# PERSONALIZED PORTFOLIO RESUME G GENERATOR

A Real-Time / Field-Based Research Project (22AM284) report submitted to the J Jawaharlal Nehru Technological University, Hyderabad

Submitted by

**V.PAVANI - 22B81A6628**

**S.JYOTHSNA - 22B81A6619**

**CH.SRAVANI - 22B81A6646**

Under the guidance of

**Dr. Srinivas Mekala**

**Associate Professor**

****

**DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

# CVR COLLEGE OF ENGINEERING

**(*An Autonomous Institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli(V),Ibrahimpatnam(M),

Rangareddy (D), Telangana- 501 510

**JUNE 2024**

# CVR COLLEGE OF ENGINEERING

**(*An Autonomous institution , NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli(V),Ibrahimpatnam(M), Rangareddy (D), Telangana- 501 510

**DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

****

**CERTIFICATE**

This is to certify that the Real time/ Field-Based research project(22AM284) report entitled **“Personalized Portfolio Resume Generator”** is a record of work carried out by **Jyothsna,Pavani,Sravani** submitted to Department of **CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING) ,** CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University, Hyderabad during the year 2023-2024.

**Project Guide Project Coordinator Head of the Department**

**Dr.Srinivas Mekala Dr.Lakshmi H N**

Associate Professor Professor & HOD

Artificial Intelligence & Machine Learning CSE(AI&ML,DS,CS)

**DECLARATION**

We hereby declare that the Real time/ Field-Based research project (22AM284) report entitled **“Personalized Portfolio Resume Generator”** is an original work done and submitted to **CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University, Hyderabad and it is a record of bonafide project work carried out by us under the guidance of **Dr.Srinivas Mekala, Associate Professor, Artificial Intelligence & Machine Learning.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this Institute or any other Institute or University.

Signature of the Student

**S.Jyothsna**

**22B81A6619**

Signature of the Student

**V.Pavani**

**22B81A6628**

Signature of the Student

**CH.Sravani**

**22B81A6646**

**Date: 01-06-2024**

**Place: Hyderabad**

**ACKNOWLEDGEMENT**

We are thankful and fortunate enough to get constant encouragement, support, and guidance from all **Teaching staff of CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** Departmentwho helped us in successfully completing this project work.

We thank **DR.B.Prasanthi**, Project Coordinator and **Mr.P.Sudheer, Mrs.K.Navaneetha, Mr.G.Venkatesh,** Project Review Committee members for their valuable guidance and support which helped us to complete the project work successfully.

We respect and thank our internal guide, **Dr.Srinivas Mekala, Associate Professor, Artificial Intelligence & Machine Learninng**, for giving us all the support and guidance, which made us complete the project duly.

We would like to express heartfelt thanks to **Dr. H. N. Lakshmi,** Professor & Head of the Department, for providing us an opportunity to do this project and extending support and guidance.

We thank our Vice-Principal **Prof. L. C. Siva Reddy** for providing excellent computing facilities and a disciplined atmosphere for doing our work.

We wish a deep sense of gratitude and heartfelt thanks to **Dr. Rama Mohan Reddy,** Principal and the **Management** for providing excellent lab facilities and tools. Finally, we thank all those guidance helpful to us in this regard.

**ABSTRACT**

This document presents the design and development of a personalized portfolio resume generator aimed at simplifying the process of creating customized resumes for individuals. The generator utilizes user input to dynamically generate tailored resumes, highlighting skills, experiences, and achievements. Through an intuitive interface, users can input their information, select templates, and generate professional resumes efficiently. The system incorporates machine learning algorithms to enhance resume customization based on user preferences. The generator aims to streamline the resume creation process, empowering individuals to showcase their unique strengths effectively in the job market.

This project automates the resume creation process using intuitive web technologies, significantly reducing the time and effort required by users. It enhances the personalization and relevancy of resumes, thereby increasing the likelihood of securing job interviews. This tool represents a significant advancement in personal branding and job application processes, offering a streamlined and effective solution for modern job seekers. It is designed to help users create highly personalized resumes and portfolios that align with their career goals and industry requirements.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No.** | | **Contents** | **Page No.** |
|  |  | Certificate | i |
|  |  | Declaration | ii |
|  |  | Acknowledgement | iii |
|  |  | Abstract | iv |
| 1 |  | **Introduction** | 1-4 |
|  | 1.1 | Problem Statement | 2 |
|  | 1.2 | Project Objectives | 3 |
|  | 1.3 | Software &Hardware specifications | 4 |
|  | 1.3.1 | Software requirements | 4 |
|  | 1.3.2 | Hardware requirements | 4 |
| 2 |  | **Design Methodology** | 5-8 |
|  | 2.1 | System Architecture | 5 |
|  | 2.2 | Data flow Diagram or Flowchart | 6 |
|  | 2.3 | Technology Description | 7-8 |
| 3 |  | **Implementation & Testing** | 9-13 |
|  | 3.1 | Code snippets | 9-12 |
|  | 3.2 | Test cases | 13 |
| 4 |  | **Conclusion** | 14 |
|  |  | **Bibliography** | 15 |
|  |  | **Appendix: ( Source code )** | 16-25 |

**1.INTRODUCTION**

In the contemporary job market, standing out among a sea of applicants is crucial. One effective way to achieve this is by presenting a personalized and visually appealing portfolio or resume. The rise of digital tools has made it easier than ever to create customized resumes that reflect not only your skills and experiences but also your personal brand. This document introduces the concept of a Personalized Portfolio Resume Generator, detailing its significance, functionality, and the steps to create one using modern programming languages and frameworks.

A Personalized Portfolio Resume Generator is a tool designed to assist individuals in creating customized resumes that reflect their personal style, professional experiences, and career aspirations. By integrating features such as customizable templates, multimedia support, and secure data handling, this tool aims to enhance the overall job application experience.

