A black background with a black square

Description automatically generated with medium confidence

**Department of**

**Master of Computer Applications**

**MCA112IA**

**WEB APPLICATION DEVELOPMENT**

**ASSIGNMENT**

**RESUME creation**

**Submitted by**

HONEY J

RVCE24MCA005

MCA I A

**RV COLLEGE OF ENGINEERING®, BENGALURU – 560059**

*(Autonomous Institution Affiliated to VTU, Belagavi)*

**Department of Master of Computer Applications**

**CERTIFICATE**

This is to certify that Ms HONEY J USN RVCE24MCA005 of **1st Semester**

Master of Computer Applications program has satisfactorily completed the course of Experiential Learning through Assignment **WEB APPLICATION PROGRAMMING- MCA112IA** prescribed for the academic year **2024 – 2025**.

|  |  |  |
| --- | --- | --- |
| Assignment marks | | |
| Phases | Max Marks | Obtained Marks |
| **1** | **30** |  |
| **2** | **30** |  |
| **Total 60** | |  |
| **Reduced to 40** | |  |

**Signature of Student Signature of faculty In-Charge Signature of Director**

**KEY TECHNOLOGIES USED**

**HTML, CSS, and JavaScript: The Core Building Blocks of Web Development**

**HTML5**

• Version: HTML5 (Latest specification)

• Description: Forms the structural foundation of the resume builder, utilizing semantic

elements like <section>, proper form controls, and modern input types (email, tel, file).

HTML5 provides the document structure and user input capabilities.

**CSS3**

• Version: CSS3 (Latest specification)

• Description: Handles all styling aspects including layout, colors, fonts, and responsive

design. The application uses features like flexbox for alignment, media queries for print

formatting, custom form styling, and box shadows for visual depth. CSS3 is critical for

both the UI aesthetics and the print-ready formatting.

**JavaScript**

• Version: ES6+ (Modern JavaScript)

• Description: Provides all interactive functionality through DOM manipulation. Key

features include dynamic field addition/removal, form validation, preview generation,

and image handling. The code uses modern JavaScript patterns including event listeners,

arrow functions, template literals, and the FileReader API.

**Bootstrap**

• Version: Not used

• Description: Despite its mention in your question, the code doesn't actually use

Bootstrap. All styling is handled through custom CSS rather than a framework.

**File Upload & Image Handling**

• Description: Implemented using HTML5's native file input and JavaScript's FileReader

API. This allows users to upload profile photos, preview them before submission, and

embed them in the final resume. The implementation includes proper image sizing and

positioning.

**Browser Print Feature**

• Description: Utilizes the browser's native print capability (window.print()) combined

with custom @print media queries in CSS. This creates a print-friendly version of the

resume that formats properly for paper output, hiding UI elements and adjusting

layout/spacing.

**SOFTWARE REQUIREMENTS**

**Web Browser :**

* Modern browsers (Chrome, Firefox, Safari, Edge) with support for:

ES6 JavaScript features

FileReader API

CSS3 featuresS

HTML5 input types

* Browser version requirements:

Chrome 49+

Firefox 45+

**Development Environment :**

* Any text editor or IDE for web development (VSCode, Notepad, etc.).
* Local web server (optional) for testing.
* Modern browser with developer tools for debugging.

**Deployment Requirements :**

* Basic web hosting with HTML support.
* No server-side technology required (static site compatible).
* No database requirement (all data remains client-side).

**HARDWARE REQUIREMENTS**

**User Hardware :**

• Any device capable of running a modern web browser.

• Minimum 512MB RAM (1GB+ recommended).

• Display resolution of at least 800x600 (responsive, but optimal at

1024x768 or higher).

• Input devices: keyboard and mouse/touchpad.

• Camera or existing photo files for profile upload.

• Printer access (physical or PDF printer) for resume output.

**Development Hardware :**

• Any computer capable of running a modern web browser and text

editor.

• Minimum 4GB RAM recommended for development tools.

• Display with resolution of 1280x720 or higher recommended.

• Storage: Minimal (< 5MB for the entire application).

**Server Hardware (if deployed) :**

• Minimal requirements - any basic web hosting.

• Bandwidth requirements: Low (primarily HTML/CSS/JS download).

• Storage: < 1MB for the application files.

• No special processing requirements (static content).

**FRONT PAGE (LANDING PAGE)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CV Creator</title>

<style>

header {

color: #ccc;

text-align: center;

padding: 2rem 0;

font-family: cursive;

animation: fadeIn 2s ease-in-out;

background-size: cover;

background-position: center;

background-repeat: no-repeat;

color: white;

position: relative;

overflow: hidden;

background-color: grey;

}

header::before {

content: "";

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.5);

z-index: 0;

}

header h1 {

font-family:Lucida Handwriting;

font-size: 3em;

font-weight: normal;

position: relative;

z-index: 1;

}

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

body {

font-family: sans-serif;

margin: 0;

padding: 0;

background:linear-gradient(black,grey,black);

background-size: cover;

background-position: center;

background-repeat: no-repeat;

color: black;

position: relative;

}

body::before {

content: "";

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(255, 255, 255, 0.5);

z-index: -1;

}

main {

padding: 20px;

display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 20px;

position: relative;

z-index: 0;

}

.cv-type {

border: 1px solid #ccc;

padding: 10px;

text-align: center;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

cursor: pointer;

transition: transform 0.3s ease, box-shadow 0.3s ease;

width: calc(50% - 20px);

max-width: 400px;

height: 350px;

display: flex;

flex-direction: column;

justify-content: space-between;

box-sizing: border-box;

background-color: grey;

color:white;

