

Blue River 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

AP23110010410 - Surikuchi Phani sri sowmya

AP23110011349 - Kavali Sindhu

AP23110011584 - Vuuyuru Shruthi Deepika

AP23110011657 - Penugonda Devi Harshitha



Under the Guidance of

Yatharth Shahrawat

SRM University-AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[December, 2025]

Acknowledgement

We express our sincere gratitude to Mr. Yatharth Shahrawat, our respected faculty, for his continuous guidance, valuable insights, and encouragement throughout the course which helped us in developing the project *Blue River Hotel Booking System*. His expertise and support played an important role in helping us understand the concepts of full-stack development and successfully implement this system.

We would also like to thank our department and institution for providing the necessary resources and a supportive learning environment that enabled us to work effectively on this project. Finally, we extend our appreciation to our peers and everyone who contributed directly or indirectly to the completion of this work.

This project has been a valuable learning experience, allowing us to gain practical knowledge in React.js, JSON Server, state management, and modern web development practices.

Team members

AP23110010410 - Surikuchi Phani Sri Sowmya

AP23110011349 - Kavali Sindhu

AP23110011584 - Vuyyuru Shruthi Deepika

AP23110011657 - Penugonda Devi Harshitha

Table of contents

S.NO.	Contents	Page no.
1	Introduction	4
2	Scenario based introduction	5
3	Target audience	6
4	Project goals and objectives	7
5	Prerequisites	8
6	Project structure	11
7	Project flow	14
8	Code description	15
9	Project execution	19
10	Outputs	20
11	Conclusion	25
12	References	26

1. Introduction

Traveling and booking accommodations is a universal experience, yet navigating the modern digital space for hotel reservations can often be frustrating. Despite the multitude of hotel booking websites and platforms available today, users frequently face inconsistent layouts, hidden pricing, cluttered interfaces, and complex booking processes. Modern travelers also expect additional features that make planning easier, such as room filtering, service bookings, real-time availability, pricing calculations, and secure account management.

The growth of web applications has created opportunities to simplify and enhance everyday tasks. Among these, digital hotel booking systems are particularly valuable. However, many existing platforms either focus only on listing rooms or overwhelm users with unnecessary content and advertisements, disrupting the booking flow. Very few systems offer a combination of structured room presentation, interactive utilities, personalized booking management, and smooth navigation within a single interface.

To address these limitations, Blue River Hotel Booking System has been developed as a modern, user-friendly, and feature-rich web application. Built with React and a JSON-based backend, the platform offers a clean and organized hotel exploration experience.

This project demonstrates how thoughtful UI/UX design, modular React components, and modern web technologies can transform hotel booking into a guided, interactive, and enjoyable experience. The Blue River Hotel Booking System lays the foundation for a scalable digital hospitality platform that can expand into complete travel planning, service management, and personalized guest experiences in the future.

2.Scenario-Based Introduction

Imagine a traveler planning a weekend getaway, unsure where to stay or which hotel suits their preferences. They can open the Blue River Hotel Booking System. Within a few clicks, they can explore available rooms, filter by type (Suite, Deluxe, Budget), check prices, amenities, and availability, and instantly view detailed room descriptions with high-quality images.

Meanwhile, the user can book rooms directly through the platform, select check-in and check-out dates, calculate total pricing automatically, and even reserve additional hotel services such as spa sessions, private beach access, or dining packages. Features like dynamic filtering, search, and responsive design make browsing effortless on any device, while secure login ensures personalized booking management and easy access to past reservations.

Hotel administrators can also use the system to manage room details, update availability, and monitor bookings efficiently through the backend. Notifications appear in real-time to confirm bookings, cancellations, or errors, keeping both users and admins informed.

The Blue River Hotel Booking System simulates a real-world hotel booking experience and provides an integrated digital solution for discovering, reserving, and managing hotel stays easily and efficiently. By combining modern web technologies, thoughtful UI/UX design, and interactive components, it transforms the traditional hotel reservation process into a guided, enjoyable, and seamless experience.

