A
MINOR PROJECT REPORT
on
DYNAMIC RESUME BUILDER

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
Computer Science and Engineering
Submitted
by

B.MEGHANA (22UP1A0515)

Under the Guidance
of
Mrs. P. Rupa
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN
(An Autonomous Institution)

(Affiliated to Jawaharlal Nehru Technological University Hyderabad, Accredited by NBA, NAAC with A+)
Kondapur (Village), Ghatkesar (Mandal), Medchal (Dist.)
Telangana-501301

(2024-2025)

VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN
(An Autonomous Institution)
(Sponsored by Lavu Educational Society)
[Affiliated to JNTUH, Hyderabad & Approved by AICTE, New Delhi]
Kondapur (V), Ghatkesar (M), Medchal –Malkajgiri (D)- 501301.Phone:9652910002/3

NAAC A+ | Accredited by NBA CSE & ECE

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled "**DYNAMIC RESUME BUILDER**"submitted by **B.Meghana(22UP1A0515)** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**, **Vignan's Institute of Management and Technology for Women** is a record of bonafide work carried by them under our guidance and supervision. The results embodied in this project report have not been submitted to any other University or institute for the award of any degree.

PROJECT GUIDE | **THE HEAD OF DEPARTMENT**

Mrs.P.Rupa
(Assistant Professor)

Mrs. M. Parimala
(Associate Professor)

(External Examiner)

DECLARATION

I hereby declare that the results embodied in the project entitled **"DYNAMIC RESUME BUILDER"** is carried out by me during the year 2024-2025 in partial fulfillment of the award of **Bachelor of Technology** in **Computer Science and Engineering** from **Vignan's Institute of Management and Technology for Women** is an authentic record of my work under the guidance Mrs.P.Rupa. I have not submitted the same to any other institute or university for the award of any other Degree.

B.MEGHANA(22UP1A0515)

ACKOWLEDGEMENT

I would like to express sincere gratitude to **Dr G. APPARAO NAIDU Principal**, **Vignan's Institute of Management and Technology for Women** for his timely suggestions which helped me to complete the project in time.

I would also like to thank our madam **Mrs.M.Parimala, Head of the Department and Associate Professor, Computer Science and Engineering** for providing me with constant encouragement and resources which helped me to complete the project in time.

I would also like to thank our Project guide **Mrs.P.Rupa Assistant Professor, Computer Science and Engineering**, for providing me with constant encouragement and resources which helped me to complete the project in time with her valuable suggestions throughout the project. I am indebted to her for the opportunity given to work under her guidance.

My sincere thanks to all the teaching and non-teaching staff of Department of Computer Science and Engineering for their support throughout my project work.

B.MEGHANA(22UP1A0515)

INDEX

List Of Figures

ABSTRACT

In today's fast-paced, digitally driven world, the process of applying for jobs, internships, and academic positions demands a professional and well-structured resume. However, for many individuals, especially students and entry-level job seekers, creating a resume from scratch can be a daunting task due to lack of design skills, formatting knowledge, or access to professional tools. To address this challenge, our project titled "**Quick Resume Builder**" provides an intuitive and efficient solution that simplifies the resume creation process through a dynamic, theme-based web application.

The Quick Resume Builder is a user-friendly web platform developed using HTML, CSS, and JavaScript that enables users to quickly input their personal, academic, and professional details and generate a well-organized resume in real time. Unlike traditional tools, this application offers a selection of visually appealing themes — Classic, Modern, and Professional — that users can choose from, allowing for customization based on preference and industry standards. The system instantly formats the entered data and presents it in a clean and structured layout, providing a live preview before download.

A standout feature of the application is its integration with the html2pdf.js JavaScript library, which allows users to download their resumes as PDF files directly from the browser without any server-side processing. The implementation of this system also emphasizes accessibility and responsiveness, making it compatible with both desktop and mobile devices. Users do not need to create accounts or log in, making the process quick and hassle-free. The system's modular structure allows for easy future expansion, including potential features like cloud-based saving, user accounts, AI-generated content suggestions, and more.

CHAPTER 1
INTRODUCTION

1.1 Objective

The primary objective of the Quick Resume Builder project is to develop an efficient, scalable, and user-centric web-based application that simplifies the resume creation process for individuals seeking employment, internships, or academic opportunities. The system is designed to provide a professional platform that allows users to build structured, visually appealing resumes without the need for advanced technical skills or external software.

This application aims to address the common challenges associated with traditional resume-building methods by offering an intuitive interface, real-time resume preview, customizable templates, and instant PDF export functionality — all accessible within a lightweight browser environment. The goal is to enhance the user experience while ensuring the output meets modern standards of professionalism and clarity.

Key Objectives:

- To provide a clean, responsive user interface that allows users to input personal, educational, and professional information in a structured manner.
- To implement real-time resume generation with live preview capability, allowing users to visualize changes instantly.
- To offer multiple professional templates/themes to accommodate different user preferences and industry requirements.
- To enable one-click PDF export functionality using client-side JavaScript libraries (e.g., html2pdf.js) without relying on server-side processing.
- To ensure data privacy and accessibility by operating entirely on the client-side with no user registration or data storage.
- To create a modular and scalable system architecture that allows for future enhancements such as AI-based suggestions, user authentication, and cloud saving.

By achieving these objectives, the Quick Resume Builder project strives to deliver a comprehensive, modern solution that empowers users to create high-quality resumes quickly and effectively, improving their chances of making a strong first impression in the competitive job market.

1.2 Existing System

The current ecosystem of resume-building tools offers a range of options—from traditional document editors to advanced online platforms and mobile applications. However, these existing systems lack an optimal balance between usability, flexibility, functionality, and user privacy.

Traditional tools like **Microsoft Word** and **Google Docs** require users to invest significant time in formatting, design, and alignment. They are not optimized for dynamic resume generation, and the lack of professional design elements often leads to generic or visually weak resumes. Moreover, users must download external templates or create their own from scratch, which can be overwhelming for non-technical individuals.

Online platforms such as **Canva**, **Zety**, and **Novoresume** simplify resume design through drag-and-drop interfaces and pre-designed templates. While they are easy to use, they often:

- Lock key features like template access and PDF export behind paywalls.
- Require users to create accounts, raising **data privacy and security concerns**.
- Offer limited layout flexibility and lack **real-time customization**.
- Depend on internet connectivity and backend processing, reducing accessibility and performance in offline environments.

Professional tools like **Adobe Illustrator** or **InDesign** allow full control over layout and design but are not beginner-friendly. They require users to be proficient in graphic design and are not practical for quick, on-the-go resume creation. Additionally, these tools are costly and resource-intensive.

Mobile applications provide convenience but suffer from **screen limitations**, **limited editing controls**, **ad interruptions**, and **poor formatting in export files**, which significantly impact the final resume quality.

1.2.1 Limitations of Existing System

Despite the availability of various tools for resume creation, the existing systems exhibit several limitations that hinder their effectiveness, especially for users seeking a quick, professional, and flexible solution. These limitations are summarized below:

Absence of Real-Time Preview

Many existing systems require users to submit or save their information before they can preview the final resume. This lack of instant visual feedback increases the effort required to make corrections and adjustments, thereby reducing efficiency.

Restricted Customization Options

Free and template-based systems often impose constraints on layout, color schemes, font styles, and section organization. Users are forced to adhere to pre-defined structures, limiting their ability to personalize the resume based on job profiles or design preferences.

Dependence on Paid Services

Most online resume platforms operate under a freemium model where key features — such as premium templates, unlimited downloads, and high-resolution PDF export — are locked behind subscription plans. This creates a barrier for users who are unwilling or unable to pay for basic resume functionality.

Data Privacy and Security Concerns

Several platforms require users to create accounts and store personal data on cloud servers. This raises legitimate concerns about data ownership, unauthorized access, and misuse of sensitive information, especially when resumes include confidential contact or employment details.

Complexity of Professional Design Tools

Advanced design software such as Adobe Illustrator or InDesign offers great flexibility but demands significant technical expertise. These tools are not practical for users without a design background and often require paid licenses, making them inaccessible to the average job seeker.

Limited Mobile Responsiveness

Many resume-building platforms are optimized for desktop use and lack full support for mobile devices. On the other hand, mobile apps often provide a restricted feature set, suffer from layout limitations, and produce outputs that lack professional quality.

Manual Formatting Challenges

When using word processors, users must manually format each section — such as aligning text, adjusting margins, and setting font sizes — which is not only time-consuming but also prone to errors and inconsistencies that may affect the visual appeal of the resume.

Lack of Intelligent Assistance

Existing systems typically do not provide automated suggestions for content, grammar, or structure improvement. The absence of smart features such as keyword optimization, relevance checking, or section prioritization limits their utility in enhancing resume effectiveness.

1.3 PROPSED SYSTEM

To overcome the limitations identified in the existing systems, the Quick Resume Builder is proposed as a client-side, responsive, and user-friendly web application that empowers users to create professional resumes effortlessly. This system is specifically designed to provide real-time customization, multiple layout themes, and instant PDF export — all without requiring user registration or backend infrastructure.

The proposed system streamlines the entire resume-building process through a guided input form and dynamic preview pane, reducing both time and effort while delivering high-quality results. Users can input their personal and professional information, select from pre-defined themes (Classic, Modern, Professional), and instantly generate a visually appealing resume ready for download.

By utilizing modern front-end technologies such as HTML5, CSS3, and JavaScript, along with the html2pdf.js library for document export, the application ensures optimal performance, accessibility, and design flexibility — all within a single-page browser environment.

Key Features of the Proposed System

- Interactive User Interface
  Simple and intuitive form fields for entering resume data such as personal details, education, skills, projects, and certifications.
- Theme-Based Customization
  Users can choose from three professionally designed themes:
  - Classic – Formal and traditional
  - Modern – Sleek and minimal
  - Professional – Clean and corporate-style layout
- Live Preview
  Real-time rendering of the resume as users input or update their information, eliminating the need for repeated downloads or format adjustments.
- Client-Side PDF Export
  Generates downloadable PDF files using the html2pdf.js JavaScript library — no server required, ensuring full privacy and instant results.
- Mobile Responsiveness
  Fully functional on desktops, tablets, and mobile devices, allowing users to build resumes on the go.
- No Login or Data Storage
  Operates entirely on the client-side, enhancing data security and eliminating the need for account creation or cloud-based storage.

1.3.1 Advantages of Proposed Systems

Real-Time Preview Functionality

The application allows users to view changes to their resume live, without the need for reloading or navigating between tabs. This feature significantly reduces editing time and improves user satisfaction by offering instant feedback.

Multiple Professionally Designed Themes

Users can select from a range of polished themes — such as Classic, Modern, and Professional — ensuring their resume aligns with their industry, role, or personal brand. Each theme is tailored using responsive CSS styling for clarity and aesthetics.

Seamless PDF Export

With integrated support for the html2pdf.js library, the system enables one-click PDF generation, producing high-quality, print-ready resumes directly from the browser. This eliminates the need for third-party converters or plugins.

Client-Side Processing for Privacy

All operations are executed entirely on the client side, ensuring that no user data is transmitted, stored, or tracked. This design prioritizes data privacy and security, which is crucial for handling personal and professional information.

User-Friendly and Accessible Interface

The form-based layout guides users step by step through the resume-building process, making it ideal for individuals with limited technical or design expertise. Clear field labels and placeholders enhance usability.

Lightweight and Fast Performance

Built using standard web technologies (HTML, CSS, JavaScript), the system is lightweight, responsive, and requires no installation. It loads quickly on all modern browsers and is optimized for performance.

Responsive and Cross-Platform Design

The application is fully responsive, ensuring compatibility across a wide range of devices, including desktops, laptops, tablets, and smartphones. This makes it suitable for users on the go.

Free and Offline Capable

Unlike many commercial resume builders, this system is completely free to use. It can also be hosted locally and used without internet access — a valuable feature for users in bandwidth-constrained environments.

