Hotel Booking System

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

D. Gopika - AP23110011571
Y. Lakshmi Maheswari - AP23110011590
CH.Himakshi - AP23110011596
P. Durga Sravanthi - AP23110011597

Under the Guidance of

Yatharth Shahrawat Sir

Department of Computer Science and Engineering

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[December, 2025]

INTRODUCTION

In today's digital era, the hospitality industry has rapidly moved toward automation to provide a smooth and convenient booking experience for customers. Traditional hotel booking methods - such as calling the reception, checking room availability manually, or physically visiting the hotel - are time-consuming and often lead to errors, double bookings, or miscommunication. To overcome these limitations, online hotel reservation platforms have become a modern necessity.

The Hotel Booking System developed in this project aims to provide a complete, user-friendly, and efficient solution for customers to search, view, and book hotel rooms online. It combines a visually appealing frontend built using React.js, a lightweight backend powered by JSON Server, and seamless API communication through Axios. This ensures that the system behaves just like a real-world application, with features such as authentication, booking management, service reservations, and order tracking.

This system allows users to create an account, log in securely, browse different room categories, view room details, check amenities, and make bookings based on their preferred dates and number of guests. Users can also book additional services like spa or dining reservations, reflecting real hotel operations. The application further provides features such as editing a booking, cancelling a booking, managing food orders, and accessing a personalized user dashboard.

On the technology side, the project demonstrates key full-stack concepts:

- Frontend Interaction using reusable React components
- State Management and authentication using React Context API
- Backend Data Handling through RESTful API endpoints
- Database Operations using the JSON Server (db.json)
- Form validations, price calculations, and real-time feedback using Toast notifications

Overall, this Hotel Booking System showcases how modern web technologies can streamline hotel operations, reduce manual workload, and greatly enhance user satisfaction. It not only serves as a practical full-stack development project but also reflects how the hospitality domain is evolving by adopting digital solutions for everyday tasks.

SCENARIO-BASED INTRO

Imagine a traveler named ananya who plans a weekend getaway to a beachside resort. She searches online, but most hotel websites show outdated information, limited photos, and no real-time availability. She calls the hotel reception and is kept on hold. When she finally reaches someone, the receptionist manually checks room availability, asks her to wait again, and confirms only one option. By the time she decides, the room is already booked by another guest. Frustrated, Ananya wishes there was a simple, modern, reliable system that could show real-time room details and allow instant booking.

This is not just her problem—hotels also face challenges. Without a digital system, staff must handle bookings through phone calls, handwritten records, and spreadsheets. This often leads to:

- Double bookings
- Miscommunication about room capacity or pricing
- Difficulty tracking customer requests
- Inability to manage dining or spa reservations efficiently

To solve these real-world issues, the **Hotel Booking System** was designed.

This system acts as a complete digital assistant for both guests and hotel management. A user can log in, browse rooms, check availability, view amenities, compare prices, and instantly book a stay—all from their device. If Ananya wants to book a spa session or restaurant table along with her room, the system manages that too. Guests can also modify or cancel bookings anytime, reducing the burden on hotel staff.

For the hotel, this platform ensures that all rooms, services, food orders, and customer details are stored in a structured database. Every action—from booking to cancellation—is recorded accurately, reducing human errors and improving operational efficiency.

By simulating a real hotel workflow using **React.js**, **JSON Server**, and **REST APIs**, this project demonstrates how modern technology transforms a simple hotel into a smart, automated, and customer-friendly digital experience.

TARGET AUDIENCE

1. Regular Travelers and Tourists

These are individuals like families, couples, students, and solo travelers who frequently plan vacations or short stays.
They look for:

- Quick access to room information
- Real-time pricing
- Seamless booking without calling the hotel
- Ability to manage or cancel bookings independently

The system helps them compare rooms, check availability, and book instantly.

2. Business Travelers

Many professionals travel for conferences, meetings, or corporate stays.
They expect:

- Reliable reservation system
- Clear pricing and facilities
- On-time check-in and check-out
- Options to book additional services such as dining or spa

The system ensures fast booking with accurate details, which is essential for time-sensitive business trips.

3. Hotel Customers Seeking Additional Services

Some guests want more than just a room—they want a complete hospitality experience.
These include:

- Guests booking spa services
- Guests reserving dining tables

- Customers placing food orders

Your system's Service Booking and My Orders modules cater exactly to these users.

4. Hotel Staff and Management

Even though this version is user-focused, hotel staff indirectly benefit from it.
They need:

- A structured way to view bookings
- Accurate guest details
- Reduced manual workload
- Clear records of food orders and dining reservations

Automation helps staff spend less time on administrative tasks and more on guest service.

