processes that embody the application.



DFD Level 1: Resume Builder System



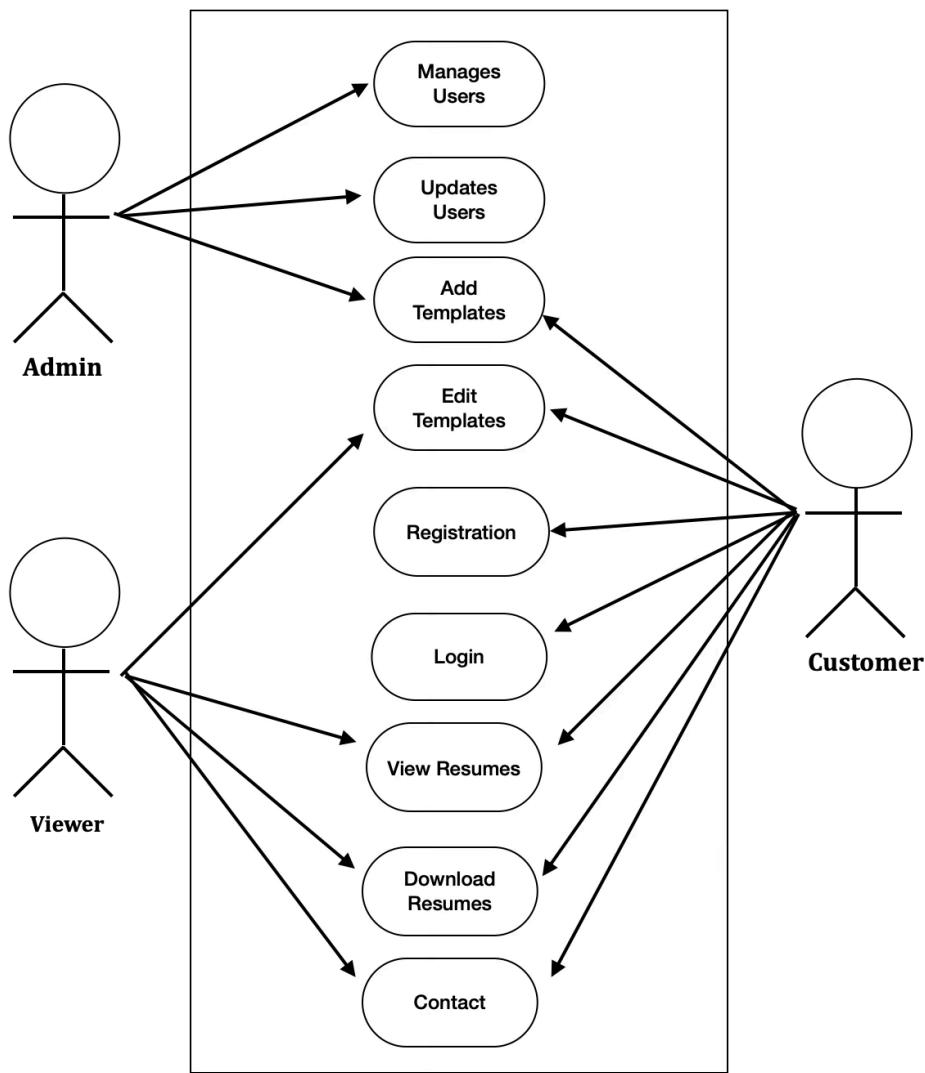
3.5 UseCase Diagram

Use Case Model is an approach that is a combination of text and pictures in order to improve the understanding of requirements. A use case model describes the complete functionality of a system by identifying how everything that is outside the system interacts with it. A Use Case Diagram is given below that relates to this application.

This project is a web application that manages a system of building resumes online. Actors It has 3 actors.

