DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

L&T

CIA 2&3 – Angular Coding Qs -Theory + Practical

B. Tech – Computer Science and Engineering
School of Engineering and Technology,

CHRIST (Deemed to be University),

Kumbalagodu, Bengaluru-560 074

Nov 2025

JavaScript Frontend Framework Project
Hotel Booking and Room Management System

Submitted By:

Team Member: Dinesh Babu R S
Roll Number: 2460360
College Email: dinesh.babu@btech.christuniversity.in

Team Member: Prisha S
Roll Number: 2460424
College Email: prisha.s@btech.christuniversity.in

Team Member: Jashwanth B S
Roll Number: 2460374
College Email: jashwanth.bs@btech.christuniversity.in

Team Member: Aishwarya S
Roll Number: 2460315
College Email: Aishwarya.s@btech.christuniversity.in

Course: Advanced Angular Frontend JavaScript
Instructor: Mr. Balagopal / Mr. Athaulla
University: Christ University
Department: Btech in Computer Science Engineering

INDEX:

Abstract:

The Hotel Booking and Room Management System is a comprehensive, web-based Single Page Application (SPA) designed and developed using Angular and TypeScript to model the functionality of a modern hotel reservation platform. The primary objective of this system is to provide users with a seamless and interactive interface for searching hotels, viewing detailed room information, checking availability, and making reservations, while also offering administrative capabilities for efficient room and booking management.

The application adopts Agular's component-driven architecture, which ensures modularity, scalability, and maintainability of the codebase. Core Angular concepts such as routing, dependency injection, reactive programming using RxJS observables, and form handling are extensively implemented to simulate real-world enterprise application behavior. TypeScript is utilized to enforce strong typing through interfaces and models, thereby improving code reliability, readability, and ease of maintenance.

To enhance user experience and ensure visual consistency across devices, Angular Material is integrated as the primary UI framework. Material components such as cards, tables, dialogs, toolbars, and snack bars provide a responsive, accessible, and professional user interface that aligns with modern design standards. Both template-driven and reactive forms are used within the application to handle user input efficiently, incorporating robust client-side validation mechanisms for fields such as booking dates, guest count, and user details.

Objectives:

The main objectives of this project are:

1. **Practical Angular Experience**
   To gain hands-on experience in building a real-world Angular application using modern best practices.

2. **Component-Based Architecture**
   To understand how complex applications are broken down into reusable components such as hotel listing, booking form, and admin panel.

3. **TypeScript Mastery**
   To apply TypeScript concepts such as interfaces, strong typing, and object-oriented principles for cleaner and safer code.

4. **Routing & Navigation**
   To implement seamless navigation using Angular Router and route guards.

5. **UI/UX Development**
   To design a responsive and accessible user interface using Angular Material.

Scope of the Project:

The project scope is limited to the **frontend and simulated backend layer**, focusing on user interaction and business logic.

Included in Scope

- Viewing hotels and rooms
- Booking and managing reservations
- Admin management of rooms
- Client-side form validation
- Mock backend integration

Excluded from Scope

- Real-time payment processing
- Real authentication servers
- Production database deployment

This controlled scope makes the project ideal for learning, testing, and portfolio demonstration.

Tools & Technologies Used:

Angular Framework

Angular is used as the core framework due to its:

- Component-based structure
- Built-in dependency injection
- Strong routing and form handling

TypeScript

TypeScript enhances JavaScript by adding:

- Static typing
- Interfaces for models (Hotel, Room, Booking)
- Better tooling and error detection

Angular Material

Used to implement:

- Responsive UI components
- Material design guidelines
- Consistent theming

JSON Server

Acts as a mock backend, simulating:

- API calls
- CRUD operations
- Asynchronous data flow
- Component-based structure
- Built-in dependency injection
- Strong routing and form handling

TypeScript

TypeScript enhances JavaScript by adding:

- Static typing
- Interfaces for models (Hotel, Room, Booking)
- Better tooling and error detection

Angular Material

Used to implement:

- Responsive UI components
- Material design guidelines
- Consistent theming

JSON Server

Acts as a mock backend, simulating:

- API calls
- CRUD operations
- Asynchronous data flow

Application Architecture:

The application follows Angular’s layered architecture:

1. Modules

- AppModule
- Feature modules (Hotels, Booking, Admin)

2. Components

Each feature is divided into components responsible for UI rendering and user interaction.

3. Services

Services handle business logic and data communication using HttpClient.

4. Models

TypeScript interfaces define data structures.

HTML Structure Overview:

Root Layout

The app.component.html serves as the shell:

<app-navbar></app-navbar>

<router-outlet></router-outlet>

Hotel List Component

- Uses *ngFor to display hotels dynamically.
- Data binding displays hotel name, price, and rating.

Hotel Detail Component

- Uses route parameters (/hotel/:id).
- Displays available rooms using Material Expansion Panels.

Booking Form Component

- Uses Reactive Forms.
- Form validation provides real-time feedback.

CSS Styling Strategy:

Styling is done using:

- Component-specific CSS
- Angular Material themes
- Flexbox layouts

Conditional styling is applied using:

```
<div [ngClass]="{'booked': room.isBooked}">
```

This improves visual feedback and user experience.

Key Features:

The Hotel Booking and Room Management System incorporates several essential features that collectively simulate a real-world hotel reservation platform:

1. Hotel Listing and Search

The system displays a list of available hotels using dynamic data binding. Users can search and filter hotels based on parameters such as location, price range, and rating. Angular directives like *ngFor and custom pipes are used to efficiently render and filter hotel data in real time.

2. Detailed Hotel and Room View

Each hotel has a dedicated detail page that provides comprehensive information, including hotel images, amenities, and available room categories (Standard, Deluxe, Suite). Route parameters enable dynamic loading of hotel-specific data without page reloads.

3. Room Availability and Booking

Users can select rooms and make reservations through a booking form. Reactive forms are implemented to handle date selection, number of guests, and room preferences, along with robust validation to ensure data accuracy.

