<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Groups Management</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

}

h1 {

text-align: center;

}

.search-container {

text-align: center;

margin-bottom: 20px;

}

.search-input {

padding: 10px;

width: 200px;

}

.search-button, .create-button, .edit-button {

padding: 10px 20px;

background-color: #007BFF; /\* Blue color \*/

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.edit-button:hover, .create-button:hover, .search-button:hover {

background-color: #0056b3; /\* Darker blue on hover \*/

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

th, td {

border: 1px solid #ddd;

padding: 10px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

/\* Modal styles \*/

.modal {

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgb(0,0,0);

background-color: rgba(0,0,0,0.4);

}

.modal-content {

background-color: #fefefe;

margin: 15% auto;

padding: 20px;

border: 1px solid #888;

width: 50%;

}

.close {

color: #aaa;

float: right;

font-size: 28px;

font-weight: bold;

}

.close:hover,

.close:focus {

color: black;

text-decoration: none;

cursor: pointer;

}

</style>

</head>

<body>

<h1>Groups Management</h1>

<div class="search-container">

<input type="text" class="search-input" id="groupSearch" placeholder="Search for groups..." oninput="searchGroups()">

<button class="search-button" onclick="searchGroups()">Search</button>

<button class="create-button" onclick="openCreateGroupModal()">Create Custom Group</button>

</div>

<div id="groupsResultsContainer">

<table>

<thead>

<tr>

<th>Group Name</th>

<th>Type</th>

<th>Action</th>

</tr>

</thead>

<tbody id="groupsResults">

<!-• Groups will be inserted here -->

</tbody>

</table>

</div>

<!-• Modal for Creating Custom Groups -->

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

<div class="modal-content">

<span class="close" onclick="closeCreateGroupModal()">&times;</span>

<h2>Create Custom Group</h2>

<input type="text" id="newGroupName" placeholder="Enter new group name" style="width: 100%; padding: 10px; margin-bottom: 10px;">

<div class="group-list">

<select id="availableRoles" multiple>

<option>Admin</option>

<option>Editor</option>

<option>Viewer</option>

<option>Subscriber</option>

<option>Contributor</option>

</select>

<div>

<button class="add-button" onclick="addRole()">Add &gt;&gt;</button>

<button class="add-button" onclick="removeRole()">Remove &lt;&lt;</button>

</div>

<select id="newGroupRoles" multiple>

<!-• Roles for the new group will be populated here -->

</select>

</div>

<button class="search-button" onclick="createCustomGroup()">Create Group</button>

</div>

</div>

<script>

// Sample groups data

const allGroups = [

{ name: 'Admin', type: 'Default' },

{ name: 'Editor', type: 'Default' },

{ name: 'Viewer', type: 'Default' },

{ name: 'Subscriber', type: 'Default' },

{ name: 'Contributor', type: 'Default' },

{ name: 'Custom Group 1', type: 'Custom' },

{ name: 'Custom Group 2', type: 'Custom' },

{ name: 'Custom Group 3', type: 'Custom' },

];

function displayGroups(groupList) {

const resultsTable = document.getElementById('groupsResults');

resultsTable.innerHTML = ''; // Clear previous results

groupList.forEach(group => {

const row = document.createElement('tr');

row.innerHTML = `

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

<td>${group.type}</td>

<td>${group.type === 'Custom' ? `<button class="edit-button" onclick="editGroup('${group.name}')">Edit</button>` : ''}</td>

`;

resultsTable.appendChild(row);

});

}

function searchGroups() {

const searchValue = document.getElementById('groupSearch').value.toLowerCase();

const filteredGroups = allGroups.filter(group =>

group.name.toLowerCase().includes(searchValue)

);

displayGroups(filteredGroups);

}

function editGroup(groupName) {

alert(`Edit functionality for ${groupName} goes here.`);

// Implement your edit functionality here.

}

function openCreateGroupModal() {

document.getElementById('createGroupModal').style.display = 'block';

}

function closeCreateGroupModal() {

document.getElementById('createGroupModal').style.display = 'none';

document.getElementById('newGroupName').value = ''; // Clear input field

document.getElementById('newGroupRoles').innerHTML = ''; // Clear roles

}

function addRole() {

const availableRolesSelect = document.getElementById('availableRoles');

const newGroupRolesSelect = document.getElementById('newGroupRoles');

Array.from(availableRolesSelect.selectedOptions).forEach(option => {

newGroupRolesSelect.appendChild(option);

});

}

function removeRole() {

const newGroupRolesSelect = document.getElementById('newGroupRoles');

const availableRolesSelect = document.getElementById('availableRoles');

Array.from(newGroupRolesSelect.selectedOptions).forEach(option => {

availableRolesSelect.appendChild(option);

});

}

function createCustomGroup() {

const groupName = document.getElementById('newGroupName').value;

const newGroupRolesSelect = document.getElementById('newGroupRoles');

const roles = Array.from(newGroupRolesSelect.options).map(option => option.value);

if (groupName.trim() === '') {

alert('Group name cannot be empty.');

return;

}

// Add the new group to the allGroups array

allGroups.push({ name: groupName, type: 'Custom' });

alert(`Custom group created: ${groupName} with roles: ${roles.join(', ')}`);

closeCreateGroupModal();

displayGroups(allGroups); // Refresh the displayed groups

}

// Display all groups on page load

window.onload = () => {

displayGroups(allGroups);

};

</script>

</body>

</html>