}

.cv-type:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

.cv-type img {

max-width: 100%;

height: 200px;

object-fit: cover;

}

.cv-type h3 {

margin: 10px 0;

}

.cv-type p {

margin-bottom: 10px;

}

.cv-type a {

text-decoration: none;

color: inherit;

display: block;

height: 100%;

}

.create-button {

background-color: black;

border: none;

color: white;

padding: 10px 20px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

margin-top: 8px;

cursor: pointer;

border-radius: 5px;

transition: background-color 0.3s ease, opacity 0.3s ease;

opacity: 1;

}

.create-button:hover {

background-color: lightblue;

opacity: 0.7;

}

@media (max-width: 600px) {

main {

flex-direction: column;

}

.cv-type {

width: 100%;

height: auto;

max-width: none;

}

.cv-type img {

height: auto;

}

}

footer {

background-color: #333;

color: #fff;

text-align: center;

padding: 10px;

position: relative;

bottom: 0;

width: 100%;

font-size: 0.8em;

z-index: 2;

}

h3,p {

font-family:Cursive;

}

</style>

</head>

<body>

<header>

<h1>BUILD YOUR CV</h1>

<p>Dedicated with a passion for Work</p>

</header>

<main>

<div class="cv-type">

<a href="business\_cv\_form.html" target="\_blank">

<imgsrc="https://cloudinary.hbs.edu/hbsit/image/upload/s--EmT0lNtW--/f\_auto,c\_fill,h\_375,w\_750,/v20200101/6978C1C20B650473DD135E5352D37D55.jpg" alt="Business CV">

<h3>Business CV</h3>

<p>For corporate jobs and professional roles.</p>

</a>

<a href="business\_cv\_form.html" target="\_self" class="create-button">Create Now</a>

</div>

<div class="cv-type">

<a href="freelance\_cv\_form.html" target="\_blank">

<imgsrc="https://lh5.googleusercontent.com/proxy/hlep460zq-AXkCG66\_hshIAA7W1d5mZetRqWVAkQxXSLNK9YG418wHBJy0uvYLCduCsciM6gkGRXg1hYE0Ie98zhmI1ZkS-v6u\_jzCJDue\_\_KGpwhLSmYRFehl-S1HI" alt="Freelance CV">

<h3>Freelance CV</h3>

<p>Highlighting skills and project experience.</p>

</a>

<a href="freelance\_cv\_form.html" target="\_self" class="create-button">Create Now</a>

</div>

<div class="cv-type">

<a href="student\_cv\_form.html" target="\_blank">

<imgsrc="https://gteducation.edu.in/wp-content/uploads/2020/12/iStock-1210745478-conflicted-scaled-1.jpg" alt="Student CV">

<h3>Student CV</h3>

<p>Focusing on education, skills, and activities.</p>

</a>

<a href="student\_cv\_form.html" target="\_self" class="create-button">Create Now</a>

</div>

<div class="cv-type">

<a href="other\_cv\_form.html" target="\_blank">

<imgsrc="https://ofhsoupkitchen.org/wp-content/uploads/2022/12/why-is-it-important-to-respect-other-opinions-1024x683.jpg" alt="Other CV">

<h3>Other CV</h3>

<p>Customized for specific purposes.</p>

</a>

<a href="other\_cv\_form.html" target="\_self" class="create-button">Create Now</a>

</div>

</main>

<footer>

© 2025 CV Creation. All rights reserved.

</footer>

</body>

</html>

**BUSINESS CV INPUT FORM**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Business CV Form</title>

<style>

body {

font-family: sans-serif;

margin: 0;

padding: 0;

background:white;

background-size: cover;

background-position: center;

background-repeat: no-repeat;

color: #333;

position: relative;

overflow-x: hidden;

}

body::before {

content: "";

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(255, 255, 255, 0.5);

z-index: -1;

}

.go-back-button {

display: inline-block;

padding: 10px 20px;

background-color:thistle;

color: white;

text-decoration: none;

border-radius: 5px;

margin: 10px;

transition: background-color 0.3s ease;

position: relative;

z-index: 1;

}

.go-back-button:hover {

background-color: Blue;

}

form {

width: 80%;

margin: 20px auto;

padding: 20px;

background: linear-gradient(to top,grey,grey);

border-radius: 5px;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2);

position: relative;

z-index: 1;

transition: transform 0.3s ease-in-out, box-shadow 0.3s ease-in-out;

}

form:hover {

transform: translateY(-5px);

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);

}

label {

display: block;

margin-bottom: 5px;

font-weight: bold;

color:white;

}

input[type="text"],

input[type="email"],

input[type="tel"],

input[type="url"],

textarea,

input[type="file"] {

width: 100%;

padding: 8px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 4px;

box-sizing: border-box;

transition: border-color 0.3s ease;

}

input[type="text"]:hover,

input[type="email"]:hover,

input[type="tel"]:hover,

input[type="url"]:hover,

textarea:hover,

input[type="file"]:hover {

border-color: #66afe9;

outline: 0;

}

input[type="text"]:focus,

input[type="email"]:focus,

input[type="tel"]:focus,

input[type="url"]:focus,

textarea:focus,

input[type="file"]:focus {

border-color: #66afe9;

outline: 0;

box-shadow: 0 0 5px rgba(102, 175, 233, .5);

}

.error {

color: thistle;

font-size: 0.8em;

margin-top: -10px;

margin-bottom: 5px;

display: none;

}

input:invalid + .error,

textarea:invalid + .error {

display: block;