3.Target Audience

The Blue River Hotel Booking System is designed to cater to a wide range of users involved in the travel and hospitality domain. The primary target audiences include:

1. Travelers:

Individuals or families planning vacations or weekend getaways who want a simple, fast, and interactive platform to browse available rooms, compare amenities, and make bookings without visiting multiple websites or making phone calls.

2. Business Travelers:

Professionals seeking convenient accommodations with specific facilities like high-speed WiFi, conference rooms, and quick service bookings. The platform allows them to filter and select rooms efficiently and manage bookings easily.

3. Hotel Service Users:

Guests looking for additional services such as spa sessions, private beach access, or dining packages. The integrated service booking feature helps them plan and reserve services alongside their stay.

4. Tech-Savvy Travelers:

Users who prefer modern, mobile-friendly, and interactive web applications with responsive design, real-time notifications, and personalized booking management.

By addressing the needs of these diverse users, the Blue River Hotel Booking System provides a seamless and enjoyable experience for both customers and hotel administrators, ensuring convenience, transparency, and efficiency in the hotel booking process.

4. Project Goals and Objectives

The Blue River Hotel Booking System is designed to provide a seamless, interactive, and efficient platform for hotel reservations and service management. The main goals and objectives are:

Seamless User Experience:

Create an intuitive and responsive interface that allows users to explore hotel rooms, view amenities, and make bookings effortlessly. The platform ensures easy navigation, clear presentation of information, and smooth interaction across all devices.

Smart Booking and Service Management:

Enable users to search, filter, and book rooms and additional hotel services such as spa sessions, private beach access, or dining packages. The system supports automatic pricing calculations, date selection, and personalized booking management.

Efficient Admin and Data Handling:

Provide hotel staff with tools to manage room details, update availability, and monitor bookings effectively through a structured backend. This ensures accurate, real-time data for both users and administrators.

Modern Web Technologies:

Utilize **React.js** for a modular, component-based frontend, combined with a JSON-based backend for data storage. Advanced features such as Context API for weather fetching and toast notifications create a scalable, maintainable, and interactive platform.

By achieving these goals, the Blue River Hotel Booking System aims to simplify the hotel reservation process, improve transparency, and enhance the overall experience for both travelers and hotel administrators.

5. Prerequisites

Before running the Blue River Hotel Booking System, the following tools, software, and technical knowledge are required. Since the project uses React.js for the frontend and a JSON file (app.db.json) as a mock backend, the setup is lightweight and beginner-friendly, without requiring any Node.js server or SQL database.

Code Editor (VS Code Recommended)

A professional code editor is required to write and manage your project files:

- Visual Studio Code is ideal because it supports JSX, modern JavaScript, React components, and JSON formatting.
- Useful extensions include Prettier, ESLint, React Snippets, and Live Server for quick testing.

Web Browser (Chrome / Firefox / Edge)

A modern browser is necessary to run and debug your React-based hotel booking application:

- Google Chrome is preferred due to strong DevTools and React Developer Tools extension.
- Firefox or Edge can be used for cross-browser UI testing.

React.js Framework Knowledge

Since the Blue River Hotel Booking System is built with React, developers should understand:

- Functional components and JSX syntax

- Props and State
- Hooks like `useState`, `useEffect`, and `useContext`
- Routing using React Router v7
- Event handling and dynamic rendering
- Conditional rendering and list rendering using `.map()`

API & JSON Data Handling

Your project fetches data from a local JSON file (`app.db.json`), so developers should know:

- Reading and writing data using `fetch` or similar mock API calls
- Handling JSON responses
- Performing basic CRUD operations on the JSON file in development/testing:
 - `GET` → fetch all rooms, users, or bookings
 - `POST` → add a new booking or user
 - `PATCH` → update booking or room details
 - `DELETE` → remove bookings

Basic HTML, CSS, and JavaScript Skills

Even though React handles most of the UI, developers should know:

- Writing HTML-like structure using JSX
- Styling using Tailwind CSS or standard CSS
- JavaScript ES6+ features such as:
 - Arrow functions
 - Array methods (`map`, `filter`, `find`)
 - Conditional logic and dynamic rendering
- These skills are essential for managing rooms, bookings, filters, forms, and UI updates.