- 1) Administrator

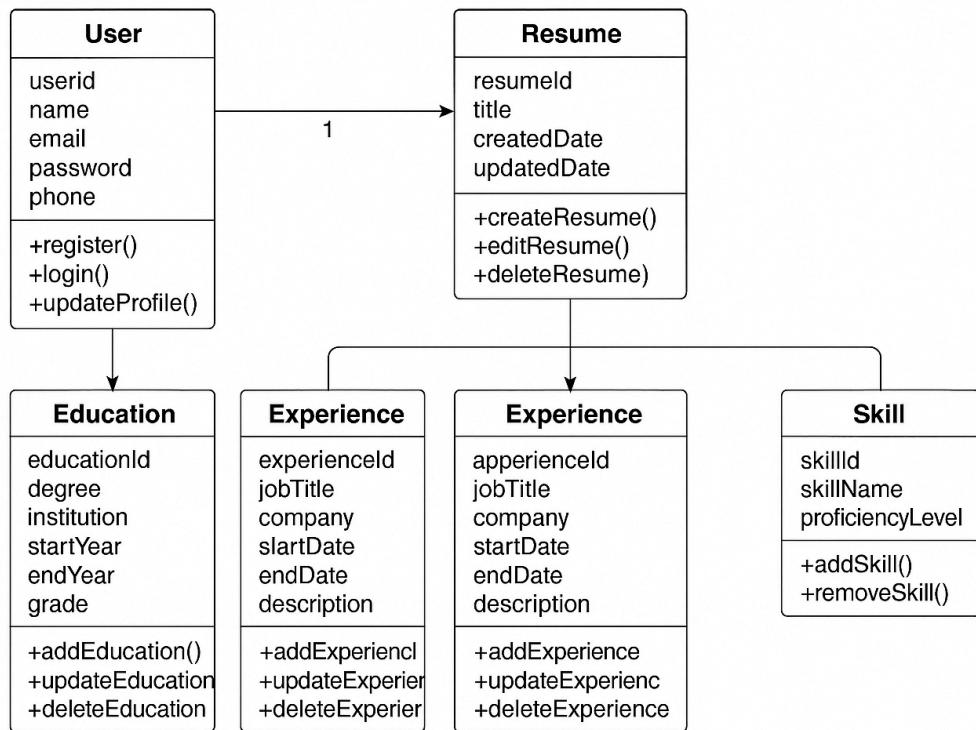
- 2) Customer
3) Viewer



UseCase Diagram

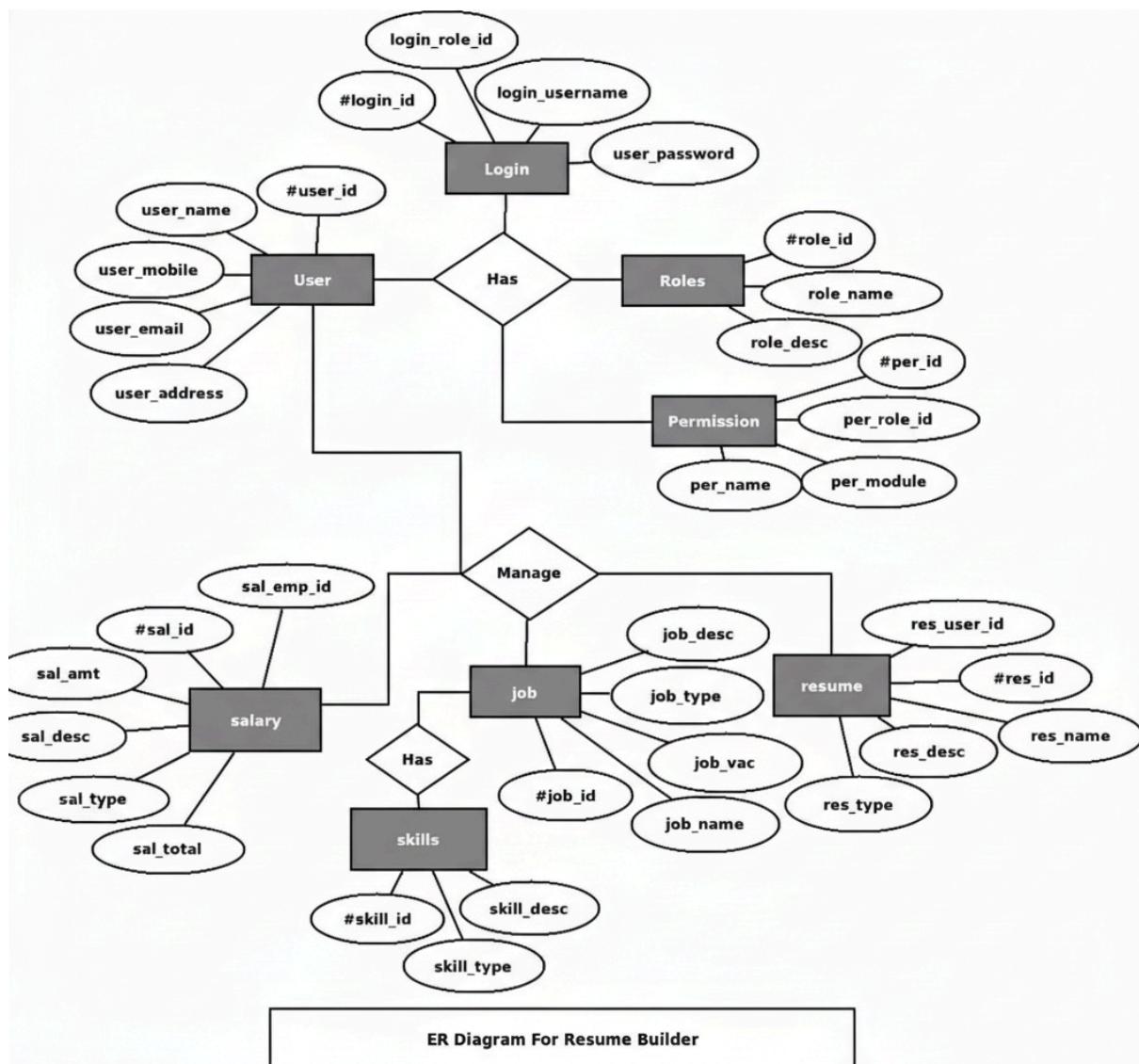
3.6 Class Diagram

A class diagram shows relationships among classes used in the system. Every class has some contents, attributes and objects associated with it. Every class produces some outputs and uses some inputs. Outputs generated are consumed by other classes and inputs consumed are generated by other classes. The class diagram shows which class is producing what and consuming what.



3.7 ER Diagram

The object/relationship pair is the cornerstone of the data model. These pairs are represented graphically using E-R diagrams. A set of primary components are identified for the ERD: data objects, attributes, relationships and various type indicators. The primary purpose of ERD is to represent data objects and their relationships.



CHAPTER 4

SYSTEM DESIGN

4.1 System Design

System design is a solution, a way to approach the creation of a new system. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Emphasis is on translating the performance requirements into design specifications. Design goes through logical and physical stages of development. Logical design reviews the present physical system; prepared input and output specifications; details the implementation plan; and prepares a logical design walkthrough. The physical design maps out the details of the physical system, plans the system implementation, devises a test and implementation plan, and specifies any new hardware and software.

