A

PROJECT REPORT

ON

Feature-rich, Resume Builder Application

SUBMITTED TO

SHIVAJIUNIVERSITY, KOLHAPUR

IN THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF DEGREE BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

MR.	AARYAN PRASANNA HOGADE	21UCS031
MR.	SHREYANSH NITIN KADAGE	21UCS037
MR.	PRATIK MAHESH KUMBHAR	21UCS047

UNDER THE GUIDANCE OF

Mrs. P. S. MORE

DKTE

Promoting Excellence in
Teaching, Learning & Research

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DKTE SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

2023-2024

D.K.T.E. SOCIETY'S

TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI
(AN AUTONOUMOUS INSTITUTE)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DKTE
Promoting Excellence in
Teaching, Learning & Research

CERTIFICATE

This is to certify that, project work entitled

Feature-rich, Resume Builder Application

is a bonafide record of project work carried out in this college by

MR.	AARYAN PRASANNA HOGADE	21UCS031
MR.	SHREYANSH NITIN KADAGE	21UCS037
MR.	PRATIK MAHESH KUMBHAR	21UCS047

is in the partial fulfillment of award of degree Bachelor in Engineering in Computer Science & Engineering prescribed by Shivaji University, Kolhapur for the academic year 2023-2024.

Mrs. P. S. MORE
(PROJECT GUIDE)

PROF.(DR.) D.V. KODAVADE
(HOD CSE DEPT.)

PROF. DR. L. S. ADMUTHE
(DIRECTOR)

EXAMINER: ________________

____

DECLARATION

We hereby declare that, the project work report entitled Feature-rich, Resume Builder Application which is being submitted to D.K.T.E. Society's Textile and Engineering Institute Ichalkaranji, affiliated to Shivaji University, Kolhapur is in partial fulfillment of degree B.tech (CSE). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy / Cyber / IPR Act amended from time to time.

Signature

MR. AARYAN PRASANNA HOGADE	21UCS031
MR. SHREYANSH NITIN KADAGE	21UCS037
MR. PRATIK MAHESH KUMBHAR	21UCS047

ACKNOWLEDGEMENT

With great pleasure we wish to express our deep sense of gratitude to Mrs. P. S. MORE for his valuable guidance, support and encouragement in completion of this project report. Also, we would like to take opportunity to thank our head of department Dr. D. V. Kodavade for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of Computer Science and Engineering Department for their support, help and assistance which they extended as and when required.

Thank you,

MR. AARYAN PRASANNA HOGADE	21UCS031
MR. SHREYANSH NITIN KADAGE	21UCS037
MR. PRATIK MAHESH KUMBHAR	21UCS047

ABSTRACT

In today's digital landscape, the significance of a well-crafted resume cannot be overstated, serving as a crucial tool for individuals to showcase their expertise and experiences to potential employers. Addressing the growing demand for streamlined resume creation processes, the Feature-rich Resume Builder Application emerges as a comprehensive solution. Harnessing the power of HTML, CSS, JavaScript, Bootstrap for frontend development, and Node.js, Express.js, and MySQL for backend functionalities, this project aims to provide users with an intuitive platform to effortlessly craft professional resumes.

The application enables dynamic interaction through Node.js and Express.js, ensuring smooth data processing and storage. HTML and CSS are employed to structure and style the web pages, offering an aesthetically pleasing and responsive layout. JavaScript enriches user experience by implementing interactive features and robust form validations. Additionally, the integration of Bootstrap framework ensures optimal performance across diverse screen sizes and devices, enhancing accessibility for users.

With the Feature-rich Resume Builder Application, users can input their personal details, educational background, work experiences, skills, and achievements into customizable templates. The system then generates a formatted resume document, ready for download or further customization. Furthermore, robust authentication mechanisms are incorporated to safeguard user data and ensure privacy.

In conclusion, this project showcases proficiency in a variety of frontend and backend technologies, while also addressing the practical need for efficient resume creation tools in today's competitive job market.

INDEX

1. Introduction

In today's fast-paced and highly competitive job market, the importance of a well-crafted resume cannot be overstated. It serves as a gateway for individuals to showcase their skills, experiences, and qualifications to potential employers. However, crafting a professional resume can often be overwhelming and time-consuming, particularly for those with limited design or formatting expertise.

To address this challenge, the Feature-rich Resume Builder Application harnesses the power of modern web technologies to offer users an intuitive platform for effortlessly creating polished resumes. By leveraging HTML, CSS, JavaScript, and Bootstrap for frontend development, and Node.js, Express.js, and MySQL for backend functionalities, this project aims to streamline the resume creation process and empower users to effectively showcase their professional profiles.

The application facilitates dynamic interaction through Node.js and Express.js, enabling seamless data handling and manipulation. HTML and CSS are employed to structure and style the web interface, ensuring a visually appealing and user-friendly experience. JavaScript enriches the platform's interactivity by implementing features such as real-time previews and robust form validations. Additionally, the integration of Bootstrap framework optimizes the application for various screen sizes and devices, promoting accessibility and usability.

Through the Feature-rich Resume Builder Application, users can input their personal information, educational background, work experience, skills, and achievements into customizable templates. The system then generates a formatted resume document, ready for download or further customization. Furthermore, robust authentication mechanisms are implemented to safeguard user data and privacy, ensuring a secure and reliable environment.

This project not only showcases proficiency in a range of web development technologies but also addresses the practical need for efficient resume creation tools in today's competitive job market. By providing users with a seamless and intuitive platform, the Feature-rich Resume Builder Application empowers individuals to present themselves effectively and stand out amidst the digital crowd.

A. Problem definition

In today's competitive job market, the ability to craft a polished and professional resume is paramount for job seekers striving to differentiate themselves. However, many individuals encounter obstacles in effectively designing and formatting their resumes, often due to limited access to tools or expertise in document design. Moreover, with the increasing reliance on digital platforms in the job application process, there's a rising demand for online resume builders that streamline resume creation.

The challenge addressed by the Feature-rich Resume Builder Application is the necessity for a user-friendly, web-based platform that enables individuals to effortlessly generate well-designed resumes tailored to their qualifications and experiences. This project endeavours to offer a solution that simplifies the resume creation process, eliminates the need for advanced design skills, and produces professional-looking resumes that are easily shareable or downloadable in various formats.

Key challenges to be tackled include:

- User Experience: Designing an intuitive user interface that enables seamless input of personal information, education background, work experience, skills, and achievements.
- Dynamic Formatting: Implementing features to dynamically format and structure resume content according to a single predefined template, ensuring consistency and readability.
- Personalization: Providing options for users to customize specific sections of their resumes to align with their preferences and the requirements of diverse industries or job positions.
- Cross-Platform Compatibility: Ensuring compatibility across various web browsers and devices to accommodate users accessing the platform from desktops, laptops, tablets, and smartphones.
- Data Security: Implementing robust authentication mechanisms to safeguard user data and privacy, recognizing that resumes may contain sensitive personal information.

By addressing these challenges, the Feature-rich Resume Builder Application aims to empower users to create professional resumes efficiently, thereby enhancing their prospects in the job market and facilitating smoother transitions in their career paths.

B. Aim and objective of the project

The Feature-rich Resume Builder Application endeavours to create a user-friendly web platform that simplifies the process of crafting professional resumes. Through the integration of HTML, CSS, JavaScript, Bootstrap for frontend development, and Node.js, Express.js, and MySQL for backend functionalities, the project seeks to empower users to generate polished resumes tailored to their unique qualifications and experiences.

Objectives:

- User Interface Development: Develop an intuitive and user-friendly interface allowing seamless input of personal information, educational background, work experience, skills, and achievements.

- Dynamic Content Handling: Implement backend routing with Node.js and Express.js to facilitate dynamic content generation, processing, and storage, ensuring smooth interaction between users and the application.

- Responsive Design: Utilize HTML, CSS, and Bootstrap to craft a responsive layout adaptable to various screen sizes and devices, thereby enhancing accessibility and user experience.

- Interactive Features: Employ JavaScript to introduce interactive elements such as form validations, auto-formatting, and real-time previews of resume content, fostering usability and user engagement.

- Downloadable Output: Enable users to download their resumes in diverse formats (e.g., PDF) for offline utilization or sharing with prospective employers, ensuring compatibility and professional presentation.

- Security Measures: Implement robust authentication mechanisms utilizing Node.js and MySQL to safeguard user data and privacy, thereby protecting sensitive personal information stored within the application.

- Testing and Quality Assurance: Conduct comprehensive testing and debugging procedures to ensure the reliability, functionality, and performance of the application across diverse environments and usage scenarios.

Through the attainment of these objectives, the Feature-rich Resume Builder Application aspires to furnish users with an inclusive tool for crafting professional resumes, thereby enhancing their job prospects and facilitating their career progression objectives.

C. Scope and limitation of the project

Scope:

- User Interface: The project will feature a user-friendly interface allowing users to seamlessly input their personal details, educational background, work experience, skills, and achievements.

