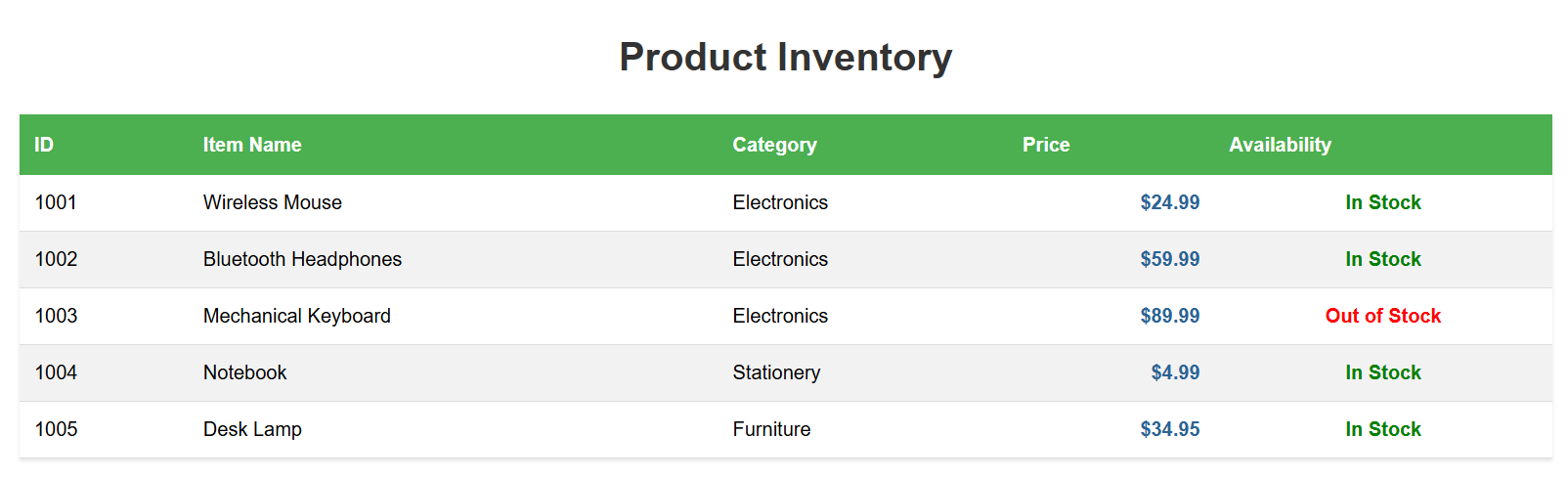
**Q1: Can you create a simple HTML table to display a list of items with basic styling using CSS?**

****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Styled Item List</title>

<style>

body {

font-family: Arial, sans-serif;

line-height: 1.6;

margin: 20px;

}

h1 {

color: #333;

text-align: center;

}

.item-table {

width: 100%;

border-collapse: collapse;

margin: 20px 0;

box-shadow: 0 2px 3px rgba(0,0,0,0.1);

}

.item-table th {

background-color: #4CAF50;

color: white;

padding: 12px;

text-align: left;

}

.item-table td {

padding: 10px 12px;

border-bottom: 1px solid #ddd;

}

.item-table tr:nth-child(even) {

background-color: #f2f2f2;

}

.item-table tr:hover {

background-color: #e9e9e9;

}

.price {

text-align: right;

font-weight: bold;

color: #2a6496;

}

.availability {

text-align: center;

}

.in-stock {

color: green;

font-weight: bold;

}

.out-of-stock {

color: red;

font-weight: bold;