}

button[type="submit"] {

background-color: thistle;

color: white;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

transition: background-color 0.3s ease;

}

button[type="submit"]:hover {

background-color: green;

}

@keyframes fadeIn {

from {

opacity: 0;

transform: translateY(-20px);

}

to {

opacity: 1;

transform: translateY(0);

} }

form {

animation: fadeIn 0.5s ease-out;

}

.qualification-container {

margin-bottom: 15px;

border: 1px solid #ccc;

padding: 10px;

border-radius: 4px;

background-color: #f9f9f9;

}

.add-more-button {

background-color: #5cb85c;

color: white;

padding: 8px 15px;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 14px;

margin-top: 5px;

}

.add-more-button:hover {

background-color: #449d44;

}

.remove-button {

background-color: #d9534f;

color: white;

padding: 5px 10px;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 12px;

margin-left: 5px;

}

.remove-button:hover {

background-color: #c9302c;

}

</style>

</head>

<body>

<a href="front.html.html" class="go-back-button">Go Back</a>

<form id="businessCVForm">

<h2>Business CV Form</h2>

<label for="profilePicture">Profile Picture:</label>

<input type="file" id="profilePicture" name="profilePicture" accept="image/\*">

<div class="error">Please upload a profile picture.</div>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName" pattern="[A-Za-z\s]+" required>

<div class="error">Please enter only letters and spaces.</div>

<label for="email">Email:</label>

<input type="email" id="email" name="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}" required>

<div class="error">Please enter a valid email address.</div>

<label for="phone">Phone Number:</label>

<input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required>

<div class="error">Please enter a 10-digit phone number (numbers only).</div>

<label for="linkedIn">LinkedIn Profile:</label>

<input type="url" id="linkedIn" name="linkedIn" pattern="https:\/\/.\*" >

<div class="error">Please enter a valid LinkedIn URL.</div>

<label for="jobTitle">Desired Job Title:</label>

<input type="text" id="jobTitle" name="jobTitle" required>

<div class="error">This field is required.</div>

<label for="yearsExperience">Years of Experience:</label>

<input type="text" id="yearsExperience" name="yearsExperience" pattern="[0-9]+" required>

<div class="error">Please enter a number for years of experience.</div>

<label for="objective">Career Objective:</label>

<textarea id="objective" name="objective" rows="3" required></textarea>

<div class="error">Please enter your career objective.</div>

<div id="keyAchievementsSection">

<label>Key Achievements:</label>

<div class="qualification-container">

<textarea name="keyAchievements[]" rows="3"></textarea>

<div class="error">Please enter your key achievements.</div>

</div>

<button type="button" class="add-more-button" onclick="addKeyAchievement()">Add More Achievement</button>

</div>

<div id="professionalAffiliationsSection">

<label>Professional Affiliations:</label>

<div class="qualification-container">

<textarea name="professionalAffiliations[]" rows="2"></textarea>

<div class="error">List any professional affiliations here.</div>

</div>

<button type="button" class="add-more-button" onclick="addProfessionalAffiliation()">Add More Affiliation</button></div>

<label for="skills">Key Skills (separate by commas):</label>

<textarea id="skills" name="skills" rows="3" required></textarea>

<div class="error">This field is required.</div>

<label for="summary">Professional Summary:</label>

<textarea id="summary" name="summary" rows="4" required></textarea>

<div class="error">This field is required.</div>

<button type="submit">Create CV</button>

</form>

<script>

const form = document.getElementById("businessCVForm");

function populateForm() {

const urlParams = new URLSearchParams(window.location.search);

for (const [key, value] of urlParams.entries()) {

const element = document.getElementById(key);

if (element) {

if (element.type === "text" || element.type === "email" || element.type === "tel" || element.type === "url" || element.tagName.toLowerCase() === "textarea")

{ element.value = value;

} } } }

window.addEventListener('load', populateForm);

function addKeyAchievement() {

addQualification("keyAchievements", "Key Achievements"); }

function addProfessionalAffiliation() {

addQualification("professionalAffiliations", "Professional Affiliation"); }

function addQualification(sectionId, labelText) {

const section = document.getElementById(sectionId + "Section");

const container = document.createElement("div");

container.classList.add("qualification-container");

const textarea = document.createElement("textarea");

textarea.name = sectionId + "[]";

textarea.rows = 3;

const errorDiv = document.createElement("div");

errorDiv.classList.add("error");

errorDiv.textContent = `Please enter your ${labelText.toLowerCase()}.`;

const removeButton = document.createElement("button");

removeButton.type = "button";

removeButton.classList.add("remove-button");

removeButton.textContent = "Remove";

removeButton.onclick = function() {

section.removeChild(container);

};

container.appendChild(textarea);

container.appendChild(errorDiv);

container.appendChild(removeButton);

section.insertBefore(container, section.querySelector(".add-more-button"));

}

form.addEventListener("submit", function(event) {

event.preventDefault(); // Prevent default submission to handle image

if (!form.checkValidity()) {

form.reportValidity();

return;

}

const formData = new FormData(this);

const data = Object.fromEntries(formData.entries());

const keyAchievements = formData.getAll('keyAchievements[]');

data.keyAchievements = keyAchievements;

const professionalAffiliations = formData.getAll('professionalAffiliations[]');

data.professionalAffiliations = professionalAffiliations;

const profilePictureFile = formData.get('profilePicture');

if (profilePictureFile && profilePictureFile.size > 0) {

const reader = new FileReader();

reader.onload = function (e) {

data.profilePicture = e.target.result;

window.location.href="preview\_cv.html?data="+encodeURIComponent(JSON.stringify(data)); }

reader.readAsDataURL(profilePictureFile);

