**Code Guide For Mile Stone 1 : Code Review**

**Local Storage - JSON data**

**0 . Members**

**[**

**{**

**"id": "M001",**

**"firstName": "John",**

**"lastName": "Doe",**

**"dateOfBirth": "1990-05-15",**

**"image": "member1.jpg"**

**},**

**{**

**"id": "M002",**

**"firstName": "Jane",**

**"lastName": "Smith",**

**"dateOfBirth": "1985-08-22",**

**"image": "member2.jpg"**

**},**

**{**

**"id": "M003",**

**"firstName": "Michael",**

**"lastName": "Johnson",**

**"dateOfBirth": "1992-11-30",**

**"image": "member3.jpg"**

**},**

**{**

**"id": "M004",**

**"firstName": "Emily",**

**"lastName": "Williams",**

**"dateOfBirth": "1995-04-10",**

**"image": "member4.jpg"**

**},**

**{**

**"id": "M005",**

**"firstName": "Nelson",**

**"lastName": "Mandella",**

**"dateOfBirth": "1965-04-17",**

**"image": "member5.jpg"**

**}**

**]**

**1.Books**

[

{

"bookId": "B001",

"bookName": "The Great Gatsby",

"dateOfPublish": "1925-04-10",

"isbn": "9780743273565",

"image":"book1.jpg"

},

{

"bookId": "B002",

"bookName": "To Kill a Mockingbird",

"dateOfPublish": "1960-07-11",

"isbn": "9780060935467",

"image":"book2.jpg"

},

{

"bookId": "B003",

"bookName": "1984",

"dateOfPublish": "1949-06-08",

"isbn": "9780451524935",

"image":"book3.jpg"

},

{

"bookId": "B004",

"bookName": "Pride and Prejudice",

"dateOfPublish": "1813-01-28",

"isbn": "9780141040349",

"image":"book4.jpg"

}

]