The need for personalization in resumes cannot be overstated. Personalized resumes allow job seekers to tailor their applications to specific job descriptions, emphasizing the most relevant skills and experiences. Additionally, personalized resumes provide an opportunity for candidates to showcase their creativity and design skills, which is particularly advantageous in fields such as marketing, design, and multimedia.

The core features of a Personalized Portfolio Resume Generator include a user-friendly interface, extensive customization options, integration capabilities, and secure data management. Customization options are abundant, allowing users to choose from a variety of professional templates, color schemes, fonts, and layouts to create a resume that truly represents their personal brand.

* 1. **PROBLEM STATEMENT**

The Personalized Portfolio Resume Generator project addresses this challenge by providing a solution that enables users to create customized and visually appealing resumes and portfolios. This tool will be designed with a user-friendly interface that guides individuals through the process of entering their personal and professional information, ensuring that the data is both relevant and well-organized.

The primary problem this project seeks to distinguish jobseekers from the multitude of applicants vying for the same positions. Traditional resumes, often generic and lacking in personalization, fail to adequately highlight an individual's unique skills, experiences, and personal brand. This lack of distinction and personalization in resumes presents a significant challenge for job seekers aiming to make a memorable and impactful impression.

Security and data protection are paramount in this project. The Personalized Portfolio Resume Generator will implement robust encryption and secure storage practices to protect user data, ensuring compliance with privacy regulations and giving users control over their personal information. Additionally, the tool will offer various export options, including PDF and generate shareable links for the online portfolio, facilitating easy distribution and accessibility.

The project aims to empower job seekers by providing them with a powerful tool to create unique, professional, and impactful resumes and portfolios. By addressing the need for personalization and leveraging modern technology, this project will enhance the job application process, helping individuals to better present their qualifications and stand out in the competitive job market.

* 1. **PROJECT OBJECTIVES**

**Develop a User-Friendly Interface:**

* Create an intuitive and easy-to-navigate interface that guides users through the resume creation process.
* Provide clear instructions and helpful tips to assist users in entering their personal information, work history, education, skills, and other relevant details.

**Offer Extensive Customization Options:**

* Provide a variety of professional templates that users can choose from to start their resume.
* Enable users to add, remove sections to tailor the resume to specific job applications and industries.

**Ensure Secure Data Handling:**

* Implement robust encryption and secure storage practices to protect user data and ensure privacy.
* Comply with data protection regulations and provide users with control over their personal information.

**Facilitate Easy Export and Sharing:**

* Offer multiple export options, including PDF and to accommodate different application requirements and preferences.
  1. **SOFTWARE AND HARDWARE SPECIFICATIONS**
     1. **Software Requirements**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification

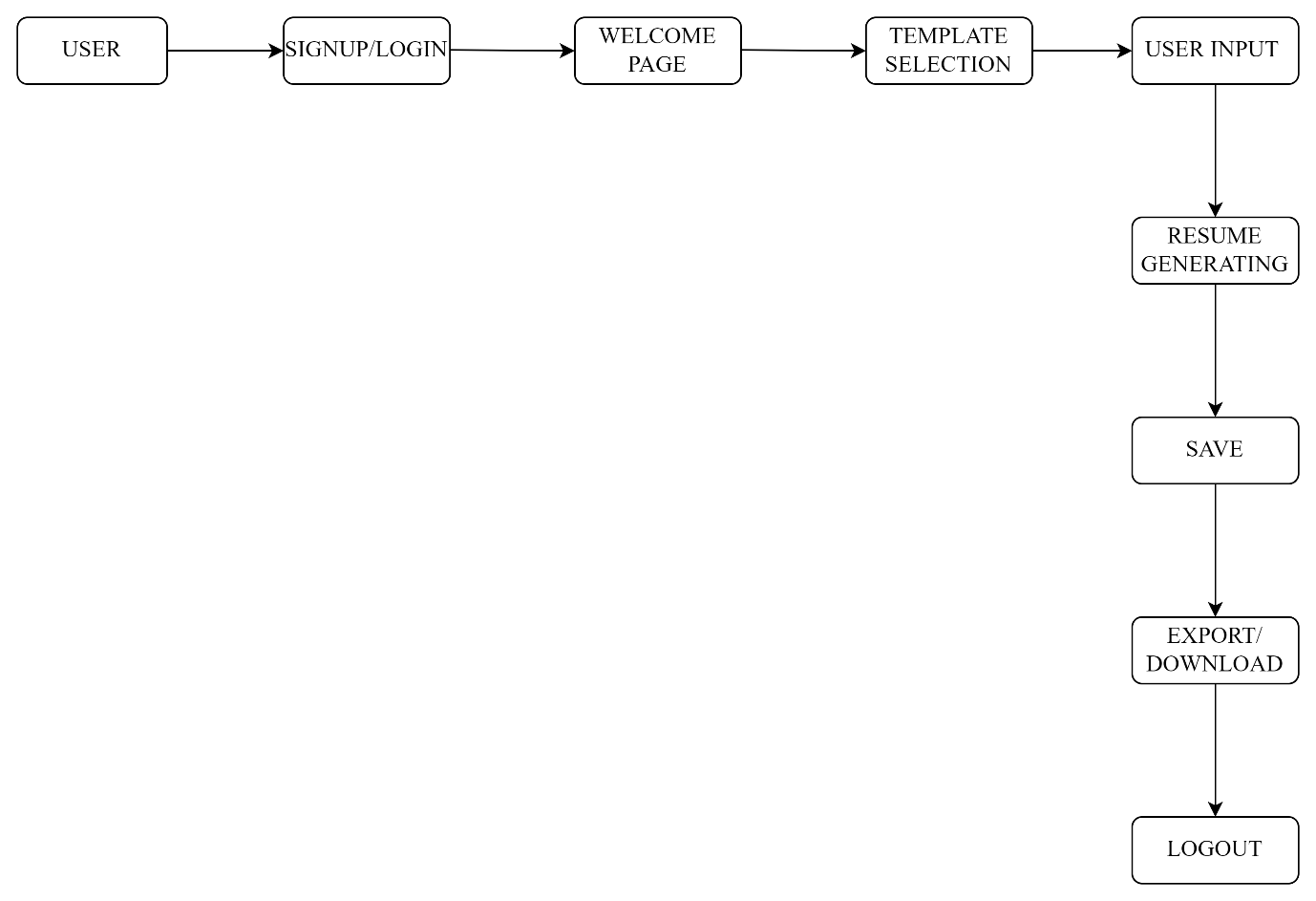
* **Languages:** HTML, CSS, JavaScript and Node.js
* **Database:** MongoDB
* **Tools/IDE:** Visual Studio Code, or any preferred code editor
  + 1. **Hardware Requirements:**