4.2 System Architecture

The application employs a three-tier architecture, which is standard for MERN applications:

1. **Client Tier (Presentation Layer):** A React-based Single Page Application (SPA) that runs in the user's browser. It handles the UI, manages frontend state, and communicates with the server via RESTful API calls.
2. **Application Tier (Logic Layer):** A Node.js and Express.js server that acts as the backend. It contains the business logic, handles API requests from the client, processes data, manages user authentication, and interacts with the database.
3. **Data Tier (Database Layer):** A MongoDB database that stores all application data, including user credentials and resume information, in a flexible, JSON-like BSON format.

4.3 Database Design and Workflow

1. Request Flow

- 1.Client makes HTTP requests to the Express server
- 2.Requests pass through middleware stack:

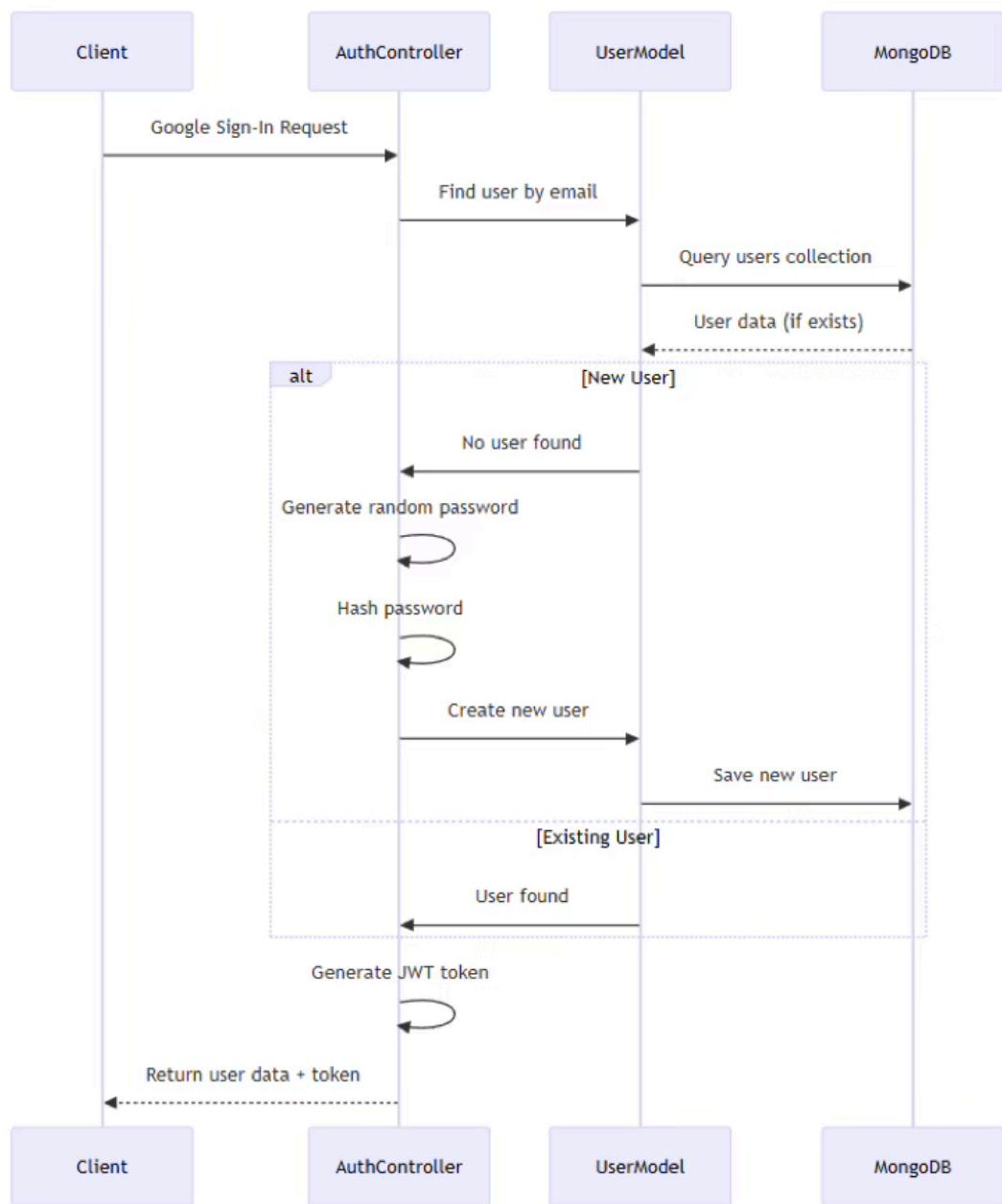
CORS handling

JSON parsing

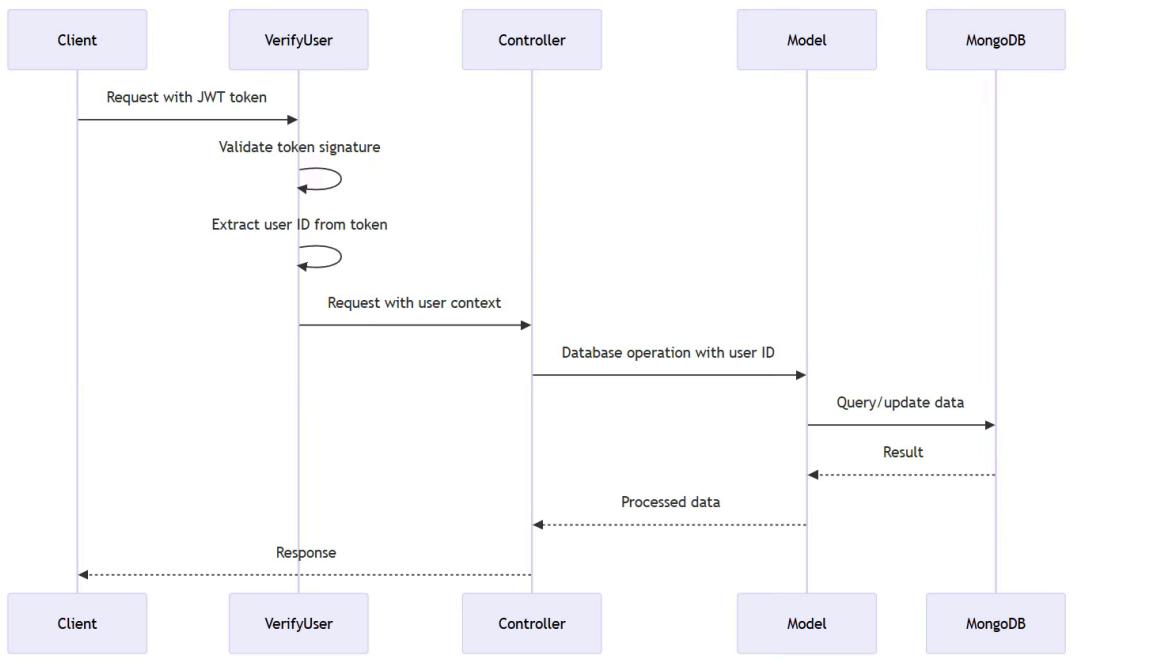
Authentication (for protected routes)