- Dynamic Content Handling: Node.js and Express.js will be utilized to handle dynamic content generation, processing, and storage, ensuring efficient interaction and data management.

- Responsive Design: Development will employ HTML, CSS, JavaScript, and Bootstrap to ensure a responsive layout adaptable to various devices and screen sizes.

- Single Template Customization: Users will have the option to customize the design, layout, and styling of their resumes within a single predefined template, aligning with their preferences and industry standards.

- Downloadable Output: The project will enable users to download their resumes in formats such as PDF for offline use or sharing with potential employers.

- Security Measures: Authentication mechanisms utilizing Node.js and MySQL will be implemented to safeguard user data and privacy, ensuring the confidentiality of sensitive personal information.

Limitations:

- Limited Template Options: Due to resource constraints, the project will offer a single predefined template, potentially limiting customization options for users.

- Basic Feature Set: Initially, the project may offer a basic set of features for resume creation and customization, with potential limitations in advanced functionalities such as custom layouts or advanced design elements.

- Browser Compatibility: While efforts will be made to ensure compatibility with major web browsers, the application may experience limitations or inconsistencies in functionality across certain browser versions or platforms.

- Data Security: Despite implemented security measures, the project may still be susceptible to security threats such as data breaches, requiring ongoing monitoring and updates to mitigate risks.

- Performance: Depending on server resources and network conditions, the performance of the application may vary, potentially leading to delays or interruptions in service during peak usage periods.

- Scalability: The project may have limitations in scalability, particularly in handling a large volume of concurrent users or expanding to accommodate additional features and functionalities in the future.

D. Timeline of the project

Phase 1: Getting Started (2 weeks)

- Figure out what we want to achieve with the project and what tasks need to be done.
- Look at what similar projects are doing and what people want from them.
- Choose the right tools and technology for our project.
- Talk to everyone involved to understand exactly what they need from the project.
- Write down all our plans and ideas in documents we can refer back to later.

Phase 2: Designing the Look (3 weeks)

- Draw sketches and pictures of how the website will look and work.
- Plan how the information will be stored in a database.
- Keep improving our designs until they're just right.
- Make a basic version of the website to show people and get feedback.
- Make everything look polished and professional.

Phase 3: Building the Website (6 weeks)

- Get everything ready for building the website.
- Use HTML, CSS, JavaScript, and other tools to make the website work.
- Use Node.js, Express.js, and MySQL to handle the data and make the website interactive.
- Make sure only the right people can access certain parts of the website.
- Let users pick different styles for their resumes and download them in different formats.

Phase 4: Testing and Fixing (3 weeks)

- Test every part of the website to make sure it works like it's supposed to.
- Fix any problems we find and make sure everything runs smoothly together.
- Make sure the website is easy and enjoyable to use for everyone who tries it.

Phase 5: Launching and Helping Out (2 weeks)

- Put the website online so everyone can use it.
- Make sure everything is perfect before we tell people about it.
- Keep an eye on how the website is doing and help anyone who has questions or problems.

E. Project Management Plan

Project Initiation:

- Kick off the project with a meeting to introduce team members and stakeholders.
- Review project goals, what needs to be done, and when it should be done.
- Define who will do what in the project team.
- Set up ways for everyone to communicate and share information.
- Get the tools and systems ready to help manage the project.

Planning:

- Figure out exactly what needs to be done for the project.
- Break down the project into smaller tasks to make it easier to manage.
- Make a plan with dates for each task and when they need to be finished.
- Work out what tasks depend on each other and which ones are most important.
- Make sure we have enough people, money, and equipment for the project.
- Plan how to deal with problems or changes that might come up.
- Decide how we'll make sure everything we do meets our standards for quality.
- Set up a way to deal with any changes that happen during the project.

Execution:

- Give each person on the team their jobs and make sure they have what they need to do them.
- Make the parts of the resume builder application we planned, making sure they look and work the way they should.
- Have regular meetings to see how things are going and fix any problems.
- Check that everything we're doing meets our standards and the plan we made.
- Keep everyone who's interested in the project up to date and listen to what they think.
- Keep track of everything we're doing so we can learn from it later.

Monitoring and Control:

- Keep an eye on how well the project is going by looking at important numbers and how things are going.
- Have regular meetings to see if there are any problems we need to fix.
- Change the plan if we need to because something isn't going the way we expected.
- Make sure we're doing what we said we would, on time and on budget.
- Tell everyone if something changes in the project.
- Make sure everything we're doing meets our standards for quality.

Project Closure:

- Have a meeting to talk about what we've achieved in the project.
- Get the people who asked for the project to say they're happy with what we've done.
- Write down what we learned from doing the project so we can do things better next time.
- Keep everything we've done in case we need it later.
- Stop spending money on the project and close the accounts.
- Celebrate the good work everyone did on the project.

2.Background study and literature overview

a. Literature overview

Numerous studies underscore the significance of resume builder applications in aiding individuals with crafting impactful resumes. Research conducted by Shakti Arora and colleagues highlights the pivotal role of these applications, leveraging advancements in natural language processing (NLP) and machine learning to streamline the resume creation journey for users1. These platforms provide a myriad of features, including template selection, content suggestion, and formatting assistance, thereby simplifying the task of showcasing one's skills and experiences.

As technological advancements continue to unfold, resume builder applications are undergoing significant evolution, incorporating innovative features to further enhance user experience. Companies like Nvidia are at the forefront of developing cutting-edge technologies aimed at revolutionizing the resume creation process, signalling a promising future for these applications.

b. Critical appraisal of other people's work:

Despite the considerable attention given to resume builder applications, the existing literature exhibits noteworthy limitations. Some studies have focused narrowly on specific features or algorithms, potentially constraining the overall effectiveness of these applications. Additionally, concerns have been raised regarding the potential introduction of biases into generated resumes, raising questions about the fairness of the recruitment process.

Further exploration through research is imperative to investigate diverse and robust approaches to resume building, taking into account factors such as user demographics and ethical considerations. Collaborative efforts across disciplines are indispensable for the development of inclusive systems that not only streamline the resume creation process but also uphold principles of fairness and accuracy.

c. Investigation of the current project and related work:

Resume.io:

Resume.io is a forward-thinking resume builder platform engineered to simplify the resume creation process for users. Through its intuitive interface, Resume.io provides users with an array of tools and functionalities to craft compelling resumes swiftly and effectively.

With Resume.io, users have access to a diverse selection of templates, enabling them to personalize layouts and receive immediate feedback on content and formatting choices. The platform harnesses advanced algorithms to suggest pertinent keywords, optimize section organization, and maintain coherence throughout the resume.

Moreover, Resume.io equips users with valuable insights into prevailing job market trends and skill demands, empowering them to tailor their resumes to align with industry standards and employer expectations.

By harnessing cutting-edge technology and adhering to user-centric design principles, Resume.io endeavours to redefine the resume building experience, offering a seamless and efficient solution for individuals aiming to propel their careers forward.

3.Requirement analysis

a. Requirement Gathering

1. Functional Requirements:

- Determine the breadth of functionalities offered by the resume builder, encompassing features such as template selection, content suggestions, and formatting assistance.
- Identify the various forms of input the resume builder should accommodate, ranging from user-input text to uploaded documents or imported data from professional networking sites.
- Define mechanisms for the resume builder to learn and enhance its capabilities over time, possibly through user feedback mechanisms or machine learning algorithms.
- Specify the user interface design for the resume builder, including navigation, input methods, and feedback mechanisms.
- Outline methods for presenting information to users, such as previewing resumes, highlighting errors, and suggesting improvements.

2. Technical Requirements:

- Determine algorithms and methodologies for parsing and analysing user-provided data, particularly text analysis for identifying skills and experiences.
- Define necessary natural language processing (NLP) capabilities to effectively understand and interpret user input.
- Specify how the resume builder will utilize context to offer relevant suggestions and recommendations to users.
- Identify integration points for the resume builder with external systems, such as professional networking platforms or job search engines.
- Determine hardware and software prerequisites for the resume builder, ensuring compatibility with various operating systems and web browsers.