System Requirements

To run the Blue River Hotel Booking System smoothly:

- Minimum 4GB RAM (8GB recommended)
- Dual-core processor
- 5GB free disk space
- Stable internet connection for downloading project files and testing the app

6. Project Structure

```
✓ blue-river-hotel
  > node_modules
  ✓ public
    logo.png
  ✓ src
    > assets
    > components
    > context
    > pages
    ✓ services
      JS api.js
      # App.css
      App.jsx
      # index.css
      main.jsx
      .gitignore
      {} db.json
      O eslint.config.js
      <> index.html
      {} package-lock.json
      {} package.json
      JS postcss.config.js
      i README.md
      JS tailwind.config.js
      ⚡ vite.config.js
```

The Blue River Hotel Booking System project is organized into frontend and backend components for modularity, scalability, and ease of maintenance.

Frontend (React.js)

Contains all UI elements, pages, and state management for the hotel booking system.

- **Components:**

- Navbar, Footer, RoomCard, Toast, WeatherWidget
- Booking forms, search/filter inputs, carousel, and interactive buttons

- **Pages:**

- Home, Rooms, RoomDetails, Bookings, ServiceBooking, Login, Register

- **Contexts:**

- AuthContext — manages global user authentication state
- ToastContext — manages global toast notifications for success/error/info messages

- **Services:**

- `api.js` — Axios instance for API requests to JSON Server
- Helper functions for date calculations, booking management, and localStorage handling

Backend (JSON Server)

- db.json stores hotel rooms, user data, and bookings
- JSON Server handles CRUD operations for rooms and bookings without needing a full-fledged database

Assets (Media Files)

- Images for rooms, hero carousel, facilities, and services
- Icons used in React components (react-icons)
- Toast notification icons and weather widget graphics

Configuration Files

- package.json — project dependencies, scripts, and dev commands
- vite.config.js — Vite build and development configuration
- .gitignore — files and folders to be ignored by Git
- index.html — root HTML file for React
- db.json — local JSON database for mock API

7.Project Flow

ProjectDemo:https://drive.google.com/file/d/1CwMuHGKrCAc-74AIIZ38CAZ_RMpOJKDX/view?usp=sharing

Demo Link (code):

<https://drive.google.com/file/d/1upCB1jHc2govjzmPHxzImFc5kl7KWT1J/view?usp=sharing>

App.jsx component:

```
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import { AuthProvider } from './context/AuthContext';
import { ToastProvider } from './context/ToastContext'; // NEW: Our custom toast system
import Navbar from './components/Navbar';
import Footer from './components/Footer';
import Home from './pages/Home';
import Login from './pages/Login';
import Register from './pages/Register';
import Rooms from './pages/Rooms';
import RoomDetails from './pages/RoomDetails';
import Bookings from './pages/Bookings';
import ServiceBooking from './pages/ServiceBooking';

function App() {
  return (
    <AuthProvider>
      <ToastProvider> {/* NEW: Wrap app with ToastProvider */}
        <Router>
          <div className="min-h-screen bg-slate-50 flex flex-col font-sans">
            <Navbar />
            <main className="flex-grow container mx-auto px-4 py-8">
              <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="/bookings" element={<Bookings />} />
                <Route path="/service-booking" element={<ServiceBooking />} />
              </Routes>
            </main>
            <Footer />
          </div>
        </Router>
      </ToastProvider>
    </AuthProvider>
  );
}
```

8.Code Description - App.jsx

Imports all essential libraries

- **BrowserRouter, Routes, Route (react-router-dom):**
Used to handle navigation between different pages of your hotel booking system.
- **AuthProvider:**
Provides global authentication state (login, logout, user info) to the entire application.
- **ToastProvider:**
Custom toast/notification system used to show success, error, and alert messages to users.