Logging

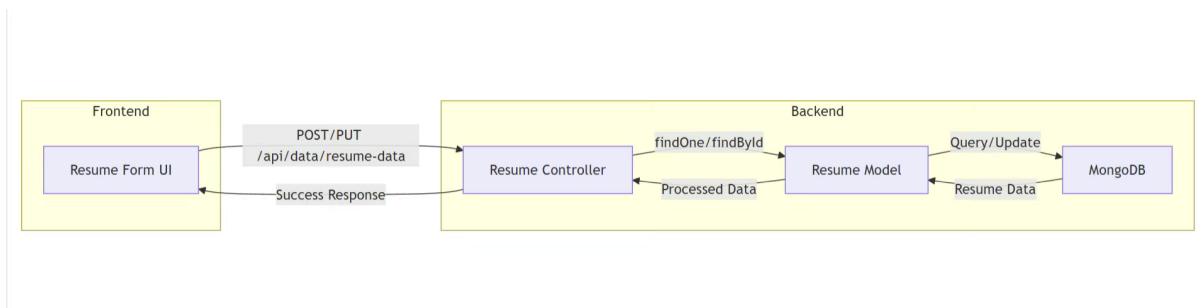
2.Authentication Process



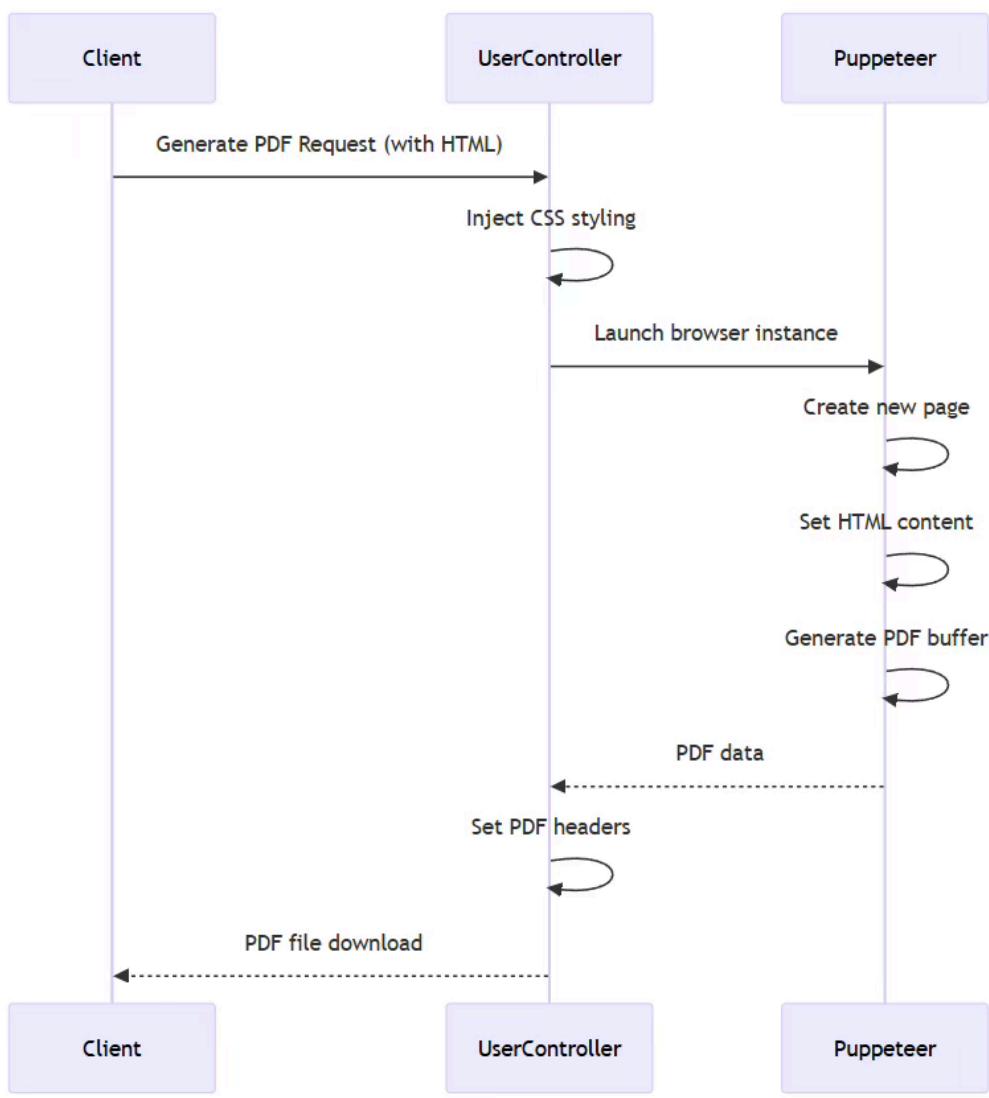
3. Protected Route Access



4.Resume Data Management



5.PDF Generation Process



Key Components Interaction

1. Database Models

- **User Model:** Stores user credentials and profile information
- **Resume Model:** Stores all resume data linked to users via userId

2. Security Layers

- **JWT Authentication:** Token-based authentication for all protected routes
- **Password Hashing:** BCrypt for secure password storage
- **Route Protection:** Middleware verifies tokens before allowing access

3. File Generation

- **Puppeteer:** Headless Chrome browser for converting HTML to PDF
- **CSS Injection:** Dynamic styling applied to resume templates

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

The application was developed in a modular fashion. React components were created for each distinct UI element (e.g., forms, buttons, preview pane). The backend was built using an MVC (Model-View-Controller) pattern to organize the code, with routes, controllers, and models separated for clarity and maintainability. RESTful API endpoints were established for all CRUD (Create, Read, Update, Delete) operations related to resumes.

5.2 Testing

A multi-level testing strategy was employed to ensure the application's quality and reliability.

- **Unit Testing:** Individual components (especially helper functions and API utility functions) were tested in isolation using frameworks like Jest.
- **Integration Testing:** The interaction between the frontend and backend was tested. API endpoints were tested using tools like Postman to verify that data was being correctly sent, received, and processed.
- **System Testing:** End-to-end testing was performed manually from a user's perspective. This involved testing the entire application workflow, from user registration to downloading a PDF, to ensure all modules work together as expected. Usability testing was also conducted to gather feedback on the user interface.