5. First-Time Users Who Want a Simple Booking Experience

Some people feel overwhelmed by complex hotel websites.
Your system appeals to them because:

- The design is clean and easy to navigate
- Booking steps are straightforward
- Instant confirmation builds trust

This makes the platform approachable even for non-technical users.

PROJECT GOALS AND OBJECTIVES

1. **To provide an easy and seamless online room booking experience** where users can browse rooms, view details, check availability, and book instantly.
2. **To automate hotel operations** by reducing manual work, minimizing errors, and maintaining accurate digital records of bookings and services.
3. **To offer secure user authentication** through login and registration, enabling personalized access to bookings, orders, and profile details.
4. **To allow users to manage their bookings easily**, including viewing, editing, and cancelling reservations through a user-friendly interface.
5. **To integrate additional hotel services**, such as dining reservations, spa/service bookings, and food orders within a single platform.
6. **To ensure real-time data handling** using a lightweight backend (JSON Server) for smooth CRUD operations and consistent data storage.

Modern Tech Stack

1. Frontend Technologies

- **React.js** – Component-based UI library for building responsive and interactive user interfaces.
- **React Router** – Handles client-side routing and navigation smoothly.
- **Tailwind CSS** – Utility-first CSS framework for modern, clean, and responsive UI design.
- **React Icons** – Provides modern iconography for enhanced user experience.
- **React Toastify / Custom Toast System** – Displays real-time notifications for actions like login, booking, or cancellations.

2. Backend Technologies

- **JSON Server** – Lightweight REST API backend used for simulating real server behavior, storing data, and handling CRUD operations.
- **Axios** – For API communication (GET, POST, PATCH, DELETE requests).

3. State & Authentication

- **React Context API** – Manages global state for user authentication and toast notifications.
- **LocalStorage** – Stores logged-in user information for persistent sessions.

4. Development & Tools

- **JavaScript (ES6+)** – Core programming language for both frontend and API interactions.
- **Node.js Environment** – Used for running JSON Server and development tooling.
- **Git / GitHub** – Version control and project collaboration.
- **VS Code** – Development environment for coding and debugging.

Key Features

1. User Authentication

- Secure user registration with name, email, and password.
- Login system with credential validation.
- User session stored in LocalStorage for persistent login.
- Logout option to clear session securely.

2. Room Browsing & Search

- Users can view all available rooms with images, prices, capacity, and amenities.
- Room categories like Deluxe, Suite, Premium, Villa, Royal, etc.
- Detailed room page showing:
    - Full description
    - Amenities
    - High-quality images
    - Price per night
    - Capacity
- Users can search based on date, number of guests, and navigate easily.

3. Room Booking System

- Select check-in and check-out dates.

- Real-time validation:
    - Check-out must be after check-in
    - Guests must not exceed room capacity
- Automatic calculation of:
    - Number of nights
    - Total cost
- Booking confirmation with toast notification and alert.
- Booking saved instantly to backend (JSON Server).

4. Booking Management

- Users can view all their bookings in one place.
- Options include:
    - **Edit booking** (change dates, guests)
    - **Cancel booking** (delete from backend)
- Updated price recalculation after editing.
- Booking list displayed with:
    - Room image
    - Room name
    - Booking dates
    - Total price
    - Status

5. Service Booking (Spa / Additional Services)

- Users can book extra hotel services like spa, wellness sessions, etc.
- Select:
    - Date
    - Time slot
    - Number of guests
    - Special requests
- Price auto-calculated based on number of guests.
- Service booking stored separately with `type: "service"`.

6. Dining Reservations

- Users can reserve tables at the hotel restaurant.
- Capture details like date, time, guests, and requests.
- Confirmed reservation gets stored in backend.

- Users can view and cancel reservations anytime.

7. Food Ordering System

- Users can place food orders directly from the dining/restaurant menu.
- Order details include:
    - Item name
    - Price
    - Quantity
    - Total cost
    - Order date and status
- Orders can be cancelled by the user.

8. User Dashboard

- Displays user information (name, email).
- Quick navigation to:
    - My Bookings
    - My Orders
    - Browse Rooms
- Users can update profile information (name, email).
- Option to logout from dashboard.

9. Home Page Features

- Beautiful hero section with luxury hotel visuals.
- Floating search bar for quick availability check.
- Highlights of hotel facilities (WiFi, Spa, Dining, Pool).
- Testimonials from guests.
- Featured rooms display.

10. Toast & Notification System

- Custom toast system plus React-Toastify integration.
- Real-time feedback for:
    - Login/Logout
    - Registration
    - Booking success
    - Cancellation
    - Errors

11. Modern UI/UX Features

- Built with Tailwind CSS for responsive design.
- Smooth animations and hover effects.
- Sticky booking cards and forms.
- Mobile-friendly pages.
- Clean layout with readable fonts and visual consistency.

