<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>To-Do List Website</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" integrity="sha512-Fo3rlrZj/k7ujTnHqT+Z2+5b1OZ4z2k4Tv6SP3o52rrw+lZ/zm+w0Rd4C72BCq8GhK7G7hW5NTg5L8ZmQDl/cA==" crossorigin="anonymous" referrerpolicy="no-referrer" />

    <style>

        body {

            font-family: Arial, sans-serif;

            background: #121212;

            color: #e0e0e0;

            margin: 0;

            padding: 0;

            display: flex;

            flex-direction: column;

            align-items: center;

        }

        .container {

            max-width: 800px;

            width: 100%;

            margin: 20px auto;

            padding: 20px;

            background: #1e1e1e;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);

            border-radius: 8px;

        }

        .header {

            text-align: center;

            padding: 20px;

            background: #333;

            color: #fff;

            border-radius: 8px 8px 0 0;

        }

        .header h1 {

            margin: 0;

            font-size: 2rem;

        }

        .auth-container {

            display: flex;

            justify-content: space-around;

            margin: 20px 0;

        }

        .auth-container div {

            width: 45%;

        }

        .auth-container input, .auth-container button {

            width: 100%;

            padding: 10px;

            margin: 10px 0;

            box-sizing: border-box;

            background: #333;

            color: #e0e0e0;

            border: 1px solid #444;

            border-radius: 4px;

        }

        .auth-container button {

            background: #6200ea;

            border: none;

            cursor: pointer;

            transition: background 0.3s;

        }

        .auth-container button:hover {

            background: #3700b3;

        }

        .todo-container {

            text-align: center;

        }

        .todo-container input, .todo-container textarea {

            width: 100%;

            padding: 10px;

            margin: 10px 0;

            box-sizing: border-box;

            background: #333;

            color: #e0e0e0;

            border: 1px solid #444;

            border-radius: 4px;

        }

        .todo-container ul {

            list-style: none;

            padding: 0;

        }

        .todo-container ul li {

            background: #2c2c2c;

            padding: 10px;

            margin: 10px 0;

            display: flex;

            justify-content: space-between;

            align-items: center;

            border-radius: 4px;

        }

        .todo-container ul li.dragging {

            opacity: 0.5;

        }

        .todo-container ul li .actions {

            display: flex;

            gap: 10px;

        }

        .todo-container ul li .actions .fa {

            cursor: pointer;

            color: #e0e0e0;

        }

    </style>

</head>

<body>

    <div class="header">

        <h1>To-Do List Website</h1>

    </div>

    <div class="container">

        <!-- Registration & Login -->

        <div class="auth-container">

            <div class="register-container">

                <h2>Register</h2>

                <input type="text" id="register-username" placeholder="Username" required>

                <input type="password" id="register-password" placeholder="Password" required>

                <button onclick="register()">Register</button>

            </div>

            <div class="login-container">

                <h2>Login</h2>

                <input type="text" id="login-username" placeholder="Username" required>

                <input type="password" id="login-password" placeholder="Password" required>

                <button onclick="login()">Login</button>

            </div>

        </div>

        <!-- To-Do List -->

        <div class="todo-container">

            <h2>To-Do List</h2>

            <input type="text" id="task-title" placeholder="Task Title">

            <textarea id="task-desc" placeholder="Task Description"></textarea>

            <button onclick="addTask()">Add Task</button>

            <ul id="task-list">

                <!-- Tasks will appear here -->

            </ul>

        </div>

    </div>

    <script>

        let tasks = [];

        function register() {

            const username = document.getElementById('register-username').value;

            const password = document.getElementById('register-password').value;

            alert(`User ${username} registered successfully!`);

            // Save user data to localStorage or database (not implemented)

        }

        function login() {

            const username = document.getElementById('login-username').value;

            const password = document.getElementById('login-password').value;

            alert(`User ${username} logged in successfully!`);

            // Authenticate user (not implemented)

        }

        function addTask() {

            const title = document.getElementById('task-title').value;

            const desc = document.getElementById('task-desc').value;

            const task = { title, desc };

            tasks.push(task);

            renderTasks();

            clearTaskInputs();

        }

        function renderTasks() {

            const taskList = document.getElementById('task-list');

            taskList.innerHTML = '';

            tasks.forEach((task, index) => {

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

                li.draggable = true;

                li.innerHTML = `

                    <div>

                        <h3>${task.title}</h3>

                        <p>${task.desc}</p>

                    </div>

                    <div class="actions">

                        <i class="fa fa-edit" onclick="editTask(${index})"></i>

                        <i class="fa fa-trash" onclick="deleteTask(${index})"></i>

                        <i class="fa fa-check" onclick="completeTask(${index})"></i>

                    </div>

                `;

                li.addEventListener('dragstart', () => li.classList.add('dragging'));

                li.addEventListener('dragend', () => li.classList.remove('dragging'));

                taskList.appendChild(li);

            });

            enableDragAndDrop();

        }

        function editTask(index) {

            const task = tasks[index];

            document.getElementById('task-title').value = task.title;

            document.getElementById('task-desc').value = task.desc;

            deleteTask(index);

        }

        function deleteTask(index) {

            tasks.splice(index, 1);

            renderTasks();

        }

        function completeTask(index) {

            alert(`Task "${tasks[index].title}" marked as completed!`);

            // Implement task completion logic

        }

        function clearTaskInputs() {

            document.getElementById('task-title').value = '';

            document.getElementById('task-desc').value = '';

        }

        function enableDragAndDrop() {

            const taskList = document.getElementById('task-list');

            const draggables = document.querySelectorAll('.todo-container ul li');

            draggables.forEach(draggable => {

                draggable.addEventListener('dragover', e => {

                    e.preventDefault();

                    const afterElement = getDragAfterElement(taskList, e.clientY);

                    if (afterElement == null) {

                        taskList.appendChild(draggable);

                    } else {

                        taskList.insertBefore(draggable, afterElement);

                    }

                });

            });

        }

        function getDragAfterElement(container, y) {

            const draggableElements = [...container.querySelectorAll('.todo-container ul li:not(.dragging)')];

            return draggableElements.reduce((closest, child) => {

                const box = child.getBoundingClientRect();

                const offset = y - box.top - box.height / 2;

                if (offset < 0 && offset > closest.offset) {

                    return { offset: offset, element: child };

                } else {

                    return closest;

                }

            }, { offset: Number.NEGATIVE\_INFINITY }).element;

        }

    </script>

</body>

</html>