The hardware requirements may serve a s the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shoals what the system does and not how it should be implemented.

* **System Specifications:** At least 4GB RAM and 128GB Hard Disk
* **Operating System:** Windows, macOS or Linux
* **Processor:** Intel Core i5 or equivalent

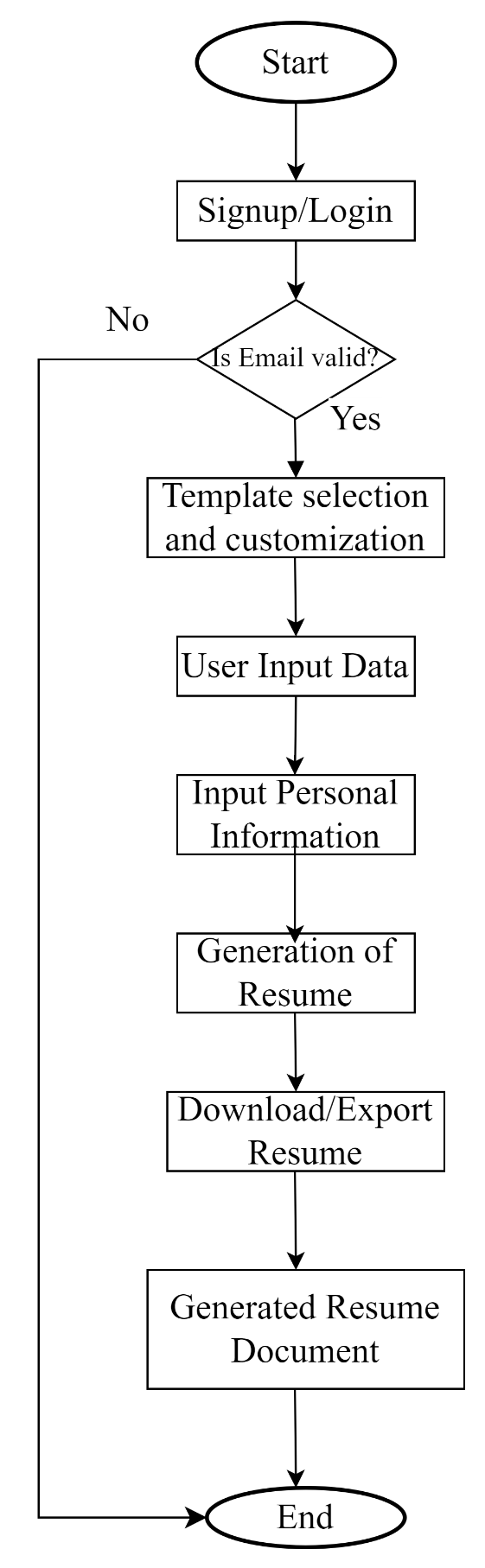
**2.DESIGN METHODOLOGY**

**2.1 SYSTEM ARCHITECTURE:**



A personalized portfolio resume generator enables users to create customized, professional portfolios through an intuitive interface. Upon signing up or logging in, users can select from various templates tailored to different professions. They can add and customize sections such as Profile, Experience, Projects, and Skills by filling out forms and uploading multimedia files. The interface allows for real-time previews and design customization, including color schemes, fonts, and layouts. All changes are saved automatically or manually, allowing users to edit their portfolios anytime. Once finalized, users can export their portfolios as PDF and user can download it.

**2.2 DATA FLOW DIAGRAM**



* 1. **TECHNOLOGY DESCRIPTION**

1. **Html & CSS :**

**Description :**

HTML is the foundational markup language for creating web pages. An HTML document typically includes a head section for metadata and a body section for the visible content. HTML tags can also have attributes that provide additional information about the elements, like **href** in links.

CSS is a stylesheet language used to describe the presentation of HTML documents. It controls the layout, colors, fonts, and overall visual style of web pages.

**Role in the Project :**

HTML and CSS play crucial roles in the creation and functionality of a resume generator. HTML is used to structure the content of the resume. It defines the layout and sections, such as the header, summary, work experience, education, skills, and contact information. Additionally, HTML forms are integral for collecting user data, allowing individuals to input their personal and professional details, which are dynamically incorporated into the resume template.

CSS is responsible for the styling and visual presentation of the resume. CSS controls the appearance of HTML elements, managing fonts, colors, margins, padding, and overall layout.

1. **Javascript :**

**Description :**

JavaScript is a high-level, versatile programming language widely used for creating interactive and dynamic content on websites. It enables developers to manipulate the Document Object Model (DOM), handle events, validate forms, and create animations directly within web browsers.

