|  |  |
| --- | --- |
| **Ex. No: 13**  **08.05.2025** | **HOTEL MANAGEMENT SYSTEM** |

# AIM:

To create an efficient Hotel Management System and perform CRUD operation using node.js

and MySql Workbench.

# HOTEL MANAGEMENT SYSTEM:

* It is a web-based application built using Node.js, Express.js, MySQL, HTML, CSS, and JavaScript that helps users manage essential home-related tasks efficiently.
* The system provides features such as task scheduling, inventory management, and personalized reminders, all accessible through an intuitive web interface.
* This project aims to demonstrate the use of backend integration, database connectivity, and responsive UI design to solve real-life household management challenges in a user-friendly and accessible way.

# CREATING ‘BOOKINGS’ TABLE:

SQL> CREATE TABLE IF NOT EXISTS bookings (

    id INT AUTO\_INCREMENT PRIMARY KEY,

    check\_in DATE NOT NULL,

    check\_out DATE NOT NULL,

  guests INT NOT NULL,

    room\_type VARCHAR(50) NOT NULL,

    name VARCHAR(100) NOT NULL,

  email VARCHAR(100) NOT NULL,

    phone VARCHAR(20) NOT NULL,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

Table Created.

# FRONT-END CODING:

**index.html (Home page)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Hotel Management System</title>

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Custom CSS -->

<link rel="stylesheet" href="styles.css">

</head>

<body>

<!-- Navigation Bar -->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container">

<a class="navbar-brand" href="#">Hotel HMS</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav ms-auto">

<li class="nav-item">

<a class="nav-link active" href="#home">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#rooms">Rooms</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#bookings">Bookings</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#contact">Contact</a>

</li>

</ul>

</div>

</div>

</nav>

<!-- Hero Section -->

<header class="hero-section text-white text-center py-5">

<div class="container">

<h1>Welcome to Hotel HMS</h1>

<p class="lead">Experience luxury and comfort at its finest</p>

<a href="#book-now" class="btn btn-primary btn-lg">Book Now</a>

</div>

</header>

<!-- Rooms Section -->

<section id="rooms" class="py-5">

<div class="container">

<h2 class="text-center mb-4">Our Rooms</h2>

<div class="row">

<div class="col-md-4 mb-4">

<div class="card">

<img src="https://via.placeholder.com/300x200" class="card-img-top" alt="Standard Room">

<div class="card-body">

<h5 class="card-title">Standard Room</h5>

<p class="card-text">Comfortable room with essential amenities.</p>

<p class="price">$99/night</p>

<button type="button" class="btn btn-primary">Book Now</button>

</div>

</div>

</div>

<div class="col-md-4 mb-4">

<div class="card">

<img src="https://via.placeholder.com/300x200" class="card-img-top" alt="Deluxe Room">

<div class="card-body">

<h5 class="card-title">Deluxe Room</h5>

<p class="card-text">Spacious room with premium amenities.</p>

<p class="price">$149/night</p>

<button type="button" class="btn btn-primary">Book Now</button>

</div>

</div>

</div>

<div class="col-md-4 mb-4">

<div class="card">

<img src="https://via.placeholder.com/300x200" class="card-img-top" alt="Suite">

<div class="card-body">

<h5 class="card-title">Suite</h5>

<p class="card-text">Luxurious suite with separate living area.</p>

<p class="price">$249/night</p>

<button type="button" class="btn btn-primary">Book Now</button>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- Booking Form Section -->