} else {

data.profilePicture = null;

window.location.href="preview\_cv.html?data="+encodeURIComponent(JSON.stringify(data)); } });

</script>

</body>

</html>

**PREVIEW OF BUSINESS CV**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CV Preview</title>

<style>

body {

font-family: 'Arial', sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

color: #333;

}

.container {

width: 21cm;

min-height: 29.7cm;

margin: 0 auto;

background-color: #fff;

box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);

display: flex;

border-radius: 5px;

overflow: hidden;

page-break-inside: avoid;

box-sizing: border-box;

}

.left-section {

background-color: #343a40;

color: #fff;

padding: 15px;

width: 35%;

flex-shrink: 0;

box-sizing: border-box;

}

.profile-img {

width: 90%;

border-radius: 50%;

margin-bottom: 15px;

display: block;

margin-left: auto;

margin-right: auto;

}

.contact-info, .skills {

margin-bottom: 15px;

}

.contact-info h3, .skills h3 {

border-bottom: 2px solid #fff;

padding-bottom: 5px;

font-size: 1.2em;

font-weight: bold;

}

.contact-info p, .skills ul {

padding-left: 0;

list-style: none;

}

.contact-info p {

margin-bottom: 4px;

word-break: break-word;

line-height: 1.3;

font-size: 1.1em;

text-align: justify;

}

.skills ul {

padding-left: 10px;

}

.skills li {

margin-bottom: 4px;

font-size: 1.1em;

word-break: break-word;

overflow-wrap: break-word;

}

.right-section {

padding: 15px;

width: 65%;

box-sizing: border-box;

}

.section-title {

font-size: 1.6em;

border-bottom: 4px solid #343a40;

margin-bottom: 15px;

padding-bottom: 5px;

font-weight: bolder;

}

.section-divider {

border-bottom: 1px solid #ccc;

margin-bottom: 15px;

}

.right-section p {

word-wrap: break-word;

overflow-wrap: break-word;

line-height: 1.5;

margin-bottom: 12px;

font-size: 1.1em;

text-align: justify;

}

.right-section > div:last-child {

margin-bottom: 0;

height: auto;

}

@media (max-width: 768px) {

.container {

flex-direction: column;

width: 95%;

}

.left-section, .right-section {

width: 100%;

}

}

@media print {

@page {

size: A4;

margin: 0;

}

body {

-webkit-print-color-adjust: exact !important;

color-adjust: exact !important;

background-color: #fff !important;

margin: 0 !important;

padding: 0 !important;

font-size: 11pt !important;

line-height: 1.3 !important;

}

.container {

width: 21cm !important;

min-height: 29.7cm !important;

max-height: 29.7cm !important;

margin: 0 !important;

padding: 0 !important;

box-shadow: none !important;

display: flex !important;

border-radius: 5px !important;

overflow: hidden !important;

page-break-inside: avoid !important;

flex-direction: row !important;

box-sizing: border-box !important;

}

.left-section {

background-color: #343a40 !important;

color: #fff !important;

padding: 15px !important;

width: 35% !important;

flex-shrink: 0 !important;

box-sizing: border-box !important;

}

.profile-img {

width: 90% !important;

border-radius: 50% !important;

margin-bottom: 15px !important;

display: block !important;

margin-left: auto !important;

margin-right: auto !important;

}

.contact-info, .skills {

margin-bottom: 15px !important;

}

.contact-info h3, .skills h3 {

border-bottom: 2px solid #fff !important;

padding-bottom: 5px !important;

font-size: 1.2em !important;

font-weight: bold !important;

}

.contact-info p {

margin-bottom: 4px !important;

word-break: break-word !important;

line-height: 1.3 !important;

font-size: 1.1em !important;

text-align: justify !important;

}

.skills ul {

padding-left: 10px !important;

}

.skills li {

margin-bottom: 4px !important;

font-size: 1.1em !important;

word-break: break-word !important;

overflow-wrap: break-word !important;

}

.right-section {

padding: 15px !important;

width: 65% !important;

box-sizing: border-box !important;

page-break-inside: avoid !important;

}

.section-title {

font-size: 1.6em !important;

border-bottom: 4px solid #343a40 !important;

margin-bottom: 15px !important;

padding-bottom: 5px !important;

font-weight: bolder !important;

}

.section-divider {

border-bottom: 1px solid #ccc !important;

margin-bottom: 15px !important;

}

.right-section p {

word-wrap: break-word !important;

overflow-wrap: break-word !important;

line-height: 1.5 !important;

margin-bottom: 12px !important;

font-size: 1.1em !important;

text-align: justify !important;

}

.download-btn {

display: none !important;

}

}

.download-btn {

background-color: #5cb85c;

border: none;

color: white;

padding: 10px 20px;

text-align: center;

text-decoration: none;

display: block;

font-size: 16px;

margin: 15px auto;

cursor: pointer;

border-radius: 5px;

transition: background-color: #4cae4c;

}

.dynamic-section {

margin-bottom: 20px;

}

</style>

</head>

<body>

<div class="container">

<div class="left-section">

<img id="previewProfilePicture" src="https://via.placeholder.com/150" alt="Profile Picture" class="profile-img">