Extensibility for Future Enhancements

The modular structure of the application allows for future feature integration, such as:

- AI-driven resume suggestions,
- Auto-formatting and grammar checks,
- LinkedIn import support,
- Cloud-based user profiles and versioning.

CHAPTER 2
LITERATURE SURVEY

Author/Source	**Title**	**Objective**	**Methodology**	**Key Findings**
Nielsen Norman Group	Human-Centered UX Design	Improve user interaction and reduce complexity in interfaces	Form-based layout, real-time feedback, responsive design	Form structure + live preview enhances usability and lowers cognitive load
ACM Research (2020)	Client-Side PDF Generation for Secure Applications	Explore secure document generation using web technologies	HTML, CSS, JavaScript + html2pdf.js library	Ensures privacy by avoiding backend usage; ideal for sensitive documents like resumes
Novoresume / Canva / Zety (Commercial Tools)	Online Resume Building Platforms	Provide pre-designed templates for resume creation	Drag-and-drop, cloud storage, registration-based access	Feature-limited unless paid; requires login; data privacy concerns

CHAPTER 3
SYSTEM ANALYSIS

System analysis is the process of studying a problem domain in detail to define the goals, functions, and feasibility of the system to be developed. This phase sets a strong foundation for development by examining the current scenario, identifying system goals, and evaluating whether the system can be successfully implemented. The following sub-sections elaborate on the **Purpose**, **Scope**, and **Feasibility** of the proposed system.

3.1 Purpose

The main purpose of the Quick Resume Builder is to create an efficient, lightweight, and interactive web-based solution that allows users to generate professional-quality resumes in real-time. Unlike conventional resume-making methods, this application aims to automate the formatting process, minimize user effort, and maximize output quality — all within a browser environment.

The system is intended to:

- Provide a fast, intuitive tool for resume creation without requiring technical or design expertise.
- Enable users to select from multiple professionally styled themes to match their industry or personal preference.
- Allow users to preview the resume in real time while entering information, ensuring transparency and immediate feedback.
- Offer one-click PDF download of the resume without using external software or servers.
- Ensure privacy and security by processing all data on the client side, avoiding any backend or cloud data storage.

The purpose also includes addressing the shortcomings of existing platforms such as dependency on paid features, lack of offline access, and complexity in formatting.

3.2 Scope

The Quick Resume Builder project is focused on designing a user-centric, responsive, and secure resume-generation tool that operates fully within a web browser. The system is primarily targeted at students, fresh graduates, and professionals who need a quick and polished resume-building experience without the hassle of traditional tools or the constraints of paid platforms.

Features within Scope:

- A form-driven interface to collect user inputs like personal info, education, experience, skills, and projects.
- Multiple themes with CSS-based styling (Classic, Modern, Professional).
- Live preview of resume as the user fills the form.
- One-click PDF generation using html2pdf.js.
- Cross-platform compatibility (desktop, tablet, mobile).
- Fully client-side operation (no login, no server, no database).

Out of Scope (for current version):

- Backend integration for user accounts or saving resumes.
- AI-powered suggestions or grammar checks.
- Resume parsing or importing from LinkedIn or documents.
- Integration with external platforms or job portals.

The current system is intended to be lightweight, fast, and accessible — with flexibility for future upgrades and features.

3.3 Feasibility Study

A feasibility study is conducted to assess the viability of a proposed system across various dimensions, ensuring that it can be successfully developed and implemented. For the Quick Resume Builder, the feasibility study includes:

3.3.1 Economic Feasibility

Economic feasibility determines whether the proposed system is cost-effective and whether the benefits outweigh the expenses over time.

Key Points:

- The system is built using free and open-source technologies (HTML, CSS, JavaScript, html2pdf.js).
- No licensing costs are required since there is no use of proprietary software.
- There is no need for a backend server or cloud storage, which significantly reduces hosting and maintenance costs.
- The system can be hosted for free on platforms like GitHub Pages, Netlify, or run locally.
- Development and testing are handled in a low-cost environment using local IDEs (e.g., VS Code).

The Quick Resume Builder is economically feasible as it incurs minimal development and deployment costs and provides significant value to users without recurring expenses.

3.3.2 Technical Feasibility

Technical feasibility evaluates whether the current technology and tools can support the development and functionality of the proposed system.

Key Points:

- The application uses core web technologies that are well-documented, stable, and widely supported across all browsers.
- The system operates entirely on the client-side, requiring no backend processing or server infrastructure.

- The resume generation feature uses the html2pdf.js library, which seamlessly converts web content into a downloadable PDF.
- The application is cross-platform compatible — working smoothly on desktops, laptops, tablets, and mobile devices.

With readily available technologies and no need for high-end infrastructure, the project is technically feasible and scalable for future enhancements.

3.3.3 Social Feasibility

Social feasibility examines the acceptability of the system by users and society, ensuring that it aligns with user needs and cultural standards.

Key Points:

- The Quick Resume Builder targets a **wide audience**, including students, job seekers, and professionals — all of whom benefit from fast and professional resume creation.
- The system enhances **digital literacy**, empowering non-technical users to create high-quality documents without relying on external help.
- It respects **user privacy** by storing no data and avoiding server-side operations — aligning with growing concerns around digital security and data protection.
- It promotes **equal opportunity**, as it is **free, open, and offline-capable**, ensuring that users from underserved areas or with limited internet access are not excluded.

The system has strong **social acceptance potential** and aligns with modern expectations for usability, privacy, and accessibility. Thus, it is **socially feasible**.

The development of the **Quick Resume Builder** is highly feasible across technical, economic, and social dimensions. Technically, the system utilizes widely supported client-side web technologies such as HTML, CSS, JavaScript, and html2pdf.js, ensuring platform independence and seamless PDF generation without requiring backend infrastructure. Economically, the project is cost-effective as it relies entirely on open-source tools and can be hosted freely on platforms like GitHub Pages or run locally without any recurring expenses.

3.4 Requirement Analysis

Requirement analysis is the process of identifying and documenting the specific needs of the system to be developed. It ensures that the software will function correctly, meet user expectations, and fulfill the objectives defined during system analysis. This section outlines both functional and non-functional requirements for the Quick Resume Builder.

3.4.1 Functional Requirements

Functional requirements describe the specific behaviors, features, and functionalities that the system must support to meet user needs.

Key Functional Requirements:

- **User Input Collection:**
  - The system must provide form fields for entering personal details, education, experience, skills, certifications, and project information.
- **Theme Selection:**
  - The user should be able to choose from multiple resume themes (e.g., Classic, Modern, Professional).
- **Live Resume Preview:**
  - As users fill out the form, a real-time preview of the resume should be displayed on the same page.
- **PDF Generation:**
  - The system should generate a high-quality, downloadable PDF version of the resume using html2pdf.js.
- **Responsive Layout:**
  - The application must adapt to different screen sizes (desktop, tablet, mobile).
- **Error Handling:**

- The system should validate required fields (like name, email, phone) and display appropriate error messages if any are missing.

3.4.2 Non-Functional Requirements

Non-functional requirements define how the system performs, focusing on quality attributes such as usability, reliability, performance, and security.

Key Non-Functional Requirements:

- **Usability:**
  - The interface must be simple, intuitive, and user-friendly, even for non-technical users.
- **Performance:**
  - The system should operate smoothly and generate PDFs within a few seconds without noticeable delay.
- **Scalability:**
  - The system design should allow for future features like AI suggestions, LinkedIn integration, or cloud storage.
- **Security & Privacy:**
  - The system must handle all operations on the client side and must not store or transmit any user data to external servers.
- **Portability:**
  - The application should be platform-independent and accessible via any modern web browser.
- **Offline Accessibility:**
  - The system should work even when hosted locally, allowing users to use it without an internet connection.

3.5 Requirements Specifications

The requirement specifications define the technical foundation upon which the system will be developed and executed. It outlines the hardware, software, and language tools necessary for building and running the **Quick Resume Builder** efficiently. This ensures smooth operation, cross-platform compatibility, and ease of maintenance during the project lifecycle.

3.5.1 Hardware Requirements

The hardware requirements are minimal due to the lightweight, client-side nature of the application. The system is accessible on standard computing devices.

- **Processor:**
  Minimum intel Pentium Processor or equivalent-Suffcient to run a browser based application.
- **RAM :**
  At least 2 GB of RAM — ensures smooth browsing, form interaction, and PDF generation.
- **Storage:**
  Around 100 MB of free disk space — mainly for browser caching and optional local development files.
- **Display:**
  A monitor with 1024×768 resolution or higher — provides enough screen space to view both the form and resume preview comfortably.
- **Input Devices:**
  Keyboard and mouse — necessary for filling out form fields and interacting with the interface.
- **Internet Connection (optional):**
  Required only if hosting the application online; not necessary for local use.

3.5.2 Software Requirements

The application is compatible with widely used operating systems and does not rely on complex software installations.

- **Operating System:**
  Windows 7 and above, Linux, or macOS — compatible with all modern operating systems.
- **Web Browser:**
  Chrome, Firefox, Microsoft Edge, Safari (latest versions) — the system runs directly in the browser using HTML and JavaScript.
- **Code Editor(for Development):**
  Visual Studio Code or any text editor — used for writing and modifying source code.
- **Hosting Platform(Optional):**
  GitHub Pages, Netlify, or Vercel — used to publish the resume builder online if needed.
- **PDF Library:**
  html2pdf.js — a JavaScript library integrated into the project to enable direct HTML-to-PDF conversion.

3.5.3 Language Specifications

The Quick Resume Builder is developed entirely using web technologies, making it portable, lightweight, and browser-friendly.

- **HTML(HyperText Markup Language):**
  Structures the form layout and resume content. It defines the input fields and sections like education, skills, and experience.
- **CSS(Cascading Style Sheets):**
  Handles the design and styling of the application, including fonts, spacing, layout, and theme customization.
- **JavaScript:**
  Adds interactivity to the form, dynamically updates the resume preview, and handles the download functionality.

CHAPTER 4
SYSTEM DESIGN

4.1 System Architecture

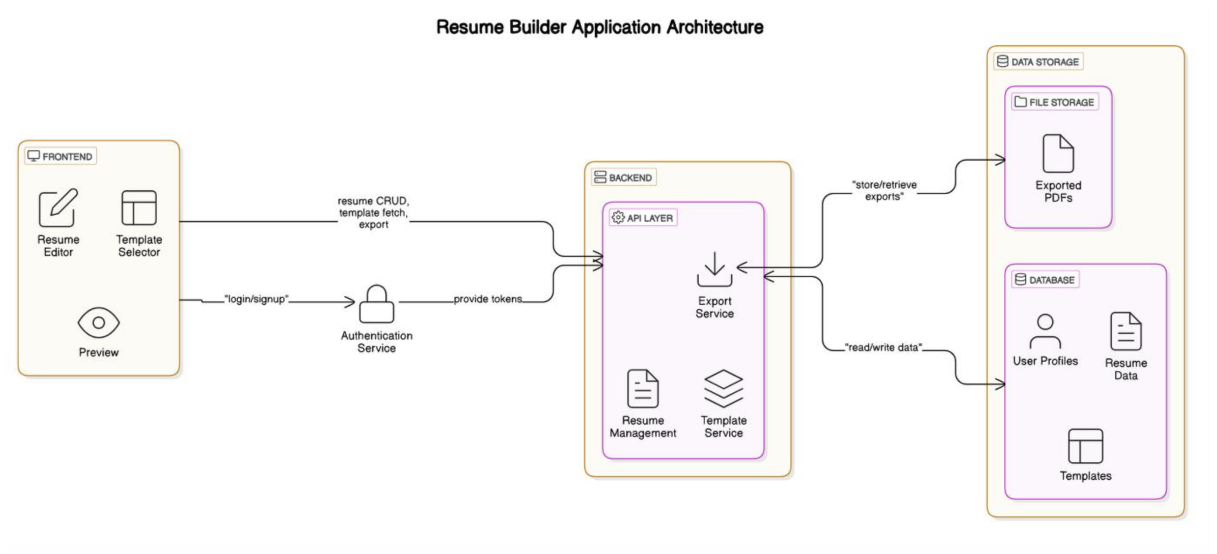


Figure 4.1 system architecture

4.2 Description

This section provides a detailed explanation of the system's structural design, outlining the core components and how they interact. The Quick Resume Builder is a purely client-side web application built using HTML, CSS, JavaScript, and the html2pdf.js library. Its architecture is simple, efficient, and designed to offer privacy, speed, and user-friendliness.