12. REST API Integration

- All operations done through Axios API calls:
    - Fetch rooms
    - Create booking
    - Update booking
    - Delete booking
    - Manage users
    - Handle orders & reservations
- Data storage using JSON Server for real-time CRUD operations.

PRE-REQUISITES

1. Software Requirements

- **Node.js (Latest LTS Version)**
  Required to run React, JSON Server, and development tools.
- **NPM or Yarn**
  Package manager used to install dependencies.
- **Code Editor (VS Code Recommended)**
  For writing and editing project files.
- **Web Browser (Google Chrome Recommended)**
  For testing and running the application.

2. Technical Knowledge

- **Basic understanding of JavaScript (ES6+)**
  Needed to understand component logic and API functions.
- **React.js fundamentals**
  Such as components, props, hooks (useState, useEffect), and routing.
- **Basic familiarity with REST APIs**
  Since the system communicates with a backend through Axios.

- **JSON format understanding**
  Required for working with `db.json` in JSON Server.
- **Basic HTML & CSS**
  Tailwind CSS knowledge is helpful for UI customization.

3. Backend Tools

- **JSON Server**
  Used as a lightweight backend for data storage and CRUD operations.
- **Axios**
  For making API requests between frontend and backend.

4. Project Dependencies Installed

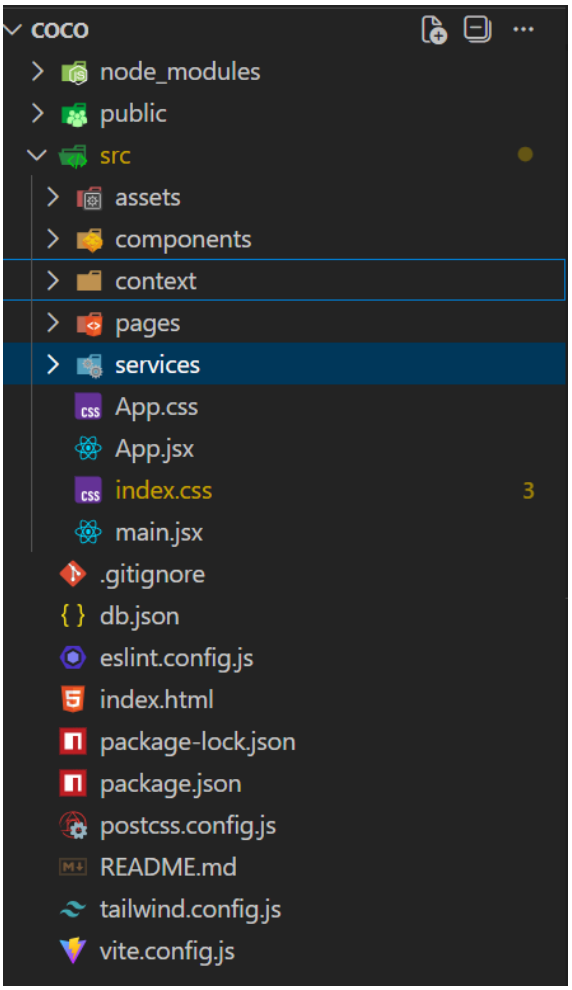
The following packages must be installed using npm:

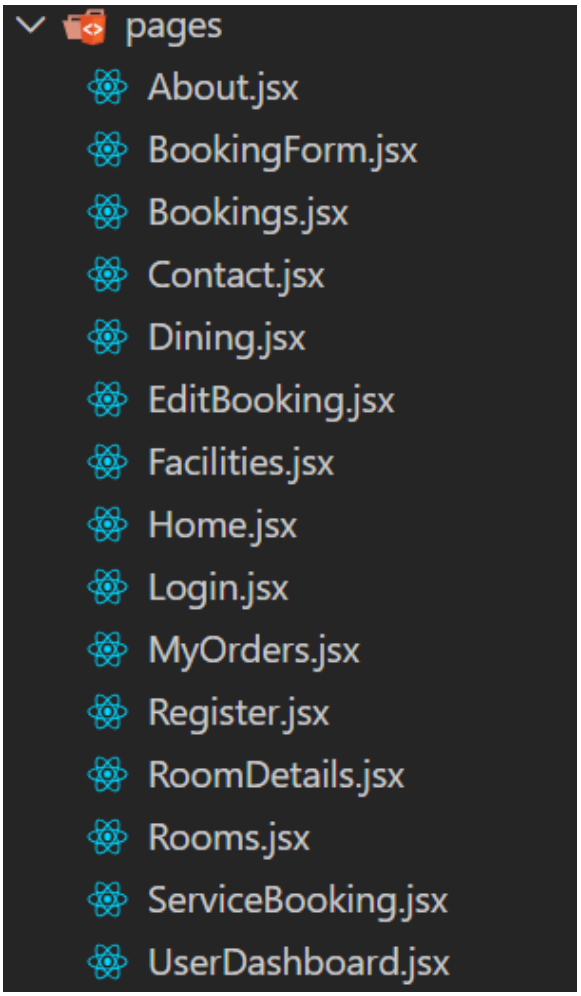
- react and react-dom
- react-router-dom
- axios
- react-icons
- react-toastify
- tailwindcss