<div class="contact-info">

<h3>Contact Information</h3>

<p id="previewFullName"><strong>Name:</strong></p>

<p id="previewEmail"><strong>Email:</strong></p>

<p id="previewPhone"><strong>Phone:</strong></p>

<p id="previewLinkedIn"><strong>LinkedIn:</strong></p>

</div>

<div class="skills">

<h3>Skills</h3>

<ul id="previewSkills">

</ul>

</div>

</div>

<div class="right-section">

<div class="objective">

<h2 class="section-title">Career Objective</h2>

<p id="previewObjective"></p>

<div class="section-divider"></div>

</div>

<div class="summary">

<h2 class="section-title">Professional Summary</h2>

<p id="previewSummary"></p>

<div class="section-divider"></div>

</div>

<div class="achievements">

<h2 class="section-title">Key Achievements</h2>

<ul id="previewAchievements">

</ul>

<div class="section-divider"></div>

</div>

<div class="affiliations">

<h2 class="section-title">Professional Affiliations</h2>

<ul id="previewAffiliations">

</ul>

<div class="section-divider"></div>

</div>

</div>

</div>

<button class="download-btn" onclick="window.print()">Download as PDF</button>

<script>

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

const urlParams = new URLSearchParams(window.location.search);

const data = JSON.parse(decodeURIComponent(urlParams.get('data')));

function setTextContent(elementId, value) {

const element = document.getElementById(elementId);

if (element) {

element.textContent = value || '';

}

}

function setInnerHTML(elementId, value) {

const element = document.getElementById(elementId);

if (element) {

element.innerHTML = value || '';

}

}

const profilePicture = document.getElementById('previewProfilePicture');

if (data.profilePicture) {

profilePicture.src = data.profilePicture;

} else {

profilePicture.src = 'https://via.placeholder.com/150';

}

setInnerHTML('previewFullName', `<strong>Name:</strong> ${data.fullName || ''}`);

setInnerHTML('previewEmail', `<strong>Email:</strong> ${data.email || ''}`);

setInnerHTML('previewPhone', `<strong>Phone:</strong> ${data.phone || ''}`);

let linkedInLink = data.linkedIn ? `<a href="${data.linkedIn}" target="\_blank">${data.linkedIn}</a>` : '';

setInnerHTML('previewLinkedIn', `<strong>LinkedIn:</strong> ${linkedInLink}`);

const skillsList = document.getElementById('previewSkills');

skillsList.innerHTML = '';

if (data.skills) {

const skillsArray = data.skills.split(',').map(skill => skill.trim());

skillsArray.forEach(skill => {

let li = document.createElement('li');

li.textContent = skill;

li.style.wordBreak = 'break-word';

skillsList.appendChild(li);

});

}

setTextContent('previewSummary', data.summary || '');

setTextContent('previewObjective', data.objective || '');

const achievementsList = document.getElementById('previewAchievements');

achievementsList.innerHTML = '';

if (data.keyAchievements && Array.isArray(data.keyAchievements)) {

data.keyAchievements.forEach(achievement => {

if (achievement) {

let li = document.createElement('li');

li.textContent = achievement;

achievementsList.appendChild(li);

}

});

}

const affiliationsList = document.getElementById('previewAffiliations');

affiliationsList.innerHTML = '';

if (data.professionalAffiliations && Array.isArray(data.professionalAffiliations)) {

data.professionalAffiliations.forEach(affiliation => {

if (affiliation) {

let li = document.createElement('li');

li.textContent = affiliation;

affiliationsList.appendChild(li);

}

});

}

});

</script>

</body>

</html>

**FREELANCER CV INPUT FORM**

// Starting of the code along with the styling is same. The updated parts for freelancing code //

<body>

<a href="front.html.html" class="go-back-button">Go Back</a>

<form id="freelanceCVForm">

<h2>Freelance CV Form</h2>

<label for="profilePicture">Profile Picture:</label>

<input type="file" id="profilePicture" name="profilePicture" accept="image/\*">

<div class="error">Please upload a profile picture.</div>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName" pattern="[A-Za-z\s]+" required>

<div class="error">Please enter only letters and spaces.</div>

<label for="email">Email:</label>

<input type="email" id="email" name="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}" required>

<div class="error">Please enter a valid email address.</div>

<label for="phone">Phone Number:</label>

<input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required>

<div class="error">Please enter a 10-digit phone number.</div>

<label for="linkedIn">LinkedIn Profile:</label>

<input type="url" id="linkedIn" name="linkedIn" pattern="https:\/\/.\*" >

<div class="error">Please enter a valid LinkedIn URL (including https://).</div>

<label for="jobTitle">Desired Job Title:</label>

<input type="text" id="jobTitle" name="jobTitle" required>

<div class="error">This field is required.</div>

<label for="yearsExperience">Years of Experience:</label>

<input type="text" id="yearsExperience" name="yearsExperience" pattern="[0-9]+" required>

<div class="error">Please enter a number for years of experience.</div>

<label for="objective">Career Objective:</label>

<textarea id="objective" name="objective" rows="3" required></textarea>

<div class="error">Please enter your career objective.</div>

<div id="keyAchievementsSection">

<label>Key Achievements:</label>

<div class="qualification-container">

<textarea name="keyAchievements[]" rows="3"></textarea>

<div class="error">Please enter your key achievements.</div>

</div>

<button type="button" class="add-more-button" onclick="addKeyAchievement()">Add More Achievement</button>

</div>

<div id="professionalAffiliationsSection">

<label>Professional Affiliations:</label>

<div class="qualification-container">

<textarea name="professionalAffiliations[]" rows="2"></textarea>

<div class="error">List any professional affiliations here.</div>

</div>

<button type="button" class="add-more-button" onclick="addProfessionalAffiliation()">Add More Affiliation</button>