4. User Dashboard

The user dashboard allows customers to view their booking history, reservation details, and booking status. This feature improves transparency and enhances the overall user experience.

5. Admin Panel

The admin panel provides hotel staff with tools to add, update, and delete room details, as well as track reservations. Angular Material tables and dialogs are used for structured data presentation and administrative actions.

6. Responsive Material UI

Angular Material components ensure a consistent, mobile-friendly, and professional user interface across all devices.

7. Notifications and Alerts

MatSnackBar is used to display booking confirmations, success messages, and error notifications, enhancing usability and system feedback.

Challenges Faced & Solutions:

1. Managing Data Across Multiple Components

Challenge: Synchronizing hotel, room, and booking data across different components.
Solution: Centralized services (HotelService, BookingService) with Dependency Injection and RxJS Observables were used to share data efficiently.

2. Form Validation and User Input Errors

Challenge: Validating complex inputs such as date ranges and guest limits.
Solution: Reactive forms with built-in and custom validators ensured accurate and user-friendly input handling.

3. Routing and Access Control

Challenge: Restricting admin access and protecting sensitive routes.
Solution: Route guards (CanActivate) were implemented to secure admin and dashboard routes.

4. UI Consistency and Responsiveness

Challenge: Maintaining a consistent UI across various screen sizes.
Solution: Angular Material's responsive components and Flexbox layouts were used to ensure device compatibility.

5. Error Handling and User Feedback

Challenge: Handling API errors gracefully.
Solution: HTTP interceptors and MatSnackBar notifications were implemented for centralized error handling and clear user feedback.

Outcome:

The successful implementation of the Hotel Booking and Room Management System resulted in a fully functional Angular-based SPA that effectively demonstrates real-world application design principles. The project achieved a clean separation of concerns through modular architecture, reusable components, and centralized services. Users can search hotels, view room availability, and make bookings seamlessly, while administrators can efficiently manage room data and reservations.

From a learning perspective, the project strengthened practical knowledge of Angular concepts such as routing, dependency injection, reactive forms, observables, and Material UI integration. Additionally, the application serves as a portfolio-ready project that reflects industry-relevant frontend development practices.

Future Enhancements:

While the current system meets its intended objectives, several enhancements can further improve functionality and scalability:

1. Backend Integration

Replacing the mock JSON server with a real backend (Node.js, Spring Boot) and database.

2. Authentication and Authorization

Implementing secure user login using JWT with role-based access control for Admin and Users.

3. Payment Gateway Integration

Adding real-time payment processing for bookings using services such as Razorpay or Stripe.

4. Real-Time Availability Updates

Using WebSockets to reflect live room availability.

5. Advanced Search and Maps

Integrating Google Maps for location-based hotel search.

6. Cloud Deployment

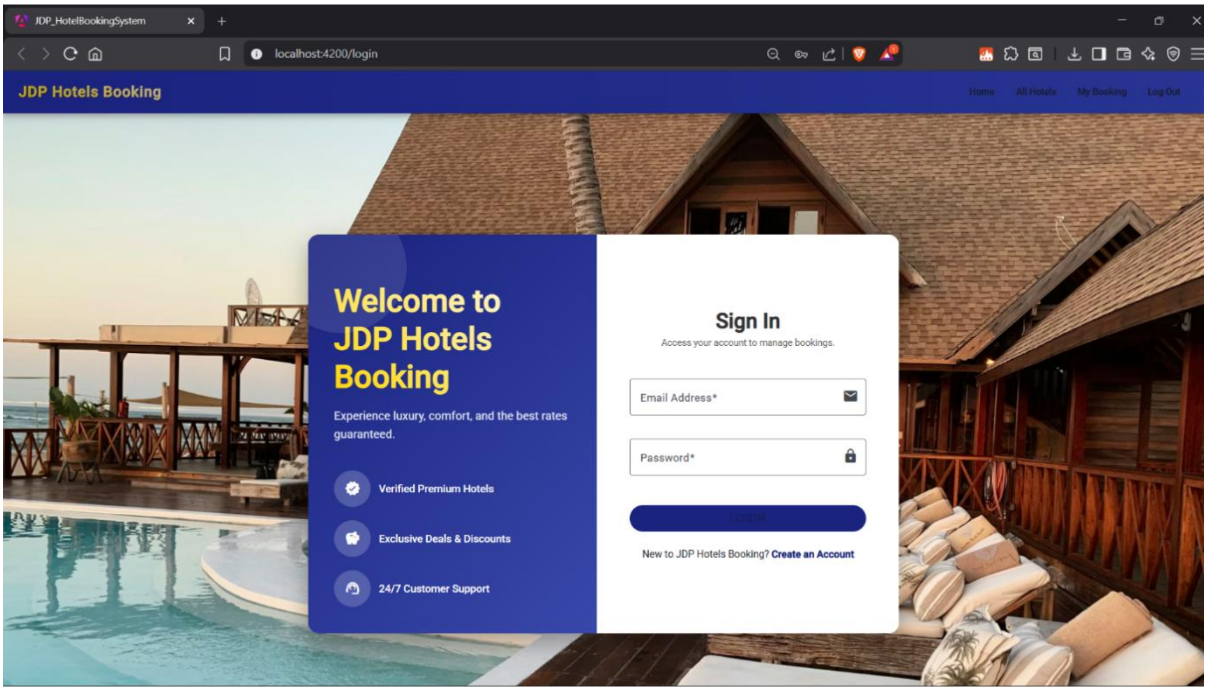
Deploying the application on cloud platforms such as AWS, Firebase, or Vercel.

Full Code:

Github Link: https://github.com/dineshbaburs/hotel-booking-system

Screenshots of Final Output:

- **Sign-In Page:**



Welcome to JDP Hotels Booking

Experience luxury, comfort, and the best rates guaranteed.

