ASSIGNMENT 14:

CODE:

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Responsive Web Page</title>

<link rel="stylesheet" href="styles.css" />

</head>

<body>

<!-- Header -->

<header>

<div class="container">

<h1>My Website</h1>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Services</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</div>

</header>

<!-- Main Content -->

<main class="container">

<section class="intro">

<h2>Welcome!</h2>

<p>This is a responsive layout demo. Resize the window to see how it adapts across devices.</p>

</section>

<section class="content">

<div class="card">

<img src="https://via.placeholder.com/300x200" alt="Placeholder Image">

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae justo nec nulla luctus bibendum.</p>

</div>

<div class="card">

<img src="https://via.placeholder.com/300x200" alt="Placeholder Image">

<p>Donec vel quam nec magna tincidunt aliquet. Suspendisse potenti. Nulla facilisi.</p>

</div>

</section>

</main>

<!-- Footer -->

<footer>

<div class="container">

<p>&copy; 2025 My Website. All rights reserved.</p>

</div>

</footer>

</body>

</html>

CSS:

/\* ========== Base Styles ========== \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Segoe UI', sans-serif;

line-height: 1.6;

background: #f9f9f9;

color: #333;

}

.container {

width: 90%;

max-width: 1200px;

margin: auto;

}

/\* ========== Header ========== \*/

header {

background-color: #007BFF;

color: white;

padding: 1rem 0;

}

header h1 {

margin-bottom: 0.5rem;

}

nav ul {

display: flex;

flex-wrap: wrap;

list-style: none;

gap: 1rem;

}

nav a {

color: white;

text-decoration: none;

font-weight: bold;

}

/\* ========== Main Content ========== \*/

main {

padding: 2rem 0;

}

.intro {

margin-bottom: 2rem;

}

.content {

display: flex;

gap: 2rem;

flex-wrap: wrap;

}

.card {

flex: 1 1 45%;

background: white;

padding: 1rem;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

.card img {

max-width: 100%;

border-radius: 4px;

margin-bottom: 1rem;

}

/\* ========== Footer ========== \*/

footer {

background-color: #222;

color: #ddd;

text-align: center;

padding: 1rem 0;

}

/\* ========== Responsive Styles ========== \*/

/\* Tablets \*/

@media (max-width: 768px) {

nav ul {

flex-direction: column;

align-items: center;

}

.content {

flex-direction: column;

}

.card {

flex: 1 1 100%;

}

}

/\* Mobile \*/

@media (max-width: 480px) {

header h1 {

font-size: 1.5rem;

text-align: center;

}

nav ul {

gap: 0.5rem;

}

nav a {

font-size: 1rem;

}

.intro h2 {

font-size: 1.2rem;

}

.card p {

font-size: 0.95rem;

}

}

OUTPUT:

A screenshot of a phone

AI-generated content may be incorrect.

TASK-2:

CODE:

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Interactive Button</title>

<style>

body {

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f2f2f2;

}

button {

padding: 15px 30px;

font-size: 16px;

cursor: pointer;

background-color: #007BFF;

color: white;

border: none;

border-radius: 5px;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<!-- Interactive Button -->

<button id="myButton">Click Me</button>

<!-- Link to JavaScript file -->

<script src="script.js"></script>

</body>

</html>

JAVASCRIPT:

// Wait for the DOM to load before attaching event listeners

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

// Get reference to the button

const button = document.getElementById("myButton");

// Add click event listener to the button

button.addEventListener("click", function () {

// Show an alert when the button is clicked

alert("Button clicked!");

});

});

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Task-3:

CODE:

HTML:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Contact Form with Validation</title>

<link rel="stylesheet" href="styles.css" />

</head>

<body>

<main class="container">

<h1>Contact Us</h1>

<form id="contactForm" novalidate>

<div class="form-group">

<label for="name">Name:</label>

<input type="text" id="name" name="name" />

<small class="error-message"></small>

</div>

<div class="form-group">

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

<input type="email" id="email" name="email" />

<small class="error-message"></small>

</div>

<div class="form-group">

<label for="message">Message:</label>

<textarea id="message" name="message" rows="5"></textarea>

<small class="error-message"></small>

</div>

<button type="submit">Submit</button>

</form>

<p id="successMessage" class="success-message"></p>

</main>

<script src="script.js"></script>

</body>

</html>

CSS:

body {

font-family: Arial, sans-serif;

background-color: #fafafa;

margin: 0;

padding: 2rem;

display: flex;

justify-content: center;

}

.container {

width: 100%;

max-width: 500px;

background: white;

padding: 2rem;

border-radius: 8px;

box-shadow: 0 0 12px rgba(0,0,0,0.1);