Imports All Components & Pages

Components:

- Navbar – Displays navigation links across the top of every page.
- Footer – Fixed footer that appears at the bottom of all pages.

Pages:

These represent the main screens of the Blue River Hotel Booking System:

- Home – Landing page showing offers, highlights, and a welcome display.
- Login – User login page for authentication.
- Register – New user signup/registration page.

- Rooms – Page that displays all available rooms with filtering options.
- RoomDetails – Detailed page for a specific room, selected using dynamic ID.
- Bookings – Shows all bookings made by the logged-in user.
- ServiceBooking – Allows users to book additional hotel services (spa, dining, beach access, etc.).

Each page is a separate UI screen loaded based on the URL route.

App Component Structure

AuthProvider (Outer Wrapper)

- Wraps the entire app to ensure authentication data is accessible everywhere.
- Any component can check login status or get the currently logged-in user.

ToastProvider

- Wraps the UI to enable global toast notifications.
- Used for showing:
 - “Room booked successfully”
 - “Login failed”
 - “Service added to booking”
 - etc.

Router Wrapper

- <Router> enables page switching without reloading the website.
- Used to render different pages based on URL paths.

Layout Structure

Inside the Router:

Navbar

Displays at the top on every page.

<main> Section

This is the main content area where all pages load.

Routing System (Routes Block)

Each <Route /> loads a different page based on the URL:

- "/" → Home Page
Displays intro, offers, hotel info.
- "/login" → Login Page
Allows users to sign in.
- "/register" → Register Page
New users can create an account.
- "/rooms" → Rooms List
Shows available rooms with filters.

- "/rooms/:id" → RoomDetails Page
Loads room details dynamically using the room ID from the URL.
- "/bookings" → My Bookings Page
Shows all bookings and history of the logged-in user.
- "/service-booking" → Service Booking Page
Allows users to book spa, dining, beach access, and other services.

Footer

- Displays at the bottom of all website screens.
- Also displays the contact information of hotel
- Social media pages of hotel

9. Project Execution

After completing the entire frontend and backend implementation of the Website, follow these steps to run the project successfully:

1. Start the Backend (JSON Server API):

```
PS C:\Users\spss0\OneDrive\Desktop\FS> cd blue-river-hotel
PS C:\Users\spss0\OneDrive\Desktop\FS\blue-river-hotel> npx json-server db.json --port 3001
JSON Server started on PORT :3001
Press CTRL-C to stop
Watching db.json...

♡( ͡° ͜ʖ ͏ )
```