}

</style>

</head>

<body>

<h1>Product Inventory</h1>

<table class="item-table">

<thead>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Category</th>

<th>Price</th>

<th>Availability</th>

</tr>

</thead>

<tbody>

<tr>

<td>1001</td>

<td>Wireless Mouse</td>

<td>Electronics</td>

<td class="price">$24.99</td>

<td class="availability in-stock">In Stock</td>

</tr>

<tr>

<td>1002</td>

<td>Bluetooth Headphones</td>

<td>Electronics</td>

<td class="price">$59.99</td>

<td class="availability in-stock">In Stock</td>

</tr>

<tr>

<td>1003</td>

<td>Mechanical Keyboard</td>

<td>Electronics</td>

<td class="price">$89.99</td>

<td class="availability out-of-stock">Out of Stock</td>

</tr>

<tr>

<td>1004</td>

<td>Notebook</td>

<td>Stationery</td>

<td class="price">$4.99</td>

<td class="availability in-stock">In Stock</td>

</tr>

<tr>

<td>1005</td>

<td>Desk Lamp</td>

<td>Furniture</td>

<td class="price">$34.95</td>

<td class="availability in-stock">In Stock</td>

</tr>

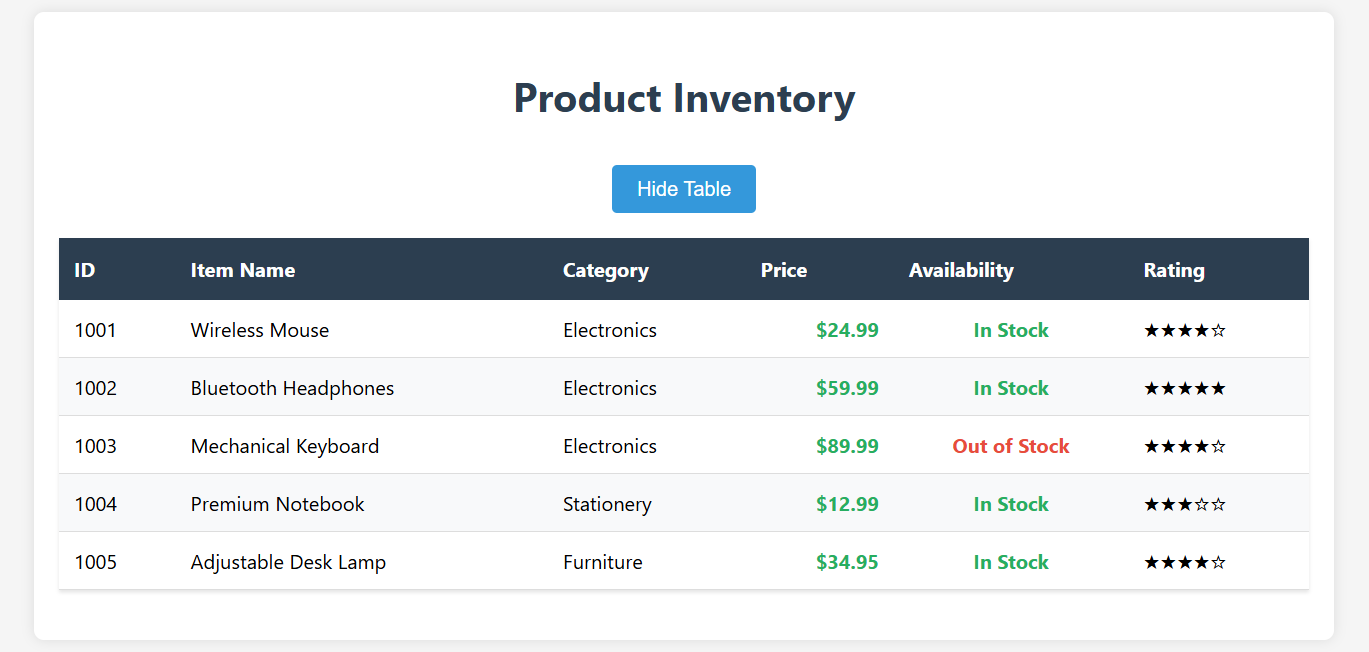
</tbody>

</table>

</body>

</html>

**Q2: Build a styled HTML table with CSS and add a JavaScript toggle button to show or hide the data.**

****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Toggleable Product Table</title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

line-height: 1.6;

margin: 20px;

background-color: #f5f5f5;

}

.container {

max-width: 1000px;

margin: 0 auto;

background: white;

padding: 20px;

border-radius: 8px;

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

}

h1 {

color: #2c3e50;

text-align: center;

margin-bottom: 30px;

}

.toggle-btn {

display: block;

margin: 0 auto 20px;

padding: 10px 20px;

background-color: #3498db;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

transition: background-color 0.3s;

}

.toggle-btn:hover {

background-color: #2980b9;

}

.item-table {

width: 100%;

border-collapse: collapse;

margin: 20px 0;

box-shadow: 0 2px 3px rgba(0,0,0,0.1);

transition: opacity 0.3s ease;

}

.item-table th {

background-color: #2c3e50;

color: white;

padding: 12px;

text-align: left;

}

.item-table td {

padding: 10px 12px;

border-bottom: 1px solid #ddd;

}

.item-table tr:nth-child(even) {

background-color: #f8f9fa;

}

.item-table tr:hover {

background-color: #e9f7fe;