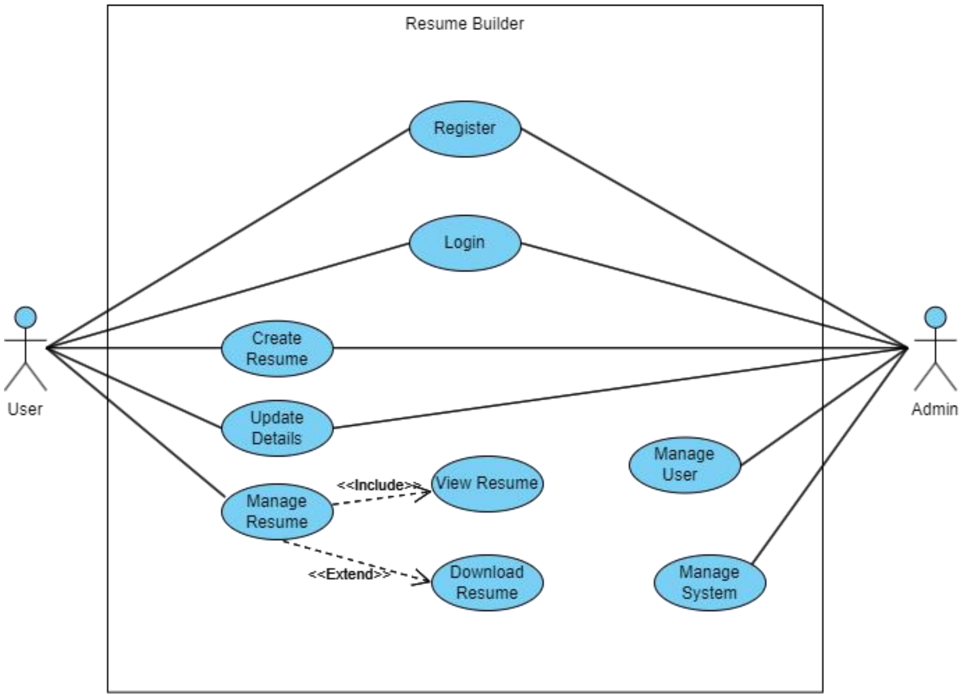
3. User Requirements:

- Identify the target user demographics for the resume builder, including career stage, specific needs, and any demographic information relevant to their usage.
- Understand user objectives and expectations when utilizing the resume builder, such as creating polished resumes, applying for jobs, or updating existing resumes.
- Identify any user preferences or requirements regarding resume formatting, content organization, or customization options.
- Understand preferred modes of interaction with the resume builder, whether through a web interface, mobile application, or integration with other platforms.

b. Requirement specification

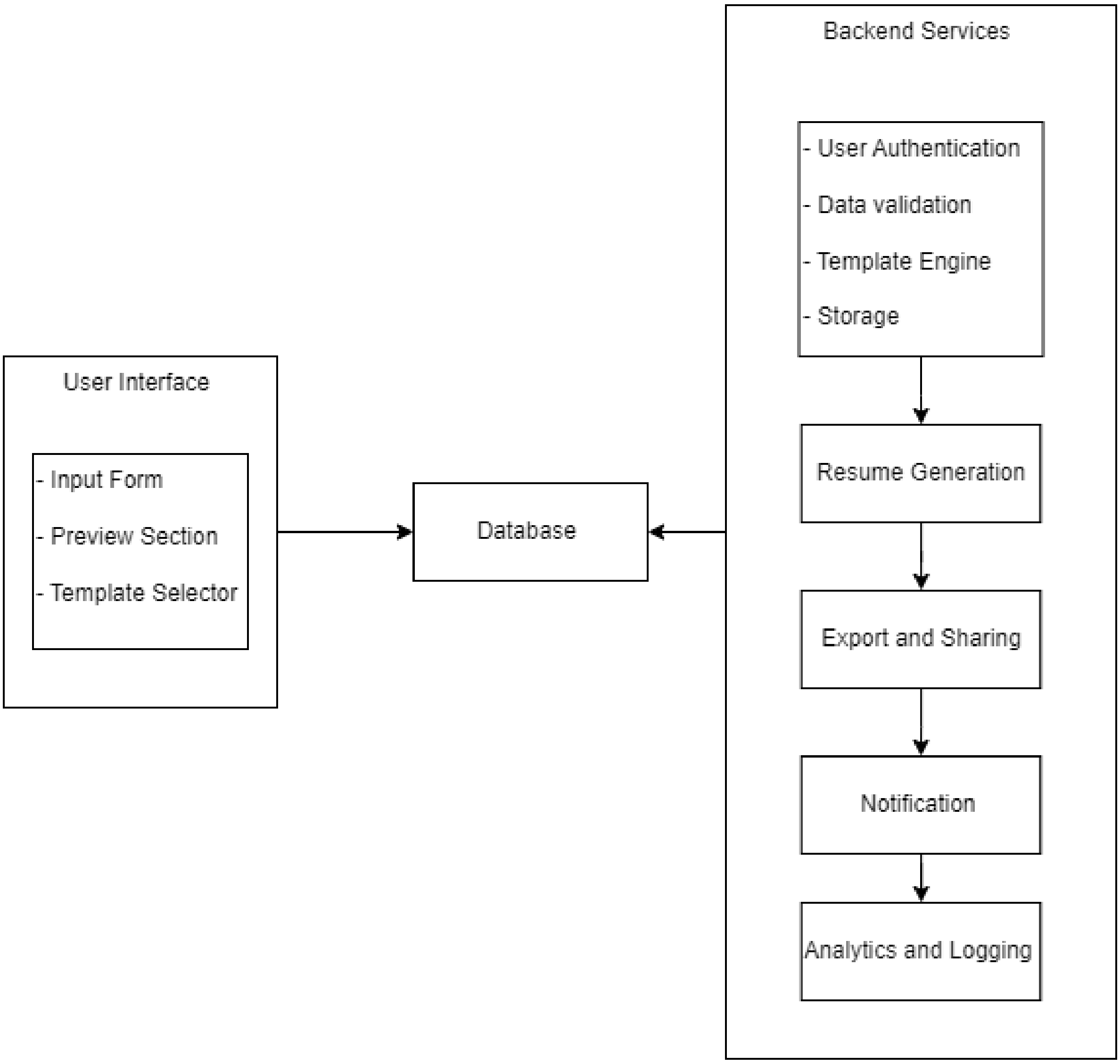
The aim of the resume builder software is to simplify the resume writing process and improve users' job application success. It offers intuitive interfaces for easy input of professional information and provides guidance on resume structure, content, and formatting. With the integration of natural language processing (NLP) capabilities, the software accurately interprets user input and suggests relevant improvements to resume content. Additionally, it seamlessly integrates with external platforms such as professional networking sites and job search engines, enriching users' job-seeking experience. Key features include template selection, content suggestion, error highlighting, and real-time previewing, empowering users to create tailored resumes aligned with their career objectives. Furthermore, the software prioritizes high performance, scalability, security, and compatibility to ensure a seamless user experience across diverse needs and platforms.

c. Use case Diagram

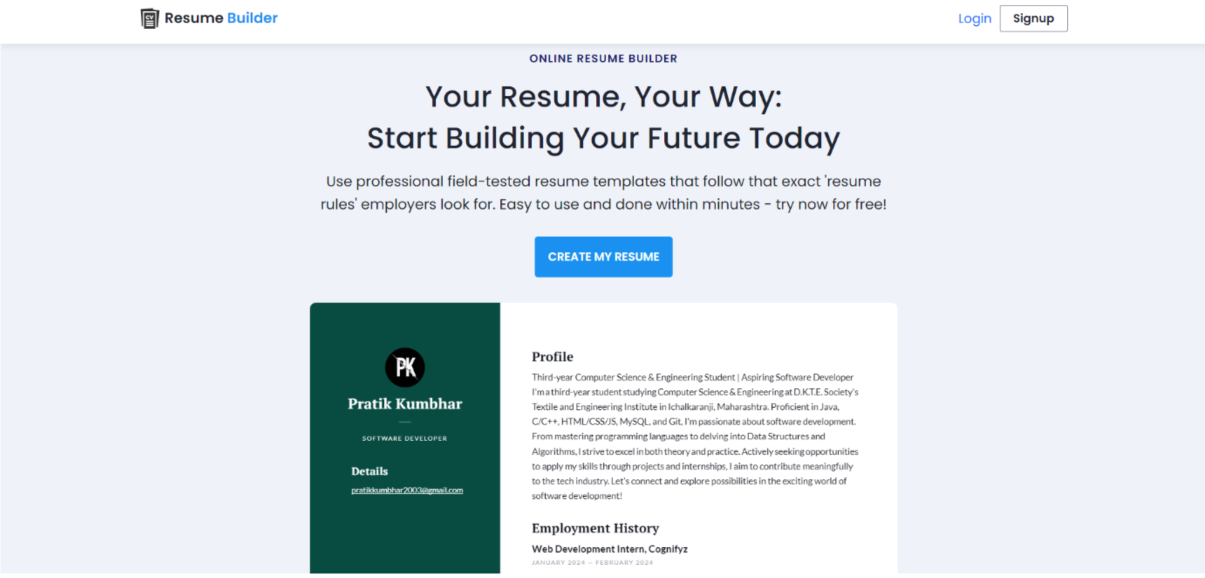


4.System Design

a. Architectural design:



b. User Interface Design:



Resume Builder

Login Signup

ONLINE RESUME BUILDER

Your Resume, Your Way:
Start Building Your Future Today

Use professional field-tested resume templates that follow that exact 'resume rules' employers look for. Easy to use and done within minutes - try now for free!