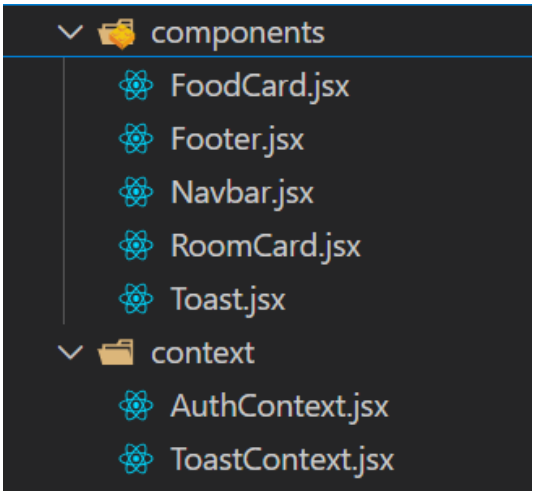
5. Hardware Requirements

- A computer with at least:
  - **4 GB RAM** (8 GB recommended for smooth development)
  - **Stable internet connection**
  - **Minimum 1 GB free storage** for project files and node modules

Project structure







DataBase: db.json

```jsx
<main className="flex-grow container mx-auto px-4 py-4">
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/login" element={<Login />} />
    <Route path="/register" element={<Register />} />
    <Route path="/rooms" element={<Rooms />} />
    <Route path="/rooms/:id" element={<RoomDetails />} />
    <Route path="/dining" element={<Dining />} />
    <Route path="/bookings" element={<Bookings />} />
    <Route path="/dashboard" element={<UserDashboard />} />
    <Route path="/facilities" element={<Facilities />} />
    <Route path="/contact" element={<Contact />} />
    <Route path="/service-booking" element={<ServiceBooking />} />
    <Route path="/about" element={<About />} />
    <Route path="/booking-form" element={<BookingForm />} />
    <Route path="/edit-booking/:id" element={<EditBooking />} />
    <Route path="/my-orders" element={<MyOrders />} />
  </Routes>
</main>
```

```jsx
import axios from 'axios';

const api = axios.create({
    baseURL: 'http://localhost:3001',
    headers: {
        'Content-Type': 'application/json',
    },
});
```

```
export const getRooms = () => api.get('/rooms');
export const getRoom = (id) => api.get(`/rooms/${id}`);
export const registerUser = (user) => api.post('/users', user);
export const loginUser = async (email, password) => {
    const response = await api.get(`/users?email=${email}&password=${password}`);
    if (response.data.length > 0) {
        return response.data[0];
    }
    throw new Error('Invalid credentials');
};
export const createBooking = (booking) => api.post('/bookings', booking);
export const getUserBookings = (userId) => api.get(`/bookings?userId=${userId}&_expand=room`);
export const deleteBooking = (id) => api.delete(`/bookings/${id}`);
export const updateBooking = (id, booking) => api.patch(`/bookings/${id}`, booking);
export const createDiningReservation = (reservation) => api.post('/diningReservations', reservation);
export const getDiningReservations = () => api.get('/diningReservations');
```

Code Description

The Hotel Booking System is built using React on the frontend and JSON Server as a lightweight backend. The code is organized into clear modules to handle booking, authentication, services, and user management.

1. API Layer (api.js)

Contains all Axios functions for communicating with the backend.
Handles fetching rooms, creating bookings, updating bookings, managing users, and service reservations.

2. Authentication (AuthContext.jsx)

Uses React Context API to manage login, logout, and user sessions.
Stores user data in LocalStorage for persistent login.

3. Routing (App.jsx)

Defines all application routes like Home, Rooms, Bookings, Dashboard, Login, Register, My Orders, and Service Booking.

4. Booking Module

- BookingForm.jsx – Creates new bookings, validates dates, calculates price.
- Bookings.jsx – Shows user bookings, allows cancellations.

- EditBooking.jsx – Lets users update booking details.

5. Room Module

- Home.jsx – Displays featured rooms and search section.
- RoomDetails.jsx – Shows full room information and booking option.

6. Services & Orders Module

- ServiceBooking.jsx – Books extra hotel services like spa sessions.
- MyOrders.jsx – Shows food orders and dining reservations with cancel options.