}

.price {

text-align: right;

font-weight: bold;

color: #27ae60;

}

.availability {

text-align: center;

}

.in-stock {

color: #27ae60;

font-weight: bold;

}

.out-of-stock {

color: #e74c3c;

font-weight: bold;

}

.hidden {

display: none;

}

</style>

</head>

<body>

<div class="container">

<h1>Product Inventory</h1>

<button id="toggleButton" class="toggle-btn">Hide Table</button>

<table id="productTable" class="item-table">

<thead>

<tr>

<th>ID</th>

<th>Item Name</th>

<th>Category</th>

<th>Price</th>

<th>Availability</th>

<th>Rating</th>

</tr>

</thead>

<tbody>

<tr>

<td>1001</td>

<td>Wireless Mouse</td>

<td>Electronics</td>

<td class="price">$24.99</td>

<td class="availability in-stock">In Stock</td>

<td>★★★★☆</td>

</tr>

<tr>

<td>1002</td>

<td>Bluetooth Headphones</td>

<td>Electronics</td>

<td class="price">$59.99</td>

<td class="availability in-stock">In Stock</td>

<td>★★★★★</td>

</tr>

<tr>

<td>1003</td>

<td>Mechanical Keyboard</td>

<td>Electronics</td>

<td class="price">$89.99</td>

<td class="availability out-of-stock">Out of Stock</td>

<td>★★★★☆</td>

</tr>

<tr>

<td>1004</td>

<td>Premium Notebook</td>

<td>Stationery</td>

<td class="price">$12.99</td>

<td class="availability in-stock">In Stock</td>

<td>★★★☆☆</td>

</tr>

<tr>

<td>1005</td>

<td>Adjustable Desk Lamp</td>

<td>Furniture</td>

<td class="price">$34.95</td>

<td class="availability in-stock">In Stock</td>

<td>★★★★☆</td>

</tr>

</tbody>

</table>

</div>

<script>

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

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

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

toggleButton.addEventListener('click', function() {

if (productTable.classList.contains('hidden')) {

productTable.classList.remove('hidden');

toggleButton.textContent = 'Hide Table';

} else {

productTable.classList.add('hidden');

toggleButton.textContent = 'Show Table';

}

});