<section id="book-now" class="py-5 bg-light">

<div class="container">

<h2 class="text-center mb-4">Book Your Stay</h2>

<div class="row justify-content-center">

<div class="col-md-8">

<form class="booking-form">

<div class="row">

<div class="col-md-6 mb-3">

<label for="check-in" class="form-label">Check-in Date</label>

<input type="date" class="form-control" id="check-in" required>

</div>

<div class="col-md-6 mb-3">

<label for="check-out" class="form-label">Check-out Date</label>

<input type="date" class="form-control" id="check-out" required>

</div>

</div>

<div class="row">

<div class="col-md-6 mb-3">

<label for="guests" class="form-label">Number of Guests</label>

<select class="form-select" id="guests" required>

<option value="1">1 Guest</option>

<option value="2">2 Guests</option>

<option value="3">3 Guests</option>

<option value="4">4 Guests</option>

</select>

</div>

<div class="col-md-6 mb-3">

<label for="room-type" class="form-label">Room Type</label>

<select class="form-select" id="room-type" required>

<option value="standard">Standard Room</option>

<option value="deluxe">Deluxe Room</option>

<option value="suite">Suite</option>

</select>

</div>

</div>

<div class="mb-3">

<label for="name" class="form-label">Full Name</label>

<input type="text" class="form-control" id="name" required>

</div>

<div class="mb-3">

<label for="email" class="form-label">Email</label>

<input type="email" class="form-control" id="email" required>

</div>

<div class="mb-3">

<label for="phone" class="form-label">Phone Number</label>

<input type="tel" class="form-control" id="phone" required>

</div>

<button type="submit" class="btn btn-primary w-100">Confirm Booking</button>

</form>

</div>

</div>

</div>

</section>

<!-- Bookings Section -->

<section id="bookings" class="py-5">

<div class="container">

<!-- Bookings will be loaded here dynamically -->

</div>

</section>

<!-- Contact Section -->

<section id="contact" class="py-5">

<div class="container">

<h2 class="text-center mb-4">Contact Us</h2>

<div class="row">

<div class="col-md-6 mb-4">

<div class="contact-info">

<h4>Get in Touch</h4>

</div>

</div>

<div class="col-md-6">

<form class="contact-form">

<div class="mb-3">

<label for="contact-name" class="form-label">Name</label>

<input type="text" class="form-control" id="contact-name" required>

</div>

<div class="mb-3">

<label for="contact-email" class="form-label">Email</label>

<input type="email" class="form-control" id="contact-email" required>

</div>

<div class="mb-3">

<label for="message" class="form-label">Message</label>

<textarea class="form-control" id="message" rows="4" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Send Message</button>

</form>

</div>

</div>

</div>

</section>

<!-- Footer -->

<footer class="bg-dark text-white py-4">

<div class="container">

<div class="row">

<div class="col-md-6">

<h5>Hotel HMS</h5>

</div>

<div class="col-md-6 text-md-end">

<p>&copy; 2024 Hotel HMS. All rights reserved.</p>

</div>

</footer>

<!-- Bootstrap JS and Popper.js -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