**Role in the Project :**

JavaScript plays a pivotal role in resume generator by providing interactivity, dynamic content updates, and seamless user experiences. It enables real-time form validation, ensuring that user inputs such as name, contact information, and job experiences are correctly formatted before submission. It also facilitates asynchronous data fetching, enabling integration with external APIs to automatically pull in additional information or validate credentials.

1. **Node.js :**

**Description :**

Node.js is a powerful, open-source runtime environment that allows developers to execute JavaScript code on the server side. Built on the V8 JavaScript engine from Google Chrome, Node.js is designed for building scalable and efficient network applications. It uses an event-driven, non-blocking I/O model, which makes it ideal for real-time applications that require high performance, such as chat applications, streaming services, and APIs. Node.js supports the development of full-stack applications using JavaScript for both client-side and server-side code, enhancing development consistency and efficiency

**Role in the Project :**

Node.js plays a crucial role in a personalized portfolio resume generator by providing a robust and scalable backend environment to handle server-side operations. It enables real-time data processing and storage, allowing users to save, retrieve, and update their resume information seamlessly. With Node.js, the generator can efficiently manage user authentication, ensuring secure access to personal data. The extensive npm ecosystem further enhances the development process by providing reusable modules and tools to streamline the creation and deployment of the portfolio resume generator.

**4. MongoDB :**

**Description :**

MongoDB is a popular, open-source NoSQL database designed for storing and managing large volumes of unstructured data. It uses a flexible, document-oriented data model, allowing data to be stored in JSON-like BSON (Binary JSON) format, which makes it highly scalable and capable of handling diverse data types. MongoDB's schema-less design enables developers to iterate quickly and accommodate evolving data requirements without the need for complex migrations.