The system follows a three-tier client-side architecture, which consists of:

1. User Interface Layer (Presentation Layer)

- Developed using HTML and CSS.
- Provides form fields for input (e.g., name, email, education, experience).
- Offers a theme selection option and live resume preview.
- Ensures responsiveness and accessibility on all devices.
- Includes buttons like "Generate Resume" and "Download PDF".

2. Application Logic Layer (Processing Layer)

- Handled by JavaScript.
- Captures user inputs and dynamically updates the resume preview section.
- Validates required fields (e.g., first name, email, phone).
- Applies the selected theme to the resume dynamically.
- Handles user events like clicking buttons and changing inputs.

3. PDF Generation Layer

- Utilizes the html2pdf.js JavaScript library.
- Converts the resume preview (DOM content) into a downloadable PDF file.
- Works entirely on the client-side (offline-capable).
- Eliminates the need for backend processing or cloud rendering.

The Quick Resume Builder is designed for simplicity and speed. It doesn't require user authentication, databases, or any server-side logic. The entire operation—from form filling to PDF download—happens inside the user’s browser.

Working Flow:

1. The user opens the application in a browser.
2. Fills out the form fields with resume details.
3. Selects a desired theme from the dropdown.
4. Clicks the "Generate Resume" button.
5. The resume preview is updated in real-time using JavaScript.
6. On clicking "Download Resume", the resume section is converted into a PDF and saved.

4.3 UML Diagrams

4.3.1 Use Case Diagram

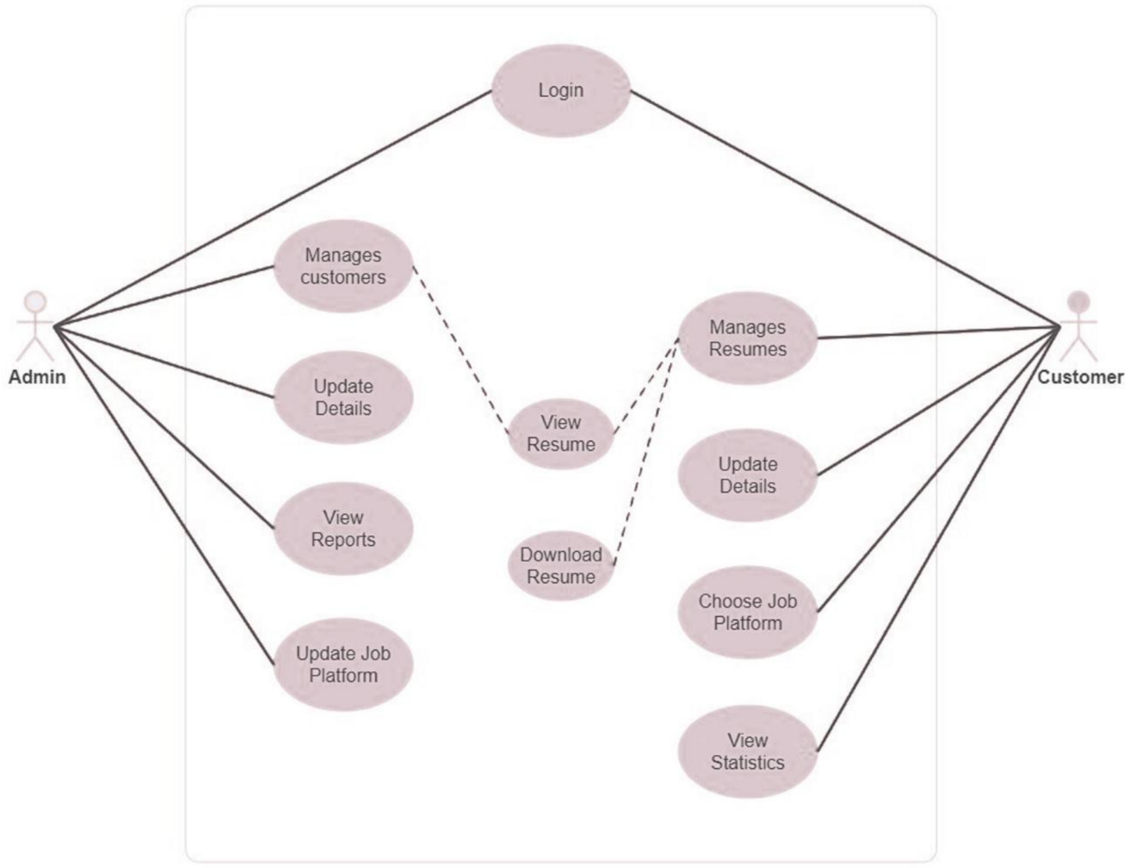


Figure: 4.3.1 use case diagram

The Use Case Diagram is a high-level representation that outlines how different users (actors) interact with the system and what functionalities they can access. For the Quick Resume Builder system, the diagram defines two primary actors: Admin and Customer. Each actor performs specific operations represented as "use cases" that the system supports.

Actors:

1. **Admin**
   The Admin is responsible for managing backend operations such as maintaining customer records, updating job platforms, and monitoring system reports. The admin ensures the platform's smooth functioning and data integrity.

2. **Customer/user**

   The Customer represents the typical user of the system who creates, views, and downloads resumes.

Use Cases:

Admin-Specific Use Cases:

- Login: Allows the admin to securely access the system.
- Manages Customers: Enables the admin to view, edit, and delete customer records.
- Update Details: Permits updating of user or system-related information.
- View Reports: Generates analytical or operational reports for system performance and usage.
- Update Job Platform: Admin can modify the available job platforms or job categories displayed to users.
- View Resume: (Dashed association) Admin may access user resumes for verification or support purposes.

Customer-Specific Use Cases:

- Login: Authenticates the customer to access personalized features.
- Manages Resumes: Allows customers to create, edit, and manage resume content.
- Update Details: Lets users modify their personal, educational, and experience data.
- Choose Job Platform: Enables selection of preferred job categories or platforms for resume tailoring.
- Download Resume: Allows exporting the resume in a PDF format using integrated tools.
- View Resume: Provides a live preview of the resume before downloading.
- View Statistics: Displays metrics like completion status or suggestions to improve resume quality.

Relationships:

- Solid lines indicate direct interaction between actors and use cases.
- Dashed lines (e.g., Admin → View Resume) represent optional or conditional interactions.

4.3.2 Activity Diagram

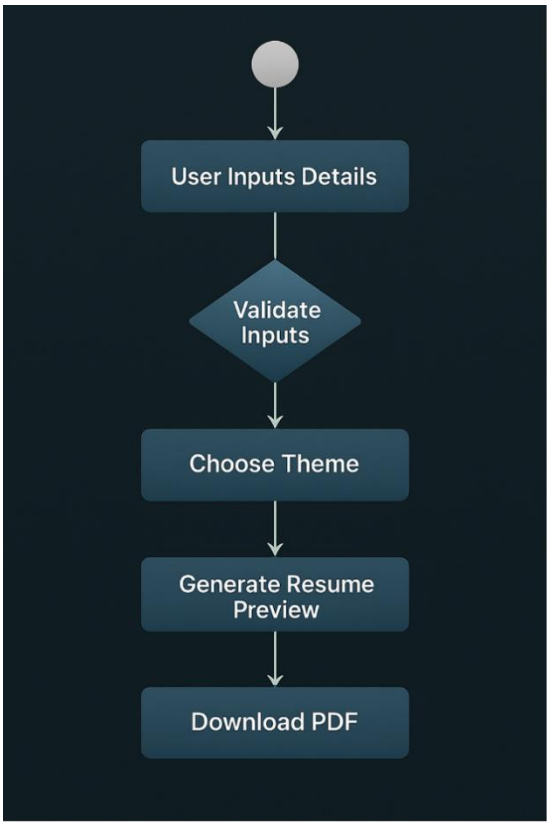


Figure 4.3.2 Activity diagram

The Activity Diagram models the dynamic workflow of the Quick Resume Builder system. It illustrates the step-by-step activities performed by the user and the system, from initiating the resume creation process to downloading the final PDF. This diagram focuses on the flow of control and highlights decision points and sequential actions.

Workflow Description:

1. **Start Node:**
   - The process begins when the user opens the application and initiates interaction.

2. **User Inputs Details:**
   - The user fills in all the required fields in the form, including personal, educational, and professional information.
   - This is the first major action step in the workflow.
3. **Validate Inputs:**
   - Once data is entered, the system checks if all mandatory fields (e.g., name, email, phone number) are correctly filled.
   - If validation fails, the user is prompted to correct the inputs before proceeding.
4. **Choose Theme:**
   - After successful validation, the user selects a resume design theme from the available options (e.g., Classic, Modern, Professional).
   - The theme defines the visual layout and styling of the resume preview.
5. **Generate Resume Preview:**
   - The system dynamically generates a live preview of the resume based on the entered information and selected theme.
   - This allows the user to review how the final resume will look before downloading.
6. **Download PDF:**
   - If the user is satisfied with the preview, they can click the "Download Resume" button.
   - The resume is then converted into a downloadable PDF file using the html2pdf.js library.
7. **End Node:**
   - The activity concludes once the resume is successfully downloaded.

4.3.3 Class Diagram

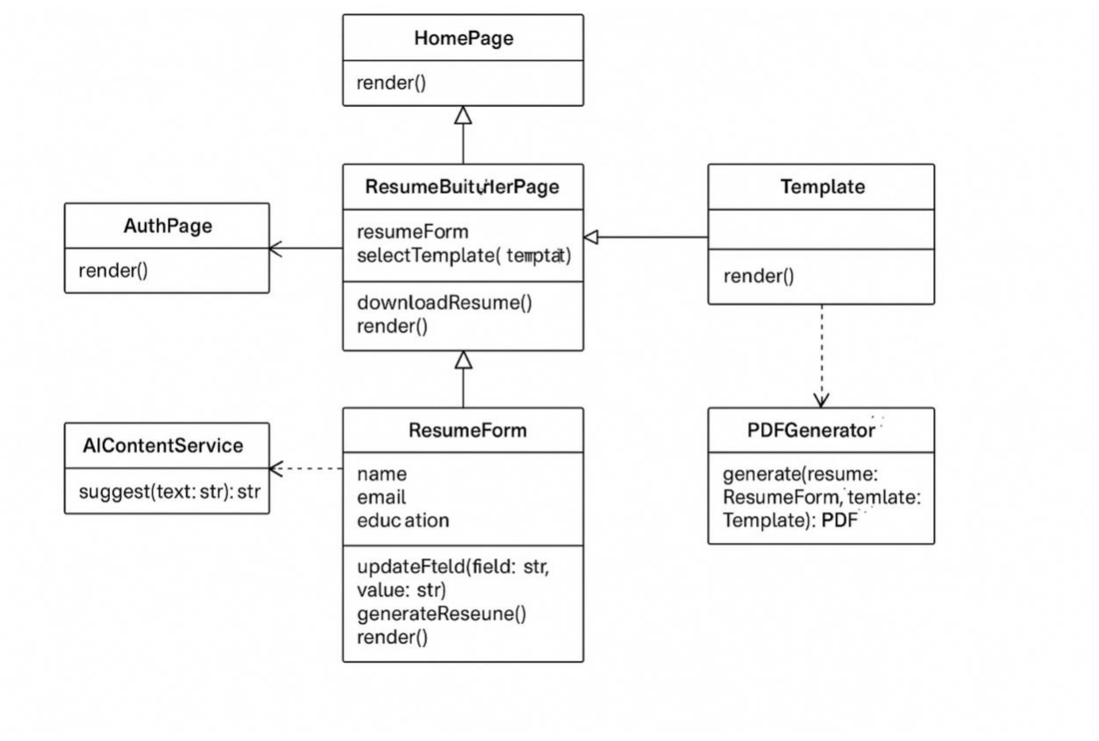


Figure 4.3.3 Class Diagram

The **Class Diagram** represents the static structure of the Quick Resume Builder system. It defines the **classes**, their **attributes**, **methods**, and the **relationships** among them. This diagram helps in understanding how different components of the application are organized and interact with each other in an object-oriented manner.