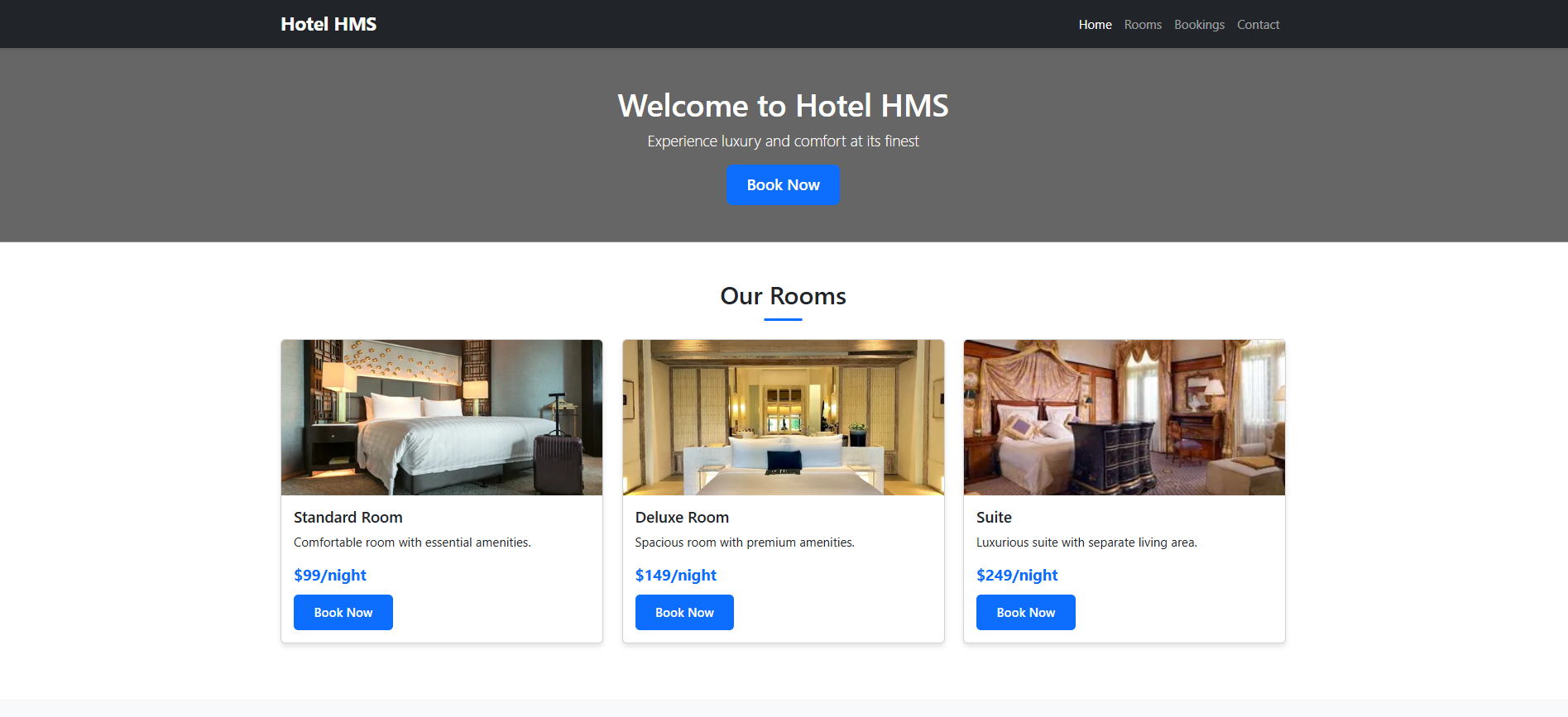
<!-- Custom JS -->

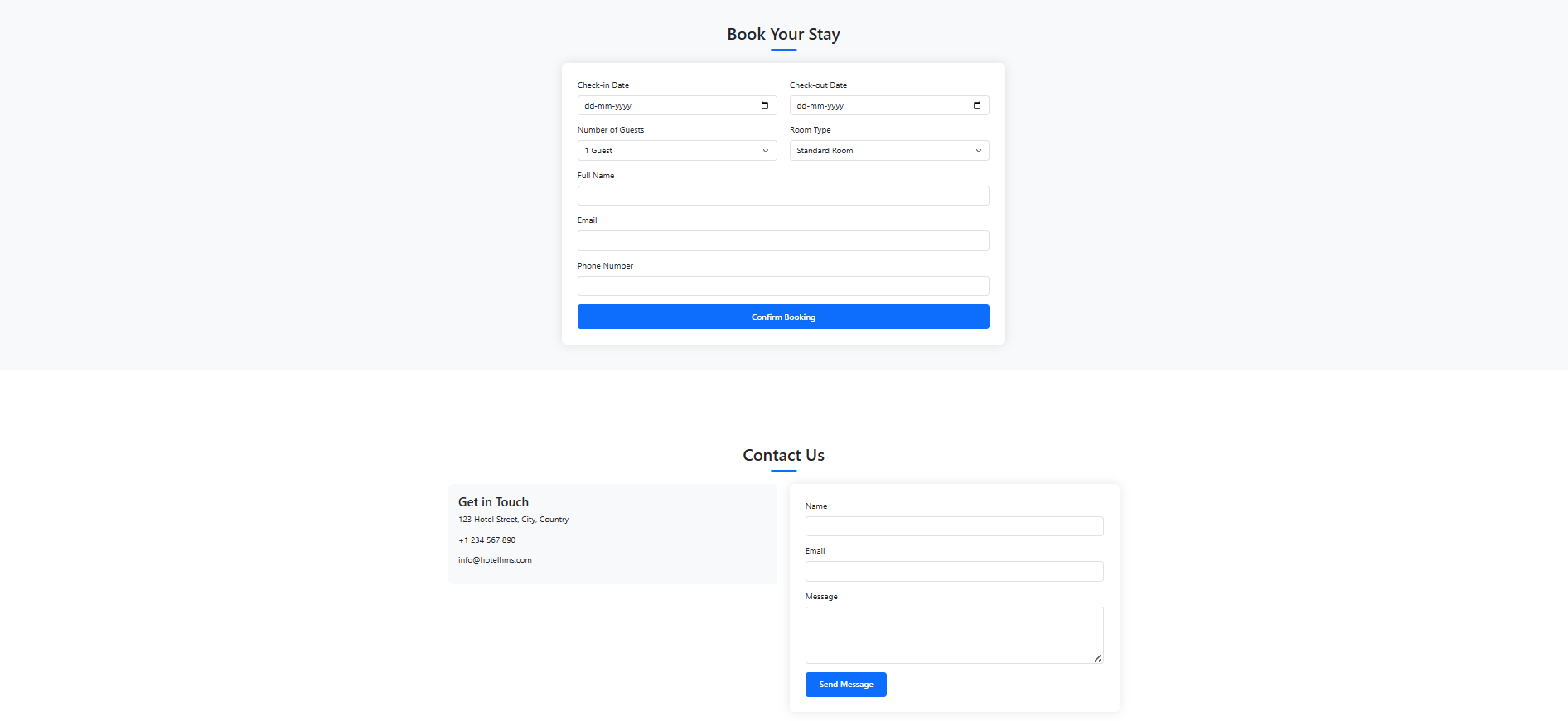
<script src="/public/js/bookings.js"></script>

</body>

</html>

**Output (Home page)**

****

****

**BACKEND CODING:**

**Node.js**

**Function to handle form submission**

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

const bookingForm = document.querySelector('.booking-form');

if (!bookingForm) {

console.error('Booking form not found');

return;

}

bookingForm.addEventListener('submit', async (e) => {

e.preventDefault();

const submitButton = e.target.querySelector('button[type="submit"]');

submitButton.disabled = true;

submitButton.innerHTML = 'Processing...';

const formData = {

check\_in: document.getElementById('check-in').value,

check\_out: document.getElementById('check-out').value,

guests: document.getElementById('guests').value,

room\_type: document.getElementById('room-type').value,

name: document.getElementById('name').value,

email: document.getElementById('email').value,

phone: document.getElementById('phone').value

};

try {

const response = await fetch('/api/bookings', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(formData)

});

const data = await response.json();

if (response.ok) {

alert('Booking created successfully!');

e.target.reset();

loadBookings(); // Reload the bookings list

} else {

alert('Error: ' + data.error);