Verified Premium Hotels

Exclusive Deals & Discounts

24/7 Customer Support

Sign In

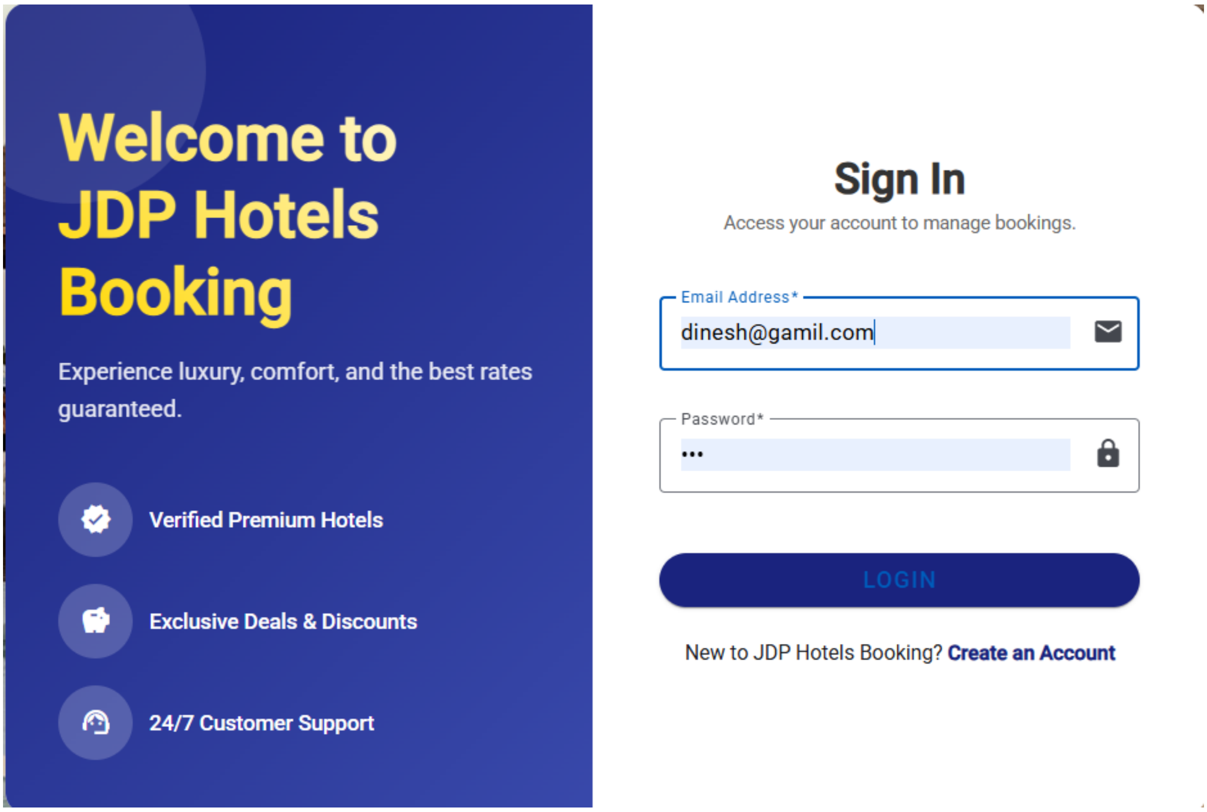
Access your account to manage bookings.

Email Address*

Password*

LOGIN

New to JDP Hotels Booking? **Create an Account**



Welcome to JDP Hotels Booking

Experience luxury, comfort, and the best rates guaranteed.

Verified Premium Hotels

Exclusive Deals & Discounts

24/7 Customer Support

Sign In

Access your account to manage bookings.