CREATE MY RESUME

Pratik Kumbhar

SOFTWARE DEVELOPER

Details

pratikkumbhar2003@gmail.com

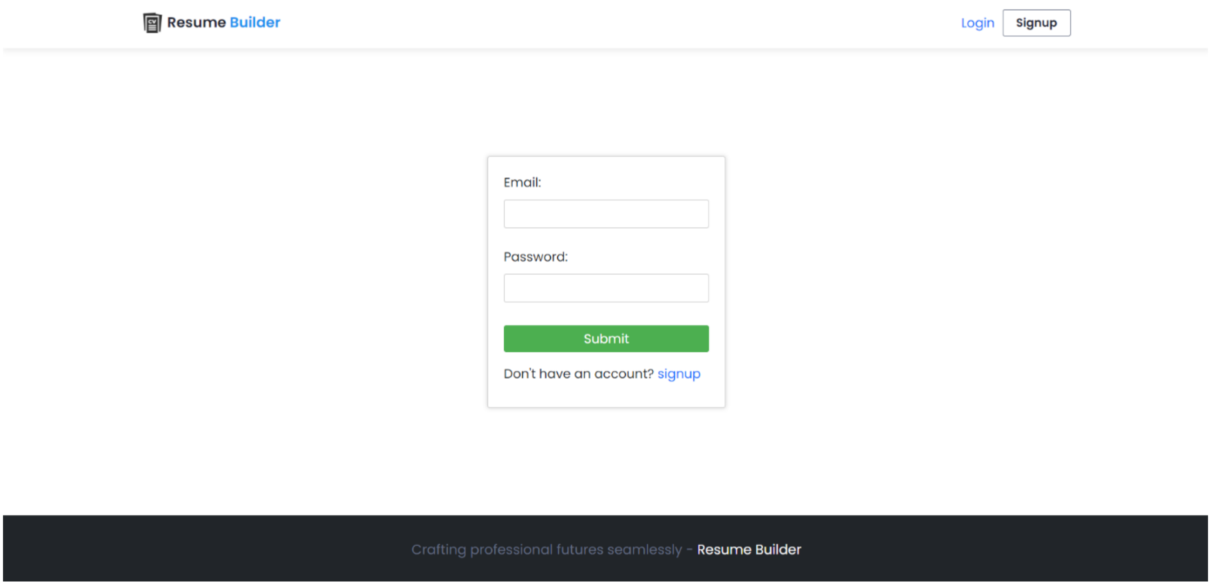
Profile

Third-year Computer Science & Engineering Student | Aspiring Software Developer
I'm a third-year student studying Computer Science & Engineering at D.K.T.E. Society's Textile and Engineering Institute in Ichalkaranji, Maharashtra. Proficient in Java, C/C++, HTML/CSS/JS, MySQL, and Git, I'm passionate about software development. From mastering programming languages to delving into Data Structures and Algorithms, I strive to excel in both theory and practice. Actively seeking opportunities to apply my skills through projects and internships, I aim to contribute meaningfully to the tech industry. Let's connect and explore possibilities in the exciting world of software development!

Employment History

Web Development Intern, Cognifyz

JANUARY 2024 – FEBRUARY 2024



Resume Builder

Login Signup

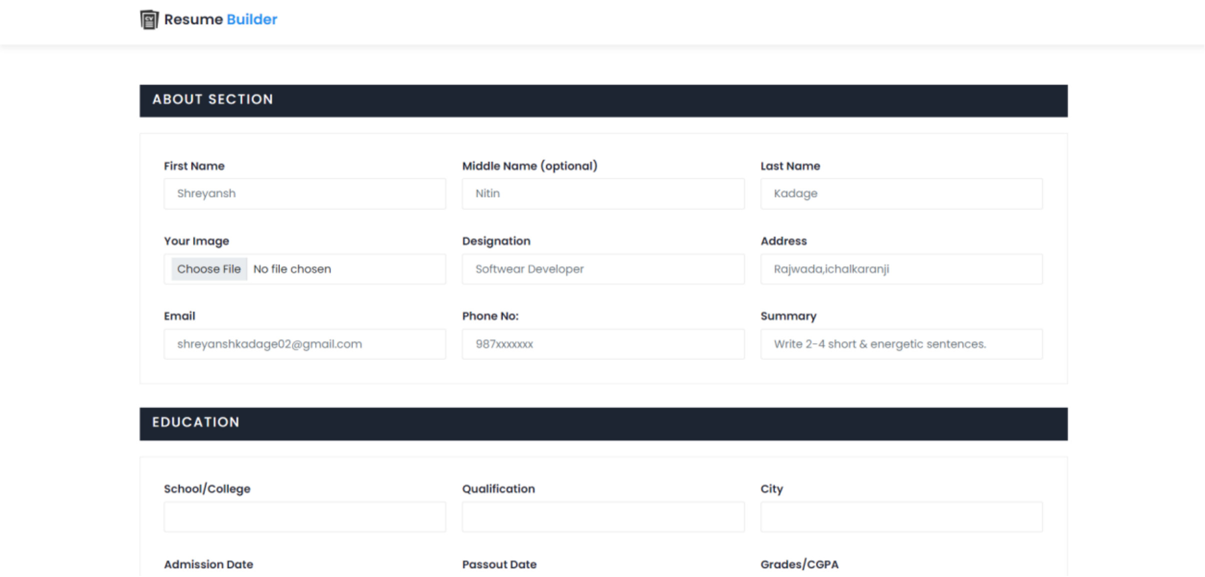
Email:

Password:

Submit

Don't have an account? signup

Crafting professional futures seamlessly - Resume Builder



Resume Builder

ABOUT SECTION

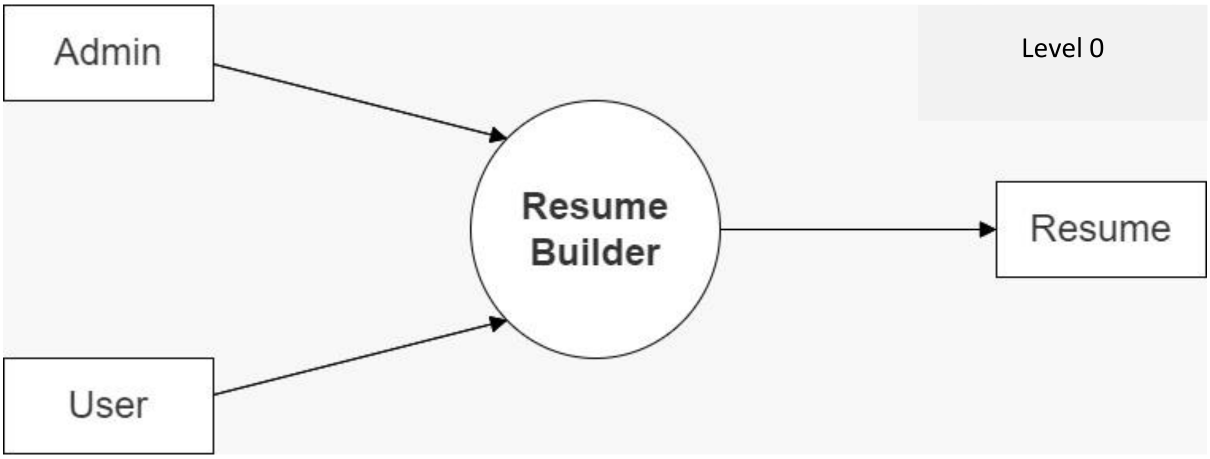
First Name Shreyansh	Middle Name (optional) Nitin	Last Name Kadage
Your Image Choose File No file chosen	Designation Softwear Developer	Address Rajwada,ichalkaranji
Email shreyanshkadage02@gmail.com	Phone No: 987xxxxxxx	Summary Write 2-4 short & energetic sentences.

