<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Management Utility</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 {

padding: 10px 20px;

}

table {

width: 100%;

border-collapse: collapse;

}

th, td {

border: 1px solid #ddd;

padding: 10px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

.edit-button, .create-button {

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

color: white;

border: none;

padding: 5px 10px;

cursor: pointer;

border-radius: 5px;

}

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

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

}

/\* 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;

}

.group-list {

display: flex;

justify-content: space-between;

align-items: center;

}

.group-list select {

width: 45%;

height: 100px;

}

.add-button {

margin: 0 10px; /\* Space around the buttons \*/

padding: 5px 10px;

background-color: #28a745; /\* Green color \*/

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.add-button:hover {

background-color: #218838; /\* Darker green on hover \*/

}

</style>

</head>

<body>

<h1>User Management Utility</h1>

<div class="search-container">

<input type="text" class="search-input" id="userSearch" placeholder="Search for users..." oninput="searchUsers()">

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

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

<button class="create-button" onclick="openGroupsPage()">Groups</button>

</div>

<div id="searchResultsContainer">

<table>

<thead>

<tr>

<th>User ID</th>

<th>Name</th>

<th>Email</th>

<th>Action</th>

</tr>

</thead>

<tbody id="searchResults">

<!-• Search results will be inserted here -->

</tbody>

</table>

</div>

<!-• Modal for Editing User -->

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

<div class="modal-content">

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

<h2>Edit User Groups</h2>

<div class="group-list">

<select id="allGroups" multiple>

<!-• Groups will be populated here -->

</select>

<div>

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

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

</div>

<select id="userGroups" multiple>

<!-• User's groups will be populated here -->

</select>

</div>

<button class="search-button" onclick="saveUserGroups()">Save</button>

</div>

</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>

const users = [

{ id: 1, name: 'John Doe', email: 'john@example.com', groups: [] },

{ id: 2, name: 'Jane Smith', email: 'jane@example.com', groups: [] },

{ id: 3, name: 'Alice Johnson', email: 'alice@example.com', groups: [] },

{ id: 4, name: 'Bob Brown', email: 'bob@example.com', groups: [] },

];

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 displayUsers(userList) {

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

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

userList.forEach(user => {

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

row.innerHTML = `

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

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

<td>${user.email}</td>

<td><button class="edit-button" onclick="editUser(${user.id})">Edit</button></td>

`;

resultsTable.appendChild(row);

});

}

function searchUsers() {

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

const filteredUsers = users.filter(user =>

user.name.toLowerCase().includes(searchValue) ||

user.email.toLowerCase().includes(searchValue)

);

displayUsers(filteredUsers);

}

function editUser(userId) {

const user = users.find(u => u.id === userId);

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

populateGroups(user);

}

function populateGroups(user) {

const allGroupsSelect = document.getElementById('allGroups');

const userGroupsSelect = document.getElementById('userGroups');

// Clear previous options

allGroupsSelect.innerHTML = '';

userGroupsSelect.innerHTML = '';

// Populate all groups

allGroups.forEach(group => {

const option = document.createElement('option');

option.value = group.name;

option.textContent = group.name;

allGroupsSelect.appendChild(option);

});

// Populate user's groups

user.groups.forEach(group => {

const option = document.createElement('option');

option.value = group;

option.textContent = group;

userGroupsSelect.appendChild(option);

});

}

function addGroup() {

const allGroupsSelect = document.getElementById('allGroups');

const userGroupsSelect = document.getElementById('userGroups');

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

userGroupsSelect.appendChild(option);

});

}

function removeGroup() {

const userGroupsSelect = document.getElementById('userGroups');

const allGroupsSelect = document.getElementById('allGroups');

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

allGroupsSelect.appendChild(option);

});

}

function saveUserGroups() {

const userId = parseInt(document.querySelector('.edit-button').onclick.toString().match(/\d+/)[0]);

const user = users.find(u => u.id === userId);

const userGroupsSelect = document.getElementById('userGroups');

user.groups = Array.from(userGroupsSelect.options).map(option => option.value);

alert(`User groups saved for ${user.name}: ${user.groups.join(', ')}`);

closeModal();

}

function closeModal() {

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

}

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;

}

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

closeCreateGroupModal();

}

function openGroupsPage() {

window.open("groups.html", "\_self");

}

// Display all users on page load

window.onload = () => {

displayUsers(users);

};

</script>

</body>

</html>