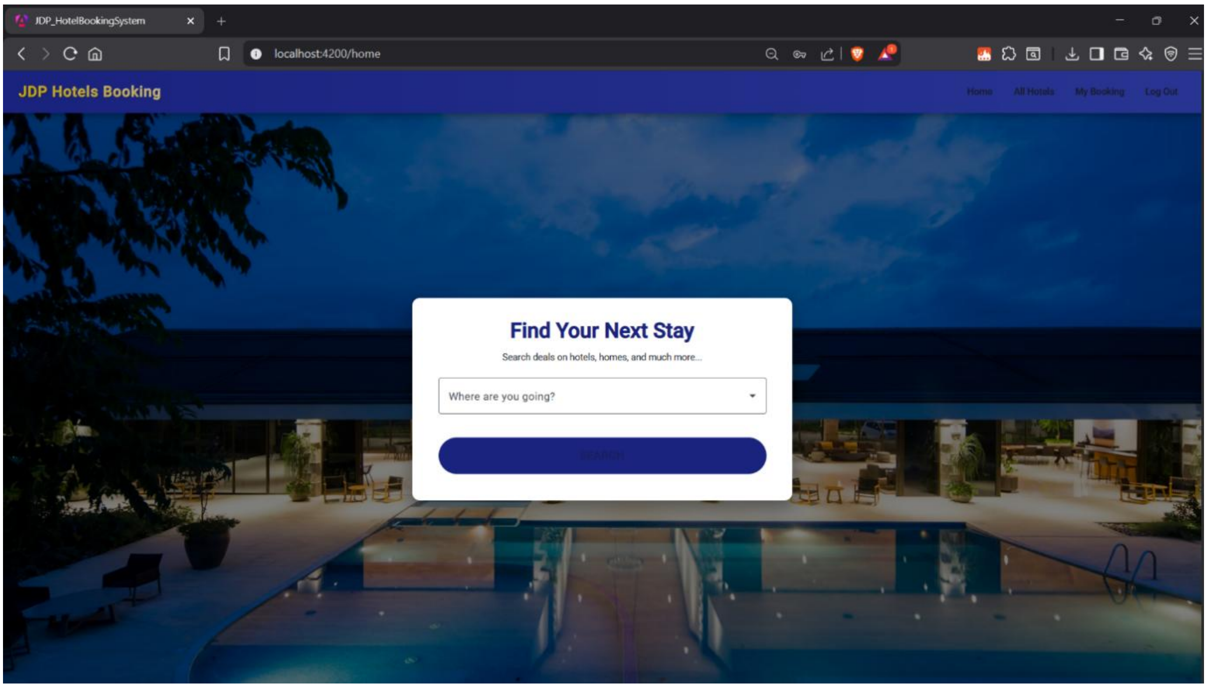
Email Address*: dinesh@gamil.com

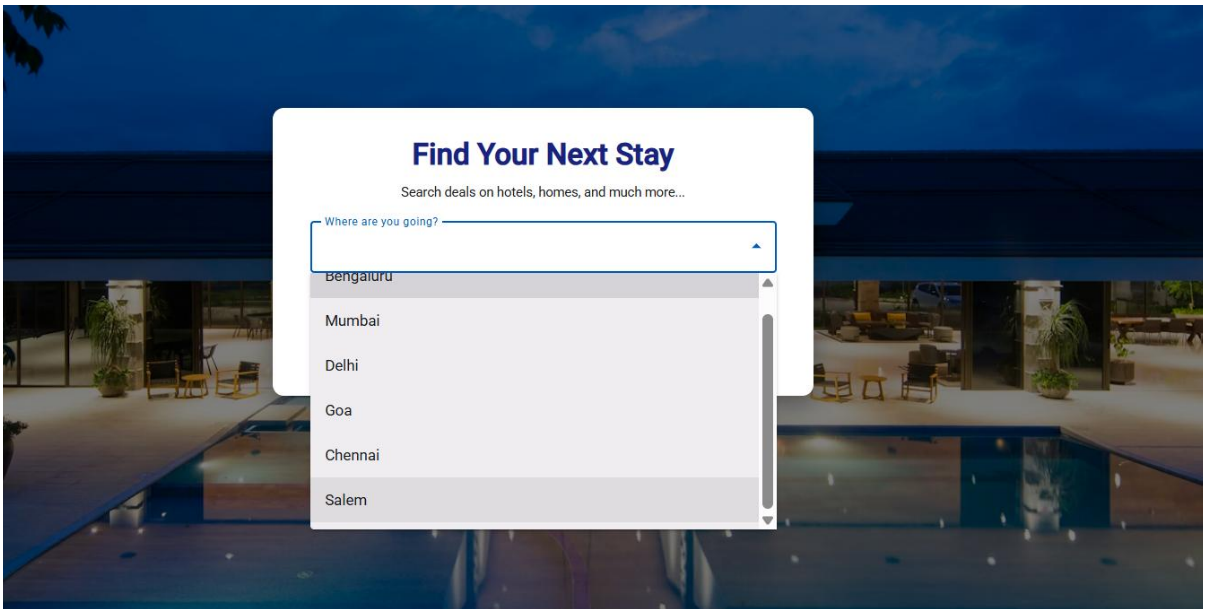
Password*: •••

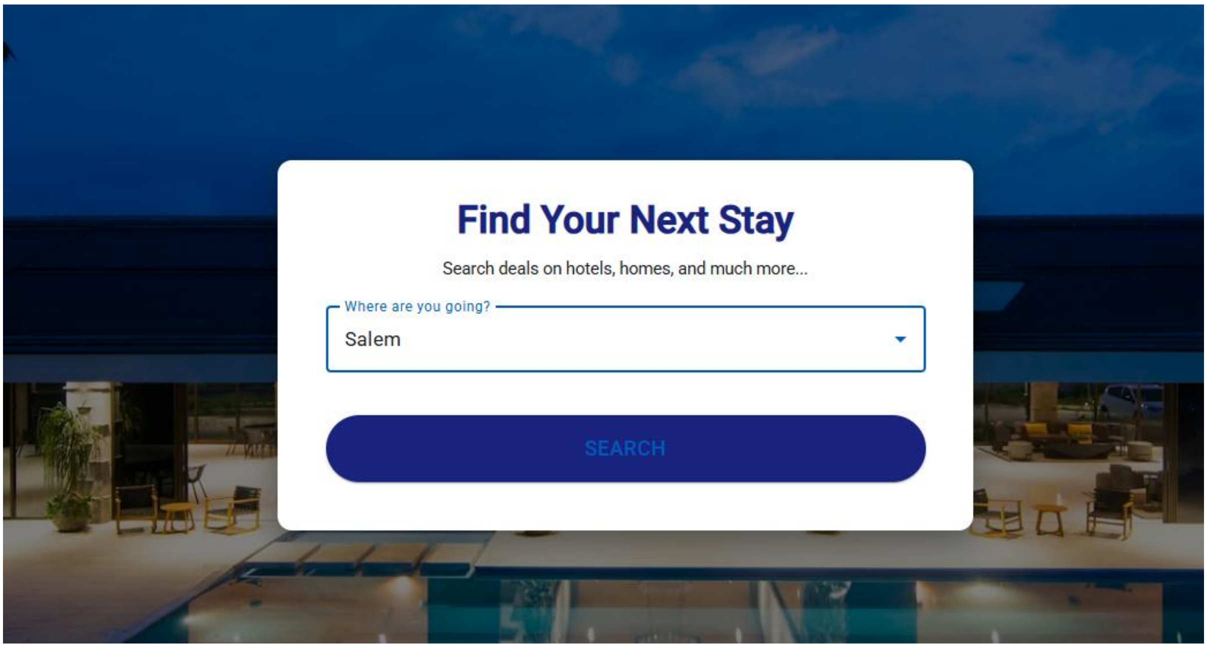
LOGIN

New to JDP Hotels Booking? **Create an Account**