}

} catch (error) {

console.error('Error:', error);

alert('Error creating booking. Please try again.');

} finally {

submitButton.disabled = false;

submitButton.innerHTML = 'Confirm Booking';

}

});

loadBookings();

});

**LOAD BOOKING(CREATES):**

**Function to load all bookings**

async function loadBookings() {

const bookingsSection = document.getElementById('bookings');

if (!bookingsSection) {

console.error('Bookings section not found');

return;

}

bookingsSection.innerHTML = '<div class="text-center"><div class="spinner-border" role="status"><span class="visually-hidden">Loading...</span></div></div>';

try {

const response = await fetch('/api/bookings');

if (!response.ok) {

throw new Error('Failed to fetch bookings');

}

const bookings = await response.json();

if (bookings.length === 0) {

bookingsSection.innerHTML = '<h2 class="text-center mb-4">Your Bookings</h2><p class="text-center">No bookings found.</p>';

return;

}

// Create a table to display bookings

const bookingsTable = document.createElement('table');

bookingsTable.className = 'table table-striped mt-4';

bookingsTable.innerHTML = `

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Check-in</th>

<th>Check-out</th>

<th>Room Type</th>

<th>Guests</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

${bookings.map(booking => `

<tr>

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

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

<td>${new Date(booking.check\_in).toLocaleDateString()}</td>

<td>${new Date(booking.check\_out).toLocaleDateString()}</td>

<td>${booking.room\_type}</td>

<td>${booking.guests}</td>

<td>

<button onclick="editBooking(${booking.id})" class="btn btn-sm btn-primary">Edit</button>

<button onclick="deleteBooking(${booking.id})" class="btn btn-sm btn-danger">Delete</button>

</td>

</tr>

`).join('')}

</tbody>

`;

bookingsSection.innerHTML = '<h2 class="text-center mb-4">Your Bookings</h2>';

bookingsSection.appendChild(bookingsTable);

} catch (error) {

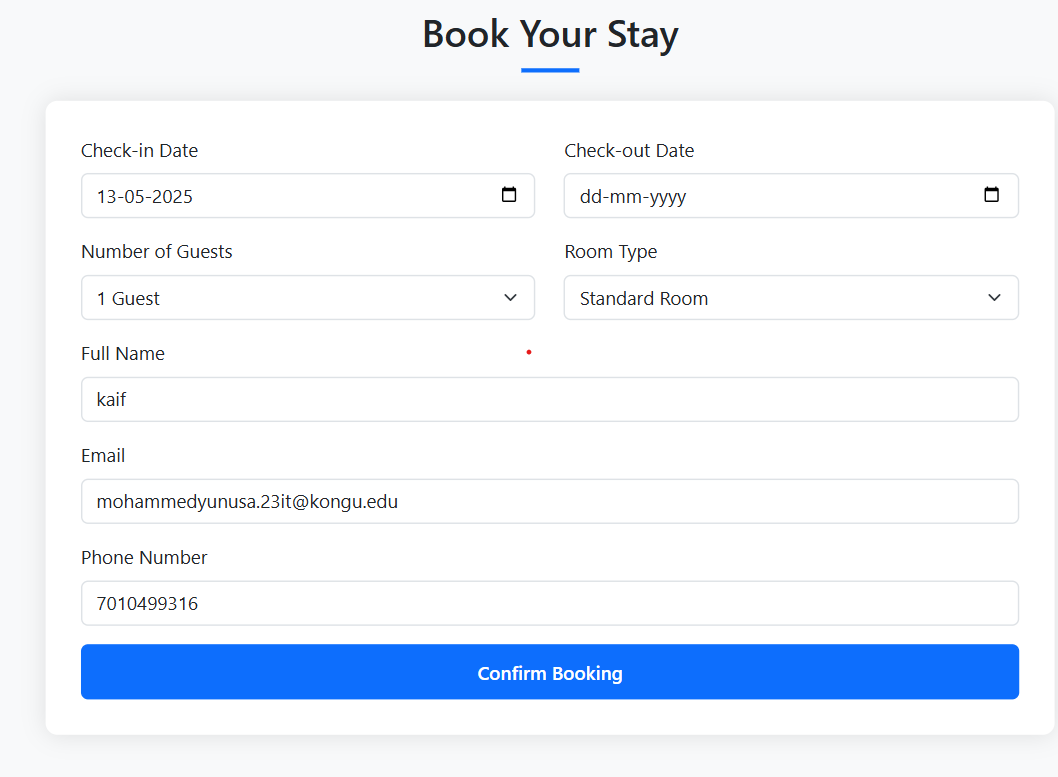
console.error('Error:', error);