**2. Videos**

**[**

**{**

**"dvdId": "DVD001",**

**"dvdTitle": "The Matrix",**

**"releaseDate": "1999-03-31",**

**"productionCompany": "Warner Bros.",**

**"image":"dvd1.jpg"**

**},**

**{**

**"dvdId": "DVD002",**

**"dvdTitle": "Inception",**

**"releaseDate": "2010-07-16",**

**"productionCompany": "Legendary Pictures",**

**"image":"dvd2.jpg"**

**},**

**{**

**"dvdId": "DVD003",**

**"dvdTitle": "The Godfather",**

**"releaseDate": "1972-03-24",**

**"productionCompany": "Paramount Pictures",**

**"image":"dvd3.jpg"**

**},**

**{**

**"dvdId": "DVD004",**

**"dvdTitle": "Pulp Fiction",**

**"releaseDate": "1994-10-14",**

**"productionCompany": "Miramax Films",**

**"image":"dvd4.jpg"**

**}**

**]**

**3. Cars**

**[**

**{**

**"carId": "CAR001",**

**"carName": "Tesla Model S",**

**"yearOfManufacture": "2022",**

**"registrationNumber": "ABC1234",**

**"image": "car1.jpg"**

**},**

**{**

**"carId": "CAR002",**

**"carName": "Ford Mustang",**

**"yearOfManufacture": "2021",**

**"registrationNumber": "XYZ5678",**

**"image": "car2.jpg"**

**},**

**{**

**"carId": "CAR003",**

**"carName": "Chevrolet Camaro",**

**"yearOfManufacture": "2020",**

**"registrationNumber": "LMN9101",**

**"image": "car3.jpg"**

**},**

**{**

**"carId": "CAR004",**

**"carName": "BMW 3 Series",**

**"yearOfManufacture": "2023",**

**"registrationNumber": "PQR3456",**

**"image": "car4.jpg"**

**}**

**]**

**4. Bikes**

**[**

**{**

**"bikeId": "BIKE001",**

**"bikeName": "Harley-Davidson Street 750",**

**"yearOfManufacture": "2022",**

**"registrationNumber": "HD1234",**

**"image": "bike1.jpg"**

**},**

**{**

**"bikeId": "BIKE002",**

**"bikeName": "Yamaha YZF-R3",**

**"yearOfManufacture": "2021",**

**"registrationNumber": "YZ5678",**

**"image": "bike2.jpg"**

**},**

**{**

**"bikeId": "BIKE003",**

**"bikeName": "Kawasaki Ninja 400",**

**"yearOfManufacture": "2020",**

**"registrationNumber": "KN9101",**

**"image": "bike3.jpg"**

**},**

**{**

**"bikeId": "BIKE004",**

**"bikeName": "Ducati Panigale V2",**

**"yearOfManufacture": "2023",**

**"registrationNumber": "DP3456",**

**"image": "bike4.jpg"**

**}**

**]**

**5. Courses**

**[**

**{**

**"courseId": "C001",**

**"courseName": "Introduction to Programming",**

**"fees": "$300",**

**"duration": "8 weeks",**

**"description": "A beginner's course that covers the basics of programming, including variables, control structures, and basic algorithms.",**

**"image": "course1.jpg"**

**},**

**{**

**"courseId": "C002",**

**"courseName": "Web Development Bootcamp",**

**"fees": "$600",**

**"duration": "12 weeks",**

**"description": "An intensive course that teaches HTML, CSS, JavaScript, and modern web development practices. Includes hands-on projects.",**

**"image": "course2.jpg"**

**},**

**{**

**"courseId": "C003",**

**"courseName": "Data Science Fundamentals",**

**"fees": "$500",**

**"duration": "10 weeks",**

**"description": "This course covers data analysis, visualization, and introduction to machine learning using Python and R.",**

**"image": "course3.jpg"**

**},**

**{**

**"courseId": "C004",**

**"courseName": "Advanced Java Programming",**

**"fees": "$450",**

**"duration": "6 weeks",**

**"description": "An advanced course focusing on complex Java concepts such as multithreading, networking, and design patterns.",**

**"image": "course4.jpg"**

**}**

**]**

**6. Gym programs**

**[**

**{**

**"programId": "P001",**

**"programName": "Weight Training",**

**"fees": "$100",**

**"duration": "4 weeks",**

**"description": "A comprehensive weight training program designed to build muscle and increase strength. Includes personalized workout plans and guidance.",**

**"image": "img1.jpg"**

**},**

**{**

**"programId": "P002",**

**"programName": "Aerobics",**

**"fees": "$80",**

**"duration": "6 weeks",**

**"description": "A high-energy aerobic exercise program to improve cardiovascular fitness and endurance. Includes group classes and expert instruction.",**

**"image": "img2.jpg"**

**},**

**{**

**"programId": "P003",**

**"programName": "Yoga",**

**"fees": "$90",**

**"duration": "8 weeks",**

**"description": "A holistic yoga program focusing on flexibility, balance, and relaxation. Includes various yoga styles and personalized instruction.",**