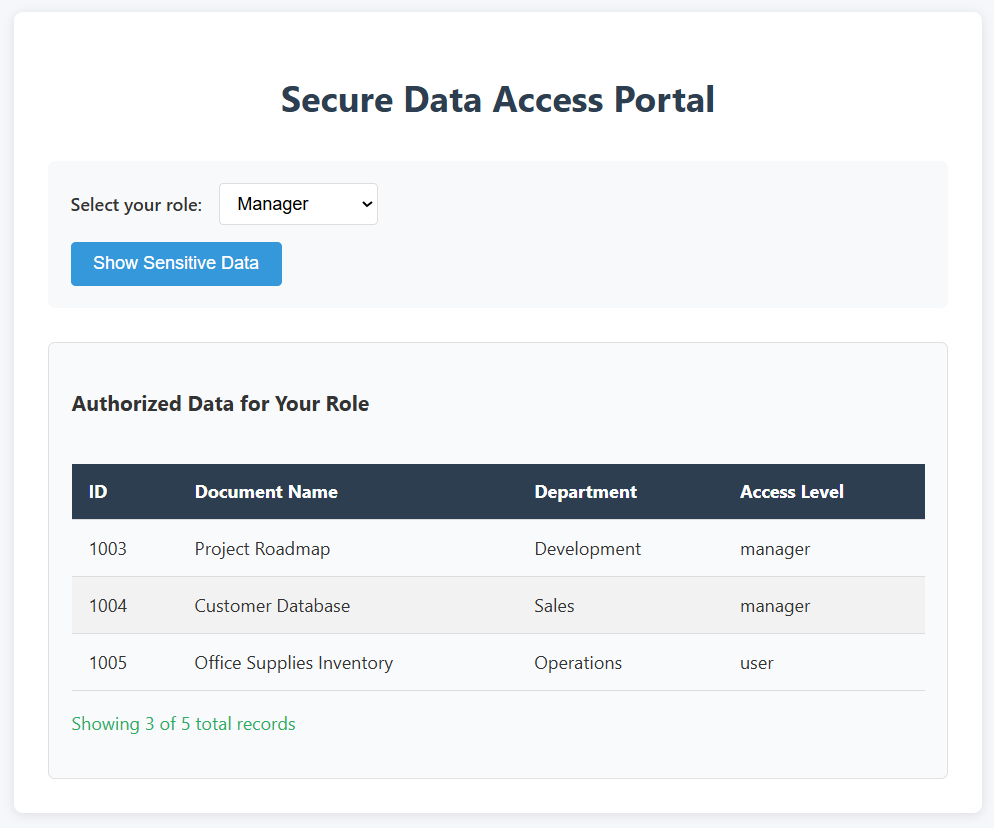
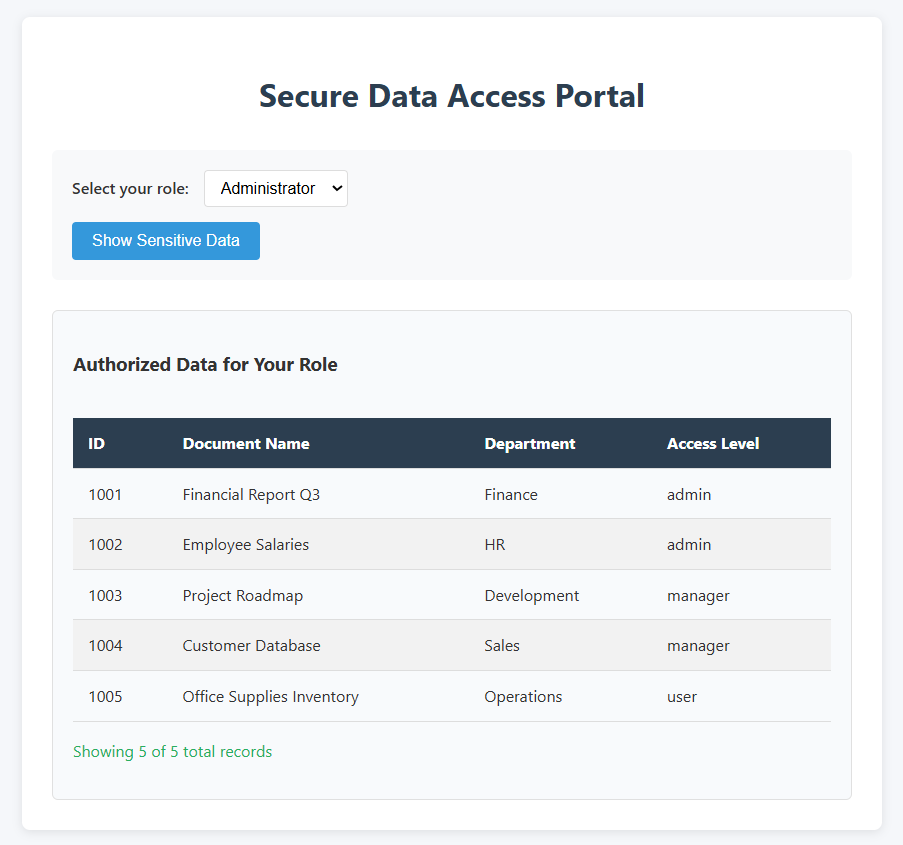