7. User Dashboard

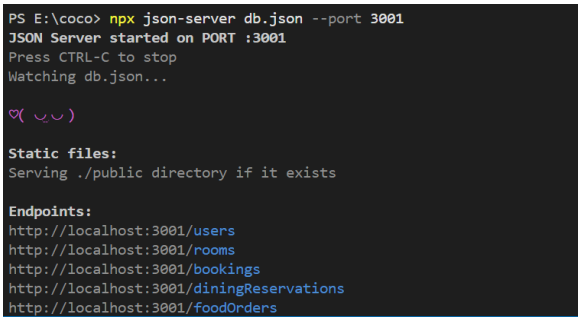
Displays user profile, quick links, and allows profile updates and logout.

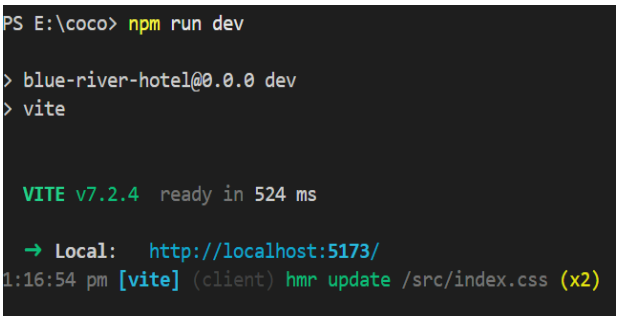
8. Database

Stores all data: users, rooms, bookings, dining reservations, and food orders.
JSON Server automatically provides REST APIs for CRUD operations.