**"image": "img3.jpg"**

**},**

**{**

**"programId": "P004",**

**"programName": "High-Intensity Interval Training (HIIT)",**

**"fees": "$110",**

**"duration": "5 weeks",**

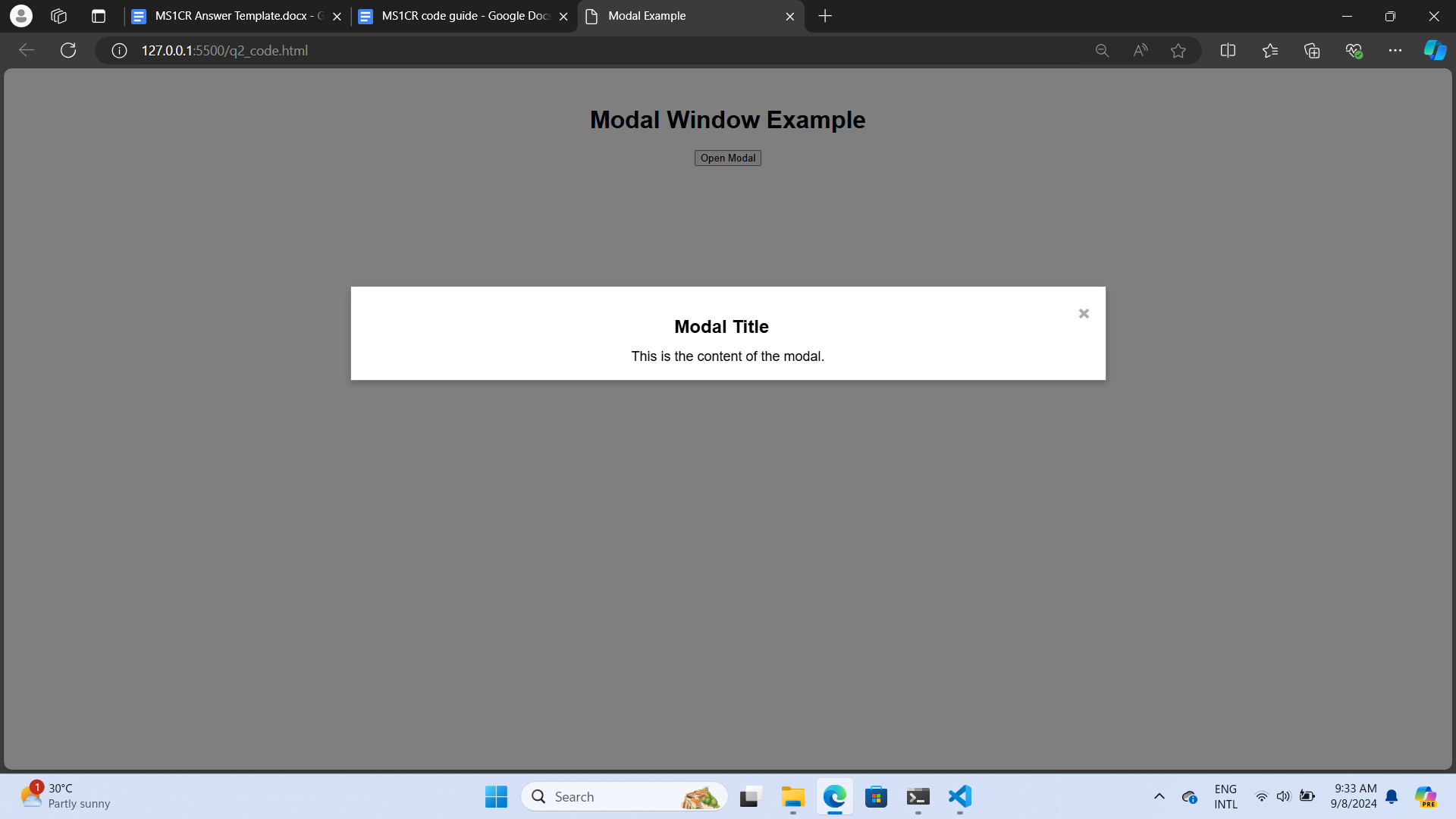
**"description": "An intense HIIT program designed to maximize calorie burn and improve overall fitness. Includes challenging workouts and recovery sessions.",**

**"image": "img4.jpg"**

**}**

**]**

**Q2. Sample Code For Modal Window**



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Modal Example</title>

<style>

/\* Basic page styling \*/

body {

font-family: Arial, sans-serif;text-align: center;margin-top: 50px;

}

/\* Modal background \*/

.modal {

display: none;background-color: rgba(0, 0, 0, 0.5);

position: fixed;z-index: 1;left: 0;top: 0;width: 100%;height: 100%;

}

/\* Modal content \*/

.modal-content {

background-color: #fff;margin: 15% auto;padding: 20px;

border: 1px solid #888;width: 50%;

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

}

/\* Close button \*/

.close-btn {

color: #aaa;float: right;font-size: 28px;font-weight: bold;cursor: pointer;

}

.close-btn:hover,

.close-btn:focus {

color: #000;text-decoration: none;

}

/\* Modal title \*/

.modal-title {

font-size: 24px;

margin-bottom: 15px;

}

/\* Modal body \*/

.modal-body {

font-size: 18px;

}

</style>

</head>

<body>

<h1>Modal Window Example</h1>

<!-- Trigger Button -->

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

<!-- Modal Structure -->

<div id="myModal" class="modal">

<div class="modal-content">

<span class="close-btn">&times;</span>

<h2 class="modal-title">Modal Title</h2>

<div class="modal-body">This is the content of the modal.</div>

</div>

</div>

<script>

// Get modal element

const modal = document.getElementById("myModal");

// Get open modal button

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

// Get close button (the "x")

const closeBtn = document.querySelector(".close-btn");

// Open modal when the button is clicked

openModalBtn.addEventListener("click", () => {

modal.style.display = "block";