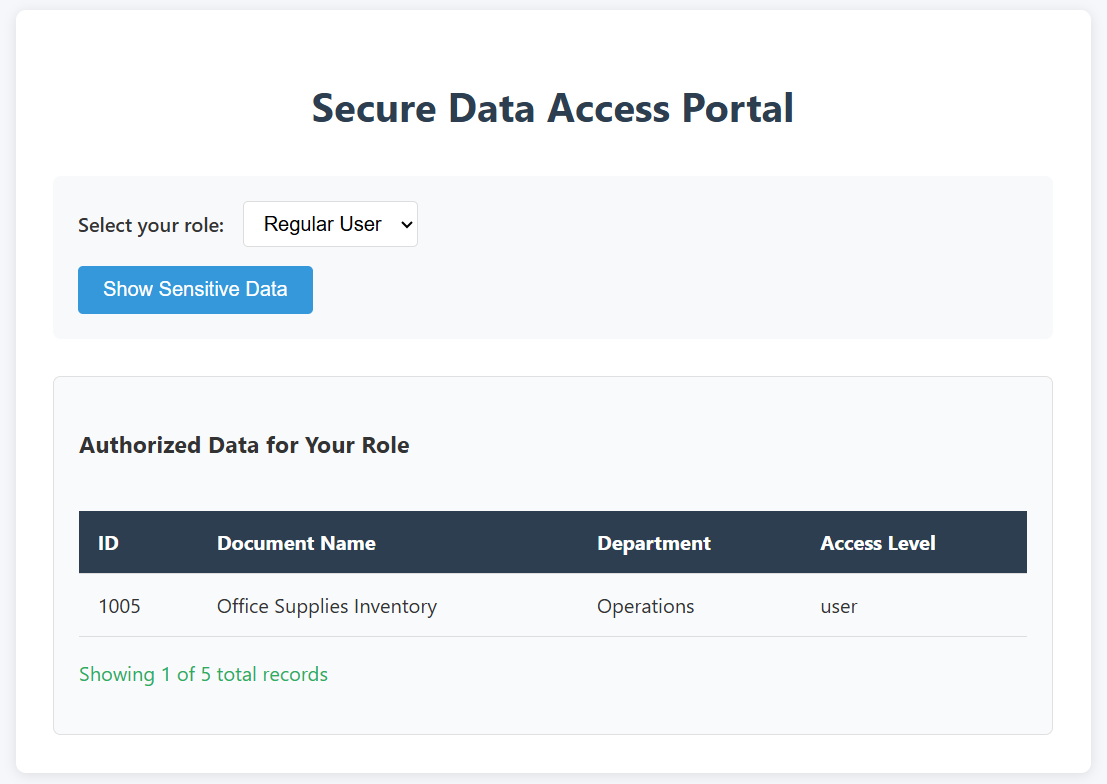
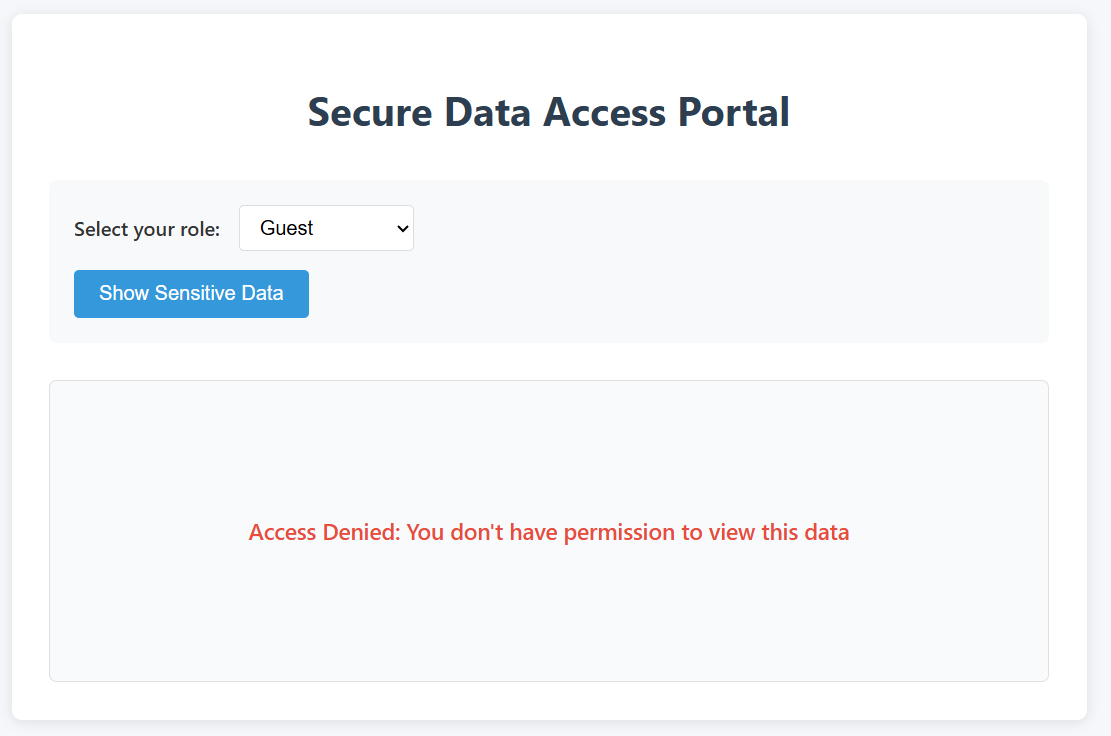
});