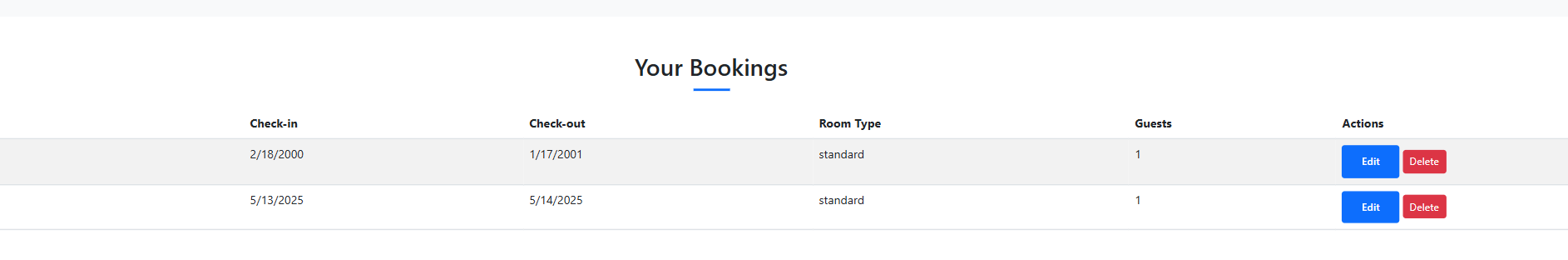
bookingsSection.innerHTML = '<h2 class="text-center mb-4">Your Bookings</h2><div class="alert alert-danger" role="alert">Error loading bookings. Please try again.</div>';

}

}

**Output (CREATES):**

****

****

**DELETE\_BOOKINGS(DELETES):**

**// Function to delete a booking**

async function deleteBooking(id) {

if (!confirm('Are you sure you want to delete this booking?')) {

return;

}

try {

const response = await fetch(`/api/bookings/${id}`, {

method: 'DELETE'

});

if (response.ok) {

alert('Booking deleted successfully!');

loadBookings(); // Reload the bookings list

} else {

const data = await response.json();

alert('Error: ' + data.error);

}

} catch (error) {

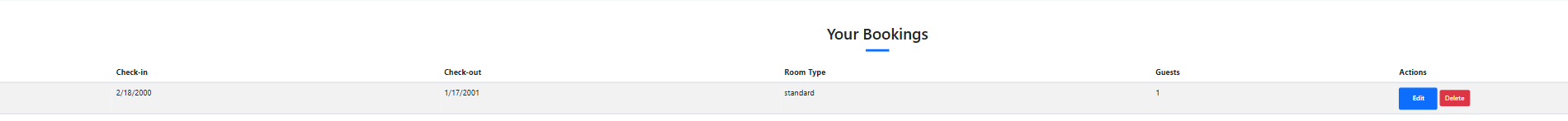
console.error('Error:', error);

alert('Error deleting booking. Please try again.');

}

}

**OUTPUT(DELETES):**

****

**EDIT\_BOOKINGS(UPDATE):**

**// Function to edit a booking**

async function editBooking(id) {

try {

const response = await fetch(`/api/bookings/${id}`);

if (!response.ok) {

throw new Error('Failed to fetch booking details');

}

const booking = await response.json();

// Populate the form with booking data

document.getElementById('check-in').value = booking.check\_in;

document.getElementById('check-out').value = booking.check\_out;

document.getElementById('guests').value = booking.guests;

document.getElementById('room-type').value = booking.room\_type;

document.getElementById('name').value = booking.name;

document.getElementById('email').value = booking.email;

document.getElementById('phone').value = booking.phone;

// Change form submission to update instead of create

const form = document.querySelector('.booking-form');

form.dataset.bookingId = id;

form.onsubmit = async (e) => {

e.preventDefault();

const submitButton = e.target.querySelector('button[type="submit"]');

submitButton.disabled = true;

submitButton.innerHTML = 'Updating...';

const formData = {

check\_in: document.getElementById('check-in').value,

check\_out: document.getElementById('check-out').value,

guests: document.getElementById('guests').value,

room\_type: document.getElementById('room-type').value,

name: document.getElementById('name').value,

email: document.getElementById('email').value,

phone: document.getElementById('phone').value

};

try {

const updateResponse = await fetch(`/api/bookings/${id}`, {

method: 'PUT',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(formData)

});

if (updateResponse.ok) {

alert('Booking updated successfully!');

form.reset();

form.removeAttribute('data-booking-id');

form.onsubmit = null; // Reset form submission handler

loadBookings(); // Reload the bookings list

}

else {

const data = await updateResponse.json();

alert('Error: ' + data.error);

}

} catch (error) {

console.error('Error:', error);

alert('Error updating booking. Please try again.');

} finally {

submitButton.disabled = false;

submitButton.innerHTML = 'Confirm Booking';

}

};