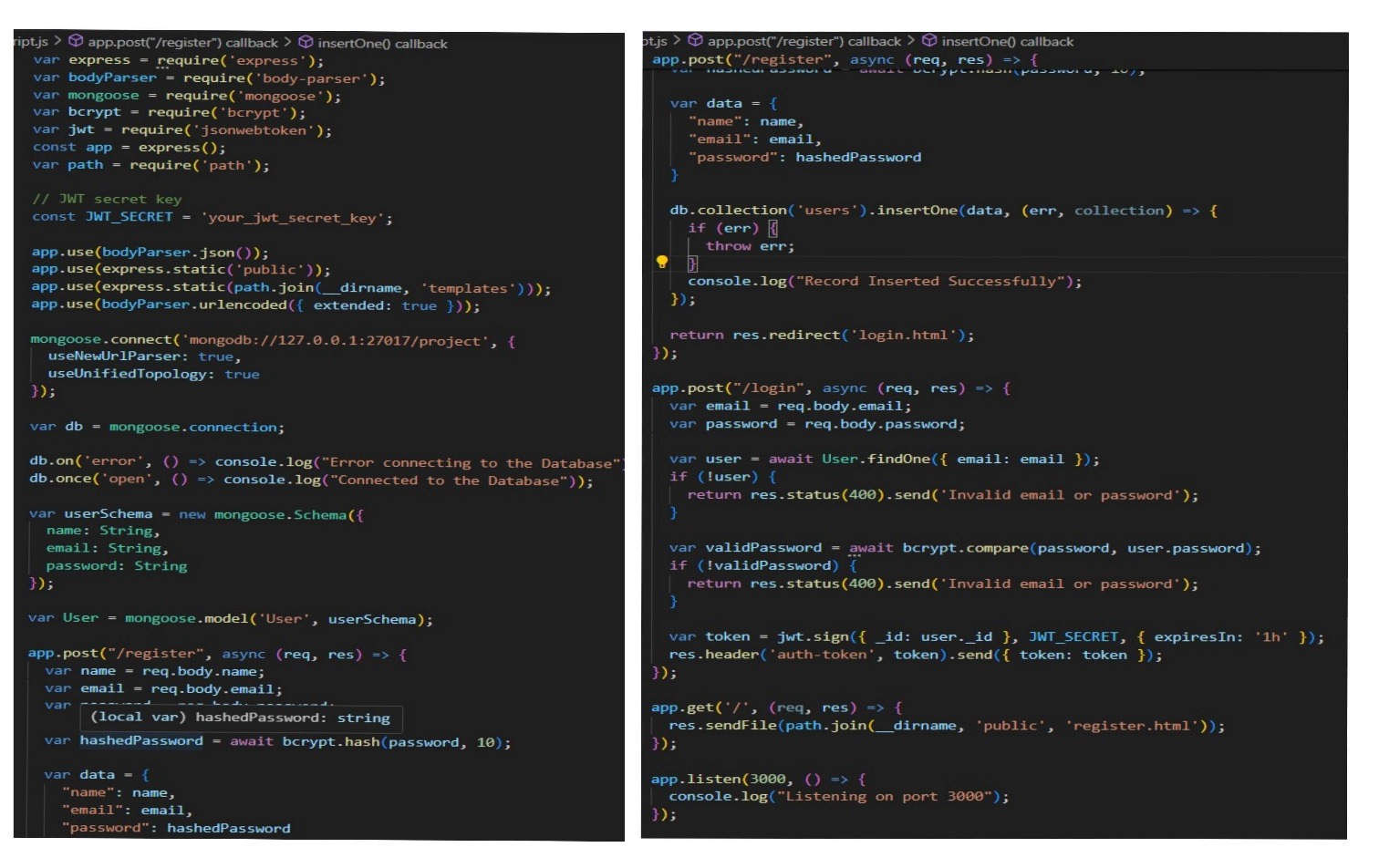
**Role in the Project :**

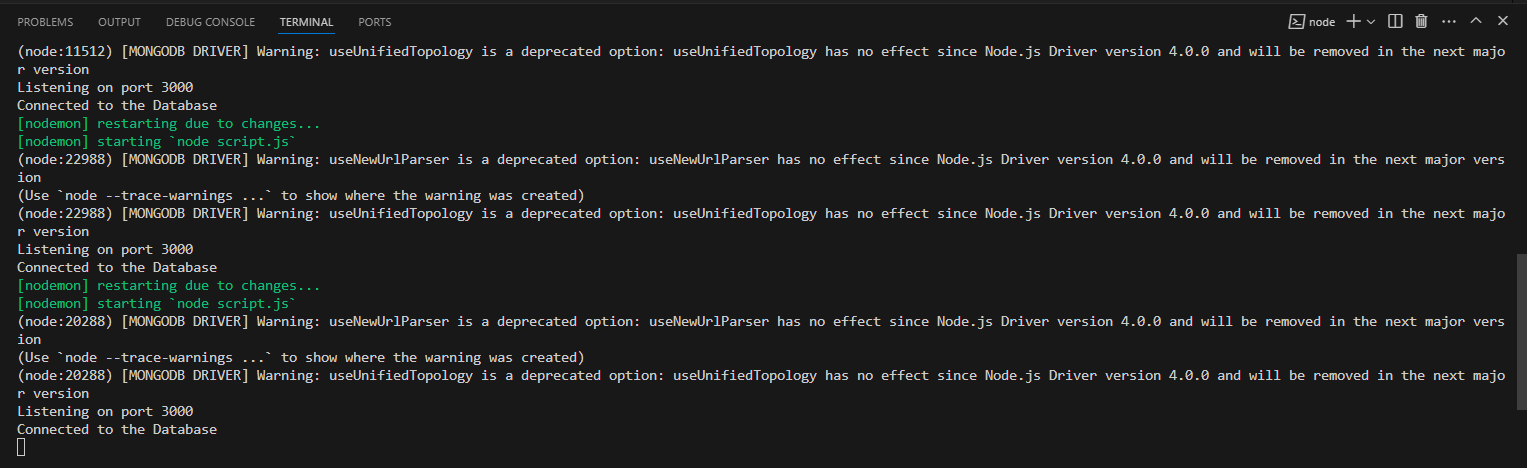
MongoDB plays a vital role in a personalized portfolio resume generator by serving as the backend database for storing and managing user data efficiently. It provides a flexible and scalable solution for storing diverse information such as user profiles, work experiences, education details, skills, and contact information. Its scalability features, including sharding and replication, ensure that the resume generator can accommodate growth and handle increasing volumes of user data effectively.