CHAPTER 6

CHALLENGES FACED AND SNAPSHOTTS

6.1 Challenges Faced

During the development of the **Resume Builder** project, several technical and implementation challenges were encountered and successfully addressed. The key challenges included:

1. Debugging PDF Generation:

The process of generating and exporting resumes in PDF format posed difficulties due to inconsistent formatting and layout issues across devices. Adjustments in CSS styling and export logic were required to ensure professional and uniform output.

2. Delay in Login System Implementation:

The authentication setup initially caused delays as integrating secure login and registration functionality demanded additional configuration and debugging to ensure user data protection.

3. Handling Dynamic Resume Sections:

Managing variable-length sections such as multiple projects, education entries, or skill sets required dynamic rendering logic and efficient state management to maintain a seamless user experience.

4. Migration to Redux Toolkit:

The transition to Redux Toolkit for state management required restructuring parts of the frontend logic. This migration improved scalability but involved refactoring and debugging existing code.

5. Synchronization Between Redux State and Backend API:

Ensuring accurate synchronization between the frontend state and backend API responses was a challenging task. Several debugging sessions were required to maintain data consistency across components.

6. Formatted Data Export for Download Feature:

Implementing the download feature with properly formatted data exports, especially for structured sections like experience and education, remained technically demanding and required fine-tuning.

7. Authentication Issues with Initial Setup:

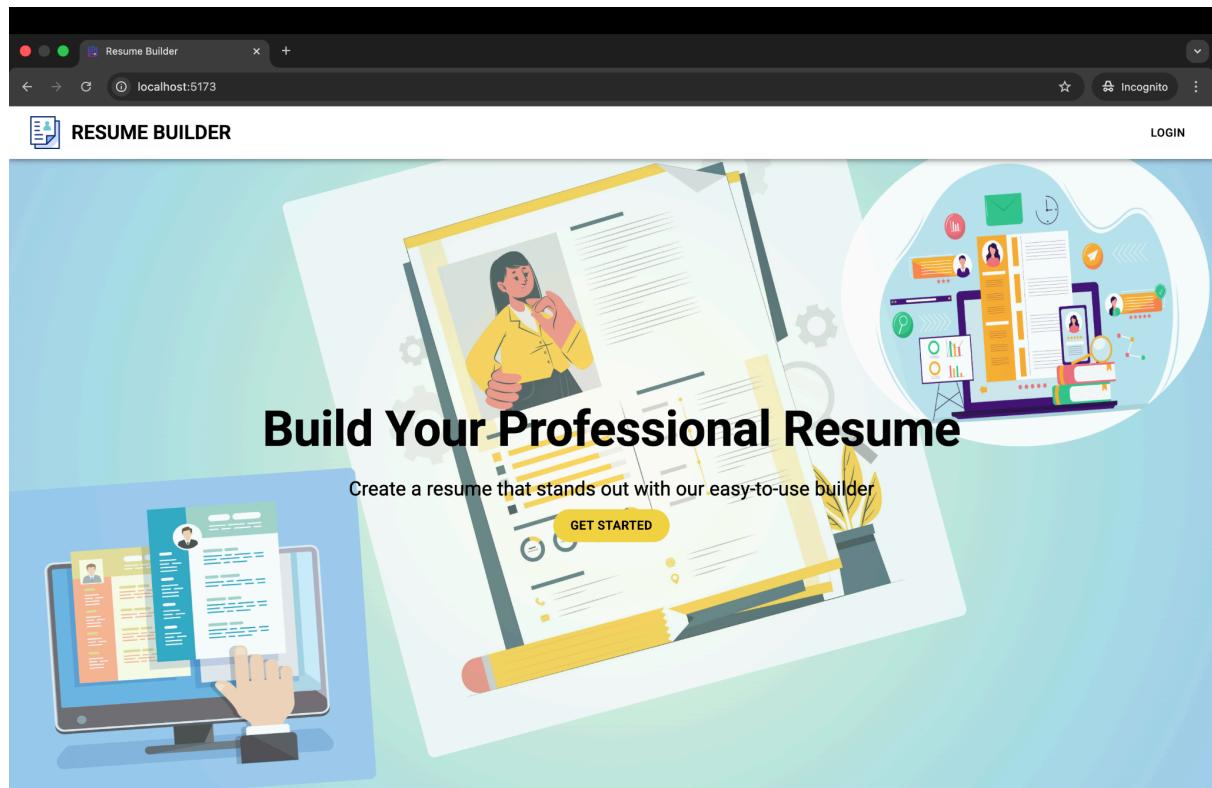
The initial authentication system encountered persistent errors, which were ultimately resolved by switching to Firebase Authentication, offering smoother integration and better reliability.