Key Classes and Their Descriptions:

1. HomePage

- **Method:** render()
- **Purpose:**
  This is the entry point of the application. It initializes the interface and redirects the user to the appropriate page (Auth or ResumeBuilder).
- **Relationship:**
  Inherited by ResumeBuilderPage.

2. AuthPage

- **Method:** render()

- **Purpose:**
  Handles authentication tasks like login/signup (if implemented). Ensures secure access to resume functionalities.

3. ResumeBuilderPage

- **Attributes/Associations:**
  - resumeForm – an instance of ResumeForm.
- **Methods:**
  - selectTemplate(template)
  - downloadResume()
  - render()
- **Purpose:**
  Acts as the main page where users build their resume. It manages input data, resume template selection, and PDF download.
- **Relationships:**
  - Has a composition relationship with ResumeForm.
  - Aggregates Template and interacts with PDFGenerator.

4. ResumeForm

- **Attributes:**
  - name, email, education – basic input fields of a resume.
- **Methods:**
  - updateField(field: str, value: str)
  - generateResume()
  - render()
- **Purpose:**
  Stores and handles user input for the resume. Contains logic to update and render resume data.

- **Relationships:**
    - Connected with AIContentService for content suggestions.
    - Passed to PDFGenerator for creating the final PDF.

5. Template

- **Method:** render()
- **Purpose:**
  Provides layout and visual structure for the resume. Used to apply different themes.
- **Relationship:**
  Used by ResumeBuilderPage and PDFGenerator.

6. PDFGenerator

- **Method:**
    - generate(resume: ResumeForm, template: Template): PDF
- **Purpose:**
  Converts the structured resume data and selected template into a downloadable PDF using html2pdf.js or similar logic.
- **Relationship:**
  Receives input from ResumeForm and Template.

7. AIContentService

- **Method:**
    - suggest(text: str): str
- **Purpose:**
  An optional AI-powered module that provides intelligent suggestions or auto-fill content (e.g., for the "About Me" section).
- **Relationship:**
  Connected to ResumeForm via a dashed line, indicating a dependency or optional interaction.**:**

Solid arrows represent direct associations or compositions.

Dashed arrows represent dependencies (non-mandatory).

Inheritance: HomePage is a superclass of ResumeBuilderPage.

The diagram emphasizes **modularity** and **separation of concerns**, improving system maintainability and scalability.

4.3.4 Sequence Diagram

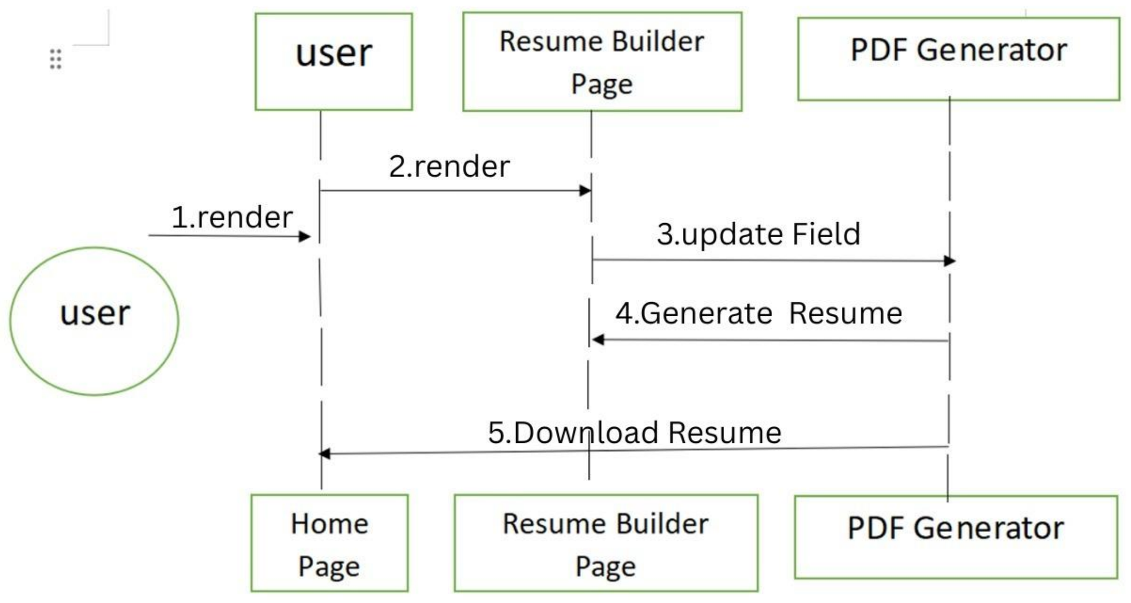


Figure 4.3.4 Sequence Diagram

The Sequence Diagram illustrates the order of interactions between the user and system components over time in the Quick Resume Builder.

Lifelines / Participants:

1. User
   - The primary actor who initiates all actions in the system.
2. Home Page
   - Initial landing page where the user starts the application.
3. Resume Builder Page

- The core interface for inputting details, selecting templates, and generating previews.

4. PDF Generator
    - Responsible for converting the resume content into a downloadable PDF.

Message Flow (Same as Image):

1. User → Home Page
    - The user accesses the application, landing first on the Home Page.
2. Home Page → Resume Builder Page
    - Navigation occurs as the user proceeds to the resume-building section.
3. User → Resume Builder Page
    - The user inputs resume details and selects a theme.
4. Resume Builder Page → PDF Generator
    - The page sends resume data and template to the PDF Generator.
5. PDF Generator → Resume Builder Page
    - The generated PDF is returned for confirmation or download.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 Methods / Algorithms

The Quick Resume Builder application follows a series of structured methods that handle user interaction, data validation, resume generation, and PDF export. The algorithms used are mostly procedural, ensuring smooth flow and usability.

1.User Input Handling Algorithm:

To collect and validate user data from the form fields.

```
Algorithm CollectAndValidateInput
    Input: User details (name, email, phone, etc.)
    Output: Validated form data or error

    Begin
        For each required field in form
            If field is empty
                Display error message
                Exit algorithm
            End If
        End For
        Return validated input data
    End
```

2. Resume Theme Selection Algorithm

To allow users to choose a resume design theme (e.g., Classic, Modern, Professional).

```
Algorithm SelectTheme
    Input: User-selected theme from dropdown
    Output: Apply selected CSS style to resume preview
Begin
        Read value of theme from <select> input
        Switch (theme)
            Case "classic": apply Classic CSS class
            Case "modern": apply Modern CSS class
            Case "professional": apply Professional CSS class
        End Switch
    End
```

3. **Resume Preview Generation**

To dynamically display a real-time preview of the resume based on user inputs.

```
Algorithm GenerateResumePreview
    Input: Validated form data, selected theme
    Output: HTML-rendered resume preview
    Begin
        Construct HTML string with user inputs
        Insert HTML into preview container
        Apply theme style
        Display the preview section
    End
```

4. **PDF Generation Using html2pdf.js**

To convert the live HTML resume preview into a downloadable PDF.

```
Algorithm DownloadResumeAsPDF
    Input: HTML resume preview element
    Output: Downloadable PDF file
    Begin
        Get reference to resume output container
        Create filename using user name + timestamp
        Call html2pdf().from(container).save(filename)
    End
```

Form Handling & Validation ensures that incomplete data is caught before generation.

Theme Selection allows flexibility in design.

Preview Generation provides immediate visual feedback.

PDF Conversion transforms the resume into a portable format for sharing or printing.

5.2 Sample Code

- **Index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>QUICK RESUME</title>
  <link rel="stylesheet" href="style.css" />
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.9.2/html2pdf.bun
dle.min.js"></script>
</head>
<body>
  <div class="container">
    <h1>QUICK RESUME</h1>
    <div class="form-section">
      <div class="form-group">
        <label for="firstName">First Name:</label>
        <input type="text" id="firstName" />
      </div>
      <div class="form-group">
        <label for="lastName">Last Name:</label>
        <input type="text" id="lastName" />
      </div>
```

```
<div class="form-group">
  <label for="email">Email:</label>
  <input type="email" id="email" />
</div>
<div class="form-group">
  <label for="phone">Phone:</label>
  <input type="tel" id="phone" />
</div>
<div class="form-group">
  <label for="about">About Me:</label>
  <textarea id="about"></textarea>
</div>
<div class="form-group">
  <label for="education">Education:</label>
  <textarea id="education"></textarea>
</div>
<div class="form-group">
  <label for="experience">Experience:</label>
  <textarea id="experience"></textarea>
</div>
<div class="form-group">
  <label for="skills">Skills:</label>
  <input type="text" id="skills" placeholder="E.g., Python, Web Dev" />
</div>
<div class="form-group">
```

```
        <label for="projects">Projects:</label>
        <textarea id="projects" placeholder="Briefly describe your
projects"></textarea>
      </div>
      <div class="form-group">
        <label for="certifications">Certifications:</label>
        <input type="text" id="certifications" />
      </div>
      <div class="form-group">
        <label for="theme">Choose Theme:</label>
        <select id="theme">
          <option value="classic">Classic</option>
          <option value="modern">Modern</option>
          <option value="professional">Professional</option>
          <option value="elegant">Elegant</option>
          <option value="minimal">Minimal</option>
        </select>
      </div>
    </div>

    <button id="generateBtn">Generate Resume</button>
    <button id="downloadBtn">Download Resume</button>

    <div id="resumeOutput" class="resume-container">
      <div id="resumeContent"></div>
    </div>
  </div>


  <script src="script.js"></script>
</body>
</html>
```

- **Style.css**

```
body {
  font-family: Arial, sans-serif;
  background: linear-gradient(to right, #2c3e50, #4ca1af);
  color: white;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.container {
  background: white;
  color: black;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
  width: 60%;
  max-height: 90vh;
  overflow-y: auto;
}

.form-section {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}

.form-group {
  width: 48%;
  margin-bottom: 10px;
}

input,
textarea,
select {
  width: 100%;
  padding: 8px;
  margin-top: 5px;
  border-radius: 5px;
  border: 1px solid #ccc;
  margin-bottom: 10px;
}
```

```
button {
  background: #4ca1af;
  color: white;
  padding: 10px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  margin-top: 10px;
  margin-right: 10px;
}

.resume-container {
  margin-top: 20px;
  padding: 20px;
  border-radius: 10px;
  display: none;
}

/* Themes */
.classic {
  background: #f4f4f4;
  color: #333;
  font-family: "Times New Roman", serif;
  border: 2px solid #333;
```

```
}
.modern {
  background: #1e1e1e;
  color: #00bcd4;
  font-family: Arial, sans-serif;
  border-left: 5px solid #00bcd4;
  padding-left: 20px;
}

.professional {
  background: #ffffff;
  color: #333;
  font-family: 'Helvetica Neue', sans-serif;
  border: 1px solid #666;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
  padding: 20px;
}

.elegant {
  background: #fffbe6;
  color: #4b3832;
  font-family: 'Georgia', serif;
  border: 1px solid #e6d3b3;
  padding: 20px;
  border-left: 5px solid #c59d5f;
```

```
}


.minimal {
  background: #ffffff;
  color: #111;
  font-family: 'Verdana', sans-serif;
  border: none;
  padding: 20px;
  border-bottom: 2px solid #888;
}


hr {
  border: 1px solid #ccc;
  margin: 10px 0;
}
h2,
h3 {
  margin-bottom: 8px;
}
p {
  margin: 5px 0;
}
```

- **Script.js**