} catch (error) {

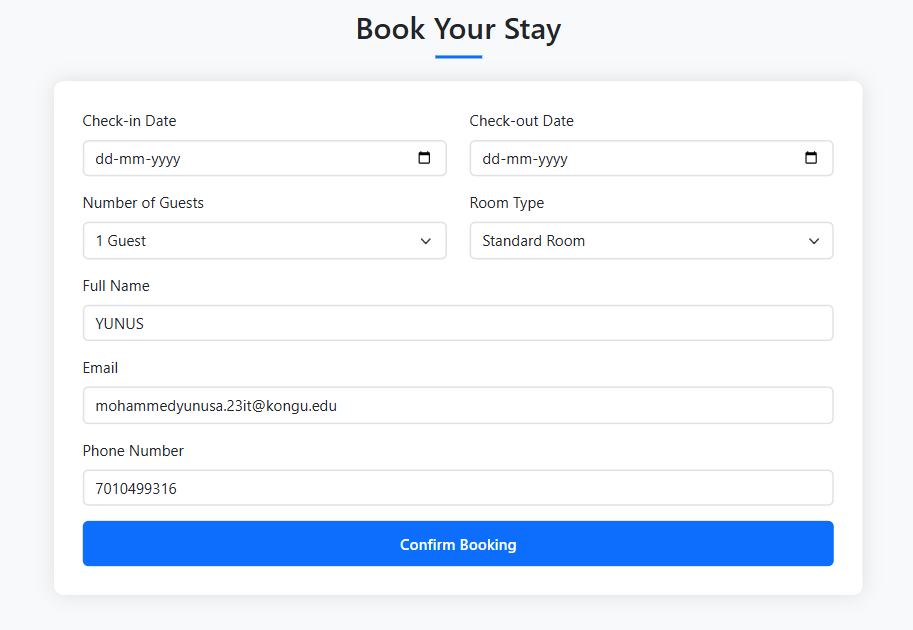
console.error('Error:', error);

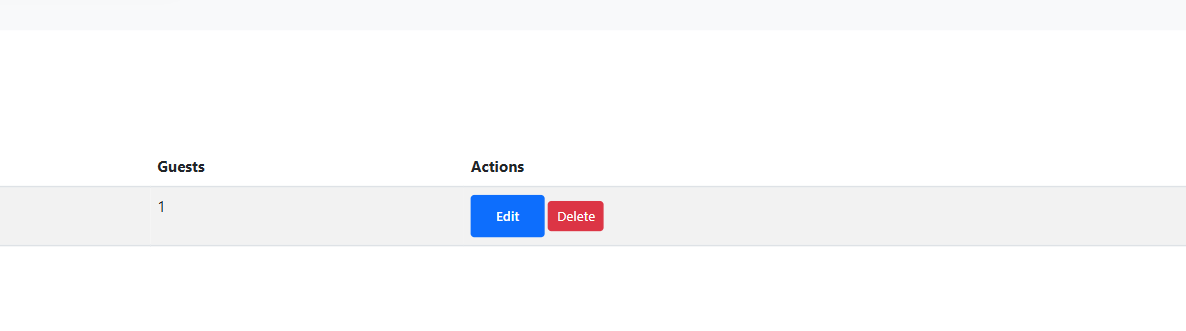
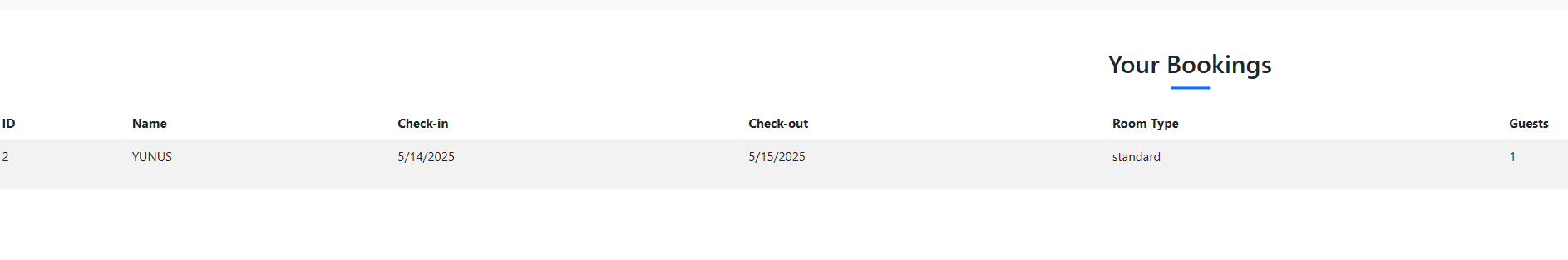
alert('Error loading booking details. Please try again.');