</div>

<label for="skills">Key Skills (separate by commas):</label>

<textarea id="skills" name="skills" rows="3" required></textarea>

<div class="error">This field is required.</div>

<label for="summary">Professional Summary:</label>

<textarea id="summary" name="summary" rows="4" required></textarea>

<div class="error">This field is required.</div>

<button type="submit">Create CV</button>

</form>

<script>

const form = document.getElementById("freelanceCVForm");

function populateForm() {

const urlParams = new URLSearchParams(window.location.search);

for (const [key, value] of urlParams.entries()) {

const element = document.getElementById(key);

if (element) {

if (element.type === "text" || element.type === "email" || element.type === "tel" || element.type === "url" || element.tagName.toLowerCase() === "textarea") {

element.value = value;

}

}

}

}

window.addEventListener('load', populateForm);

function addKeyAchievement() {

addQualification("keyAchievements", "Key Achievements");

}

function addProfessionalAffiliation() {

addQualification("professionalAffiliations", "Professional Affiliation");

}

function addQualification(sectionId, labelText) {

const section = document.getElementById(sectionId + "Section");

const container = document.createElement("div");

container.classList.add("qualification-container");

const textarea = document.createElement("textarea");

textarea.name = sectionId + "[]";

textarea.rows = 3;

const errorDiv = document.createElement("div");

errorDiv.classList.add("error");

errorDiv.textContent = `Please enter your ${labelText.toLowerCase()}.`;

const removeButton = document.createElement("button");

removeButton.type = "button";

removeButton.classList.add("remove-button");

removeButton.textContent = "Remove";

removeButton.onclick = function() {

section.removeChild(container);

};

container.appendChild(textarea);

container.appendChild(errorDiv);

container.appendChild(removeButton);

section.insertBefore(container, section.querySelector(".add-more-button"));

}

form.addEventListener("submit", function(event) {

event.preventDefault();

if (!form.checkValidity()) {

form.reportValidity();

return;

}

const formData = new FormData(this);

const data = Object.fromEntries(formData.entries());

const keyAchievements = formData.getAll('keyAchievements[]');

data.keyAchievements = keyAchievements;

const professionalAffiliations = formData.getAll('professionalAffiliations[]');

data.professionalAffiliations = professionalAffiliations;

const profilePictureFile = formData.get('profilePicture'); // Get the file

if (profilePictureFile && profilePictureFile.size > 0) { // Check if a file was selected

const reader = new FileReader();

reader.onload = function (e) {

data.profilePicture = e.target.result; // Store the image data URL

window.location.href = "preview\_cv\_f.html?data=" + encodeURIComponent(JSON.stringify(data));

}

reader.readAsDataURL(profilePictureFile); // Read the file as a data URL

} else {

data.profilePicture = null; // Set to null if no file selected, or send a default image URL

window.location.href = "preview\_cv\_f.html?data=" + encodeURIComponent(JSON.stringify(data));

}

});

</script>

</body>

</html>

**PREVIEW OF FREELANCER CV**

<body>

<div class="container">

<div class="left-section">

<img id="previewProfilePicture" src="https://via.placeholder.com/150" alt="Profile Picture" class="profile-img">

<div class="contact-info">

<h3>Contact Information</h3>

<p id="previewFullName"><strong>Name:</strong></p>

<p id="previewEmail"><strong>Email:</strong></p>

<p id="previewPhone"><strong>Phone:</strong></p>

<p id="previewLinkedIn"><strong>LinkedIn:</strong></p>

</div>

<div class="skills">

<h3>Skills</h3>

<ul id="previewSkills">

</ul>

</div>

</div>

<div class="right-section">

<div class="objective">

<h2 class="section-title">Career Objective</h2>

<p id="previewObjective"></p>

<div class="section-divider"></div>

</div>

<div class="jobTitle">

<h2 class="section-title">Desired Job Title</h2>

<p id="previewJobTitle"></p>

<div class="section-divider"></div>

</div>

<div class="yearsExperience">

<h2 class="section-title">Years of Experience</h2>

<p id="previewYearsExperience"></p>

<div class="section-divider"></div>

</div>

<div class="keyAchievements">

<h2 class="section-title">Key Achievements</h2>

<p id="previewKeyAchievements"></p>

<div class="section-divider"></div>

</div>

<div class="professionalAffiliations">

<h2 class="section-title">Professional Affiliations</h2>

<p id="previewProfessionalAffiliations"></p>

<div class="section-divider"></div>

</div>

<div class="summary">

<h2 class="section-title">Professional Summary</h2>

<p id="previewSummary"></p>

</div>

</div>

</div>

<button class="download-btn" onclick="window.print()">Download as PDF</button>

<script>

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

const urlParams = new URLSearchParams(window.location.search);

const data = JSON.parse(decodeURIComponent(urlParams.get('data')));

function setTextContent(elementId, value) {

const element = document.getElementById(elementId);

if (element) {

element.textContent = value || '';

}

}

function setInnerHTML(elementId, value) {

const element = document.getElementById(elementId);

if (element) {

element.innerHTML = value || '';

}

}

const profilePicture = document.getElementById('previewProfilePicture');

if (data.profilePicture) {

profilePicture.src = data.profilePicture;

} else {

profilePicture.src = 'https://via.placeholder.com/150';