- **Room Booking Page:**



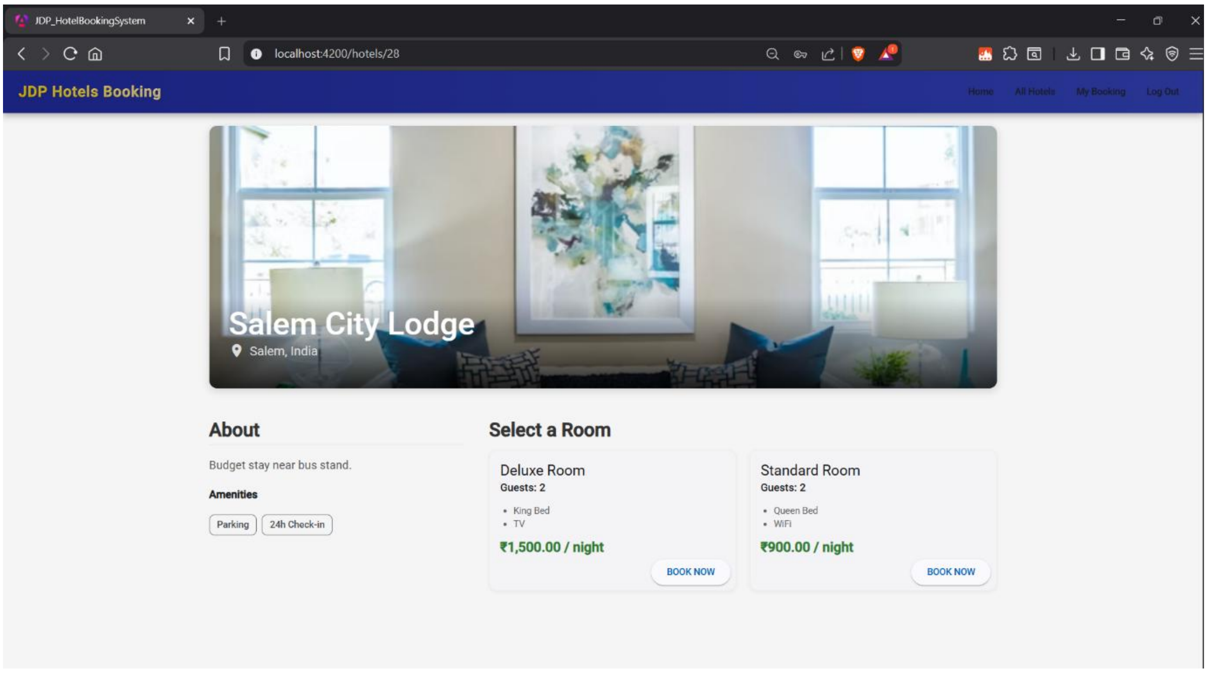




Find Your Next Stay

Search deals on hotels, homes, and much more...

Where are you going?: Salem



Salem City Lodge

Salem, India

About

Budget stay near bus stand.