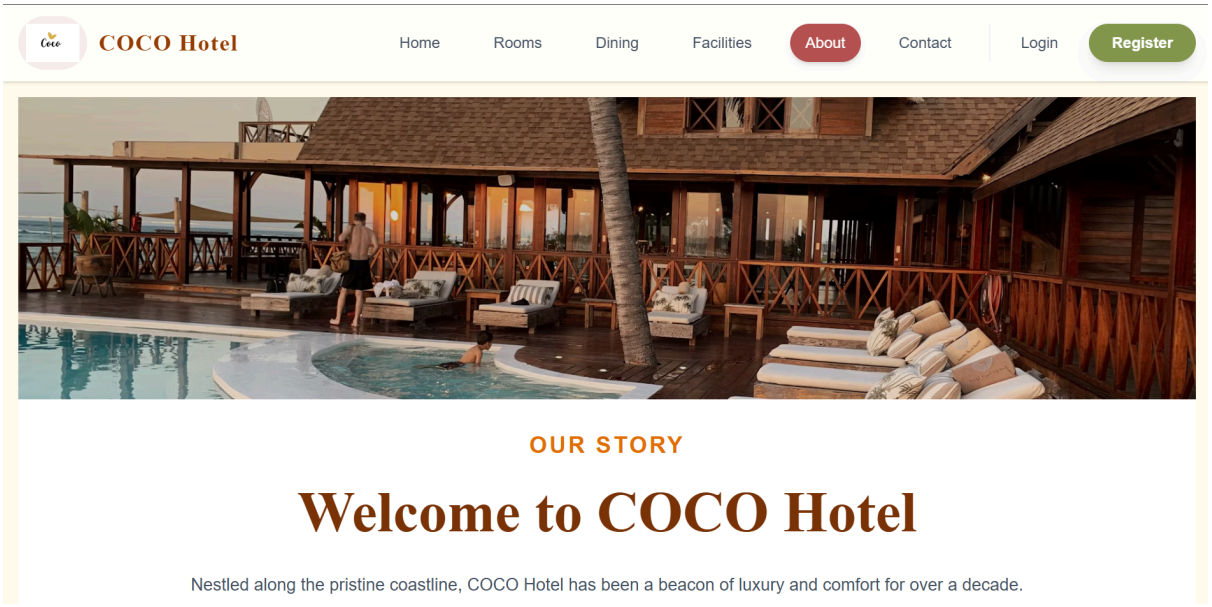
Project Execution

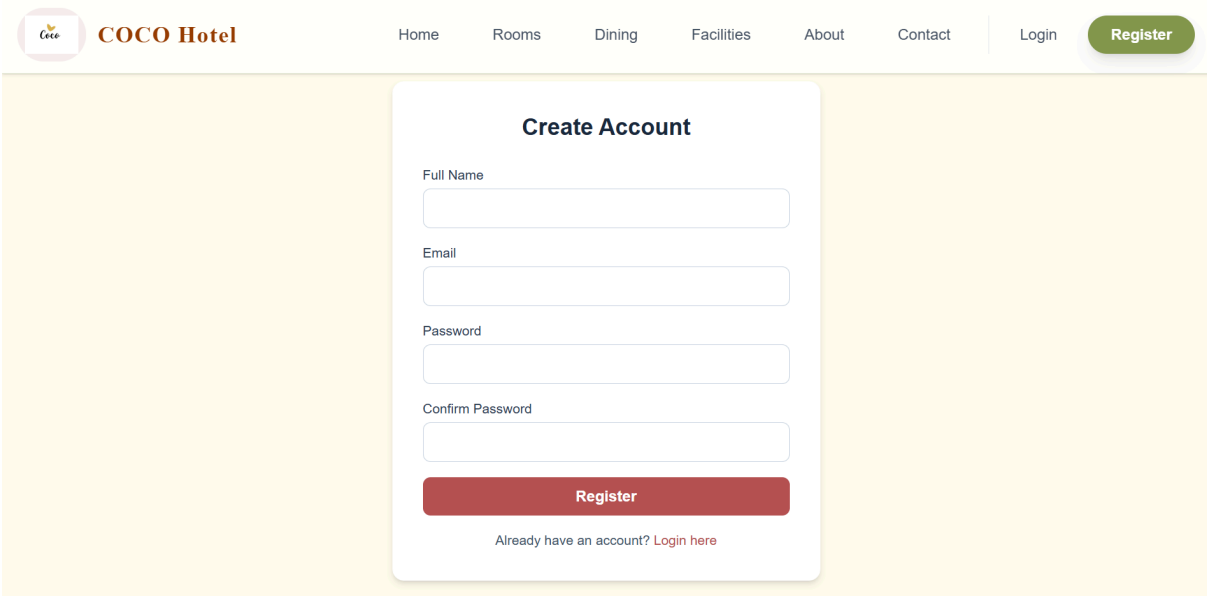
After completing the code, run your db.JSON server using “npx json-server db.json –port 3001” and start the frontend using “npm run dev”.
These commands will launch your JSON Server API and your Vite React application.