```
document.getElementById("generateBtn").addEventListener("click",
generateResume);

document.getElementById("downloadBtn").addEventListener("click",
downloadResume);


function generateResume() {
  const theme = document.getElementById("theme").value;


  const content = `
    <h2>${document.getElementById("firstName").value}
${document.getElementById("lastName").value}</h2>
    <p><strong>Email:</strong>  ${document.getElementById("email").value}  |
<strong>Phone:</strong> ${document.getElementById("phone").value}</p>
    <hr>
    <h3>About Me</h3><p>${document.getElementById("about").value}</p>


<h3>Experience</h3><p>${document.getElementById("experience").value}</p
>
    <h3>Skills</h3><p>${document.getElementById("skills").value}</p>
    <h3>Projects</h3><p>${document.getElementById("projects").value}</p>

<h3>Certifications</h3><p>${document.getElementById("certifications").value}
</p>
  `;
  const resumeOutput = document.getElementById("resumeOutput");
  resumeOutput.className = `resume-container ${theme}`;
  document.getElementById("resumeContent").innerHTML = content;
  resumeOutput.style.display = "block";
}
function downloadResume() {
  const resume = document.getElementById("resumeOutput");
  html2pdf().from(resume).save("My_Resume.pdf");
}
```

CHAPTER 6

SYSTEM TESTING

System testing is a critical phase in software development where the complete integrated system is evaluated to ensure it meets the specified requirements. In the Quick Resume Builder project, system testing was conducted to verify the correctness, functionality, usability, and performance of the application.

Purpose of Testing

The main purpose of system testing in this project was to:

- Verify that all components of the system work together seamlessly.
- Ensure that the system meets both functional and non-functional requirements.
- Identify any defects or inconsistencies before deployment.
- Validate the system's reliability, usability, and performance.

Types of Testing Performed

Unit Testing

- Each function was tested independently.
- Input handling, form validation, and PDF generation functions were tested for correct outputs.

Integration Testing

- Verified the flow of data between modules: user input → resume preview → PDF download.
- Ensured smooth interaction between UI elements and JavaScript methods.

Functional Testing

- Checked that the system performed all intended functionalities such as:
  - Generating resume preview.
  - Switching between themes.

    - Exporting to PDF.
    - Handling invalid or incomplete inputs.

User Interface Testing

- Ensured that the design and layout were consistent across devices and browsers.
- Verified element alignment, color schemes, and responsive design.

Compatibility Testing

- The application was tested on multiple browsers (Chrome, Firefox, Edge) and devices (desktop, tablet, mobile).
- Ensured that resumes were generated correctly and consistently in all environments.

After conducting various levels of testing, the Quick Resume Builder application was found to be stable and functional across all major platforms and devices. All essential features operate as expected, and no critical bugs remain. The application is ready for deployment and usage by end users.

CHAPTER 7

SCREENSHOOTS/OUTPUT

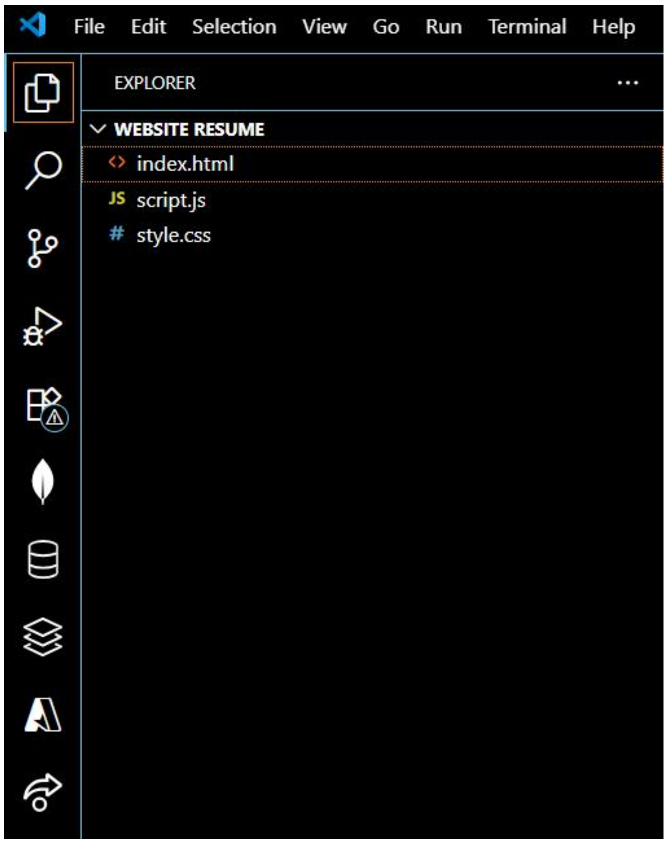
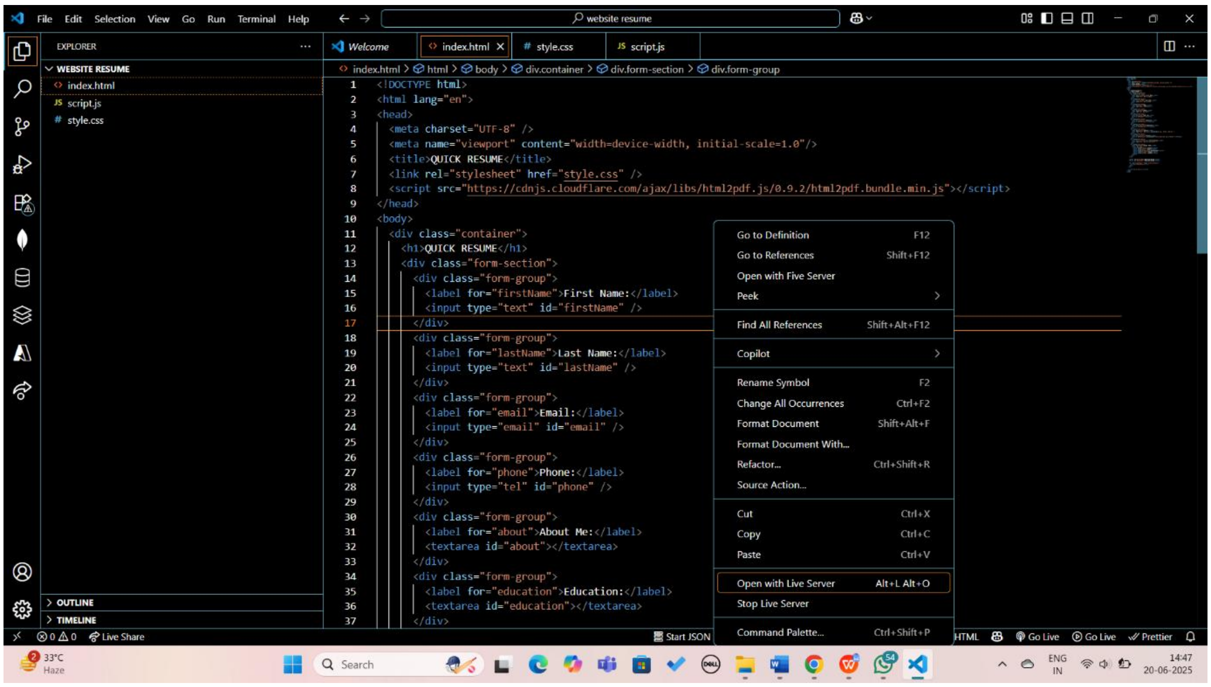


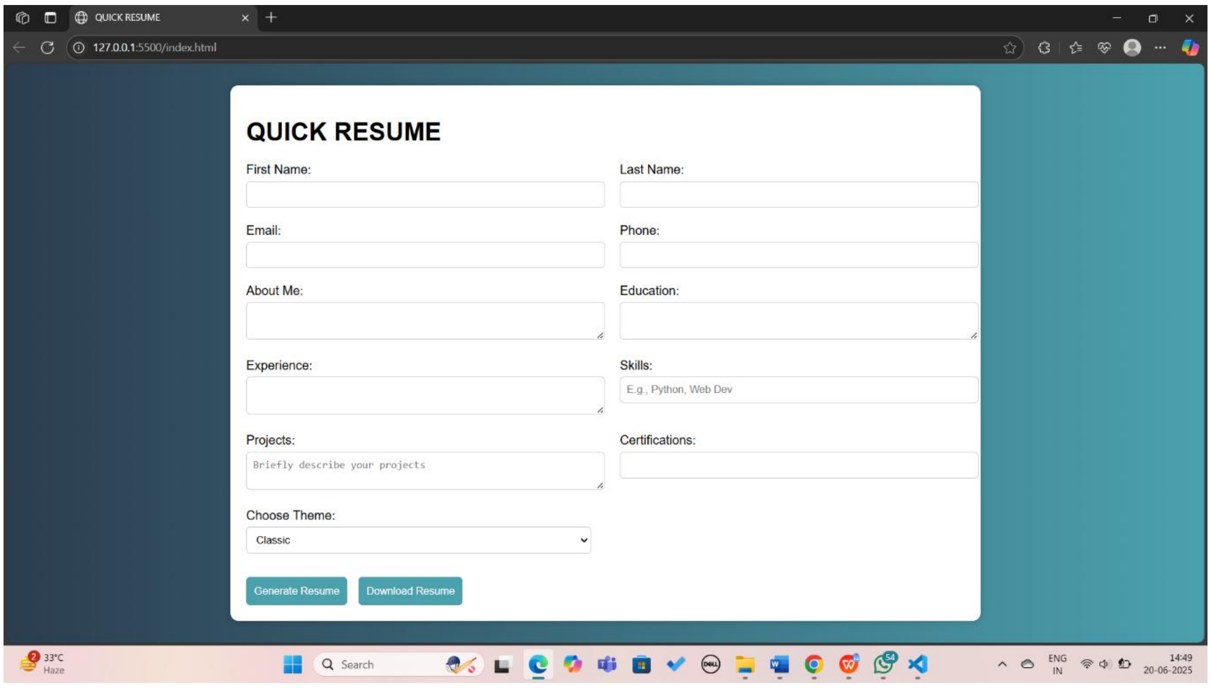
Figure 7.1 project structure



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>QUICK RESUME</title>
  <link rel="stylesheet" href="style.css" />
  <script src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.9.2/html2pdf.bundle.min.js"></script>
</head>
<body>
  <div class="container">
    <h1>QUICK RESUME</h1>
    <div class="form-section">
      <div class="form-group">
        <label for="firstName">First Name:</label>
        <input type="text" id="firstName" />
      </div>
      <div class="form-group">
        <label for="lastName">Last Name:</label>
        <input type="text" id="lastName" />
      </div>
      <div class="form-group">
        <label for="email">Email:</label>
        <input type="email" id="email" />
      </div>
      <div class="form-group">
        <label for="phone">Phone:</label>
        <input type="tel" id="phone" />
      </div>
      <div class="form-group">
        <label for="about">About Me:</label>
        <textarea id="about"></textarea>
      </div>
      <div class="form-group">
        <label for="education">Education:</label>
        <textarea id="education"></textarea>
      </div>
```

Figure 7.2 Runnig the code

(Live Server gets opened)



QUICK RESUME

First Name:

Last Name:

Email:

Phone:

About Me:

Education:

Experience:

Skills:

E.g., Python, Web Dev

Projects:

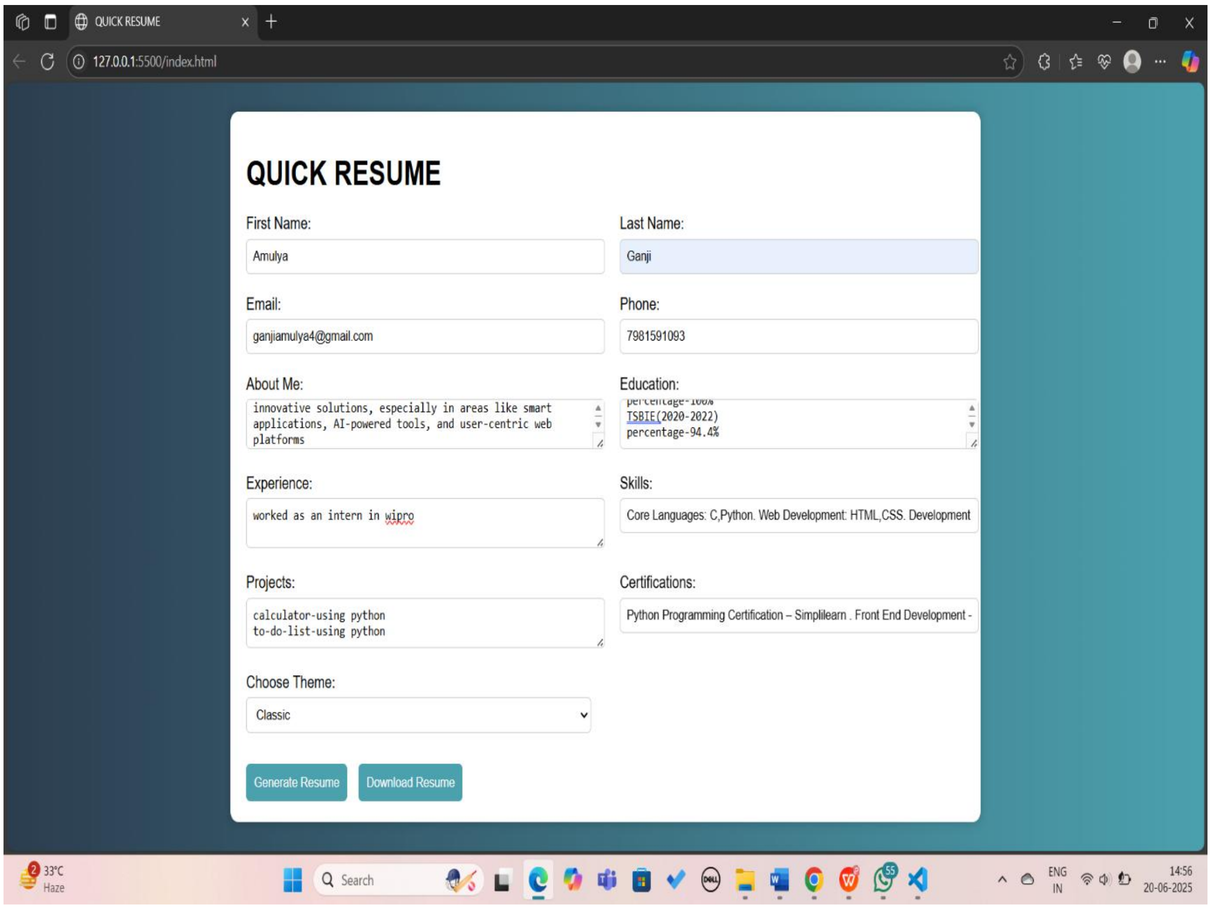
Briefly describe your projects

Certifications:

Choose Theme:

Classic

Figure 7.3 Interface of resume builder


QUICK RESUME

First Name: Amulya

Last Name: Ganji

Email: ganjiamulya4@gmail.com

Phone: 7981591093

About Me:
```
innovative solutions, especially in areas like smart
applications, AI-powered tools, and user-centric web
platforms
```

Education:
```
percentage-100%
TSBIE(2020-2022)
percentage-94.4%
```

Experience:
```
worked as an intern in wipro
```

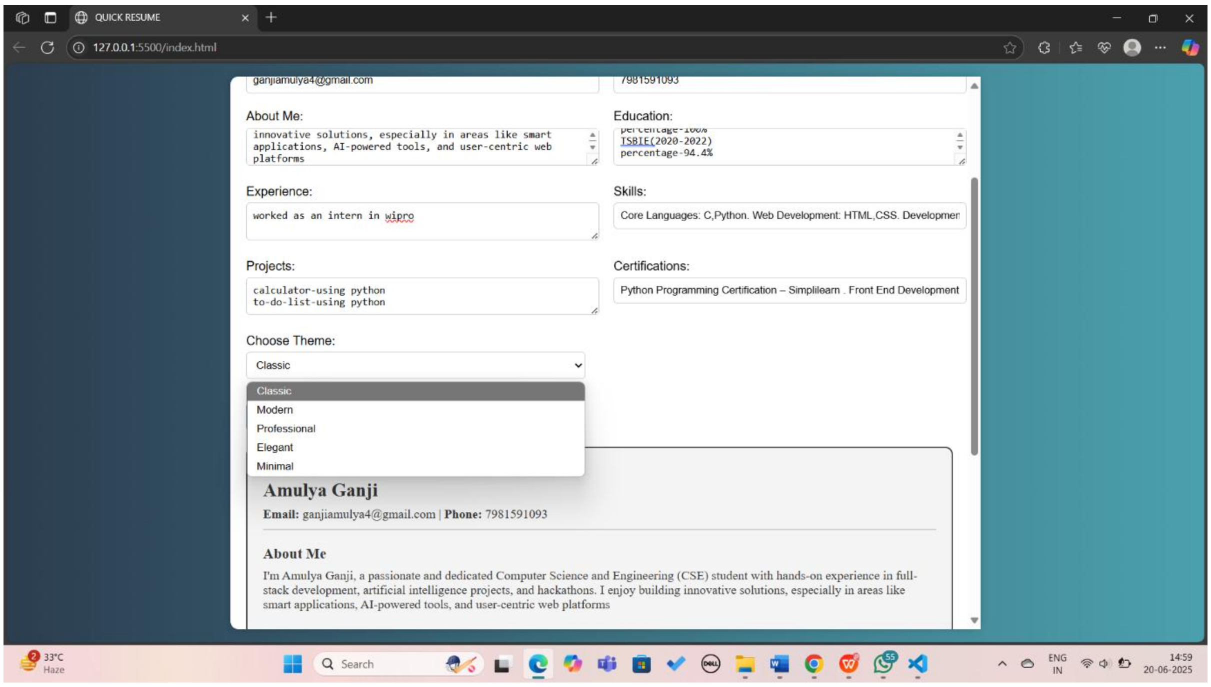
Skills: Core Languages: C,Python. Web Development: HTML,CSS. Development

Projects:
```
calculator-using python
to-do-list-using python
```

Certifications: Python Programming Certification – Simplilearn . Front End Development -

Choose Theme: Classic

Figure 7.4 Filling the Details


ganjiamulya4@gmail.com

7981591093

About Me:

```
innovative solutions, especially in areas like smart
applications, AI-powered tools, and user-centric web
platforms
```

Education:

```
percentage-100%
TSBIE(2020-2022)
percentage-94.4%
```

Experience:

```
worked as an intern in wipro
```

Skills:

Core Languages: C,Python. Web Development: HTML,CSS. Developmen

Projects:

```
calculator-using python
to-do-list-using python
```

Certifications:

Python Programming Certification – Simplilearn . Front End Development

Choose Theme:

Classic

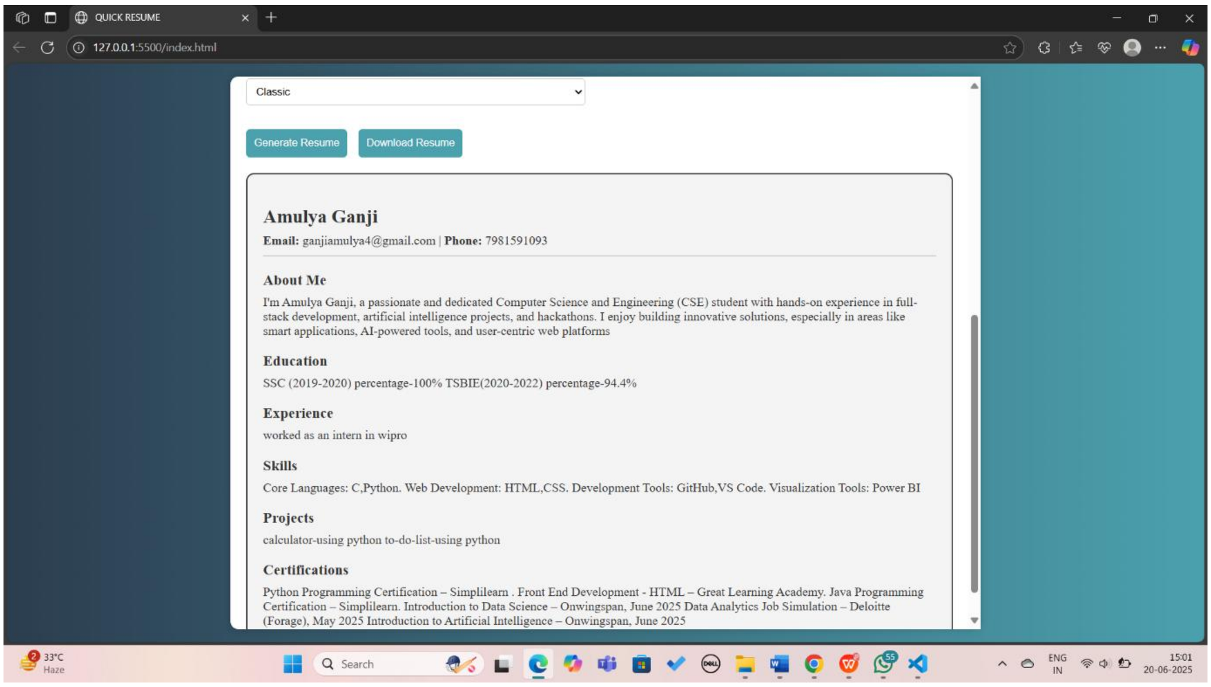
Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Figure 7.5 Choosing the Theme for resume



Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

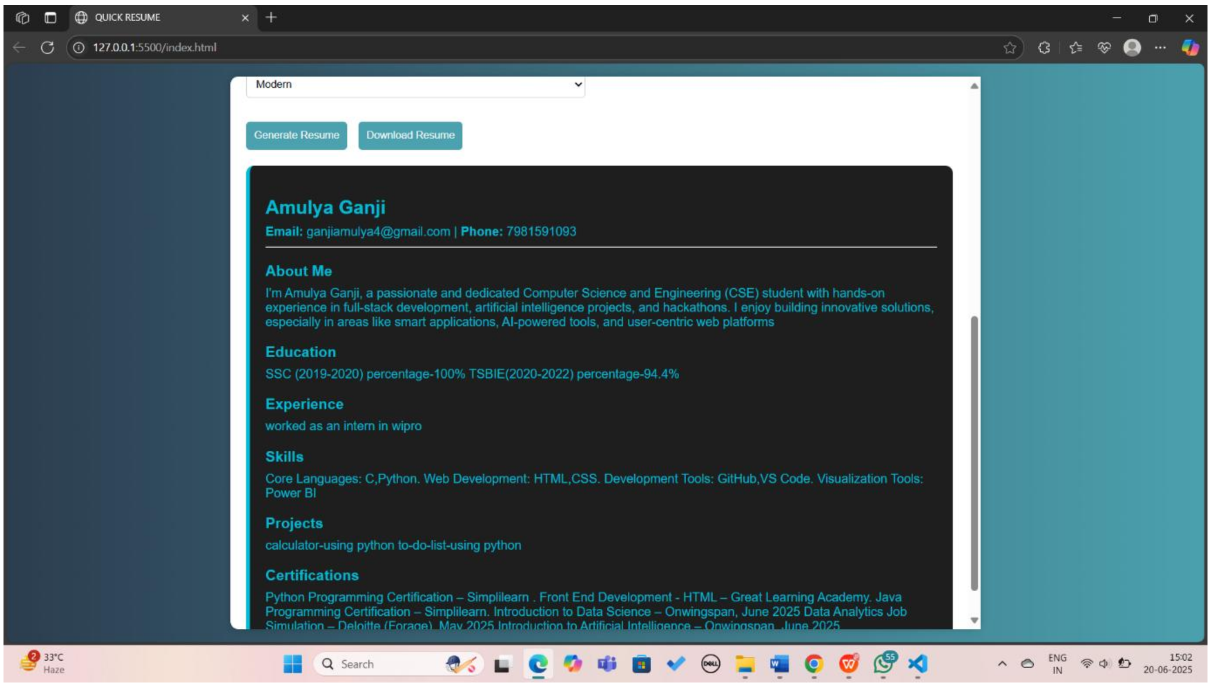
Projects

calculator-using python to-do-list-using python

Certifications

Python Programming Certification – Simplilearn . Front End Development - HTML – Great Learning Academy. Java Programming Certification – Simplilearn. Introduction to Data Science – Onwingspan, June 2025 Data Analytics Job Simulation – Deloitte (Forage), May 2025 Introduction to Artificial Intelligence – Onwingspan, June 2025

