

## Tugas Membuat Sistem Reservasi Ruangan Kelas

### 1. Index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Room Reservation</title>
7    <link rel="stylesheet" href="styles.css">
8    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
9  </head>
10 <body>
11   <video autoplay muted loop id="bg-video">
12     <source src="background.mp4" type="video/mp4">
13     Your browser does not support HTML5 video.
14   </video>
15
16   <div class="container">
17     <h1 class="text-center">Room Reservation System</h1>
18
19     <!-- Form untuk reservasi -->
20     <form id="reservation-form">
21       <div class="form-group">
22         <label for="roomId">Room ID:</label>
23         <input type="number" id="roomId" class="form-control" required>
24       </div>
25       <div class="form-group">
26         <label for="date">Date:</label>
27         <input type="date" id="date" class="form-control" required>
28       </div>
29       <div class="form-group">
30         <label for="time">Time:</label>
31         <input type="time" id="time" class="form-control" required>
32       </div>
33       <div class="form-group">
34         <label for="name">Name:</label>
35         <input type="text" id="name" class="form-control" required>
36       </div>
37       <button type="button" class="btn btn-primary" onclick="handleAddReservation()">Add Reservation</button>
38     </form>
39
40     <hr>
41
42     <!-- Daftar ruangan -->
43     <div id="room-list"></div>
44
45     <hr>
46
47     <!-- Tabel untuk daftar reservasi -->
48     <div id="reservation-list">
49       <h3>Reservation List</h3>
50       <table class="table table-bordered table-striped">
51         <thead>
52           <tr>
53             <th>#</th>
54             <th>Room Name</th>
55             <th>Date</th>
56             <th>Time</th>
57             <th>Reserved By</th>
58             <th>Actions</th>
59           </tr>
60         </thead>
61         <tbody id="reservation-table-body">
62           <!-- Daftar reservasi akan dimuat di sini -->
63         </tbody>
64       </table>
65     </div>
66   </div>
67
68   <script src="script.js"></script>
69 </body>
70 </html>
71
```

## 2. styles.css

```
1  body {
2      background-color: #f8f9fa;
3      font-family: Arial, sans-serif;
4      position: relative;
5      overflow-x: hidden;
6  }
7
8  /* Video background */
9  #bg-video {
10     position: fixed;
11     top: 0;
12     left: 0;
13     width: 100%;
14     height: 100%;
15     object-fit: cover;
16     z-index: -1;
17 }
18
19 /* Highlight form animation */
20 .highlight {
21     animation: highlight 1s ease-out;
22 }
23
24 @keyframes highlight {
25     0% {
26         background-color: #fff;
27     }
28     50% {
29         background-color: #d4edda;
30     }
31     100% {
32         background-color: #fff;
33     }
34 }
```

```

36  /* Animasi fade-in untuk reservasi */
37  .fade-in {
38      animation: fadeIn 1s ease-in;
39  }
40
41  @keyframes fadeIn {
42      0% {
43          opacity: 0;
44      }
45      100% {
46          opacity: 1;
47      }
48  }
49
50  /* Animasi slide-out untuk membatalkan reservasi */
51  .slide-out {
52      animation: slideOut 0.5s ease-out;
53  }
54
55  @keyframes slideOut {
56      0% {
57          opacity: 1;
58          transform: translateX(0);
59      }
60      100% {
61          opacity: 0;
62          transform: translateX(100%);
63      }
64  }
65

```

### 3. script.js

```

1  class Room {
2      constructor(id, name, capacity) {
3          this.id = id;
4          this.name = name;
5          this.capacity = capacity;
6          this.remainingCapacity = capacity;
7          this.reservations = [];
8      }
9
10     // Memeriksa apakah ruangan tersedia pada tanggal dan waktu tertentu
11     isAvailable(date, time) {
12         return (
13             this.remainingCapacity > 0 &&
14             !this.reservations.some(reservation => reservation.date === date && reservation.time === time)
15         );
16     }
17
18     addReservation(reservation) {
19         if (this.remainingCapacity > 0) {
20             this.reservations.push(reservation);
21             this.remainingCapacity--; // Kurangi kapasitas yang tersisa
22         }
23     }
24
25     removeReservation(reservationId) {
26         const reservationIndex = this.reservations.findIndex(reservation => reservation.id === reservationId);
27         if (reservationIndex !== -1) {
28             this.reservations.splice(reservationIndex, 1);
29             this.remainingCapacity++; // Tambah kapasitas yang tersisa
30         }
31     }
32 }
33

```