8. UI Rendering and Responsiveness:

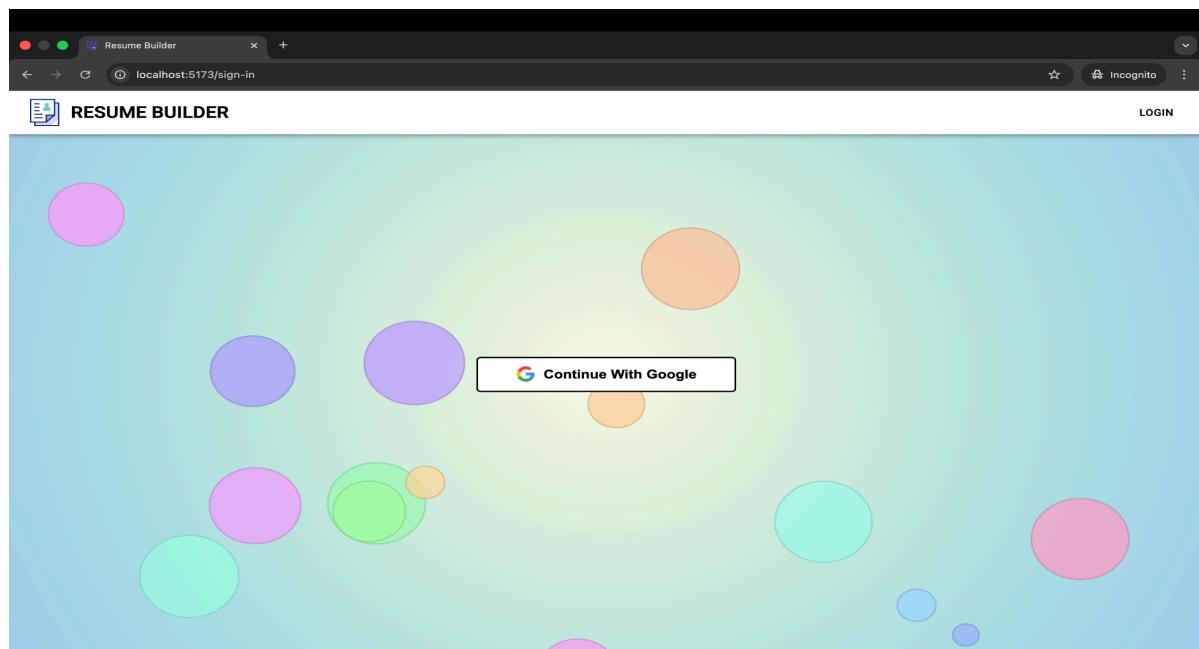
Some custom components initially faced rendering issues, particularly on different screen sizes. Adopting Material UI's responsive design system helped resolve these problems, ensuring a consistent and modern user interface.