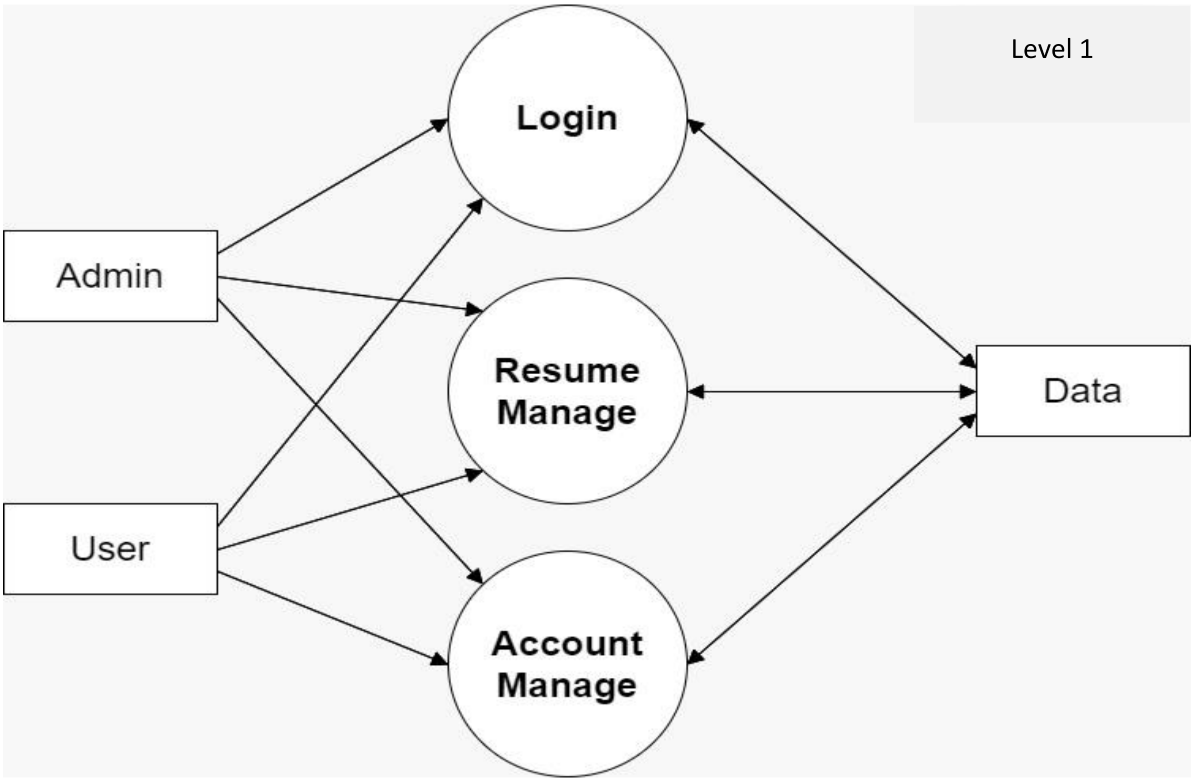
EDUCATION

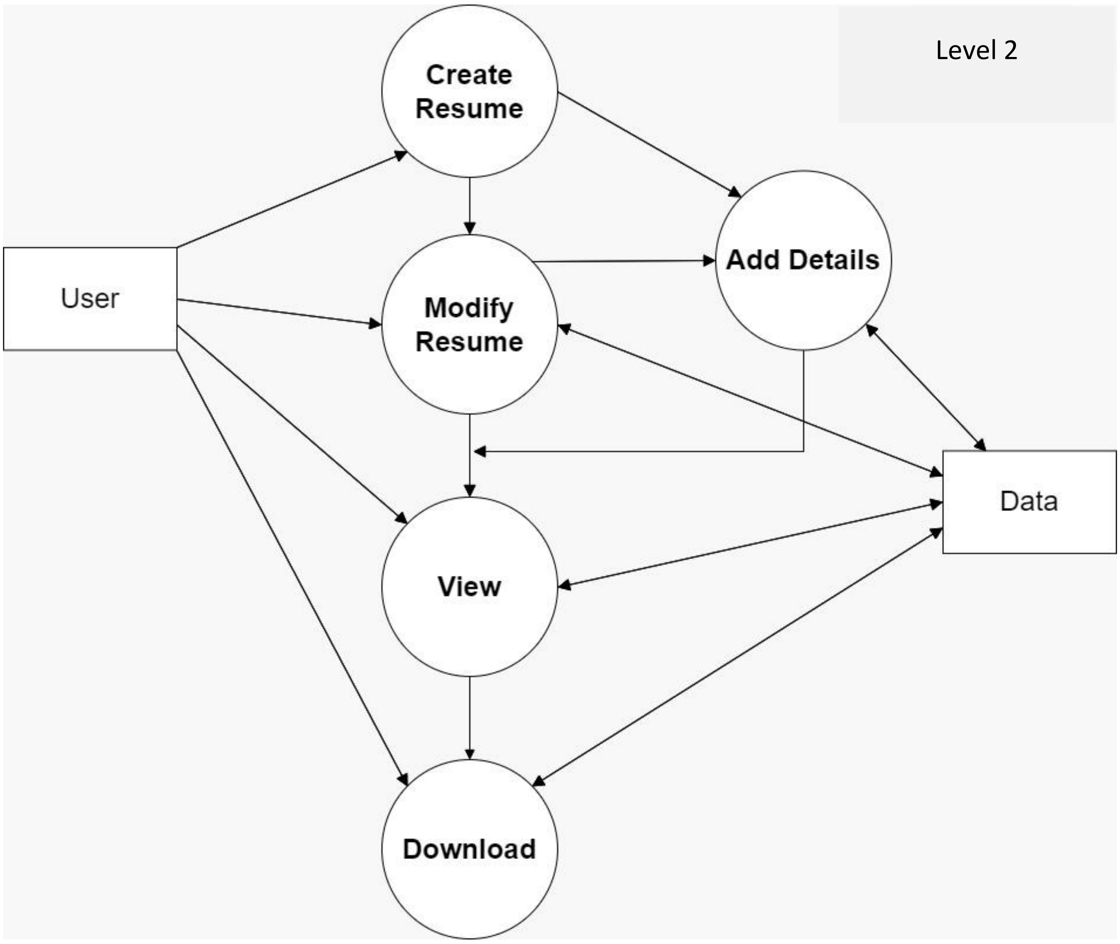
School/College	Qualification	City
Admission Date	Passout Date	Grades/CGPA

c. System Modeling:

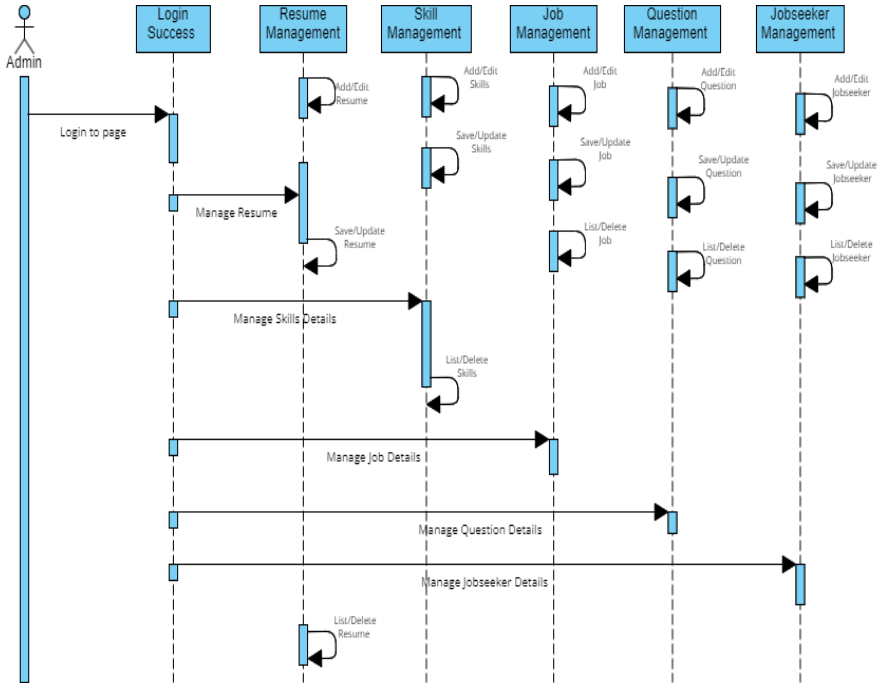
1.Dataflow Diagram:



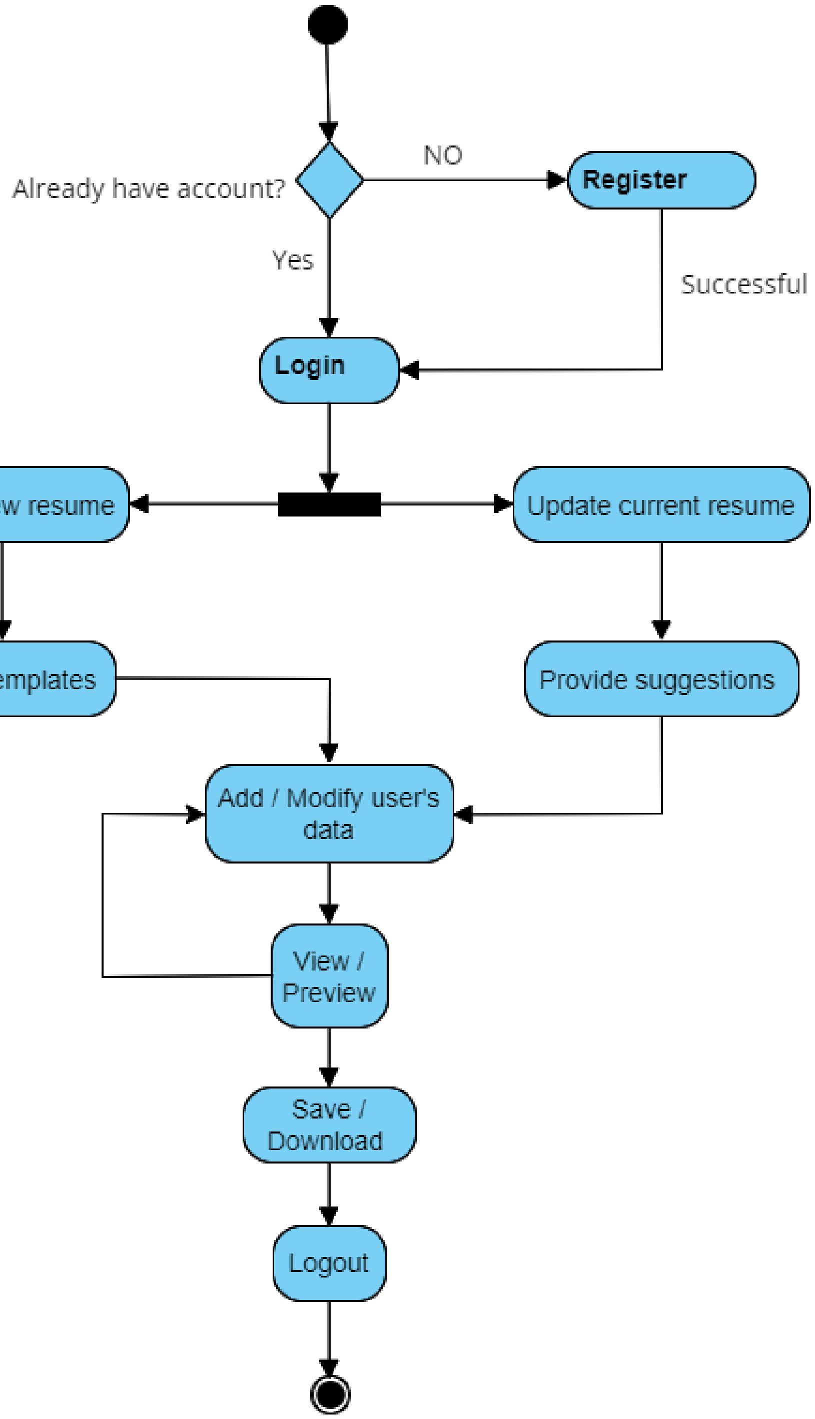



Level 2
Create Resume
User
Modify Resume
Add Details
Data
View
Download

2. Sequence Diagram:



3. Activity Diagram:



5.Implementation

a. Environmental Setting for Running the Project

Software Requirements:

- Operating System: The project is designed to run on any modern operating system such as Windows, macOS, or Linux.
- Web Browser: The website is compatible with popular web browsers including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- Server Environment: Node.js and Express.js are required for the backend routing. Ensure that Node.js and npm (Node Package Manager) are installed on your system.
- Database Management System (DBMS): MySQL is utilized for database management. Ensure that MySQL is installed and configured properly.
- Web Development Tools: Developers working on the project should have access to text editors or integrated development environments (IDEs) such as Visual Studio Code, Sublime Text, or JetBrains WebStorm for coding and debugging purposes.

Hardware Requirements:

- Processor: The project does not have specific processor requirements and should run on most modern CPUs.
- Memory (RAM): Adequate RAM is required to ensure smooth operation of the development environment and any necessary software components. A minimum of 4GB RAM is recommended.
- Storage: Sufficient disk space is needed for storing project files, dependencies, and database content. At least 1GB of free disk space is advisable.
- Internet Connection: An internet connection is necessary for accessing external libraries, dependencies, and resources, as well as for deploying the project to a web server.

Dependencies:

- Frontend Frameworks: HTML, CSS, JavaScript, Bootstrap.
- Backend Frameworks: Node.js, Express.js.
- Database Libraries: MySQL.

b. Detailed Description of Methods

Frontend Development

User Interface Design:

Design Principles and UI/UX Considerations:

- The design focuses on simplicity and ease of use, ensuring an intuitive and user-friendly interface. This is achieved through a clear layout and straightforward navigation.
- Layouts are structured to guide users seamlessly through the resume creation process.
- The color scheme and typography are chosen for readability and professional aesthetics, using Bootstrap's default styles with custom modifications.
- Visual elements such as buttons, forms, and icons are designed to enhance user experience and engagement, ensuring that users can easily interact with the application.

Technologies Used:

- HTML: Used for structuring the web pages.
- CSS: Used for styling and layout of the web pages.
- JavaScript: Used for adding interactivity and dynamic behavior to the web pages.
- Bootstrap: Used for responsive design and pre-built components, ensuring the application is mobile-friendly.
- Rationale: These technologies were chosen for their robustness, ease of use, and extensive community support. Bootstrap ensures the application is mobile-friendly and responsive, while JavaScript adds dynamic functionality.

Features Implemented:

- Resume Creation Form: Users can enter personal information through a structured form.
- Sections for Work Experience, Education, and Skills: Users can add details about their work history, academic background, and skills.
- Interactive Elements: Users can add, edit, and delete entries within their resume using dynamic form fields.
- Preview Functionality: Users can view a preview of their generated resume before finalizing it.

Backend Development

Server-Side Logic:

- Handling Requests and Processing Data: The backend, built with Node.js and Express.js, handles HTTP requests, processes data, and interacts with the MySQL database.
- RESTful API Endpoints: Designed to manage user data, resumes, experiences, education, and other resources. Each endpoint follows REST principles to ensure a standardized communication format.

Database Management:

- Database Schema and Models: The MySQL database schema includes tables for user information, resumes, work experiences, education entries, and skills. Only user login and signup information are stored in the database.
- Choice of DBMS: MySQL was chosen for its reliability, ease of use, and strong integration with Node.js applications.
- Database Interactions: Utilizing MySQL queries to manage user data and resume-related information.

Authentication and Authorization:

- Authentication Mechanisms: Implemented using Express.js middleware to handle user registration, login, and logout functionalities.
- Authorization: Ensuring secure access to user data through session management and access control mechanisms.

Form Validation with Regular Expressions:

- String Validation (strRegex): Validates that the input contains only letters and spaces.
- Email Validation (emailRegex): Ensures the input is in a valid email format.
- Phone Number Validation (phoneRegex): Supports various phone number formats including international numbers.
- Digit Validation (digitRegex): Ensures the input contains only digits.
- Dynamic Form Handling:
- Fetching User Inputs: The getUserInputs function collects all form inputs, including dynamic lists for achievements, experiences, education, projects, and skills.
- Validation Event Listeners: Event listeners are added to form fields to validate inputs in real-time, providing immediate feedback to users.

Data Display and Preview:

- Display Function (displayCV): Updates the display elements with the user's input data to show a preview of the resume.
- List Data Display (showListData): Dynamically creates and appends HTML elements to display lists of achievements, experiences, education, projects, and skills.

Image Upload and Preview:

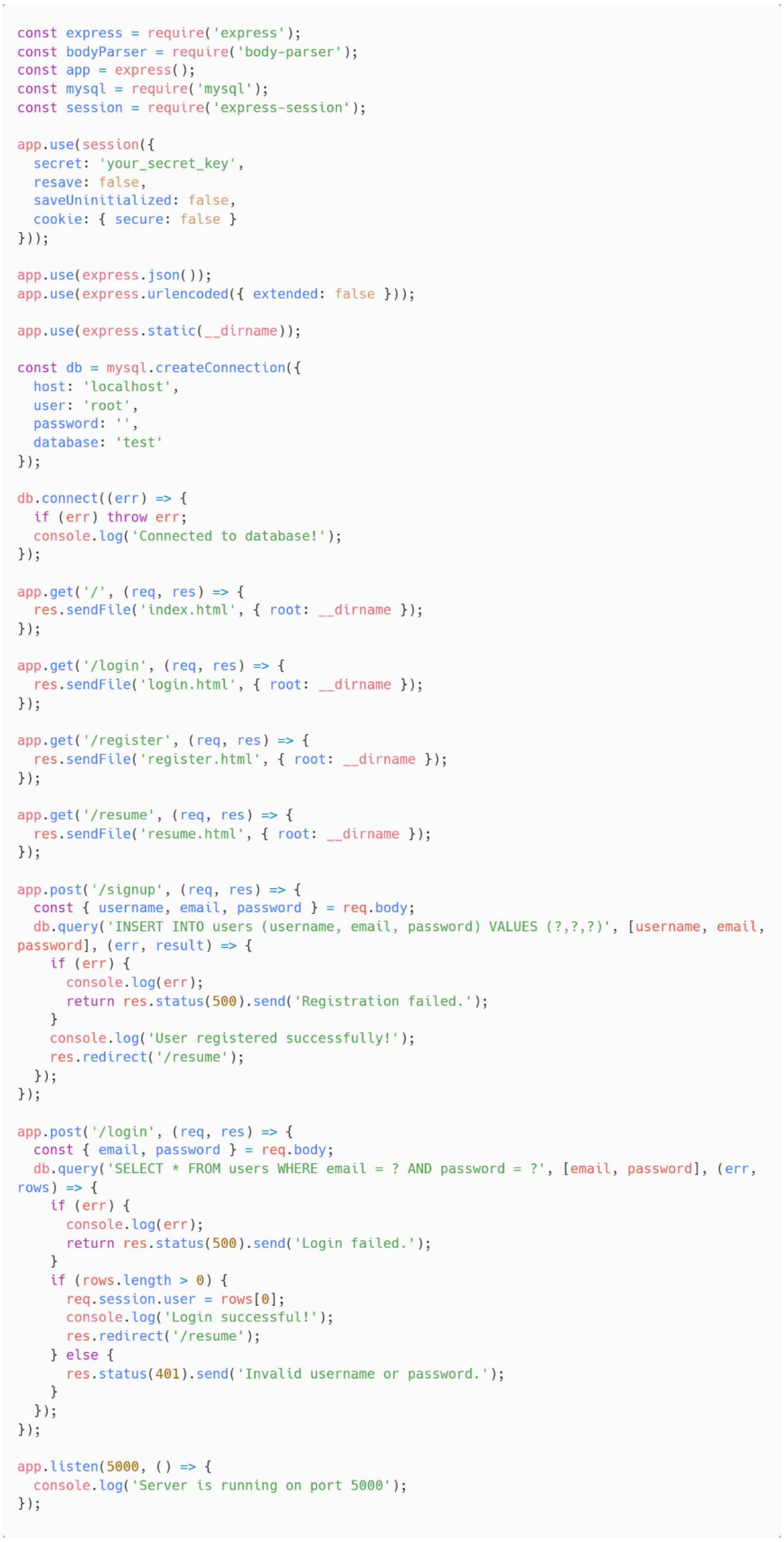
- Image Preview Function (previewImage): Uses a FileReader to read the selected image file and display it in the preview section.