}

setInnerHTML('previewFullName', `<strong>Name:</strong> ${data.fullName || ''}`);

setInnerHTML('previewEmail', `<strong>Email:</strong> ${data.email || ''}`);

setInnerHTML('previewPhone', `<strong>Phone:</strong> ${data.phone || ''}`);

let linkedInLink = data.linkedIn ? `<a href="${data.linkedIn}" target="\_blank">${data.linkedIn}</a>` : '';

setInnerHTML('previewLinkedIn', `<strong>LinkedIn:</strong> ${linkedInLink}`);

const skillsList = document.getElementById('previewSkills');

skillsList.innerHTML = '';

if (data.skills) {

const skillsArray = data.skills.split(',').map(skill => skill.trim());

skillsArray.forEach(skill => {

let li = document.createElement('li');

li.textContent = skill;

li.style.wordBreak = 'break-word';

skillsList.appendChild(li);

});

}

setTextContent('previewSummary', data.summary || '');

setTextContent('previewObjective', data.objective || '');

setTextContent('previewJobTitle', data.jobTitle || '');

setTextContent('previewYearsExperience', data.yearsExperience || '');

setTextContent('previewKeyAchievements', data.keyAchievements || '');

setTextContent('previewProfessionalAffiliations', data.professionalAffiliations || '');

});

</script>

</body>

</html>

**STUDENT CV INPUT FORM**

<body>

<a href="front.html.html" class="go-back-button">Go Back</a>

<form id="studentCVForm">

<h2>Student CV Form</h2>

<label for="profilePicture">Profile Picture:</label>

<input type="file" id="profilePicture" name="profilePicture" accept="image/\*">

<div class="error">Please upload a profile picture.</div>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName" pattern="[A-Za-z\s]+" required>

<div class="error">Please enter only letters and spaces.</div>

<label for="email">Email:</label>

<input type="email" id="email" name="email" pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}" required>

<div class="error">Please enter a valid email address.</div>

<label for="phone">Phone Number:</label>

<input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required>

<div class="error">Please enter a 10-digit phone number (numbers only).</div>

<label for="linkedIn">LinkedIn Profile:</label>

<input type="url" id="linkedIn" name="linkedIn">

<div class="error">Please enter a valid LinkedIn URL.</div>

<label for="major">Major:</label>

<input type="text" id="major" name="major" required>

<div class="error">This field is required.</div>

<label for="graduationYear">Expected Graduation Year:</label>

<input type="text" id="graduationYear" name="graduationYear" pattern="[0-9]{4}" required>

<div class="error">Please enter a valid 4-digit year.</div>

<label for="objective">Career Objective:</label>

<textarea id="objective" name="objective" rows="3" required></textarea>

<div class="error">Please enter your career objective.</div>

<div id="educationSection">

<label>Education (Institution, Degree, Expected Graduation):</label>

<div class="qualification-container">

<textarea name="education[]" rows="3"></textarea>

<div class="error">Please enter your education details.</div>

</div>

<button type="button" class="add-more-button" onclick="addEducation()">Add More Education</button>

</div>

<div id="awardsSection">

<label>Awards and Recognition:</label>

<div class="qualification-container">

<textarea name="awards[]" rows="2"></textarea>

<div class="error">List any awards or recognitions received.</div>

</div>

<button type="button" class="add-more-button" onclick="addAward()">Add More Award</button>

</div>

<div id="internshipsSection">

<label>Internships (Company, Role, Dates):</label>

<div class="qualification-container">

<textarea name="internships[]" rows="3"></textarea>

<div class="error">List your internship experience.</div>

</div>

<button type="button" class="add-more-button" onclick="addInternship()">Add More Internship</button>

</div>

<div id="projectsSection">

<label>Projects (Description, Technologies Used):</label>

<div class="qualification-container">

<textarea name="projects[]" rows="3"></textarea>

<div class="error">Describe any relevant projects.</div>

</div>

<button type="button" class="add-more-button" onclick="addProject()">Add More Project</button>

</div>

<label for="skills">Key Skills (separate by commas):</label>

<textarea id="skills" name="skills" rows="3" required></textarea>

<div class="error">This field is required.</div>

<label for="summary">Professional Summary:</label>

<textarea id="summary" name="summary" rows="4" required></textarea>

<div class="error">This field is required.</div>

<button type="submit">Create CV</button>

</form>

<script>

const form = document.getElementById("studentCVForm");

function populateForm() {

const urlParams = new URLSearchParams(window.location.search);

for (const [key, value] of urlParams.entries()) {

const element = document.getElementById(key);

if (element) {

if (element.type === "text" || element.type === "email" || element.type === "tel" || element.type === "url" || element.tagName.toLowerCase() === "textarea") {

element.value = value;

}

}

}

}

window.addEventListener('load', populateForm);

function addEducation() {

addQualification("education", "Education Details");

}

function addAward() {

addQualification("awards", "Awards and Recognition");

}

function addInternship() {

addQualification("internships", "Internship Experience");

}

function addProject() {

addQualification("projects", "Project Details");

}

function addQualification(sectionId, labelText) {

const section = document.getElementById(sectionId + "Section");

const container = document.createElement("div");

container.classList.add("qualification-container");

const textarea = document.createElement("textarea");

textarea.name = sectionId + "[]";

textarea.rows = 3;

// No required attribute here, as these are not mandatory

const errorDiv = document.createElement("div");

errorDiv.classList.add("error");

errorDiv.textContent = `Please enter your ${labelText.toLowerCase()}.`;

const removeButton = document.createElement("button");

removeButton.type = "button";

removeButton.classList.add("remove-button");

removeButton.textContent = "Remove";

removeButton.onclick = function() {

section.removeChild(container);

};

container.appendChild(textarea);

container.appendChild(errorDiv);

container.appendChild(removeButton);

section.insertBefore(container, section.querySelector(".add-more-button"));