**3.IMPLEMENTATION AND TESTING**

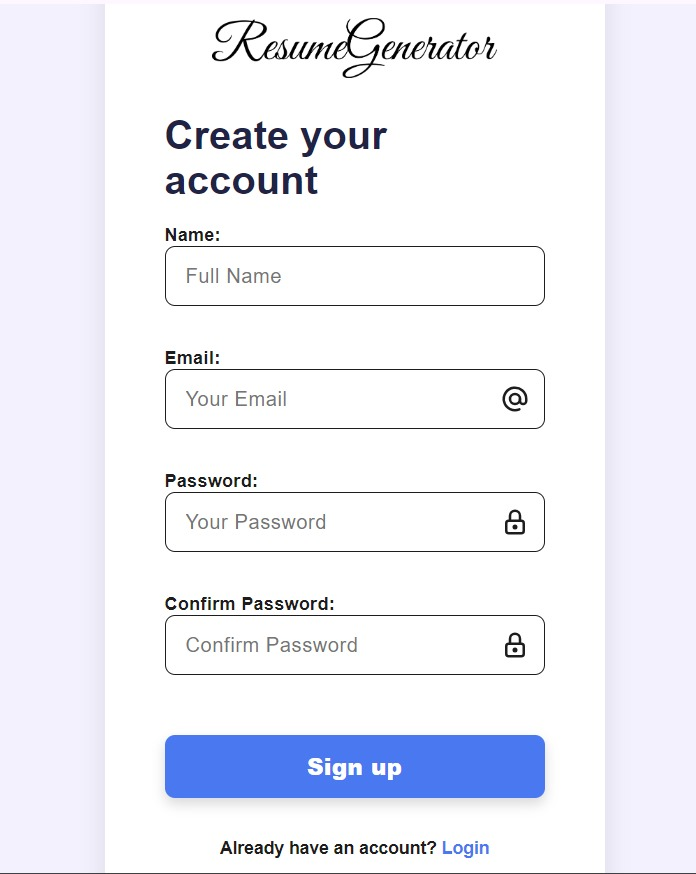
**3.1 CODE SNIPPETS**

**DATABASE CONNECTION (MONGODB) :**

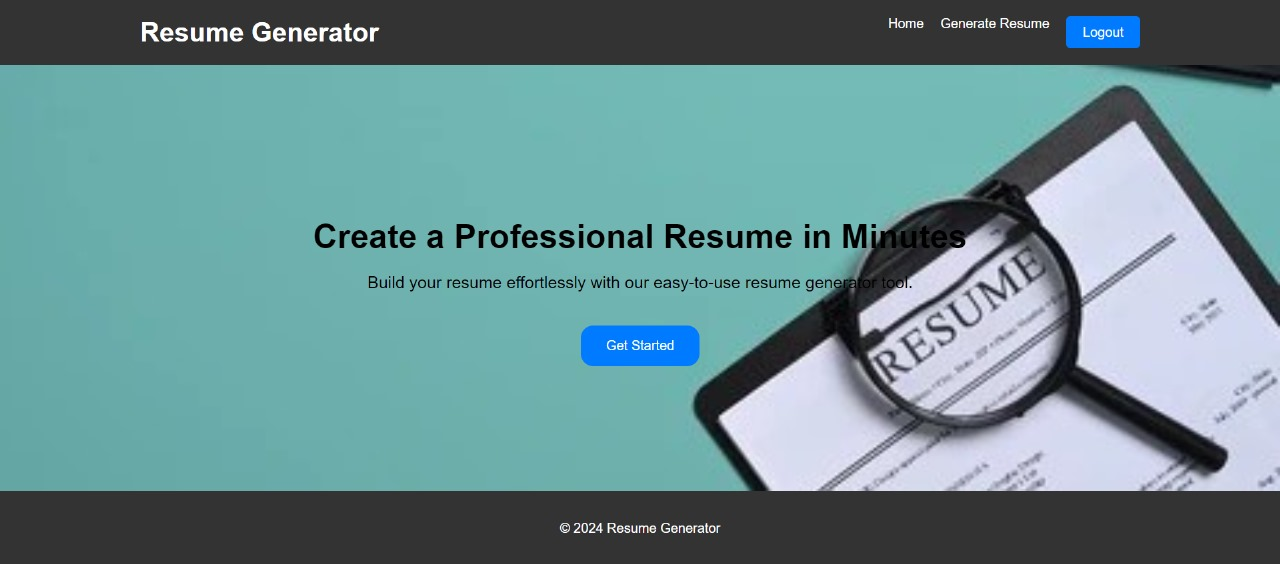
****



**REGISTER PAGE : LOGIN PAGE**

****

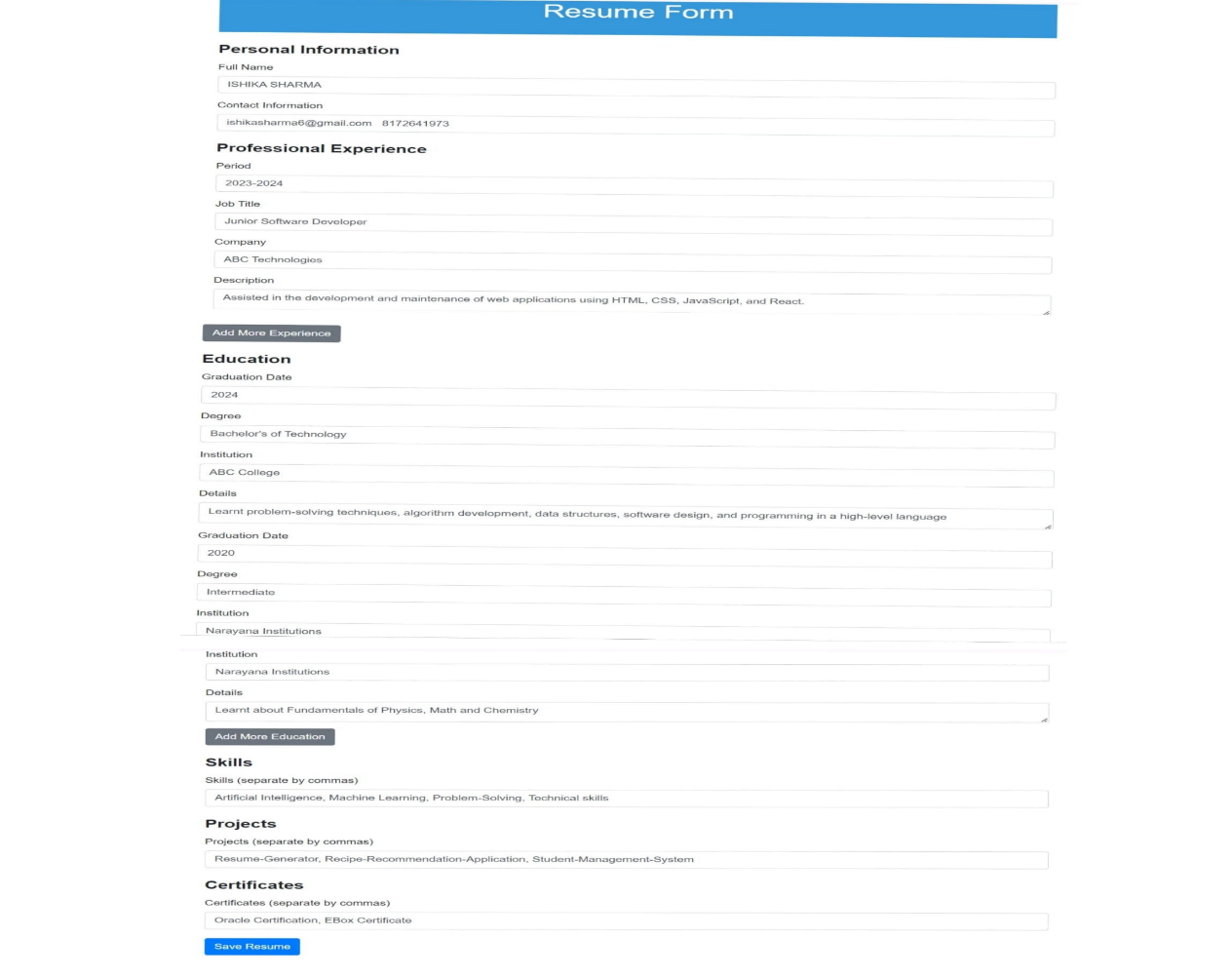
**HOME PAGE :**

****

**TEMPLATES PAGE :**

****

**FORM PAGE :**

****

**RESUME PAGE :**

****

* 1. **TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Id** | **Test Description** | **Expected Result** | **Actual Result** | **Status** | **Remarks** |
| 1. | Database Connection | Connected to the Database | Connected to the Database | Pass | - |
| 2. | User Registration | Record Inserted Successfully | Record Inserted Successfully | Pass | - |
| 3. | User Login | Redirect to Home page | Redirected to Home page | Pass | - |
| 4. | Template Selection | Redirect to Form page | Redirected to Form page | Pass | - |
| 5. | Resume Generation | Resume will generate | Resume Generated | Pass | - |

**4. CONCLUSION**

Ultimately, our personal portfolio resume builder is your gateway to crafting a great description of your career journey. With its user-friendly interface and customizable templates, this tool allows you to present your skills and knowledge in the most accurate and relevant way. By using the power of personal design and content, you can effectively communicate your strengths, aspirations and achievements to employers to stand out in the competitive market.