CV Generation and Printing:

- Generate CV Function (generateCV): Gathers all user inputs, validates them, and updates the preview section with the data.
- Print CV Function (printCV): Initiates the browser's print functionality to allow users to print their generated resume.

c. Implementation Details

The backend Node.js application is structured to handle user authentication, session management, and CRUD operations for resume data. Below is an outline of how the backend logic is implemented:



```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const mysql = require('mysql');
const session = require('express-session');

app.use(session({
  secret: 'your_secret_key',
  resave: false,
  saveUninitialized: false,
  cookie: { secure: false }
}));

app.use(express.json());
app.use(express.urlencoded({ extended: false }));

app.use(express.static(__dirname));

const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'test'
});

db.connect((err) => {
  if (err) throw err;
  console.log('Connected to database!');
});

app.get('/', (req, res) => {
  res.sendFile('index.html', { root: __dirname });
});

app.get('/login', (req, res) => {
  res.sendFile('login.html', { root: __dirname });
});

app.get('/register', (req, res) => {
  res.sendFile('register.html', { root: __dirname });
});

app.get('/resume', (req, res) => {
  res.sendFile('resume.html', { root: __dirname });
});

app.post('/signup', (req, res) => {
  const { username, email, password } = req.body;
  db.query('INSERT INTO users (username, email, password) VALUES (?,?,?)', [username, email,
password], (err, result) => {
    if (err) {
      console.log(err);
      return res.status(500).send('Registration failed.');
    }
    console.log('User registered successfully!');
    res.redirect('/resume');
  });
});

app.post('/login', (req, res) => {
  const { email, password } = req.body;
  db.query('SELECT * FROM users WHERE email = ? AND password = ?', [email, password], (err,
rows) => {
    if (err) {
      console.log(err);
      return res.status(500).send('Login failed.');
    }
    if (rows.length > 0) {
      req.session.user = rows[0];
      console.log('Login successful!');
      res.redirect('/resume');
    } else {
      res.status(401).send('Invalid username or password.');
    }
  });
});

app.listen(5000, () => {
  console.log('Server is running on port 5000');
});
```

- Session Management: Uses express-session to manage user sessions, ensuring secure login and logout functionality.
- Database Connection: Establishes a connection to a MySQL database to store user information.
- Authentication Routes: Implements routes for user registration (/signup) and login (/login), with validation and error handling.
- Static File Serving: Serves static HTML files for the main pages (index.html, login.html, register.html, resume.html).

This structure ensures a robust and scalable backend, capable of handling user authentication and resume management efficiently.

6. Integration and Testing

6.1) Description of the Integration Modules:

6.1.1) User Authentication Module:

This module ensures secure access to the web application by managing user authentication and registration processes.

Sign-up Process:

- User Interaction: Users access the sign-up form provided on the application's homepage. They input their details, including name, email, and password.
- Form Submission: Upon submission, the form data is sent to the server for validation.
- Server-Side Processing:
    - The server-side Node.js application verifies the uniqueness of the email address.
    - User credentials are securely stored in the MySQL database after hashing the password using a strong hashing algorithm like bcrypt.
- User Feedback: A success message is displayed, and users are redirected to the login page.

Login Process:

- User Interaction: Registered users enter their email and password on the login form.
- Client-Side Validation: Ensures all required fields are filled.
- Form Submission: The login credentials are sent to the server for authentication.

- Server-Side Processing:
    - Node.js scripts compare the provided credentials with the stored data in the MySQL database.
    - If the credentials are valid, a session is initiated, and users are redirected to the dashboard.
    - If the credentials are invalid, an error message is displayed.

Authentication Logic:

- Password Security: Passwords are hashed using bcrypt before storing them in the database.
- Session Management: Sessions are managed using express-session to maintain user authentication throughout their browsing session.
- Error Handling: Proper error handling mechanisms are implemented to prevent security vulnerabilities like SQL injection and cross-site scripting (XSS).

Functionality:

- Secure Authentication: Provides a secure and seamless user authentication experience.
- Data Integrity: Ensures data integrity by securely storing user credentials.
- Protection Against Threats: Protects against common security threats like brute force attacks and password guessing.

6.1.2) Resume Creation Module:

This module enables users to create, edit, and save resumes using customizable templates.

Template Selection:

- User Interaction: Users are presented with a gallery of resume templates to choose from.
- Template Loading: Upon selection, the chosen template's layout is loaded into the resume editor interface.

Resume Editing:

- Editor Interface: The editor interface comprises HTML form fields corresponding to various resume sections such as personal information, work experience, education, skills, etc.
- User Input: Users input their details into the respective form fields.

- Dynamic Updates: JavaScript functions dynamically update the preview section in real-time as users fill in the information, providing instant feedback on the layout and formatting.

Data Persistence:

- Saving Resumes: Upon completion, users have the option to save their resumes to their accounts.
- Data Serialization: The form data is serialized and stored in the MySQL database under the user's profile, allowing for easy retrieval and editing in the future.

Functionality:

- Professional Resume Creation: Facilitates the creation of professional resumes through an intuitive and customizable interface.
- Real-Time Feedback: Provides real-time feedback to users, enhancing the editing experience.
- Seamless Data Persistence: Ensures seamless data persistence for users' resumes.

6.1.3) Resume Export Module:

This module enables users to export their resumes in various formats such as PDF, DOCX, etc.

Export Options:

- User Interaction: Users are presented with a list of export formats (e.g., PDF, DOCX).
- Format Selection: Upon selection, the chosen format is passed as a parameter to the export function.

Export Functionality:

- Backend Processing: Node.js scripts generate the resume in the selected format based on the stored data in the MySQL database.
- Conversion Libraries: Libraries or APIs, such as PDFKit for PDF and docx for DOCX, may be utilized to convert the HTML/CSS representation of the resume into the desired format.
- File Generation: The generated file is then made available for download by the user.

Functionality:

- Versatile Downloads: Enables users to download their resumes in formats suitable for different purposes (e.g., online submission, printing).
- User Convenience: Provides flexibility and convenience in sharing resumes with potential employers.

6.2) Testing:

No.	Test case Title	Description	Expected Outcome	The requirement in RS that is being tested	Result
1	Successful User Verification	The login to the system should be tried with the login assigned by the admin and the correct password	Login should be successful and the user should enter the system	RS2	
2	Unsuccessful User Verification due to Wrong Password	Login to the system with a wrong password	Login should fail with an error ‘Invalid Password’	RS3	
3	Unsuccessful User Verification due to invalid login id	Login to the system with a invalid login id	Login should fail with an error ‘Invalid user id’	RS3	
4	Create resume	Creating resume using the given information	Resume	RS7, RS8, RS9	
5	Modify resume	Modifying the resume as per the requirement	Modified resume	RS12	
7	Preview and download resume	Show the preview of resume and download it	Resume in PDF format	RS10, RS11	

7. Performance Analysis:

Evaluating the performance of a resume builder web application involves a detailed examination of several key areas to ensure optimal speed, responsiveness, and scalability. Here’s a comprehensive approach to conducting performance analysis:

1) Page Load Time

- Measurement: Measure the time taken for each page to load, including the homepage, resume editor, export page, etc.
- Tools: Use browser developer tools or performance monitoring tools like Google Page Speed Insights or GTmetrix.
- Optimization:
    - Optimize assets such as images, scripts, and stylesheets to reduce loading times.
    - Utilize caching mechanisms to store frequently accessed data locally on the client-side or server-side.