Amenities

Parking | 24h Check-in

Select a Room

Deluxe Room
Guests: 2

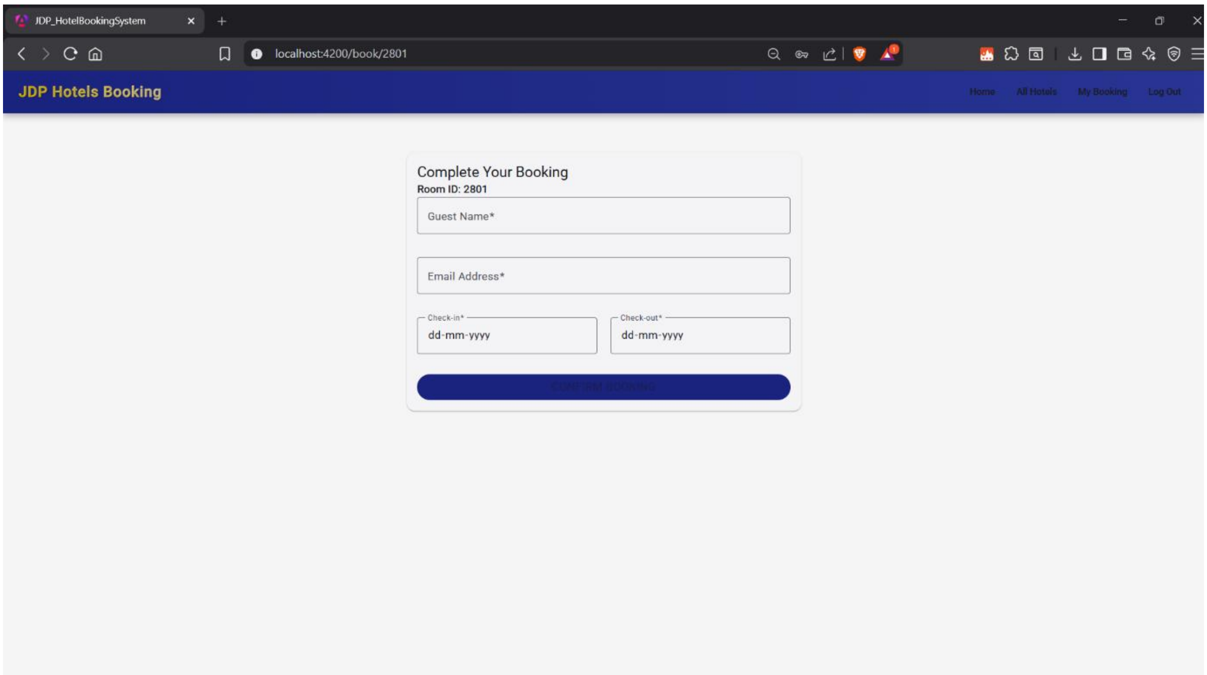
- King Bed
- TV

₹1,500.00 / night

Standard Room
Guests: 2

- Queen Bed
- WiFi

₹900.00 / night



Complete Your Booking

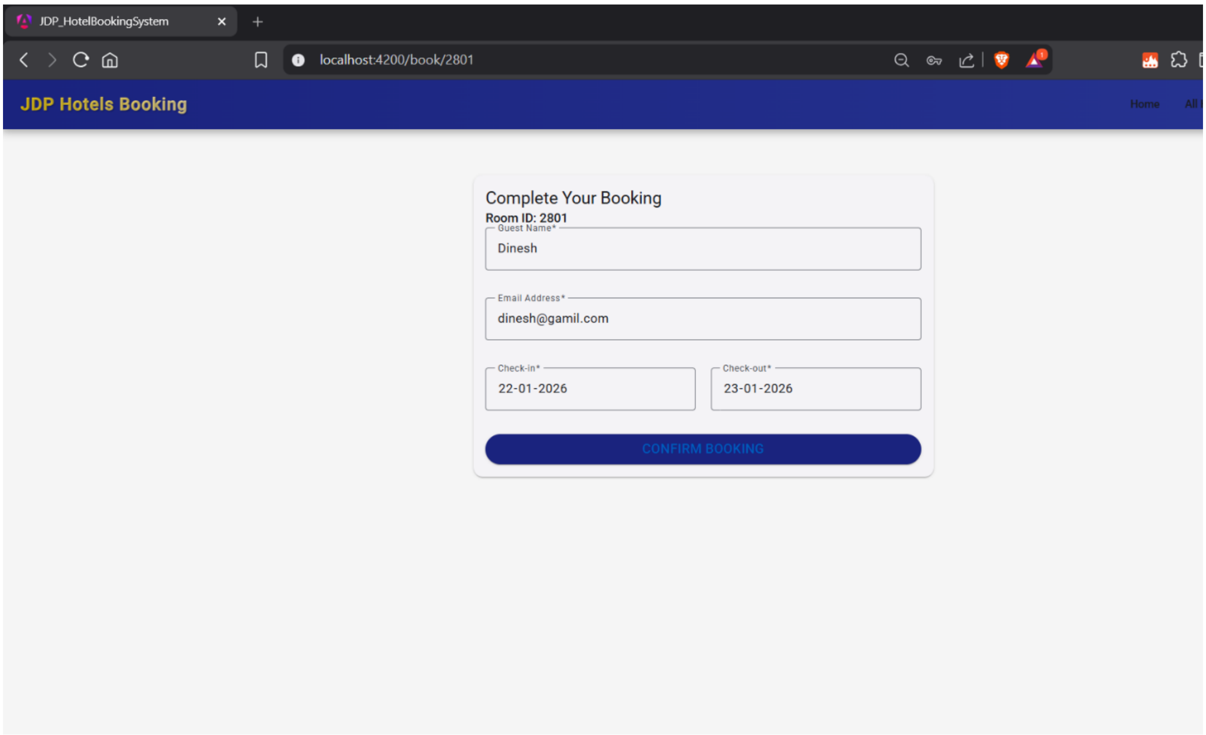
Room ID: 2801

Guest Name*

Email Address*

Check-in* dd-mm-yyyy

Check-out* dd-mm-yyyy



Complete Your Booking

Room ID: 2801

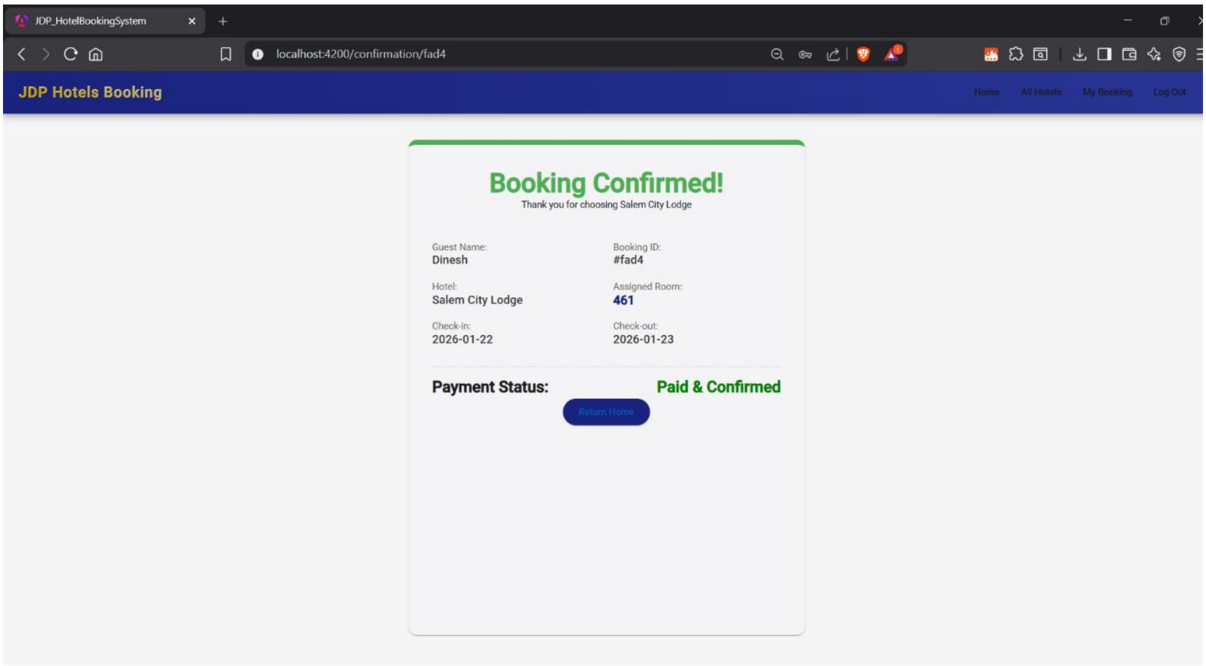
Guest Name*: Dinesh

Email Address*: dinesh@gamil.com

Check-in*: 22-01-2026

Check-out*: 23-01-2026

- **<u>Booking Confirmation Page</u>:**



Booking Confirmed!