6.2 Snapshots

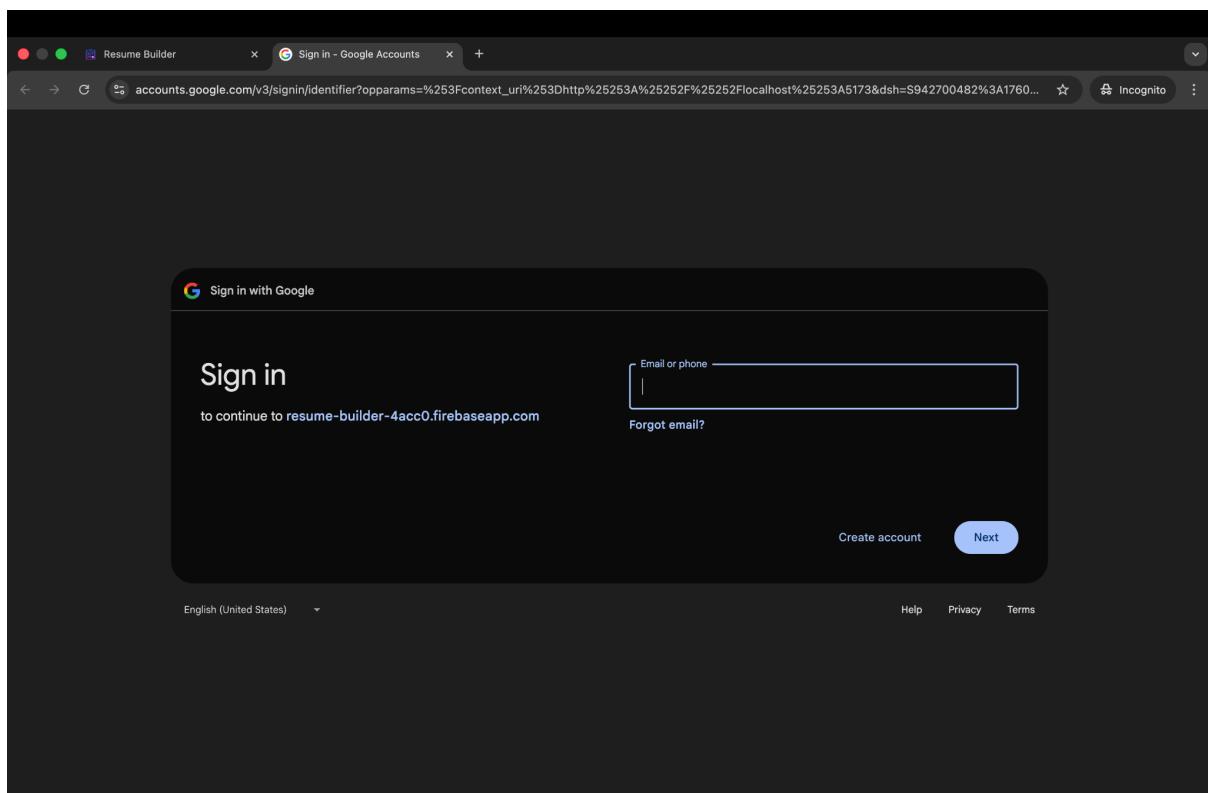
Home Page



Login



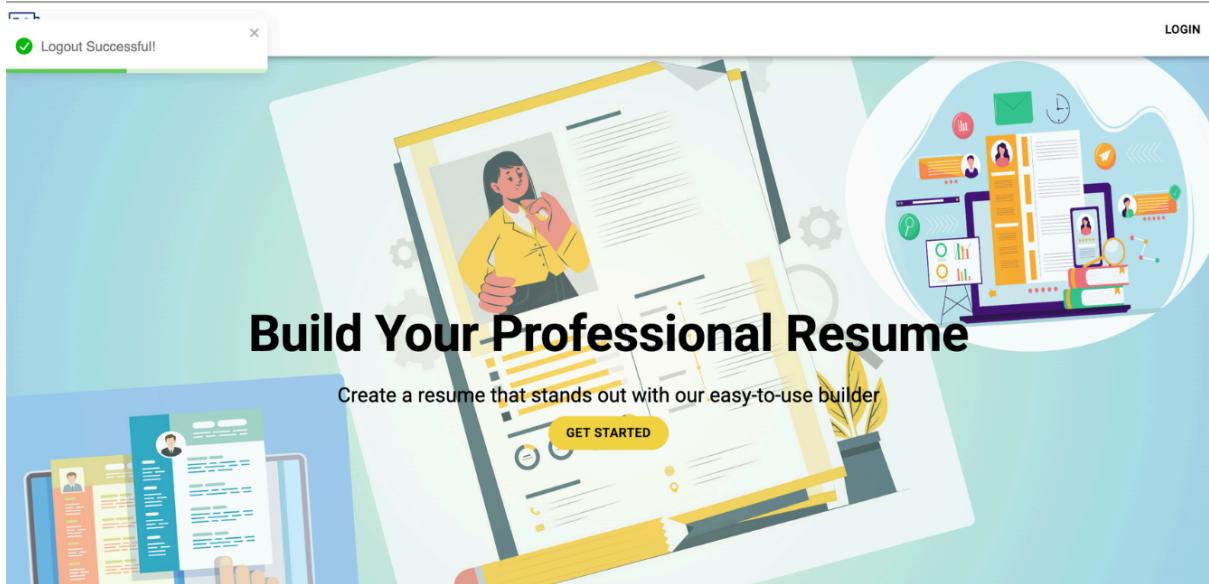
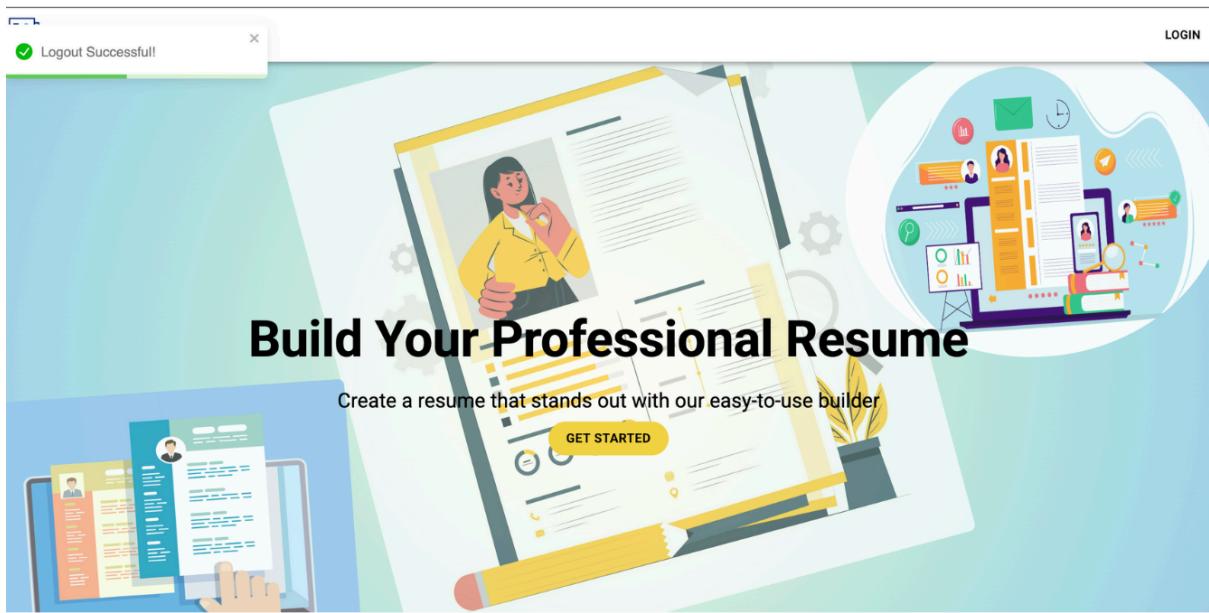
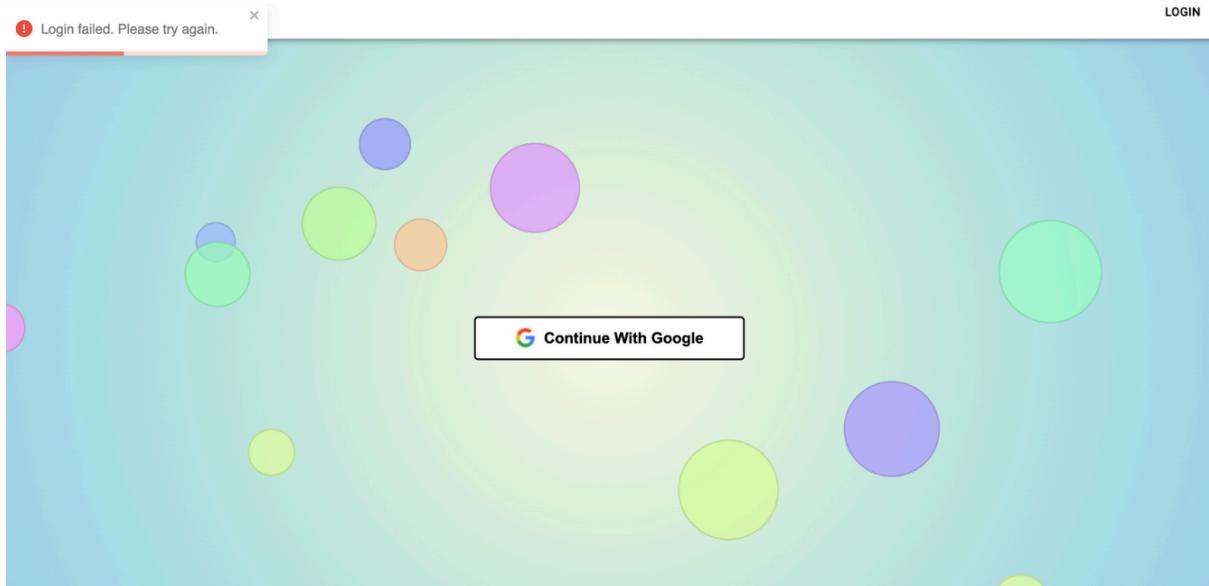
Sign in with Google ID