}

}

**OUTPUT(UPDATES):**

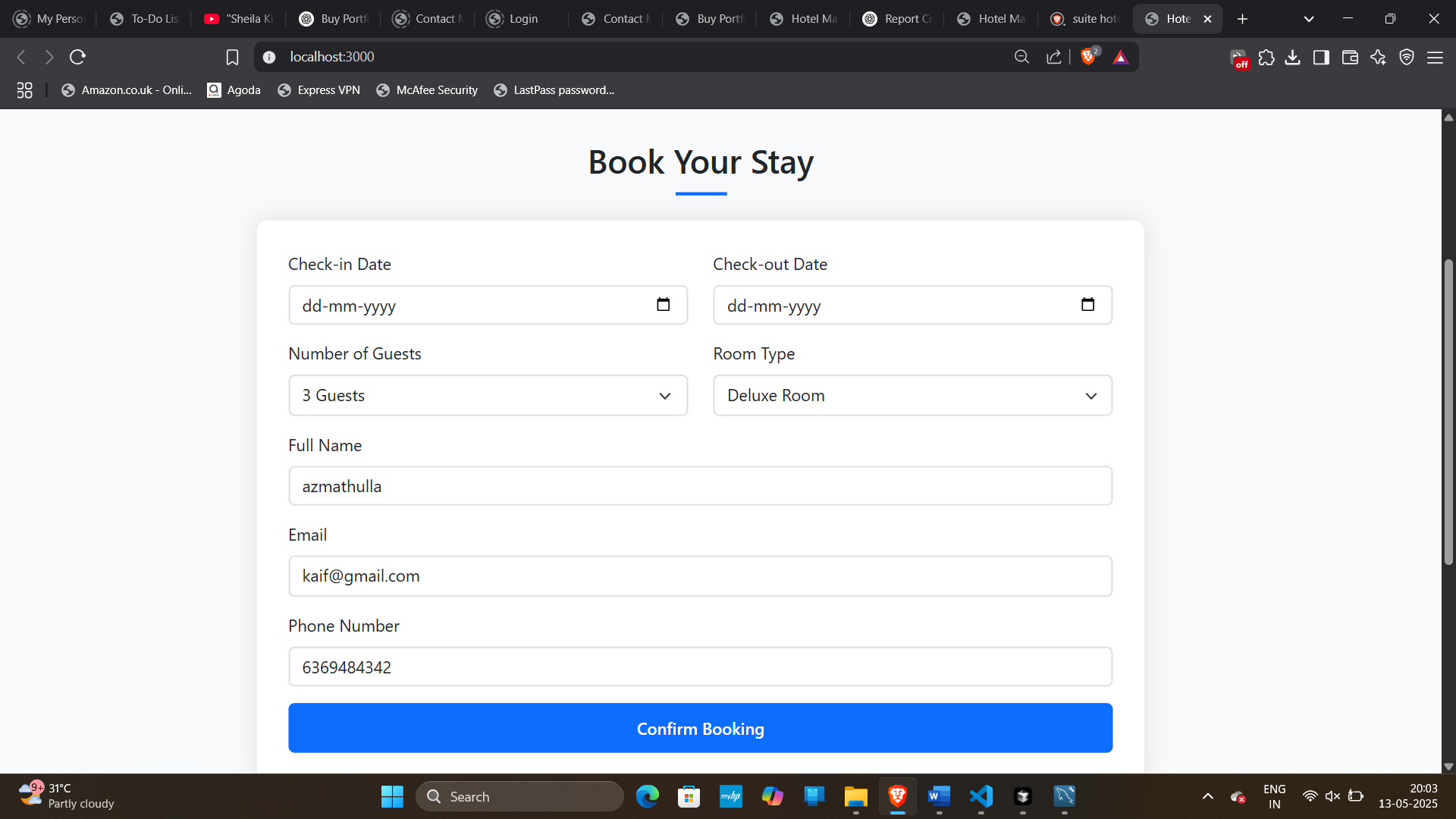
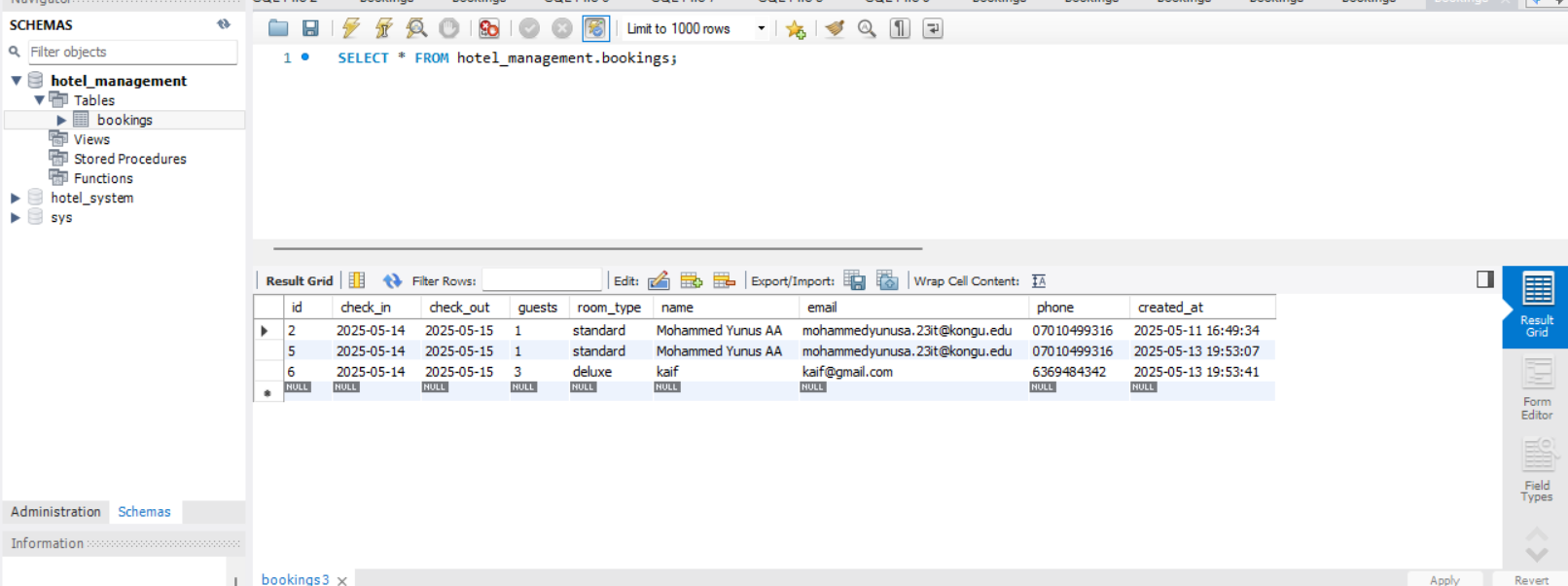
****