Figure 7.6 Choosing the Theme (Classic)



Modern

Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

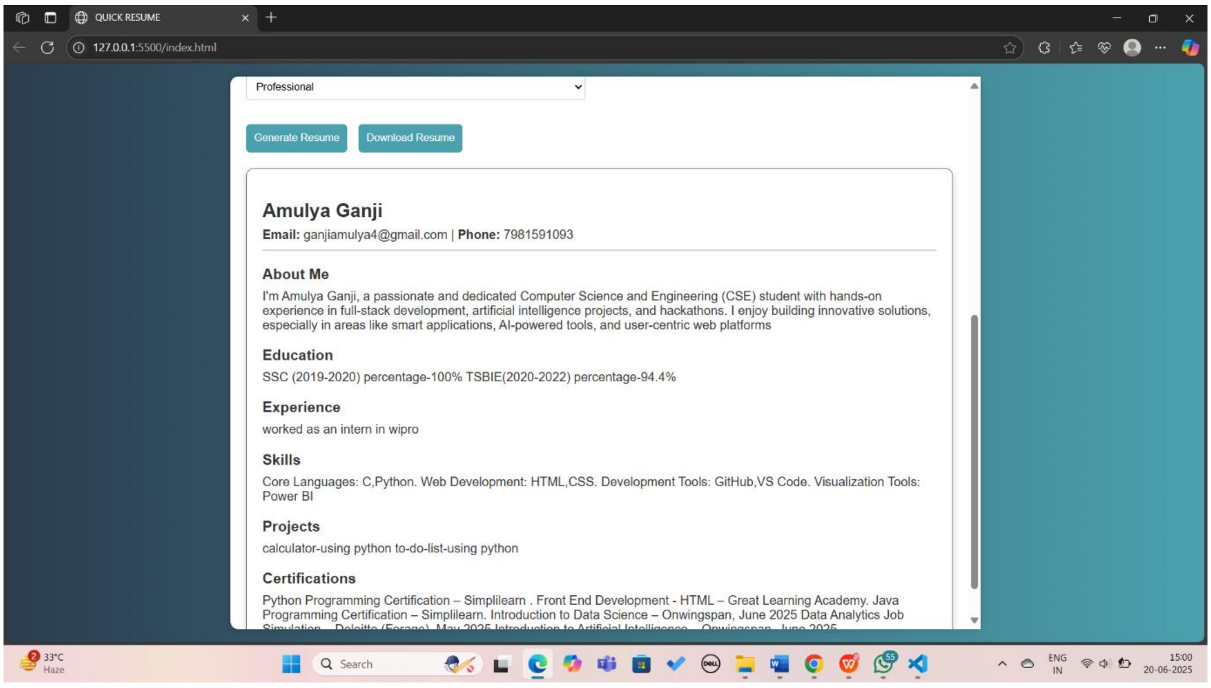
Projects

calculator-using python to-do-list-using python

Certifications

Python Programming Certification – Simplilearn . Front End Development - HTML – Great Learning Academy. Java Programming Certification – Simplilearn. Introduction to Data Science – Onwingspan, June 2025 Data Analytics Job Simulation – Deloitte (Forage), May 2025 Introduction to Artificial Intelligence – Onwingspan, June 2025

Figure 7.7 Choosing the Theme (Modern)



Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

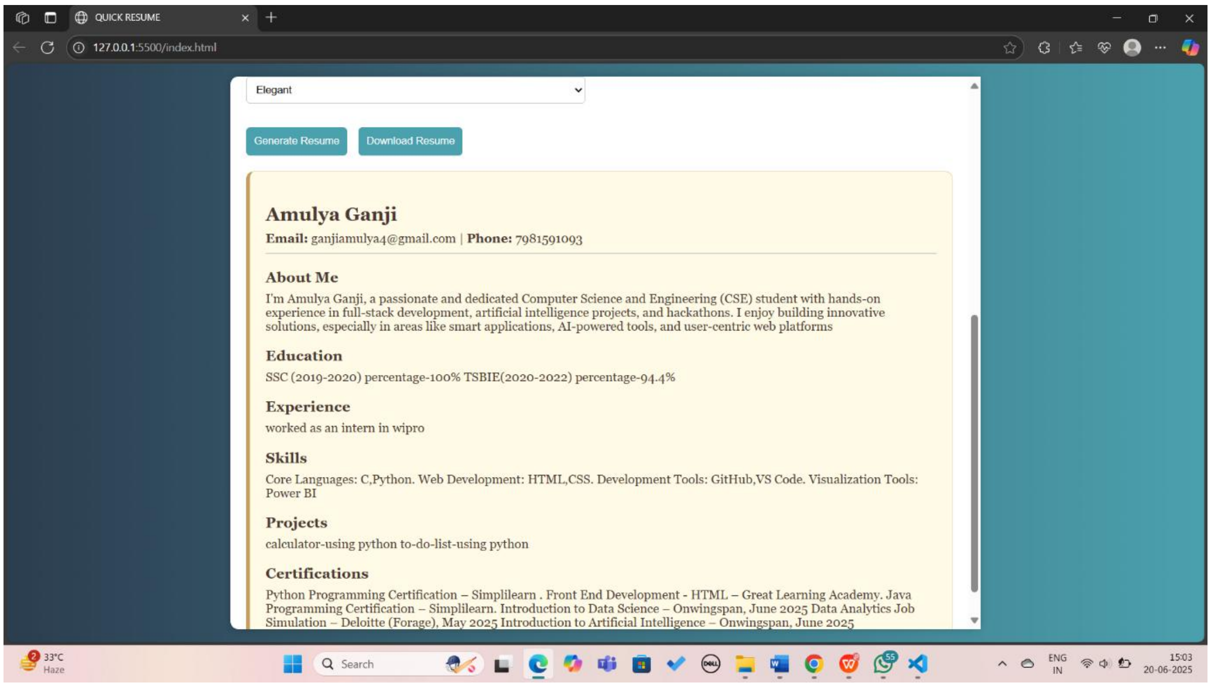
Projects

calculator-using python to-do-list-using python

Certifications

Python Programming Certification – Simplilearn . Front End Development - HTML – Great Learning Academy. Java Programming Certification – Simplilearn. Introduction to Data Science – Onwingspan, June 2025 Data Analytics Job Simulation – Deloitte (Forage), May 2025 Introduction to Artificial Intelligence – Onwingspan, June 2025

Figure 7.8 Choosing the Theme (Professional)



Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

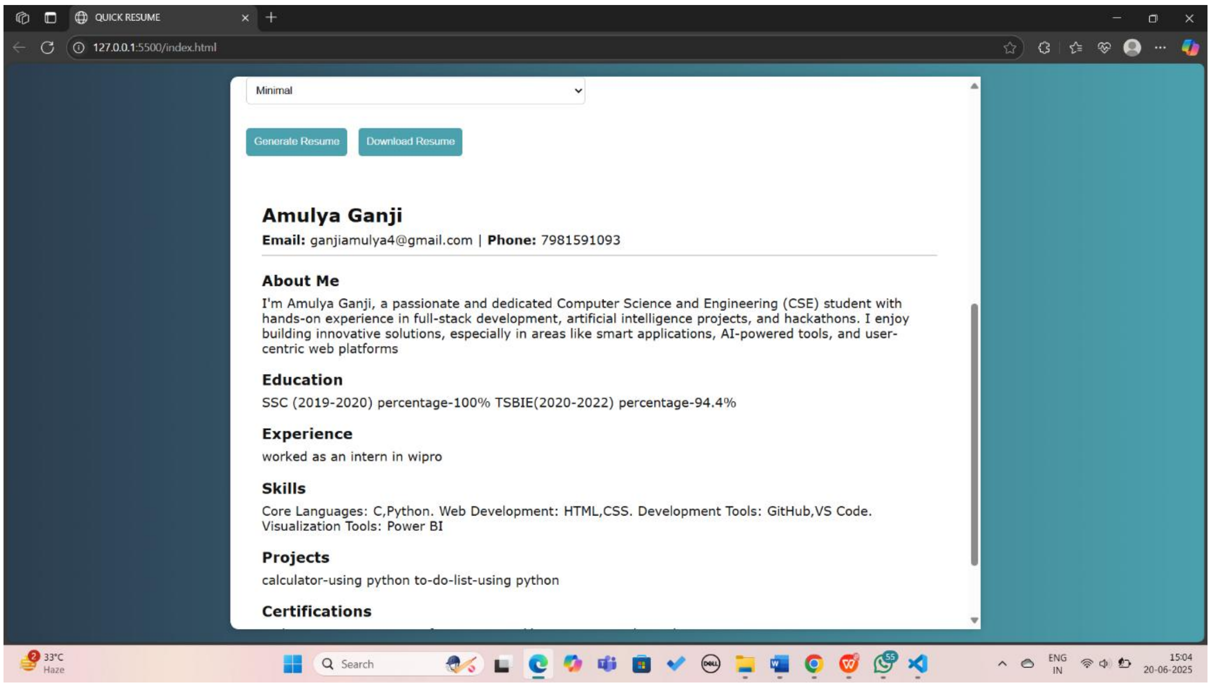
Projects

calculator-using python to-do-list-using python

Certifications

Python Programming Certification – Simplilearn . Front End Development - HTML – Great Learning Academy. Java Programming Certification – Simplilearn. Introduction to Data Science – Onwingspan, June 2025 Data Analytics Job Simulation – Deloitte (Forage), May 2025 Introduction to Artificial Intelligence – Onwingspan, June 2025

Figure 7.9 Choosing the Theme (Elegant)



Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

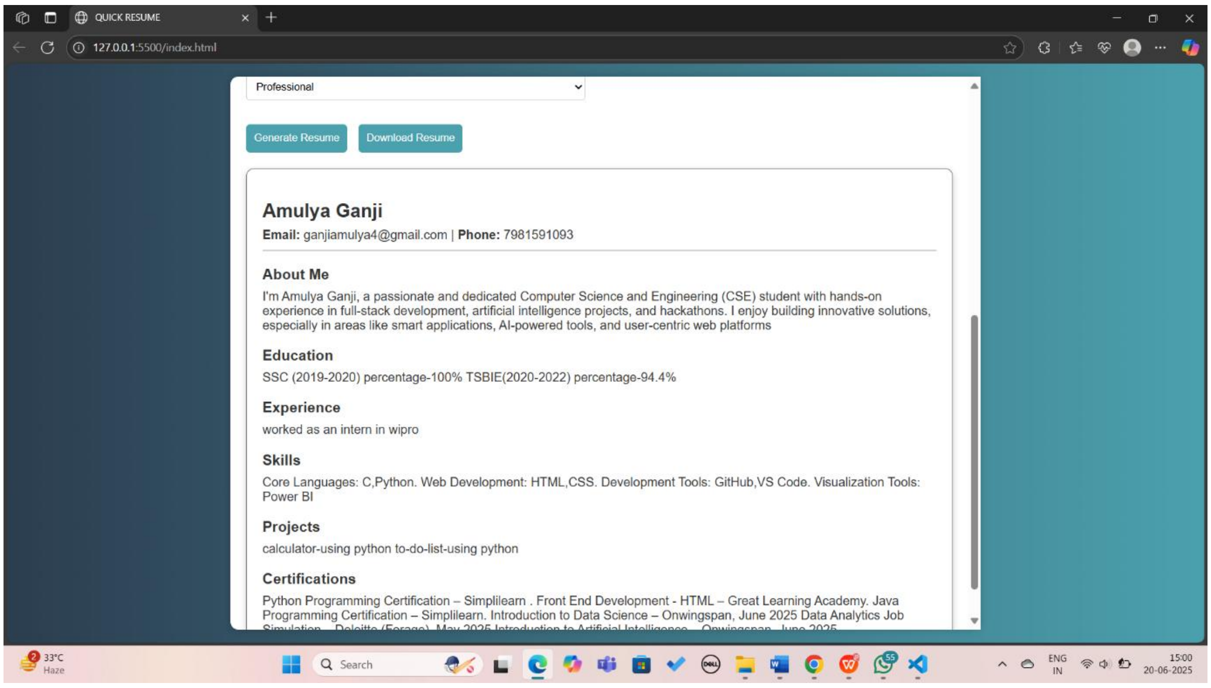
Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