}

form.addEventListener("submit", function(event) {

event.preventDefault();

if (!form.checkValidity()) {

form.reportValidity();

return; // Stop submission if invalid

}

const formData = new FormData(this);

const data = Object.fromEntries(formData.entries());

const education = formData.getAll('education[]');

data.education = education;

const awards = formData.getAll('awards[]');

data.awards = awards;

const internships = formData.getAll('internships[]');

data.internships = internships;

const projects = formData.getAll('projects[]');

data.projects = projects;

const profilePictureFile = formData.get('profilePicture');

if (profilePictureFile && profilePictureFile.size > 0) {

const reader = new FileReader();

reader.onload = function (e) {

data.profilePicture = e.target.result;

window.location.href="preview\_cv\_s.html?data="+encodeURIComponent

(JSON.stringify(data));

}

reader.readAsDataURL(profilePictureFile); // Read the file as a data URL

} else {

data.profilePicture = null;

window.location.href="preview\_cv\_s.html?data="+encodeURIComponent

(JSON.stringify(data));

} });

</script>

</body>

</html>

**PREVIEW OF STUDENT CV**

<body>

<div class="container">

<div class="left-section">

<img id="previewProfilePicture" src="https://via.placeholder.com/150" alt="Profile Picture" class="profile-img">

<div class="contact-info">

<h3>Contact Information</h3>

<p id="previewFullName"><strong>Name:</strong></p>

<p id="previewEmail"><strong>Email:</strong></p>

<p id="previewPhone"><strong>Phone:</strong></p>

<p id="previewLinkedIn"><strong>LinkedIn:</strong></p>

</div> <div class="skills">

<h3>Skills</h3>

<ul id="previewSkills"> </ul>

</div>

</div>

<div class="right-section">

<div class="objective">

<h2 class="section-title">Career Objective</h2>

<p id="previewObjective"></p>

<div class="section-divider"></div>

</div>

<div class="summary">

<h2 class="section-title">Professional Summary</h2>

<p id="previewSummary"></p>

<div class="section-divider"></div>

</div>

<div class="education">

<h2 class="section-title">Education</h2>

<ul id="previewEducation"> </ul>

<div class="section-divider"></div>

</div>

<div class="awards">

<h2 class="section-title">Awards and Recognition</h2>

<ul id="previewAwards"> </ul>

<div class="section-divider"></div>

</div>

<div class="internships">

<h2 class="section-title">Internships</h2>

<ul id="previewInternships"> </ul>

<div class="section-divider"></div>

</div>

<div class="projects">

<h2 class="section-title">Projects</h2>

<ul id="previewProjects"> </ul>

<div class="section-divider"></div>

</div>

</div>

</div>

<div class="page-break"></div>

<button class="download-btn" onclick="window.print()">Download as PDF</button>

<script>

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

const urlParams = new URLSearchParams(window.location.search);

const data = JSON.parse(decodeURIComponent(urlParams.get('data')));

function setTextContent(elementId, value) {

const element = document.getElementById(elementId);

if (element) {

element.textContent = value || '';

} }

function setInnerHTML(elementId, value) {

const element = document.getElementById(elementId);

if (element) {

element.innerHTML = value || '';

} }

const profilePicture = document.getElementById('previewProfilePicture');

if (data.profilePicture) {

profilePicture.src = data.profilePicture;

} else {

profilePicture.src = 'https://via.placeholder.com/150';

}

setInnerHTML('previewFullName', `<strong>Name:</strong> ${data.fullName || ''}`);

setInnerHTML('previewEmail', `<strong>Email:</strong> ${data.email || ''}`);

setInnerHTML('previewPhone', `<strong>Phone:</strong> ${data.phone || ''}`);

let linkedInLink = data.linkedIn ? `<a href="${data.linkedIn}" target="\_blank">${data.linkedIn}</a>` : '';

setInnerHTML('previewLinkedIn', `<strong>LinkedIn:</strong> ${linkedInLink}`);

const skillsList = document.getElementById('previewSkills');

skillsList.innerHTML = '';

if (data.skills) {

const skillsArray = data.skills.split(',').map(skill => skill.trim());

skillsArray.forEach(skill => {

let li = document.createElement('li');

li.textContent = skill;

li.style.wordBreak = 'break-word';

skillsList.appendChild(li);

}); }

setTextContent('previewSummary', data.summary || '');

setTextContent('previewObjective', data.objective || '');

const educationList = document.getElementById('previewEducation');

educationList.innerHTML = '';

if (data.education && Array.isArray(data.education)) {

data.education.forEach(item => {

if (item) {

let li = document.createElement('li');

li.textContent = item;

educationList.appendChild(li);

} }); }

const awardsList = document.getElementById('previewAwards');

awardsList.innerHTML = '';

if (data.awards && Array.isArray(data.awards)) {

data.awards.forEach(item => {

if (item) {

let li = document.createElement('li');

li.textContent = item;

awardsList.appendChild(li);

} }); }

const internshipsList = document.getElementById('previewInternships');

internshipsList.innerHTML = '';

if (data.internships && Array.isArray(data.internships)) {

data.internships.forEach(item => {

if (item) {

let li = document.createElement('li');

li.textContent = item;

internshipsList.appendChild(li);

} }); }

const projectsList = document.getElementById('previewProjects');

projectsList.innerHTML = '';

if (data.projects && Array.isArray(data.projects)) {

data.projects.forEach(item => {

if (item) {

let li = document.createElement('li');

li.textContent = item;

projectsList.appendChild(li);

} }); }

});

</script>

</body>

</html>