}

h1 {

text-align: center;

margin-bottom: 1.5rem;

}

.form-group {

margin-bottom: 1.25rem;

position: relative;

}

label {

display: block;

margin-bottom: 0.5rem;

font-weight: bold;

}

input[type="text"],

input[type="email"],

textarea {

width: 100%;

padding: 0.6rem;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 1rem;

transition: border-color 0.3s;

}

input[type="text"]:focus,

input[type="email"]:focus,

textarea:focus {

outline: none;

border-color: #007BFF;

}

.error-message {

color: #d93025;

font-size: 0.85rem;

margin-top: 0.25rem;

display: none;

position: absolute;

bottom: -18px;

left: 0;

}

input.error,

textarea.error {

border-color: #d93025;

}

.success-message {

color: #2e7d32;

font-weight: bold;

text-align: center;

margin-top: 1rem;

}

button {

display: block;

width: 100%;

padding: 0.75rem;

background-color: #007BFF;

border: none;

color: white;

font-size: 1.1rem;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s ease;

}

button:hover {

background-color: #0056b3;

}

JAVASCRIPT:

// Wait for the DOM to load

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

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

const successMessage = document.getElementById("successMessage");

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

e.preventDefault(); // Prevent form from submitting

// Clear previous success message

successMessage.textContent = "";

// Validate the form

const isValid = validateForm();

if (isValid) {

successMessage.textContent = "Form submitted successfully!";

form.reset();

clearErrors();

}

});

function validateForm() {

let valid = true;

// Get form fields

const nameInput = form.elements["name"];

const emailInput = form.elements["email"];

const messageInput = form.elements["message"];

clearErrors();

// Validate Name

if (nameInput.value.trim() === "") {

showError(nameInput, "Name is required.");

valid = false;

}

// Validate Email

if (emailInput.value.trim() === "") {

showError(emailInput, "Email is required.");

valid = false;

} else if (!isValidEmail(emailInput.value.trim())) {

showError(emailInput, "Please enter a valid email.");

valid = false;

}

// Validate Message

if (messageInput.value.trim() === "") {

showError(messageInput, "Message is required.");

valid = false;

}

return valid;

}

// Show inline error message

function showError(input, message) {

input.classList.add("error");

const errorMessage = input.parentElement.querySelector(".error-message");

if (errorMessage) {

errorMessage.textContent = message;

errorMessage.style.display = "block";

}

}

// Clear all error messages and styles

function clearErrors() {

const errorMessages = form.querySelectorAll(".error-message");

errorMessages.forEach((msg) => {

msg.textContent = "";

msg.style.display = "none";

});

const errorInputs = form.querySelectorAll(".error");

errorInputs.forEach((input) => {

input.classList.remove("error");

});

}

// Simple email format validation

function isValidEmail(email) {

// Basic regex for email validation

const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

return re.test(email.toLowerCase());

}

});

OUTPUT:

**A screen shot of a contact us form

AI-generated content may be incorrect.**

**Task-4:**

**CODE:**

**HTML:**

**<!DOCTYPE html>**

**<html lang="en" >**

**<head>**

**<meta charset="UTF-8" />**

**<meta name="viewport" content="width=device-width, initial-scale=1" />**

**<title>Dynamic List</title>**

**<style>**

**body {**

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

**padding: 2rem;**

**background: #f9f9f9;**

**max-width: 600px;**

**margin: auto;**

**}**

**h1 {**

**text-align: center;**

**margin-bottom: 1rem;**

**}**

**ul {**

**background: white;**

**padding: 1rem 1.5rem;**

**border-radius: 8px;**

**box-shadow: 0 0 10px rgba(0,0,0,0.1);**

**list-style-type: disc;**

**min-height: 100px;**

**}**

**li {**

**margin-bottom: 0.5rem;**

**font-size: 1.1rem;**

**}**

**.buttons {**

**margin-top: 1.5rem;**

**text-align: center;**

**}**

**button {**

**padding: 0.6rem 1.2rem;**

**font-size: 1rem;**

**margin: 0 0.5rem;**

**cursor: pointer;**

**border-radius: 5px;**

**border: none;**

**background-color: #007BFF;**

**color: white;**

**transition: background-color 0.3s ease;**

**}**

**button:hover {**

**background-color: #0056b3;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>Product List</h1>**

**<ul id="itemList">**

**<li>Product 1</li>**

**<li>Product 2</li>**

**<li>Product 3</li>**

**</ul>**

**<div class="buttons">**

**<button id="addBtn">Add Item</button>**

**<button id="removeBtn">Remove Item</button>**

**</div>**

**<script src="script.js"></script>**

**</body>**

**</html>**

**JAVASCRIPT:**

**document.addEventListener("DOMContentLoaded", () => {**

**const itemList = document.getElementById("itemList");**

**const addBtn = document.getElementById("addBtn");**

**const removeBtn = document.getElementById("removeBtn");**

**let itemCount = itemList.children.length;**

**// Add new item**

**addBtn.addEventListener("click", () => {**

**itemCount++;**

**const newItem = document.createElement("li");**

**newItem.textContent = `Product ${itemCount}`;**

**itemList.appendChild(newItem);**

**});**

**// Remove last item**

**removeBtn.addEventListener("click", () => {**

**if (itemList.children.length > 0) {**

**itemList.removeChild(itemList.lastElementChild);**

**itemCount--;**

**if(itemCount < 0) itemCount = 0; // safeguard**

**} else {**

**alert("No more items to remove!");**

**}**

**});**

**});**

**OUTPUT:**

**A screenshot of a product list

AI-generated content may be incorrect.**

**TASK -5:**

**CODE:**

**HTML:**

**<!DOCTYPE html>**

**<html lang="en" >**

**<head>**

**<meta charset="UTF-8" />**

**<meta name="viewport" content="width=device-width, initial-scale=1" />**

**<title>Styled Modal Popup</title>**

**<style>**

**body {**

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

**background: #f0f0f0;**

**margin: 0;**

**padding: 2rem;**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**height: 100vh;**

**}**

**button#openModalBtn {**

**padding: 1rem 2rem;**

**font-size: 1.2rem;**

**border: none;**

**background-color: #007BFF;**

**color: white;**

**border-radius: 6px;**

**cursor: pointer;**

**transition: background-color 0.3s ease;**

**}**

**button#openModalBtn:hover {**

**background-color: #0056b3;**

**}**

**/\* Modal overlay \*/**

**.modal-overlay {**

**display: none; /\* hidden by default \*/**

**position: fixed;**

**top: 0;**

**left: 0;**

**width: 100vw;**

**height: 100vh;**

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

**backdrop-filter: blur(2px);**

**z-index: 1000;**

**justify-content: center;**

**align-items: center;**

**}**

**/\* Modal container \*/**

**.modal {**

**background: white;**

**width: 90%;**

**max-width: 400px;**

**padding: 1.5rem 2rem;**

**border-radius: 10px;**

**position: relative;**

**box-shadow: 0 8px 16px rgba(0,0,0,0.25);**

**text-align: center;**

**}**

**/\* Close button \*/**

**.modal-close {**

**position: absolute;**

**top: 12px;**

**right: 12px;**

**background: transparent;**

**border: none;**

**font-size: 1.5rem;**

**font-weight: bold;**

**cursor: pointer;**

**color: #555;**

**transition: color 0.3s ease;**

**}**

**.modal-close:hover {**

**color: #000;**

**}**

**</style>**

**</head>**

**<body>**

**<button id="openModalBtn">Open Modal</button>**

**<!-- Modal structure -->**

**<div class="modal-overlay" id="modalOverlay">**

**<div class="modal" role="dialog" aria-modal="true" aria-labelledby="modalTitle">**

**<button class="modal-close" id="modalCloseBtn" aria-label="Close modal">&times;</button>**

**<h2 id="modalTitle">Modal Popup</h2>**

**<p>This is a styled modal popup. Click outside or on the &times; button to close.</p>**

**</div>**

**</div>**

**<script src="script.js"></script>**

**</body>**

**</html>**

**JAVASCRIPT:**

**document.addEventListener("DOMContentLoaded", () => {**

**const openBtn = document.getElementById("openModalBtn");**

**const modalOverlay = document.getElementById("modalOverlay");**

**const closeBtn = document.getElementById("modalCloseBtn");**

**// Function to open modal**

**function openModal() {**

**modalOverlay.style.display = "flex";**

**}**

**// Function to close modal**

**function closeModal() {**

**modalOverlay.style.display = "none";**

**}**

**// Open modal on button click**

**openBtn.addEventListener("click", openModal);**

**// Close modal on close button click**

**closeBtn.addEventListener("click", closeModal);**

**// Close modal when clicking outside modal content (overlay)**

**modalOverlay.addEventListener("click", (event) => {**

**if (event.target === modalOverlay) {**

**closeModal();**

**}**

**});**

**// Optional: close modal on Escape key press**

**document.addEventListener("keydown", (event) => {**

**if (event.key === "Escape" && modalOverlay.style.display === "flex") {**

**closeModal();**

**}**

**});**

**});**

**OUTPUT:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**\*\*\*\*\***