Projects

calculator-using python to-do-list-using python

Certifications

Figure 7.10 Choosing the Theme (Minimal)



Professional

Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

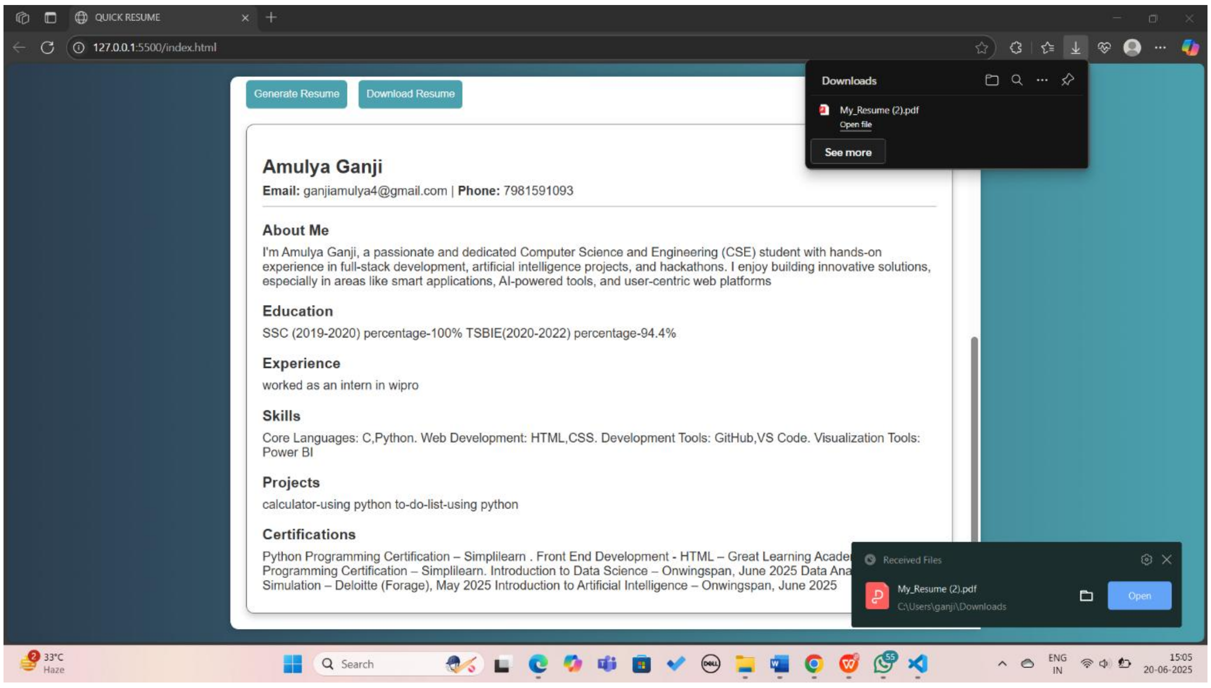
Projects

calculator-using python to-do-list-using python

Certifications

Python Programming Certification – Simplilearn . Front End Development - HTML – Great Learning Academy. Java Programming Certification – Simplilearn. Introduction to Data Science – Onwingspan, June 2025 Data Analytics Job Simulation – Deloitte (Forage), May 2025 Introduction to Artificial Intelligence – Onwingspan, June 2025

Figure 7.11 Preview of Resume



Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

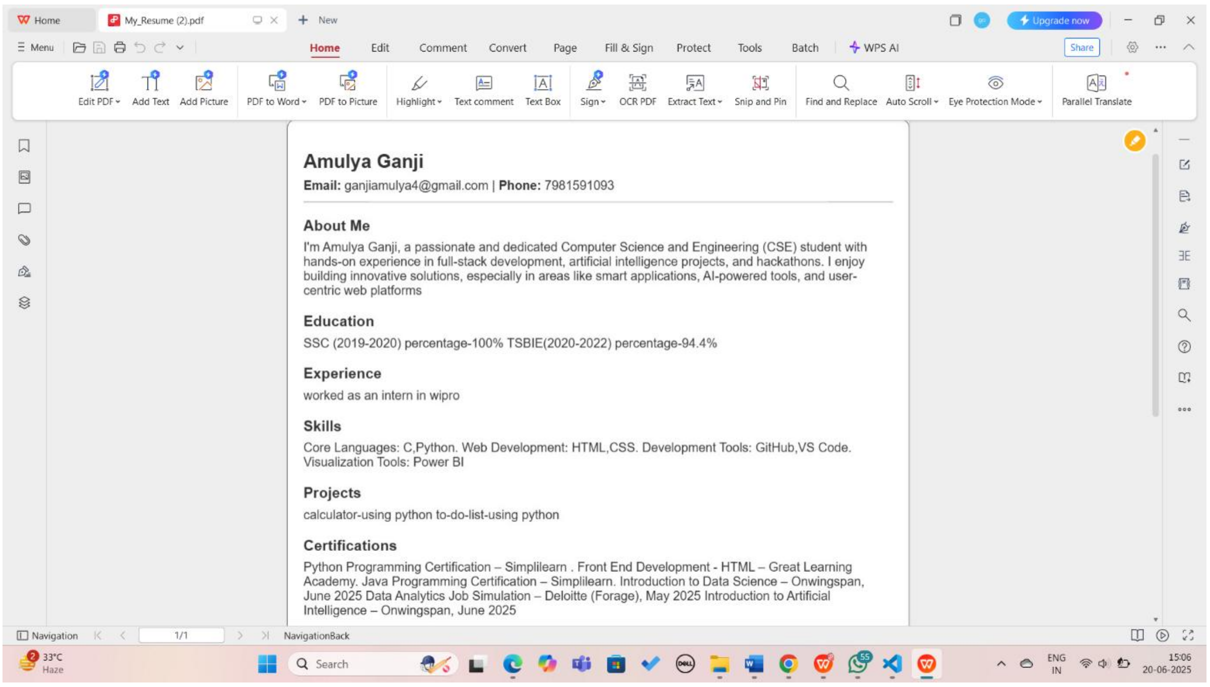
Projects

calculator-using python to-do-list-using python

Certifications

Python Programming Certification – Simplilearn . Front End Development - HTML – Great Learning Acade
Programming Certification – Simplilearn. Introduction to Data Science – Onwingspan, June 2025 Data Ana
Simulation – Deloitte (Forage), May 2025 Introduction to Artificial Intelligence – Onwingspan, June 2025

Figure 7.12 Download The Resume



Amulya Ganji

Email: ganjiamulya4@gmail.com | **Phone:** 7981591093

About Me

I'm Amulya Ganji, a passionate and dedicated Computer Science and Engineering (CSE) student with hands-on experience in full-stack development, artificial intelligence projects, and hackathons. I enjoy building innovative solutions, especially in areas like smart applications, AI-powered tools, and user-centric web platforms

Education

SSC (2019-2020) percentage-100% TSBIE(2020-2022) percentage-94.4%

Experience

worked as an intern in wipro

Skills

Core Languages: C,Python. Web Development: HTML,CSS. Development Tools: GitHub,VS Code. Visualization Tools: Power BI

Projects

calculator-using python to-do-list-using python

Certifications

Python Programming Certification – Simplilearn . Front End Development - HTML – Great Learning Academy. Java Programming Certification – Simplilearn. Introduction to Data Science – Onwingspan, June 2025 Data Analytics Job Simulation – Deloitte (Forage), May 2025 Introduction to Artificial Intelligence – Onwingspan, June 2025

Figure 7.13 Final Resume

CHAPTER 8

CONCLUSION

The Quick Resume Builder project successfully delivers a user-friendly, efficient, and accessible platform for generating professional resumes in real-time. The system provides a seamless experience from user data entry to dynamic theme selection and PDF generation.

Through structured modules such as form validation, theme customization, and automated PDF export, the system meets its primary objective: enabling users to create high-quality, customized resumes with minimal effort and time. The interface is responsive and intuitive, ensuring cross-platform compatibility and usability across various devices and browsers.

Additionally, the project underwent comprehensive system testing, ensuring that each component performs as expected and integrates well within the system. The application demonstrates the importance of modular design, user-centric development, and reliable functionality in web-based resume builders.

The development of this project has provided valuable insights into front-end development, JavaScript integration, responsive design, and the use of external libraries like **html2pdf.js**. It reflects the practical application of web technologies to solve real-life problems efficiently. The successful implementation of the Quick Resume Builder demonstrates how simple tools, when thoughtfully designed, can greatly enhance user productivity and experience. This project not only fulfills academic requirements but also showcases the potential to be scaled and deployed as a real-world solution for educational institutions, career portals, and individual job seekers.

CHAPTER 9

FUTURE SCOPE

The Quick Resume Builder application has successfully achieved its goal of providing users with a simple, fast, and effective way to create professional resumes. However, there is significant scope for future enhancement and scalability to meet evolving user needs and industry demands. One of the key areas for improvement lies in implementing user authentication and profile management. By introducing secure login and registration systems, users will be able to create personalized accounts, save multiple resumes, and retrieve or modify them at any time

Another important enhancement is the integration of cloud storage services, such as Firebase, AWS, or Google Drive. This feature would allow users to store their resumes in the cloud, providing accessibility from any device, eliminating the need for local storage, and facilitating resume sharing through public or private links. Furthermore, the application can be made more intelligent and user-centric by integrating AI-powered content recommendations. By using Natural Language Processing (NLP) techniques, the system can suggest optimized summaries, skill sets, and experience descriptions based on industry standards and user profiles.

The visual appeal of resumes can also be enhanced through advanced template customization, allowing users to choose from a wider range of professionally designed templates and modify colors, fonts, and layouts to suit their preferences. In addition, developing a mobile application version of the system for Android and iOS platforms will extend its reach and offer users the convenience of building and downloading resumes on the go.

Looking ahead, integrating the application with job portals like LinkedIn, Naukri, and Indeed could streamline the job application process, allowing users to directly apply to jobs using their generated resumes. The addition of multilingual support would further expand the accessibility of the system to users from various linguistic backgrounds. Lastly, implementing real-time collaboration features would enable multiple users, such as mentors or peers, to contribute to the resume-building process simultaneously.

CHAPTER 10

10.BIBLOGRAPHY

10.1 References

- Ian Sommerville, *Software Engineering*, 9th Edition, Pearson Education, 2011.
- Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition, McGraw-Hill, 2010.
- Herbert Schildt, *Java: The Complete Reference*, 9th Edition, McGraw-Hill Education, 2014.
- Bjarne Stroustrup, *The C++ Programming Language*, 4th Edition, Addison-Wesley, 2013.
- Pankaj Jalote, *An Integrated Approach to Software Engineering*, 3rd Edition, Springer, 2005.

10.2 Websites

- W3Schools - HTML, CSS, JavaScript Tutorials
  https://www.w3schools.com
- MDN Web Docs – Mozilla Developer Network
  https://developer.mozilla.org
- html2pdf.js Official Documentation
  https://ekoopmans.github.io/html2pdf.js/
- GeeksforGeeks – Programming & Web Technologies
  https://www.geeksforgeeks.org
- Stack Overflow – Developer Community
  https://stackoverflow.com