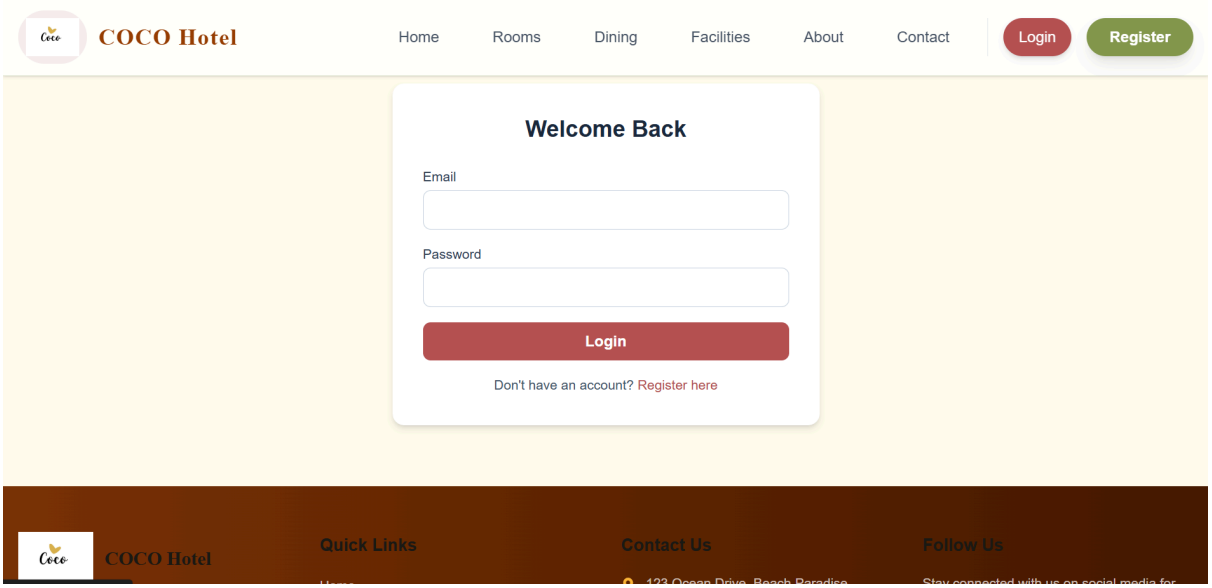


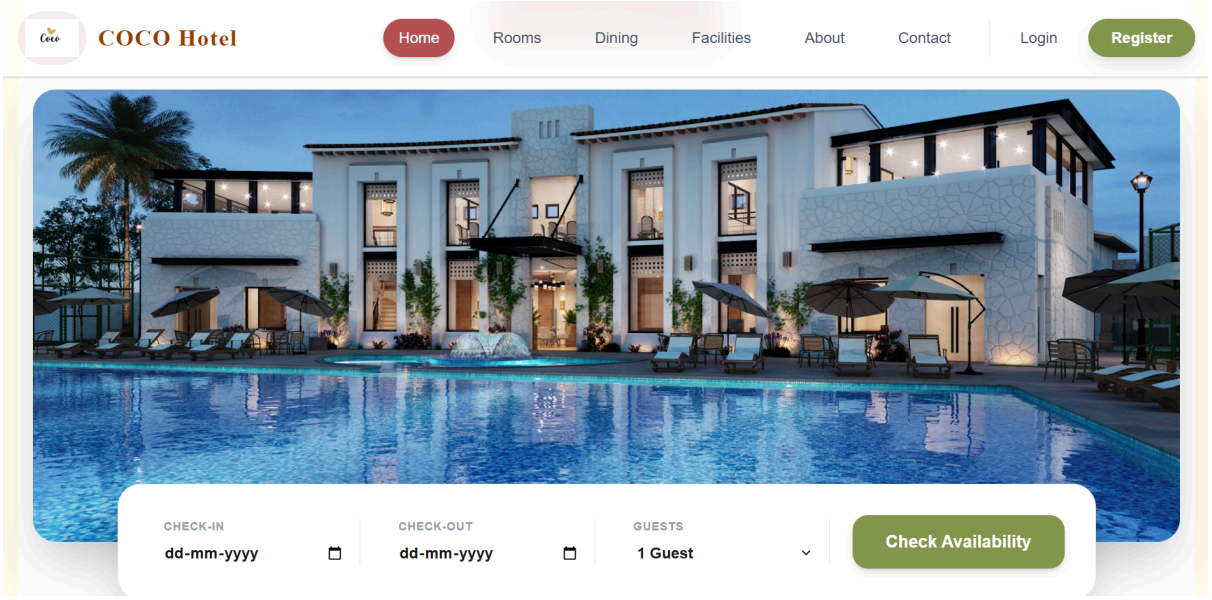


OUTPUT


COCO Hotel
Home
Rooms
Dining
Facilities
About
Contact
Login
Register
OUR STORY
Welcome to COCO Hotel
Nestled along the pristine coastline, COCO Hotel has been a beacon of luxury and comfort for over a decade.


COCO Hotel
Home
Rooms
Dining
Facilities
About
Contact
Login
Register
Create Account
Full Name
Email
Password
Confirm Password
Register
Already have an account? Login here


COCO Hotel
Home
Rooms
Dining
Facilities
About
Contact
Login
Register
Welcome Back
Email
Password
Login
Don't have an account? Register here
COCO Hotel
Quick Links
Contact Us
Follow Us


COCO Hotel
Home
Rooms
Dining
Facilities
About
Contact
Login
Register
CHECK-IN
dd-mm-yyyy
CHECK-OUT
dd-mm-yyyy
GUESTS
1 Guest
Check Availability

COCO Hotel

Home | Rooms | Dining | Facilities | About | Contact | Login | Register

Our Luxurious Rooms

Filters

Max Price: ₹150000

Guests: Any

Room Type: All Types

AC / Non-AC: Any | AC | Non-AC

Deluxe Ocean View ₹16600

A beautiful room with a stunning view of the ocean. Perfect for couples.

2 Guests | AC | WiFi

View Details

Executive Suite ₹29050

Spacious suite with separate living area and premium amenities.

4 Guests | AC | WiFi

View Details

COCO Hotel

Home | Rooms | Dining | Facilities | About | Contact | Login | Register

Breakfast | Lunch | Dinner

Royal Eggs Benedict

Poached eggs on toasted English muffin with smoked salmon and hollandaise sauce.

ORDER NOW

Belgian Waffles Deluxe

Crispy waffles served with fresh berries, whipped cream, and maple syrup.

ORDER NOW

Continental Platter

Assortment of fresh pastries, fruits, cheeses, and cold cuts.