</script>

</body>

</html>

**Q3: Create a responsive HTML, CSS, and JavaScript component that only displays data when a user with permission clicks a 'Show Data' button. Include a message if the user is not authorized.**

****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Permission-Based Data Display</title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

line-height: 1.6;

margin: 0;

padding: 20px;

background-color: #f5f7fa;

color: #333;

}

.container {

max-width: 800px;

margin: 0 auto;

background: white;

padding: 30px;

border-radius: 8px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

}

h1 {

color: #2c3e50;

text-align: center;

margin-bottom: 30px;

}

.auth-controls {

display: flex;

flex-direction: column;

gap: 15px;

margin-bottom: 30px;

padding: 20px;

background-color: #f8f9fa;

border-radius: 6px;

}

.role-selector {

display: flex;

gap: 15px;

align-items: center;

}

label {

font-weight: 600;

}

select {

padding: 8px 12px;

border: 1px solid #ddd;

border-radius: 4px;

font-size: 16px;

}

button {

padding: 10px 20px;

background-color: #3498db;

color: white;

border: none;

border-radius: 4px;

cursor: pointer;

font-size: 16px;

transition: background-color 0.3s;

align-self: flex-start;

}

button:hover {

background-color: #2980b9;

}

.data-container {

border: 1px solid #e0e0e0;

border-radius: 6px;

padding: 20px;

margin-top: 20px;

background-color: #f8fafc;

min-height: 200px;

display: flex;

flex-direction: column;

justify-content: center;

}

.message {

text-align: center;

font-size: 18px;

color: #666;

}

.error {

color: #e74c3c;

font-weight: 600;

}

.success {

color: #27ae60;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

th, td {

padding: 12px 15px;

text-align: left;

border-bottom: 1px solid #ddd;

}

th {

background-color: #2c3e50;

color: white;

}

tr:nth-child(even) {

background-color: #f2f2f2;

}

tr:hover {

background-color: #e9f7fe;

}

@media (max-width: 600px) {

.container {

padding: 15px;

}

.auth-controls {

padding: 15px;

}

.role-selector {

flex-direction: column;

align-items: flex-start;

}

th, td {

padding: 8px 10px;

font-size: 14px;

}

}

</style>

</head>

<body>

<div class="container">

<h1>Secure Data Access Portal</h1>

<div class="auth-controls">

<div class="role-selector">

<label for="userRole">Select your role:</label>

<select id="userRole">

<option value="guest">Guest</option>

<option value="user">Regular User</option>

<option value="admin">Administrator</option>

<option value="manager">Manager</option>

</select>

</div>

<button id="showDataBtn">Show Sensitive Data</button>

</div>

<div class="data-container" id="dataContainer">

<p class="message">Click the button above to view data (if authorized)</p>

</div>

</div>

<script>

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

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

const userRoleSelect = document.getElementById('userRole');

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

// Sample sensitive data

const sensitiveData = [

{ id: 1001, name: "Financial Report Q3", department: "Finance", accessLevel: "admin" },

{ id: 1002, name: "Employee Salaries", department: "HR", accessLevel: "admin" },

{ id: 1003, name: "Project Roadmap", department: "Development", accessLevel: "manager" },

{ id: 1004, name: "Customer Database", department: "Sales", accessLevel: "manager" },

{ id: 1005, name: "Office Supplies Inventory", department: "Operations", accessLevel: "user" }

];

showDataBtn.addEventListener('click', function() {

const userRole = userRoleSelect.value;

dataContainer.innerHTML = '';

// Check permissions

if (userRole === 'guest') {

showErrorMessage("Access Denied: You don't have permission to view this data");

return;

}

// Filter data based on user role

let authorizedData = sensitiveData.filter(item => {

if (userRole === 'admin') return true;

if (userRole === 'manager') return item.accessLevel !== 'admin';

if (userRole === 'user') return item.accessLevel === 'user';

return false;

});

if (authorizedData.length === 0) {

showSuccessMessage("No data available for your access level");

return;

}

displayDataTable(authorizedData);

});

function showErrorMessage(message) {

dataContainer.innerHTML = `<p class="message error">${message}</p>`;

}

function showSuccessMessage(message) {

dataContainer.innerHTML = `<p class="message success">${message}</p>`;

}

function displayDataTable(data) {

let html = `

<h3>Authorized Data for Your Role</h3>

<table>

<thead>

<tr>

<th>ID</th>

<th>Document Name</th>

<th>Department</th>

<th>Access Level</th>

</tr>

</thead>

<tbody>

`;

data.forEach(item => {

html += `

<tr>

<td>${item.id}</td>

<td>${item.name}</td>

<td>${item.department}</td>

<td>${item.accessLevel}</td>

</tr>

`;

});

html += `

</tbody>

</table>

<p class="success">Showing ${data.length} of ${sensitiveData.length} total records</p>

`;

dataContainer.innerHTML = html;

}

});

</script>

</body>

</html>