Designed to meet your career development needs, this new tool offers a simple and effective way to create professional resumes that attract the attention of recruiters and hiring managers. Whether you're a professional , our resume builder will make you look good and provide the perfect presentation of your required skills.

At its core, our personal resume builder is more than just a tool; it’s the power to get the job done. It empowers you to tell your story in an organized and compelling way, making you the best competitor in your field. Use the power of your personal story to unlock new opportunities with a resume that reveals your true potential. Your success story starts here; start building your resume today!

**BIBLIOGRAPHY**

**Frameworks and Libraries**:

Bootstrap. (2024). Retrieved from Bootstrap.

Express.js. (2024). Express - Node.js web application framework. Retrieved from Express.

Mongoose. (2024). Mongoose - Elegant MongoDB object modeling for Node.js. Retrieved from Mongoose.

bcrypt. (2024). Retrieved from npm bcrypt.

jsonwebtoken. (2024). Retrieved from npm jsonwebtoken.

**Tools and Software**:

Node.js. (2024). Node.js JavaScript runtime. Retrieved from Node.js.

MongoDB. (2024). MongoDB NoSQL database. Retrieved from MongoDB.

Visual Studio Code. (2024). Code Editing. Redefined. Retrieved from VS Code.

**Specific Code Snippets**:

Bootstrap Form Examples. Retrieved from Bootstrap Examples.

JavaScript Event Handling. Retrieved from Web Docs.

**Templates:**

Microsoft ATS templates

**APPENDIX (SOURCE CODE)**

**DATABASE CONNECTION / REGISTRATION :**

var express = require('express');

var bodyParser = require('body-parser');

var mongoose = require('mongoose');

var bcrypt = require('bcrypt');

var jwt = require('jsonwebtoken');

const app = express();

var path = require('path');

const JWT\_SECRET = 'resume\_generate';

app.use(bodyParser.json());

app.use(express.static('public'));

app.use(express.static(path.join(\_\_dirname, 'templates')));

app.use(bodyParser.urlencoded({ extended: true }));

mongoose.connect('mongodb://127.0.0.1:27017/project', {

useNewUrlParser: true, useUnifiedTopology: true

});

var db = mongoose.connection;

db.on('error', () => console.log("Error connecting to the Database"));

db.once('open', () => console.log("Connected to the Database"));

var userSchema = new mongoose.Schema({name: String, email: String, password: String});

var User = mongoose.model('User', userSchema);

app.post("/register", async (req, res) => {

var name = req.body.name;

var email = req.body.email;

var password = req.body.password;

var hashedPassword = await bcrypt.hash(password, 10);

var data = {"name": name, "email": email, "password": hashedPassword }

db.collection('users').insertOne(data, (err, collection) => {

if (err) { throw err; }

console.log("Record Inserted Successfully");

});

return res.redirect('login.html');

});

app.post("/login", async (req, res) => {

var email = req.body.email;

var password = req.body.password;

var user = await User.findOne({ email: email });

if (!user) { return res.status(400).send('Invalid email or password'); }

var validPassword = await bcrypt.compare(password, user.password);

if (!validPassword) { return res.status(400).send('Invalid email or password'); }

var token = jwt.sign({ \_id: user.\_id }, JWT\_SECRET, { expiresIn: '1h' });

res.header('auth-token', token).send({ token: token });

});

app.get('/', (req, res) => {res.sendFile(path.join(\_\_dirname, 'public', 'register.html'));});

app.listen(3000, () => {console.log("Listening on port 3000");});

**LOGIN SCRIPT :**

function login() {

const email = document.getElementById('email').value;

const password = document.getElementById('pass').value;

fetch('/login', { method: 'POST',

headers: { 'Content-Type': 'application/json'},

body: JSON.stringify({ email, password })

})

.then(response => response.json())

.then(data => {if (data.token) {

localStorage.setItem('auth-token', data.token);

window.location.href = 'home.html';

} else {alert('Invalid email or password. Please try again.'); }

})

.catch(error => console.error('Error:', error));

}

**HOME SCRIPT :**

function logout() {

localStorage.removeItem('auth-token');

window.location.href = 'login.html';

}

document.addEventListener('DOMContentLoaded', () => {

const token = localStorage.getItem('auth-token');

if (!token) { window.location.href = 'login.html'; }

});

**FORM SCRIPT :**

document.addEventListener('DOMContentLoaded', function () {

let experienceCount = 1;

let educationCount = 1;

document.getElementById('addExperience').addEventListener('click', function () {

experienceCount++;

const experienceSection = document.getElementById('experienceSection');

const newExperience = document.createElement('div');

newExperience.className = 'experience-item';

newExperience.innerHTML = `<div class="form-group">

<label for="experiencePeriod${experienceCount}">Period</label>

<input type="text" class="form-control" id="experiencePeriod${experienceCount}" required>

</div> <div class="form-group">

<label for="experienceJobTitle${experienceCount}">Job Title</label>

<input type="text" class="form-control" id="experienceJobTitle${experienceCount}" >

</div> <div class="form-group">

<label for="experienceCompany${experienceCount}">Company</label>

<input type="text" class="form-control" id= "experienceCompany${experienceCount}" >

</div> <div class="form-group">

<label for="experienceDescription${experienceCount}">Description</label>

<textarea class="form-control" id="experienceDescription${experienceCount}" rows="3" required></textarea></div>`;

experienceSection.appendChild(newExperience);