```

34 class Reservation {
35     constructor(id, roomId, date, time, name) {
36         this.id = id;
37         this.roomId = roomId;
38         this.date = date;
39         this.time = time;
40         this.name = name;
41     }
42 }
43
44 const rooms = [
45     new Room(1, 'Conference Room A', 10),
46     new Room(2, 'Conference Room B', 15),
47     new Room(3, 'Meeting Room C', 5)
48 ];
49
50 let reservationIdCounter = 1;
51
52 /* Fungsi untuk menambahkan reservasi baru */
53 function addNewReservation(roomId, date, time, name) {
54     const room = rooms.find(r => r.id === roomId);
55     if (!room) {
56         alert("Room not found.");
57         return;
58     }
59
60     if (!room.isAvailable(date, time)) {
61         alert("Room is already booked at the selected date and time.");
62         return;
63     }
64
65     const newReservation = new Reservation(reservationIdCounter++, roomId, date, time, name);
66     room.addReservation(newReservation);
67

```

```

68     // Tambahkan animasi highlight pada form saat berhasil
69     const form = document.getElementById('reservation-form');
70     form.classList.add('highlight');
71     setTimeout(() => form.classList.remove('highlight'), 1000);
72
73     alert("Reservation added successfully.");
74     renderRooms(); // Perbarui daftar ruangan
75     renderReservations(); // Perbarui daftar reservasi
76 }
77
78 /* Fungsi untuk menampilkan daftar ruangan */
79 function renderRooms() {
80     const roomList = document.getElementById('room-list');
81     roomList.innerHTML = "<h3>Room List</h3>";
82
83     rooms.forEach(room => {
84         const roomElement = document.createElement('div');
85
86         const statusText = room.remainingCapacity (parameter) room: Room
87             ? `Available - Remaining Capacity: ${room.remainingCapacity}`
88             : "Not Available - Full Capacity";
89
90         roomElement.innerHTML = `
91             <strong>${room.name}</strong> (Capacity: ${room.capacity}) - ID: ${room.id}
92             <br>Status: <span class="${room.remainingCapacity > 0 ? 'available' : 'not-available'}">${statusText}</span>
93         `;
94
95         roomList.appendChild(roomElement);
96     });
97 }
98
99 /* Fungsi untuk menampilkan daftar reservasi */

```

```

100 function renderReservations() {
101   const reservationTableBody = document.getElementById('reservation-table-body');
102   reservationTableBody.innerHTML = ''; // Kosongkan tabel sebelum menampilkan data baru
103
104   let reservationCount = 1;
105
106   rooms.forEach(room => {
107     room.reservations.forEach(reservation => {
108       const row = document.createElement('tr');
109       row.classList.add('fade-in'); // Animasi fade-in
110
111       row.innerHTML = `
112         <td>${reservationCount}</td>
113         <td>${room.name}</td>
114         <td>${reservation.date}</td>
115         <td>${reservation.time}</td>
116         <td>${reservation.name}</td>
117         <td>
118           <button class="btn btn-danger btn-sm" onclick="cancelReservation(${reservation.id}, ${room.id}, this)">Cancel</button>
119         </td>
120       `;
121       reservationTableBody.appendChild(row);
122     });
123   });
124 }
125
126 /* Fungsi untuk menghapus reservasi dengan animasi slide-out */
127 function cancelReservation(reservationId, roomId, buttonElement) {
128   const room = rooms.find(r => r.id === roomId);
129   if (room) {
130     const row = buttonElement.closest('tr');
131     row.classList.add('slide-out');
132     setTimeout(() => {
133       room.removeReservation(reservationId);
134       renderRooms();

```

```

135       renderReservations();
136     }, 500); // Tunggu hingga animasi selesai
137   } else {
138     alert("Room not found.");
139   }
140 }
141
142 /* Fungsi untuk menangani penambahan reservasi dari form */
143 function handleAddReservation() {
144   const roomId = parseInt(document.getElementById('roomId').value);
145   const date = document.getElementById('date').value;
146   const time = document.getElementById('time').value;
147   const name = document.getElementById('name').value;
148
149   if (roomId && date && time && name) {
150     addNewReservation(roomId, date, time, name);
151   } else {
152     alert("Please fill in all fields.");
153   }
154 }
155
156 // Render ruangan dan reservasi saat pertama kali halaman dimuat
157 renderRooms();
158 renderReservations();
159

```