2) Server Response Time

- Monitoring: Monitor the time taken for the server to respond to user requests.
- Tools: Use server monitoring tools to identify performance bottlenecks, such as slow database queries or resource-intensive PHP scripts.
- Optimization:
    - Optimize database queries by indexing tables, optimizing SQL queries, and caching frequently accessed data.
    - Consider using a content delivery network (CDN) to distribute static assets and reduce server load.

3) Database Performance

- Analysis: Analyse database performance metrics such as query execution time, throughput, and resource utilization.
- Optimization:
    - Identify and optimize slow-running queries by adding indexes, rewriting queries, or denormalizing data where appropriate.
    - Implement database caching techniques to reduce the load on the database server and improve response times for read-heavy operations.
- Monitoring Tools: Regularly monitor database health and performance using tools like MySQL Performance Schema or PostgreSQL's pg_stat_statements.

4) Client-Side Rendering Performance

- Evaluation: Evaluate the performance of client-side rendering frameworks or libraries used in the application.
- Optimization:
    - Identify and optimize render-blocking resources that may delay page rendering, such as large JavaScript files or CSS stylesheets.

    - Implement lazy loading techniques to defer the loading of non-critical resources until they are needed.
    - Minify and compress JavaScript and CSS files to reduce file sizes and improve load times.

5) Network Performance

- Measurement: Measure network latency and throughput using tools like Pingdom or Wireshark.
- Optimization:
    - Optimize asset delivery by minimizing the number of HTTP requests, combining and minifying files, and leveraging browser caching.
    - Utilize HTTP/2 or HTTP/3 protocols to enable multiplexing and reduce latency for simultaneous resource requests.
    - Consider implementing a content delivery network (CDN) to distribute content closer to end-users and reduce network latency.

6) Scalability and Load Testing

- Testing: Conduct load testing to evaluate how the application performs under different levels of concurrent user traffic.
- Tools: Use load testing tools like Apache JMeter or Gatling to simulate realistic user scenarios and measure response times, throughput, and resource utilization.
- Optimization:
    - Identify scalability bottlenecks, such as database connection limits, server capacity, or resource contention, and scale resources accordingly.
    - Implement horizontal scaling strategies, such as load balancers and auto-scaling groups, to handle increased traffic during peak usage periods.

7) Error Monitoring and Logging

- Monitoring: Monitor application errors and exceptions using logging and monitoring tools.
- Analysis: Analyse error logs to identify and address performance-related issues, such as slow API responses, database errors, or server crashes.
- Proactive Measures: Implement proactive alerting and notification mechanisms to detect performance anomalies in real-time and take corrective action.

8. Future Scope

1. **Market Analysis**: Conduct a comprehensive analysis of the current trends in the resume building industry. Identify areas where there is potential for growth or innovation, such as:

    - Emerging technologies like AI-driven resume optimization tools.
    - Changing user preferences towards personalized and visually appealing resume formats.
    - Gaps in existing services, such as the need for more intuitive and user-friendly interfaces.

2. **User Feedback Integration**: Incorporate feedback from users of the resume builder website to identify common pain points and feature requests. Use this feedback to propose enhancements or new features, such as:

    - Enhanced customization options for resume templates.
    - Integration of real-time feedback mechanisms to assist users in optimizing their resumes.
    - Collaboration tools for users to seek feedback from peers or professionals.

3. **Competitive Analysis**: Study competing resume builder websites to understand their strengths and weaknesses. Identify opportunities where your website can differentiate itself or offer unique value propositions, such as:

    - Advanced AI-driven resume analysis tools for personalized recommendations.
    - Seamless integration with job search platforms for streamlined application processes.
    - Comprehensive career planning resources beyond resume building, such as interview preparation guides and career coaching services.

4. **Technological Advancements**: Explore advancements in technology, such as artificial intelligence or natural language processing, to improve the resume building process. Consider integrating these technologies to enhance user experience and streamline resume creation, including:

    - AI-powered resume writing assistants that analyze user input and offer personalized suggestions.
    - Natural language processing tools to extract relevant information from user-provided documents or online profiles.
    - Blockchain technology for secure verification of credentials and achievements listed in resumes.

5. **Expansion into New Markets**: Evaluate opportunities for expanding the reach of the resume builder website into new geographic markets or target demographics. Consider factors such as cultural and linguistic considerations, and tailor the platform to effectively enter these markets. This may involve:

   - Localization of the platform to support multiple languages and cultural preferences.
   - Customization of resume templates to cater to specific industry standards or regional expectations.
   - Partnerships with local job search platforms or educational institutions to gain traction in new markets.

6. **Partnerships and Collaborations**: Explore partnerships or collaborations with other companies or organizations to enhance the resume building experience. This could include integration with job search platforms, educational institutions, or professional associations to provide additional value-added services, such as:

   - Seamless integration with LinkedIn or other professional networking platforms for importing profile data.
   - Collaboration with universities or career centers to offer resume-building workshops or seminars.
   - Integration with HR software platforms to streamline the recruitment process for employers.

7. **Long-term Vision**: Clearly articulate the long-term vision for the resume builder website, aligning it with broader industry trends and developments. Outline specific milestones and goals to achieve over the coming years, including:

   - Continual improvement of the platform based on user feedback and technological advancements.
   - Expansion into adjacent markets, such as career coaching or professional development services.
   - Establishment as a trusted authority in the resume building industry, known for innovation, reliability, and user-centric design.

9. Application

1. **Path to Job Opportunities**:

   The resume builder website serves as a gateway to accessing job opportunities across various industries. By helping users categorize their personality, skills, and experiences, the platform enables them to create compelling resumes tailored to specific job roles and industries. This personalized approach increases their chances of standing out to potential employers and securing relevant job opportunities.

2. **Skill Assessment Tools Integration**:

   Integrate skill assessment quizzes or tests within the platform to assist users in identifying their strengths and areas for improvement. These assessments can cover a wide range of skills relevant to different industries and job roles. Based on the results, the platform can provide personalized recommendations for skill development, including online courses, workshops, or resources to enhance users' qualifications and marketability.

3. **Resume Analytics Dashboard**:

   Offer users access to a comprehensive resume analytics dashboard, providing valuable insights into the performance of their resumes. Key metrics such as views, downloads, and application success rates are tracked and visualized, allowing users to gauge the effectiveness of their job application strategies. By analyzing these metrics, users can make data-driven adjustments to their resumes, optimizing them for better job search outcomes and increasing their chances of securing interviews and job offers.

10. Installation Guide and User Manual

Requirements:

- Node.js and npm installed on your system
- MySQL database server

Steps:

1. **Clone the Repository:** Clone the repository containing the Resume Builder Application to your local machine:

   ```
   git clone <repository-url>
   ```

2. **Navigate to the Project Directory:**

   ```
   cd resume-builder
   ```

3. **Install Dependencies:** Install the required Node.js dependencies using npm:

   ```
   npm install
   ```

4. **Database Setup:**
   - Ensure that your MySQL database server is running.
   - Create a new MySQL database for the application (e.g., **test**).
   - Import the database schema provided in the **database/schema.sql** file to set up the required tables.

5. **Start the Server:** Start the Node.js server:

   ```
   npm start
   ```

6. **Access the Application:**

   Open a web browser and navigate to **http://localhost:5000** to access the Resume Builder Application.

User Manual:

1. Sign Up:

- Click on the "Sign Up" button on the homepage.
- Fill in the required details (username, email, password) and submit the form.
- Upon successful registration, you will be redirected to the login page.

2. Login:

- Enter your email and password on the login page and click "Login".
- If the credentials are correct, you will be redirected to the resume editor page.

3. Create/Edit Resume:

- On the resume editor page, fill in the various sections of your resume, including personal information, work experience, education, skills, etc.
- Click on the "Save" button to save your progress. You can continue editing your resume later.

4. Preview and Download Resume:

- Once you have filled in all the necessary details, click on the "Preview" button to preview your resume.
- You can download your resume in various formats (PDF, DOCX) by selecting the desired format and clicking the "Download" button.

11. References

Academic Journals and Articles:

Identify scholarly articles related to resume building, career development, and user experience design.

Case Studies:

Seek out case studies or success stories from other resume builder platforms or similar digital products.

Industry Reports:

Look for industry reports and market analyses that discuss trends, challenges, and opportunities in the resume builder software market.

Online preference Links for Building the Website:

1. https://chatgpt.com
2. https://www.w3schools.com/tutorials/
3. https://www.geeksforgeeks.org/web-development/

12. Conclusion

This application serves the purpose to help students at universities to design and create their resumes with ease using forms and profiles that we all are so used to in terms of social media. In conclusion, our resume builder website offers a user-friendly platform for crafting professional resumes. Through intuitive design and valuable features, seamless integration with professional networks, we've empowered users to create standout resumes tailored to their career goals.