ORDER NOW

COCO Hotel Home Rooms Dining Facilities About Contact Dashboard SomeOne LOGOUT

Manage Bookings

New Booking My Bookings

Deluxe Ocean View

2025-12-01 - 2025-12-05 1 Guests

Total: ₹66400

Confirmed

Deluxe Ocean View

2025-12-06 - 2025-12-19 2 Guests

Total: ₹215800

Confirmed

COCO Hotel Home Rooms Dining Facilities About Contact Dashboard SomeOne LOGOUT

My Dashboard

SomeOne

alice@gmail.com

Edit Profile

Logout

My Bookings

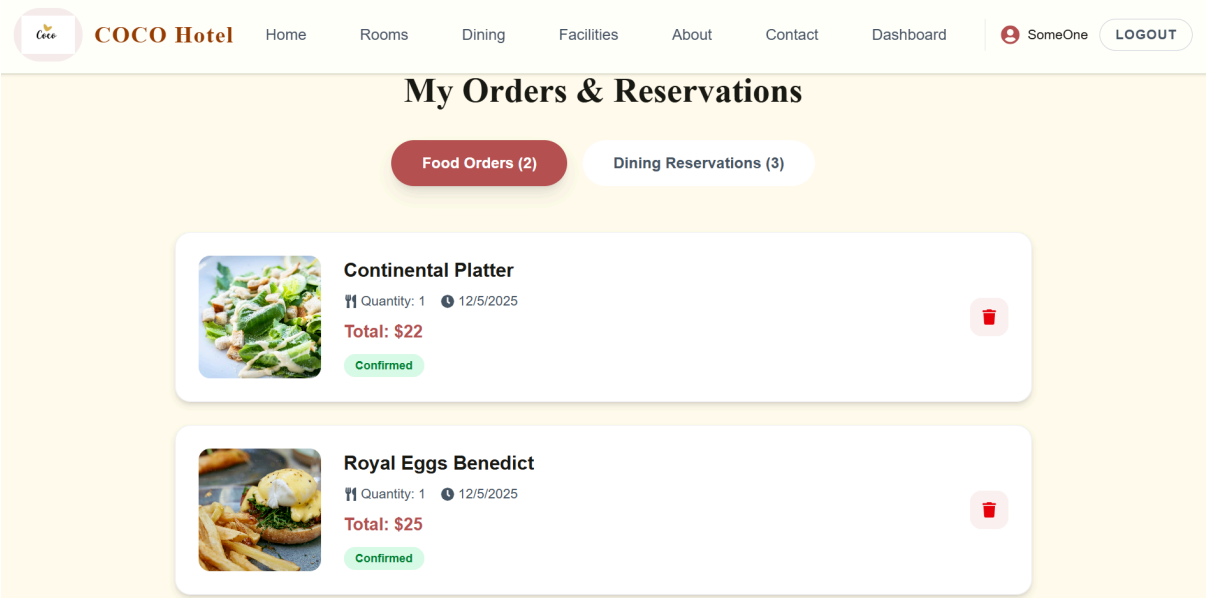
View and manage your room bookings.

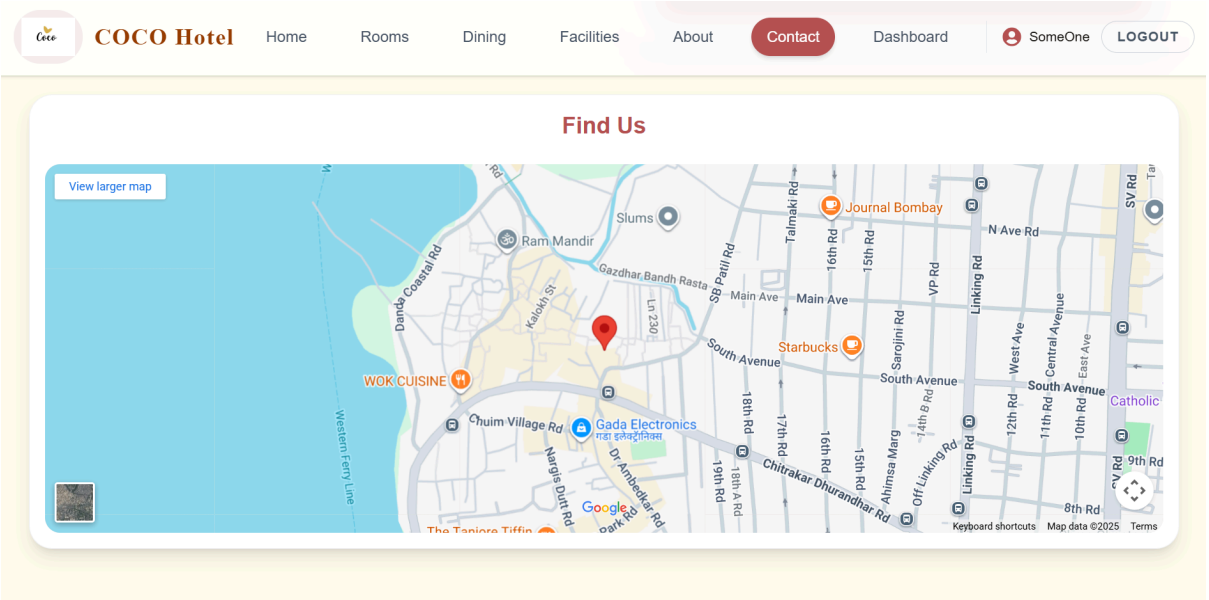
My Orders

View food orders and dining reservations.

Browse Rooms

Explore our luxury rooms.





Project Demo Link:
https://drive.google.com/file/d/1iemh8T7ESFOFqC-2xXDrFisUpStA1701/view?usp=sharing

Project Code Explanation Link:
https://drive.google.com/file/d/1SHTjjuZrHYcPVcrDN0zrO8SCSKMDgkpO/view?usp=sharing

Github Link:
https://github.com/DurgaSravanthiP/COCO_HotelBookingSystem