});

// Close modal when the close button is clicked

closeBtn.addEventListener("click", () => {

modal.style.display = "none";

});

// Close modal when clicking anywhere outside of the modal content

window.addEventListener("click", (event) => {

if (event.target === modal) {

modal.style.display = "none";

}

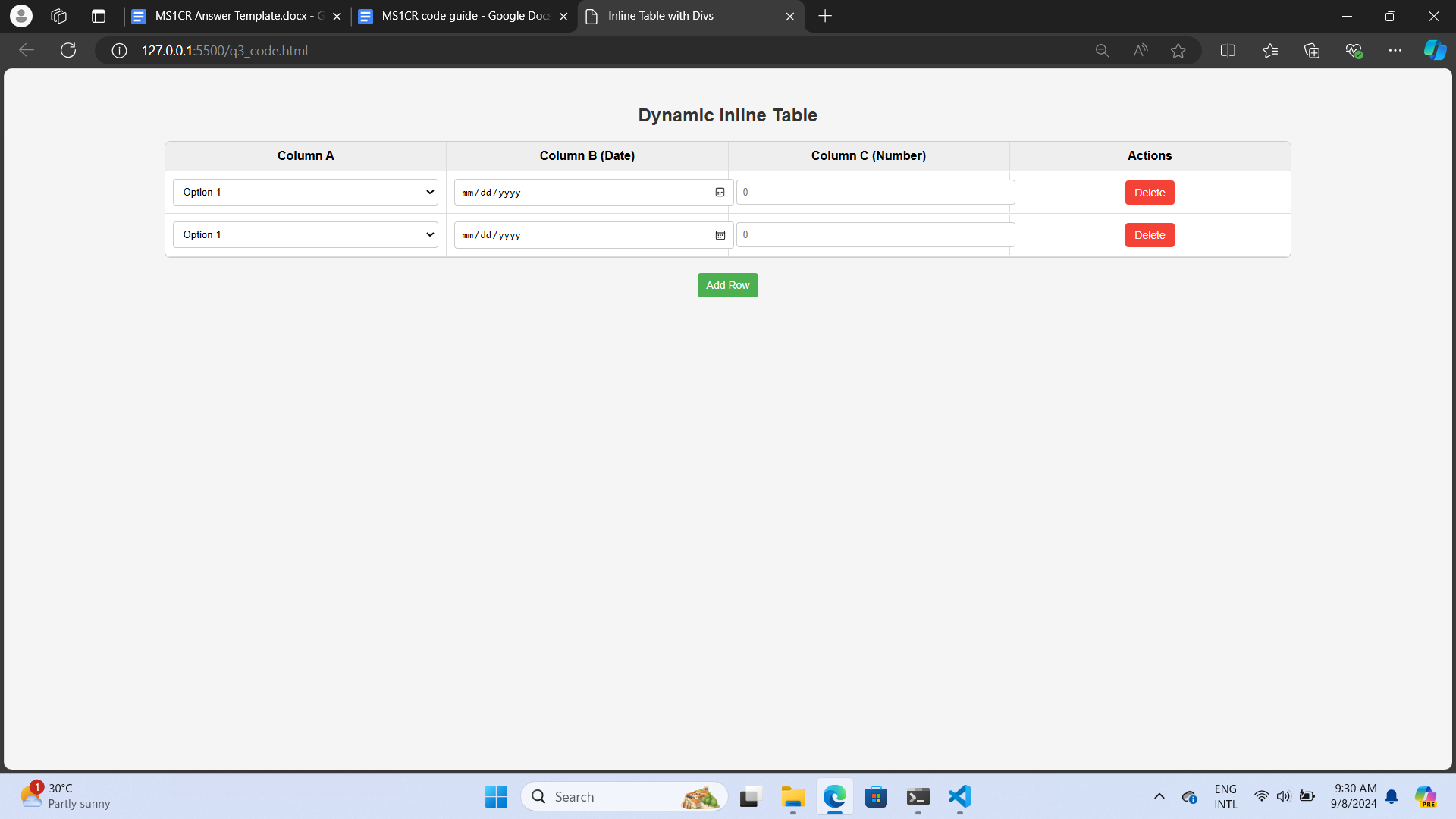
});

</script>

</body>

</html>

**Q3. Sample Code For Inline Table**



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Inline Table with Divs</title>