});

document.getElementById('addEducation').addEventListener('click', function () {

educationCount++;

const educationSection = document.getElementById('educationSection');

const newEducation = document.createElement('div');

newEducation.className = 'education-item';

newEducation.innerHTML = `<div class="form-group">

<label for="educationGraduation${educationCount}">Graduation Date</label>

<input type="text" class="form-control" id="educationGraduation${educationCount}" >

</div><div class="form-group">

<label for="educationDegree${educationCount}">Degree</label>

<input type="text" class="form-control" id="educationDegree${educationCount}" >

</div> <div class="form-group">

<label for="educationInstitution${educationCount}">Institution</label>

<input type="text" class="form-control" id="educationInstitution${educationCount}" >

</div> <div class="form-group">

<label for="educationDetails${educationCount}">Details</label>

<textarea class="form-control" id="educationDetails${educationCount}" rows="3" required></textarea> </div>`;

educationSection.appendChild(newEducation);

});

document.getElementById('resumeForm').addEventListener('submit', function (e) {

e.preventDefault();

const name = document.getElementById('name').value;

const contact = document.getElementById('contact').value;

const experiences = [];

document.querySelectorAll('.experience-item').forEach((item, index) => {

experiences.push({

period: document.getElementById(experiencePeriod${index + 1}).value,

jobTitle: document.getElementById(experienceJobTitle${index + 1}).value,

company: document.getElementById(experienceCompany${index + 1}).value,

description: document.getElementById(experienceDescription${index + 1}).value

});

});

const educations = [];

document.querySelectorAll('.education-item').forEach((item, index) => {

educations.push({

graduation: document.getElementById(educationGraduation${index + 1}).value,

degree: document.getElementById(educationDegree${index + 1}).value,

institution: document.getElementById(educationInstitution${index + 1}).value,

details: document.getElementById(educationDetails${index + 1}).value

});

});

const skills = document.getElementById('skills').value.split(',').map(skill => skill.trim());

const projects = document.getElementById('projects').value.split(',').map(project => project.trim());

const certificates = document.getElementById('certificates').value.split(',').map(certificate => certificate.trim());

const resumeData = {

name: name,contact: contact,experiences: experiences,

educations: educations,skills: skills,

projects: projects,certificates: certificates

};

localStorage.setItem('resumeData', JSON.stringify(resumeData));

alert('Resume saved successfully!');

window.location.href = 'resume.html';

});

});

**RESUME SCRIPT :**

document.addEventListener('DOMContentLoaded', function () { const resumeData = JSON.parse(localStorage.getItem('resumeData')); if (resumeData) { document.getElementById('resumeName').textContent = resumeData.name || 'Your Name'; document.getElementById('resumeContact').textContent = resumeData.contact || 'Your Contact Information'; populateExperience(resumeData.experiences); populateEducation(resumeData.educations); populateSkills(resumeData.skills); populateProjects(resumeData.projects); populateCertificates(resumeData.certificates); } document.getElementById('downloadPDF').addEventListener('click', function () { hideButtons(); const { jsPDF } = window.jspdf; const a4Width = 210; const a4Height = 297; html2canvas(document.body).then(canvas => { const imgData = canvas.toDataURL('image/png'); const pdf = new jsPDF({ orientation: 'portrait', unit: 'mm', format: 'a4' }); const imgProps = pdf.getImageProperties(imgData); const pdfWidth = a4Width; const pdfHeight = (imgProps.height \* pdfWidth) / imgProps.width; pdf.addImage(imgData, 'PNG', 0, 0, pdfWidth, pdfHeight); pdf.save('resume.pdf'); showButtons(); }).catch(() => showButtons()); }); function hideButtons() { document.getElementById('downloadPDF').classList.add('hidden'); document.getElementById('downloadWord').classList.add('hidden'); document.getElementById('printResume').classList.add('hidden'); } function showButtons() { document.getElementById('downloadPDF').classList.remove('hidden'); document.getElementById('downloadWord').classList.remove('hidden'); document.getElementById('printResume').classList.remove('hidden'); } document.getElementById('printResume').addEventListener('click', function () { hideButtons(); window.print(); showButtons(); });});function populateExperience(experiences) { const container = document.getElementById('experienceContent'); experiences.forEach(exp => { const div = document.createElement('div'); div.className = 'section-content'; div.innerHTML = `<strong>${exp.period || 'Period'}</strong><br> <strong>${exp.jobTitle || 'Job Title'} | ${exp.company || Company'}</strong><br> ${exp.description || 'Description of your responsibilities and achievements.'}`; container.appendChild(div); });}function populateEducation(educations) { const container = document.getElementById('educationContent'); educations.forEach(edu => { const div = document.createElement('div'); div.className = 'section-content'; div.innerHTML = `<strong>${edu.graduation || 'Graduation Date'}</strong><br> <strong>${edu.degree || 'Degree'}, ${edu.institution || 'Institution'}</strong><br> <ul> <li>${edu.details || 'Details about your education.'}</li> </ul>`; container.appendChild(div); });}function populateSkills(skills) { const container = document.querySelector('#skillsContent ul'); skills.forEach(skill => { const li = document.createElement('li'); li.className = 'col-md-4'; li.textContent = skill; container.appendChild(li); });}function populateProjects(projects) { const container = document.querySelector('#projectsContent ul'); projects.forEach(project => { const li = document.createElement('li'); li.className = 'col-md-4'; li.textContent = project; container.appendChild(li); });}function populateCertificates(certificates) { const container = document.querySelector('#certificatesContent ul'); certificates.forEach(certificate => { const li = document.createElement('li'); li.className = 'col-md-4'; li.textContent = certificate; container.appendChild(li); });}