Form preview

A screenshot of the "Personal Details" section of the Resume Builder application. The title "Personal Details" is centered at the top. Below it is a grid of input fields. The first row contains two fields: "FirstName*" and "LastName*". The second row contains "Email*" and "MobileNo*". The third row contains "Address" with a house icon. The fourth row contains "LinkedIn" with a LinkedIn icon and "Github" with a GitHub icon. The fifth row contains "Codechef" with a Codechef icon and "Leetcode" with a Leetcode icon. The sixth row contains "AboutMe" with a double arrow icon. At the bottom right of this section, there is a link "Education Section →". Above the "Personal Details" section, there is a navigation bar with a logo, the text "RESUME BUILDER", a "SECTIONS" button, and a red circular icon with a white letter "F".

Login Authentication



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The "Resume Builder" project successfully achieves its goal of providing freshers with a simple, effective, and modern tool for creating professional resumes. By leveraging the MERN stack, the application delivers a responsive, scalable, and feature-rich user experience. It effectively addresses the common pain points faced by fresh graduates, empowering them to craft compelling resumes that stand out to potential employers. The project demonstrates a practical application of modern web development technologies to solve a real-world problem.

7.2 Future Scope

While the current application is fully functional, there are several avenues for future enhancement:

- Enhanced AI Features: Integrate a more advanced NLP model to provide contextual suggestions, check for grammatical errors, and score the resume against specific job descriptions.
- Cover Letter Generator: Add a new module for creating cover letters that match the selected resume template and user data.
- Job Application Tracker: Implement a feature allowing users to track the jobs they have applied for using the resumes created on the platform.
- Expanded Template Library: Continuously add new and diverse templates to cater to different industries and roles.
- Sharing and Analytics: Allow users to generate a shareable link to their resume and provide basic analytics, such as the number of views.

REFERENCES

- I. <https://www.canva.com/resume-builder>
- II. <https://www.adobe.com/express/create/resume>
- III. <https://www.linkedin.com/resume-builder>
- IV. <https://www.resumebuilder.com/>