<style>

/\* Basic page styling \*/

body {

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

padding: 20px;

background-color: #f5f5f5;

}

h2 {

text-align: center;

color: #333;

}

/\* Table container styling \*/

#table-container {

width: 80%;

margin: 20px auto;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #fff;

overflow: hidden;

}

/\* Table row styling \*/

.table-header,

.table-row {

display: flex;

align-items: center;

border-bottom: 1px solid #ddd;

}

.table-header {

background-color: #eee;

font-weight: bold;

}

.table-cell {

flex: 1;

padding: 10px;

text-align: center;

border-right: 1px solid #ddd;

}

.table-cell:last-child {

border-right: none;

}

input,

select {

width: 100%;

padding: 8px;

border: 1px solid #ccc;

border-radius: 4px;

}

/\* Button styling \*/

button {

padding: 8px 12px;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 14px;

}

.add-btn {

display: block;

margin: 20px auto;

background-color: #4CAF50;

color: white;

}

.delete-btn {

background-color: #f44336;

color: white;

}

button:hover {

opacity: 0.8;

}

</style>

</head>

<body>

<h2>Dynamic Inline Table</h2>

<!-- Table Container -->

<div id="table-container">

<div class="table-header">

<div class="table-cell">Column A</div>

<div class="table-cell">Column B (Date)</div>

<div class="table-cell">Column C (Number)</div>

<div class="table-cell">Actions</div>

</div>

<!-- First row -->

<div class="table-row">

<div class="table-cell">

<select>

<option value="option1">Option 1</option>

<option value="option2">Option 2</option>

<option value="option3">Option 3</option>

</select>

</div>

<div class="table-cell"><input type="date" /></div>

<div class="table-cell"><input type="number" placeholder="0" /></div>

<div class="table-cell"><button class="delete-btn">Delete</button></div>

</div>

</div>

<!-- Add button -->

<button id="add-row-btn" class="add-btn">Add Row</button>

<script>

// Get references to the add button and table container

const addRowBtn = document.getElementById("add-row-btn");

const tableContainer = document.getElementById("table-container");

// Add event listener to add a new row

addRowBtn.addEventListener("click", () => {

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

newRow.classList.add("table-row");

// Add columns to the new row

newRow.innerHTML = `

<div class="table-cell">

<select>

<option value="option1">Option 1</option>

<option value="option2">Option 2</option>

<option value="option3">Option 3</option>

</select>

</div>

<div class="table-cell">

<input type="date" />

</div>

<div class="table-cell">

<input type="number" placeholder="0" />

</div>

<div class="table-cell">

<button class="delete-btn">Delete</button>

</div>

`;

// Append the new row to the table container

tableContainer.appendChild(newRow);

// Attach event listener to the new delete button

newRow.querySelector(".delete-btn").addEventListener("click", () => {

newRow.remove();

});

});

// Attach event listener to existing delete button

document.querySelectorAll(".delete-btn").forEach((button) => {

button.addEventListener("click", (event) => {

event.target.closest(".table-row").remove();

});

});

</script>

</body>

</html>

**Code to extract URL Parameter**

// Function to get the value of a query parameter by name

function getQueryParam(param) {

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

return urlParams.get(param);

}

// Extract videoId from the URL

const videoId = getQueryParam('videoId');

console.log(videoId); // Output the videoId value

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Video Lends Page</title>

<script>

// Function to get the value of a query parameter by name

function getQueryParam(param) {

// Get the query parameters from the URL

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

// Return the value associated with the given parameter name

return urlParams.get(param);

}

// Function to display the videoId on the page

function displayVideoId() {

// Extract videoId from the URL

const videoId = getQueryParam('videoId');

// Get the element where the videoId will be displayed

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

// Check if videoId is present in the URL

if (videoId) {

videoIdDisplay.textContent = "Extracted videoId: " + videoId;

} else {

videoIdDisplay.textContent = "No videoId found in the URL.";

}

}

// Call the displayVideoId function when the page loads

window.onload = displayVideoId;

</script>

</head>

<body>

<h1>Welcome to the Video Lends Page</h1>

<p>This page demonstrates how to extract the videoId from the URL.</p>

<!-- Placeholder for displaying the extracted videoId -->

<p id="videoIdDisplay"></p>

</body>

</html>