Thank you for choosing Salem City Lodge

Guest Name:
Dinesh

Booking ID:
#fad4

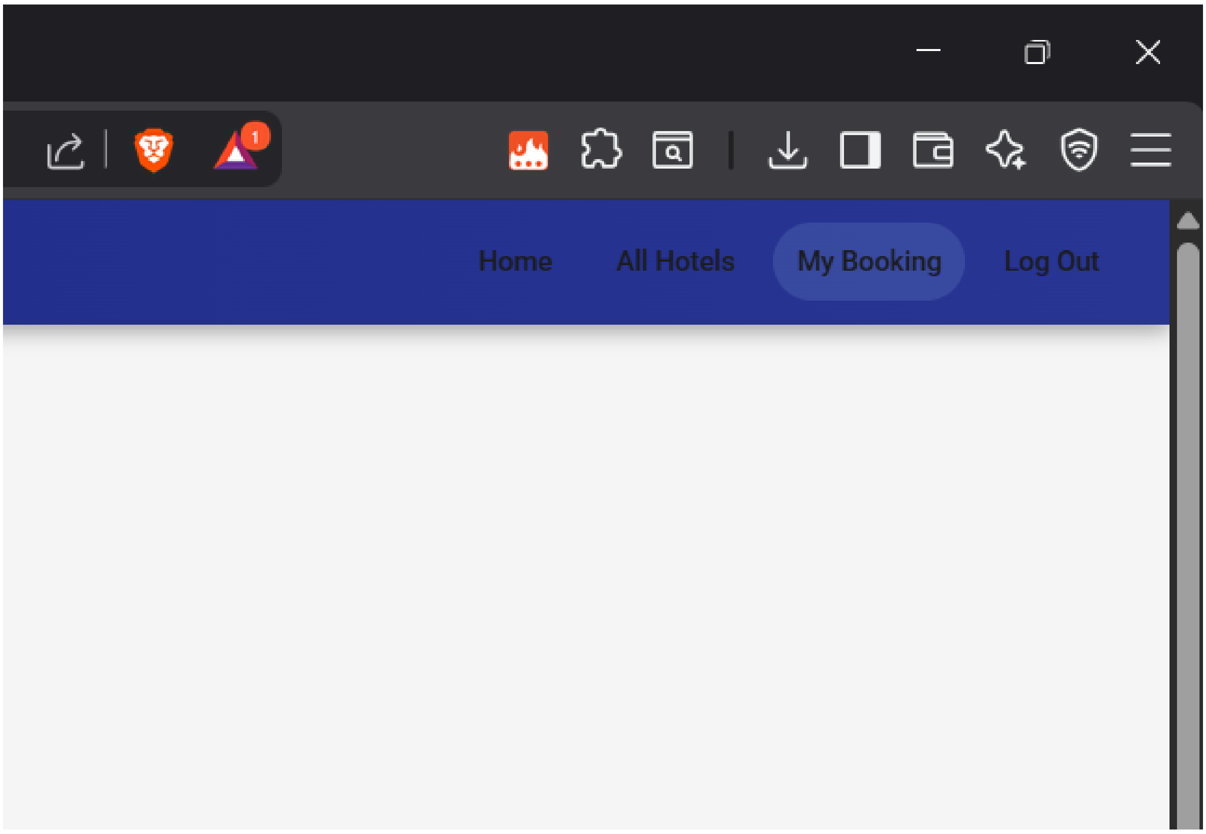
Hotel:
Salem City Lodge

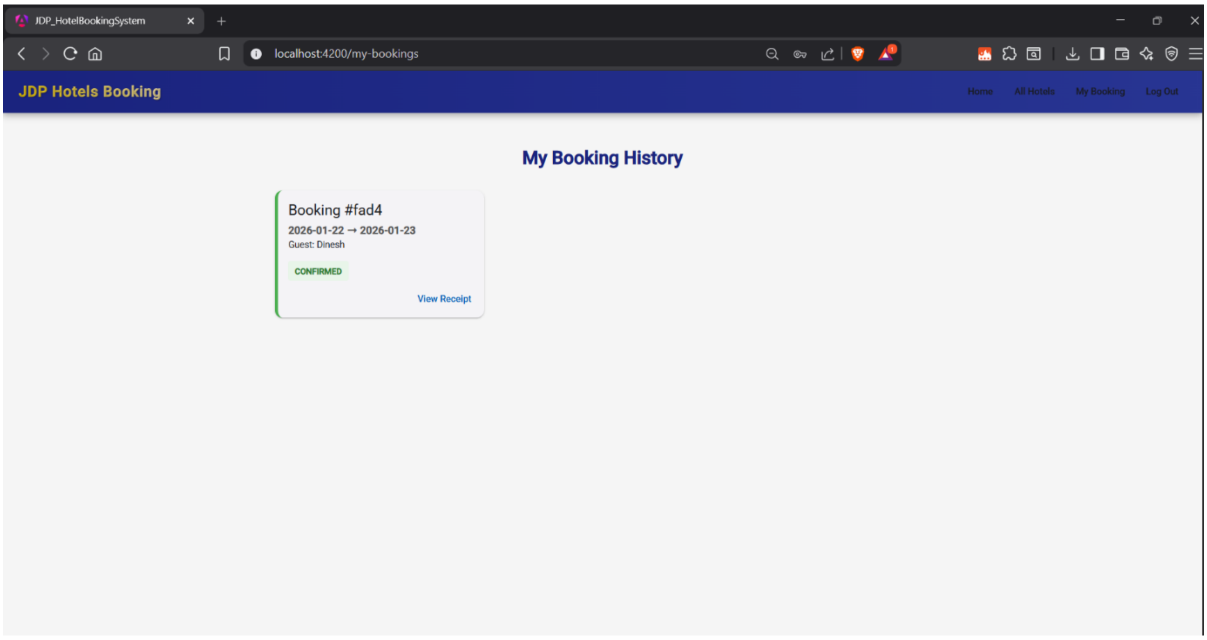
Assigned Room:
461

Check-in:
2026-01-22

Check-out:
2026-01-23

Payment Status: Paid & Confirmed





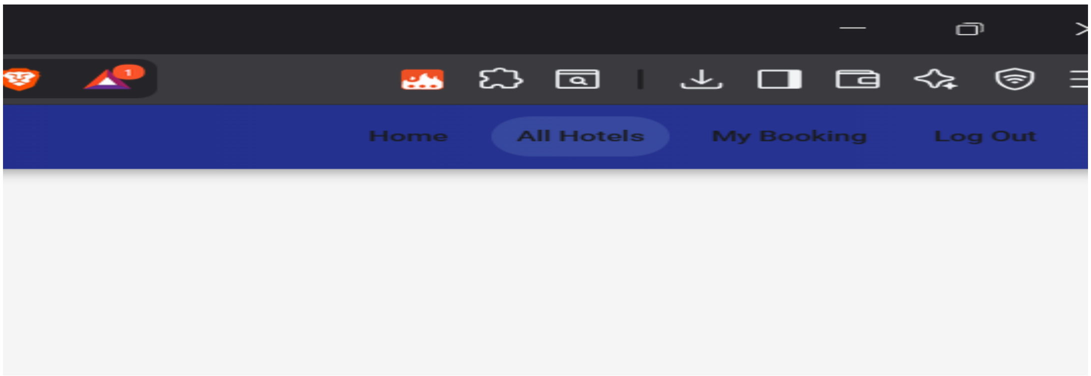
My Booking History

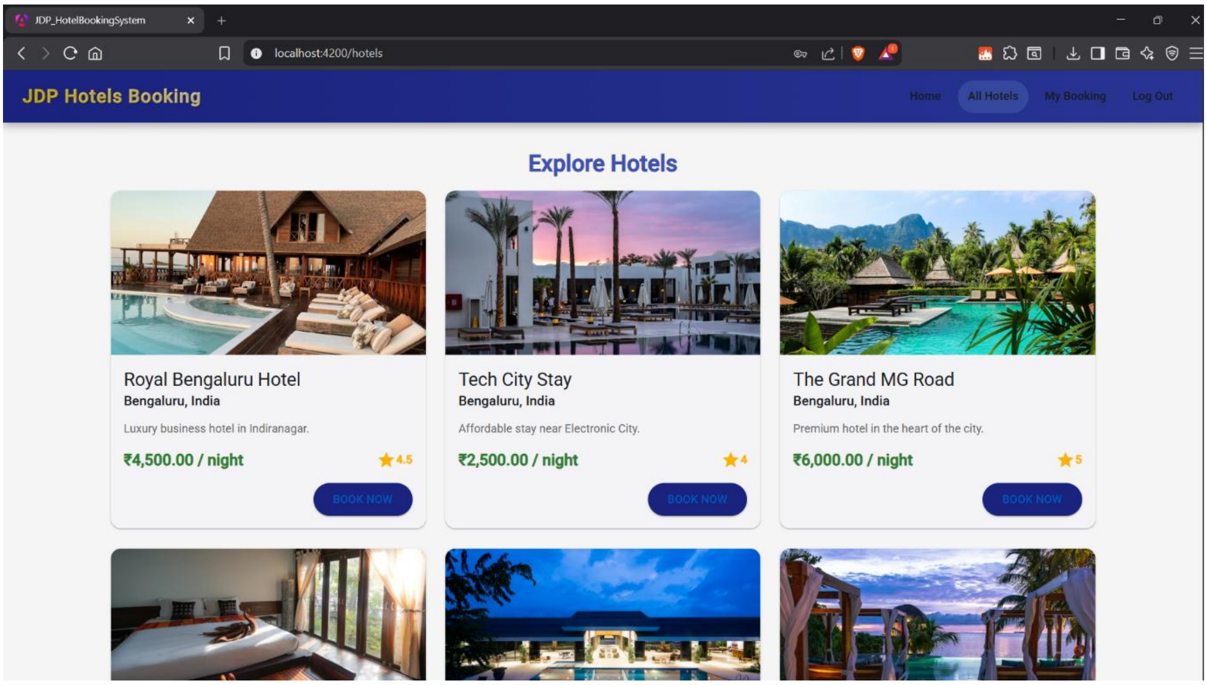
Booking #fad4

2026-01-22 → 2026-01-23

Guest: Dinesh

CONFIRMED





Explore Hotels

Royal Bengaluru Hotel
Bengaluru, India
Luxury business hotel in Indiranagar.
₹4,500.00 / night ★4.5

Tech City Stay
Bengaluru, India
Affordable stay near Electronic City.
₹2,500.00 / night ★4

The Grand MG Road
Bengaluru, India
Premium hotel in the heart of the city.
₹6,000.00 / night ★5

Conclusion:

The Hotel Booking and Room Management System successfully demonstrates the design and implementation of a modern, scalable, and user-friendly web application using Angular and TypeScript. The project effectively integrates core Angular concepts such as component-based architecture, routing, dependency injection, reactive and template-driven forms, and RxJS observables to simulate a real-world hotel reservation platform.

Through the use of Angular Material, the application achieves a professional and responsive user interface that enhances usability and accessibility across various devices. The implementation of structured services and strongly typed models ensures clean separation of concerns, improved maintainability, and ease of future expansion. Mock backend integration using JSON Server or static JSON files enables realistic simulation of asynchronous data handling and CRUD operations, providing valuable insight into client–server communication patterns.

References:

The development of the Hotel Booking and Room Management System was supported by various official documentation, learning resources, and tools. These references were used to ensure best practices, accurate implementation, and adherence to modern web development standards:

1. **Angular Official Documentation**
   Angular Team. *Angular Documentation*.
   https://angular.io/docs

2. **TypeScript Handbook**
   Microsoft. *TypeScript Documentation*.
   https://www.typescriptlang.org/docs

3. **Angular Material Documentation**
   Angular Team. *Angular Material UI Components*.
   https://material.angular.io

4. **RxJS Documentation**
   ReactiveX. *RxJS Library Documentation*.
   https://rxjs.dev

5. **JSON Server Documentation**
   Typicode. *JSON Server – Fake REST API.*
   https://github.com/typicode/json-server

6. **MDN Web Docs**
   Mozilla. *HTML, CSS, and JavaScript Reference*.
   https://developer.mozilla.org