****

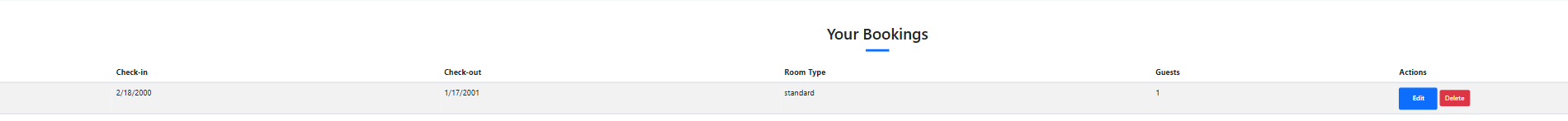
# OUTPUT:

# 

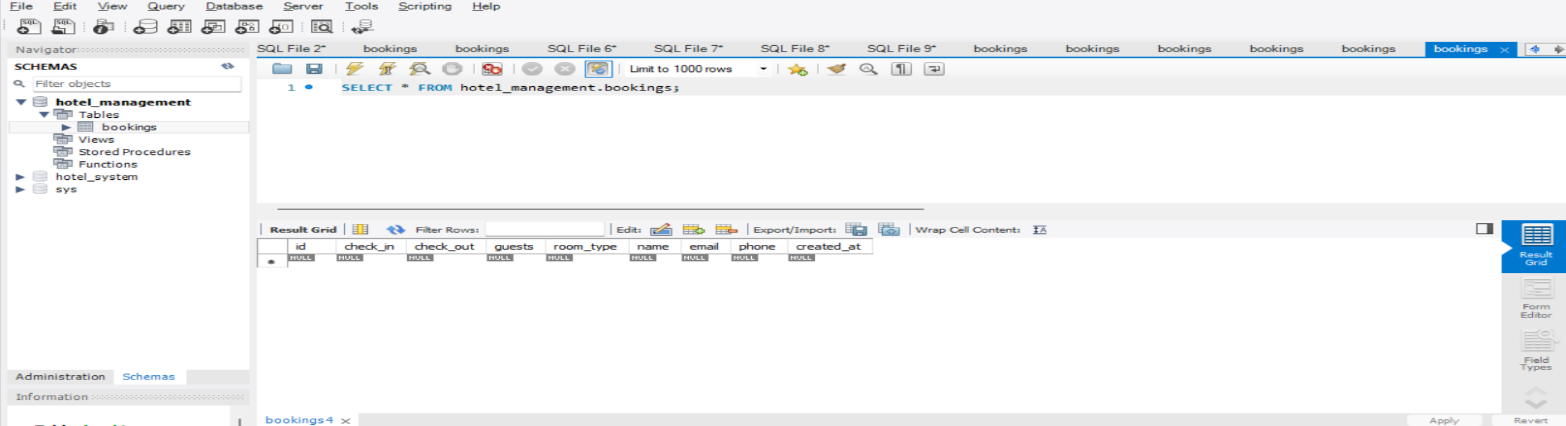
**Adding a new Booking Listing the updated Booking**

 ****

**Updating a existing track Listing the updated track**

****

**Deleting the existing Book**

****

**Listing the deletion of books**

|  |  |  |
| --- | --- | --- |
| **CONTENTS** | **MARKS ALLOTED** | **MARKS OBTAINED** |
| Aim, Algorithm, SQL, PL/SQL | 30 |  |
| Execution and Result | 20 |  |
| Viva | 10 |  |
| Total | 60 |  |

# RESULT:

Thus, the Hotel Management Expert System was successfully developed and executed, showcasing effective task automation, and database integration was created and performed CRUD operation.