Index:
<http://localhost:3001/>

2. Start the Frontend (React + Vite Application):

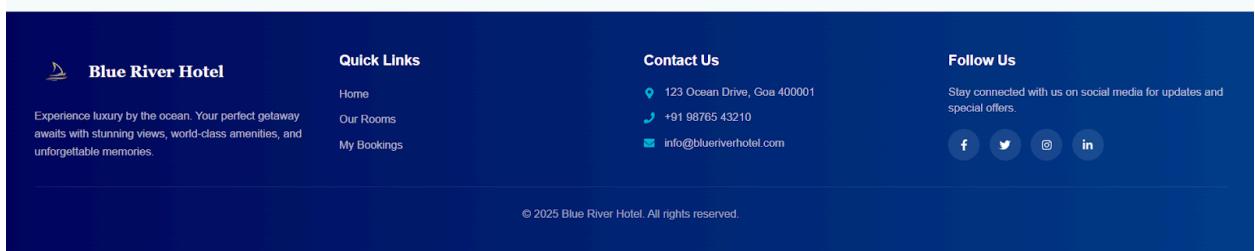
```
PS C:\Users\spss0\OneDrive\Desktop\FS\blue-river-hotel> npm run dev
> blue-river-hotel@0.0.0 dev
> vite

VITE v7.2.4 ready in 1048 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

10. Outputs

Footer:



Navbar:

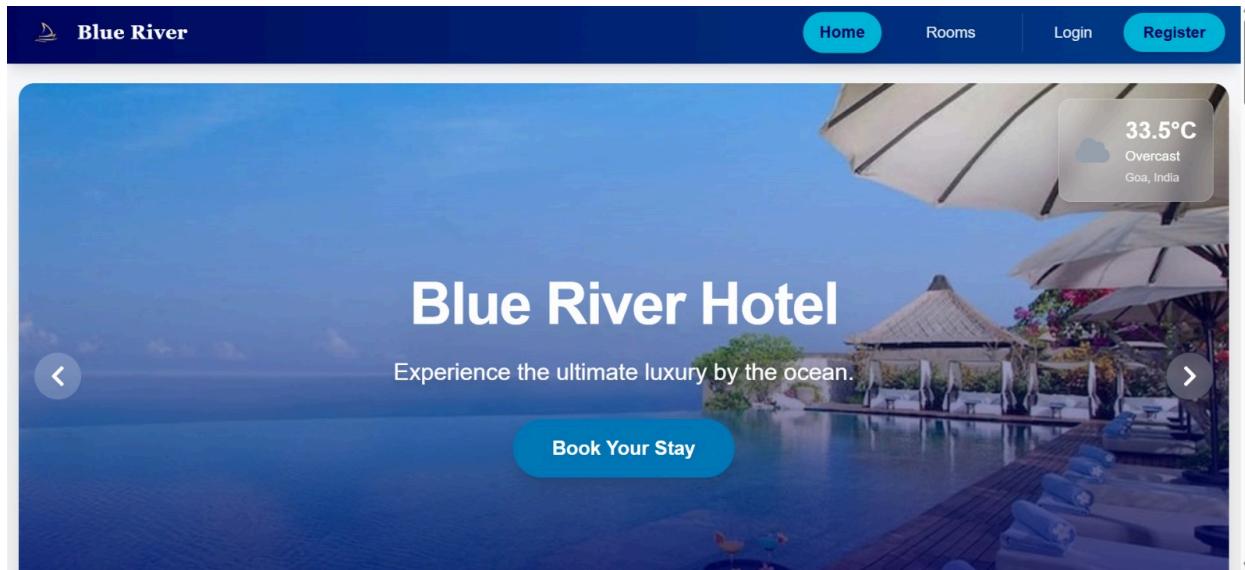


Room Cards:

The page displays three room cards side-by-side. Each card includes a photo, price, room type, guest information, and a "View Details" button.

- Deluxe Ocean View**
DELUXE
₹16,600/night
A beautiful room with a stunning view of the ocean. Perfect for couples.
2 Guests | 5 Amenities
[View Details](#)
- Executive Suite**
SUITE
₹29,050/night
Spacious suite with separate living area and premium amenities.
4 Guests | 6 Amenities
[View Details](#)
- Standard Room**
STANDARD
₹9,960/night
Comfortable room for a relaxing stay. Great value.
2 Guests | 3 Amenities
[View Details](#)

Home:



Bookings:

The image shows the "My Bookings" page from the Blue River Hotel website. The page has a dark blue header with the hotel's logo and name, and navigation links for "Home", "Rooms", "My Bookings", "Test User", and "Logout". The main content area is titled "My Bookings". It displays two confirmed bookings in cards:

- Executive Suite**:
Check-In: 8/12/2025 | Check-out: 20/12/2025 | Status: Confirmed
Total: ₹3,48,600 | Edit | Cancel Booking
- Private Beach Access**:
Date: 27/12/2025 | Time: 09:00 AM | Guests: 2 | Status: Confirmed
Total: ₹1,000 | Edit | Cancel Booking

Login page:

The screenshot shows the login page of the Blue River Hotel website. At the top, there is a dark blue header bar with the hotel's logo and name "Blue River". To the right of the logo are navigation links for "Home", "Rooms", "Login", and "Register". The main content area has a white background with a central form titled "Welcome Back". The form contains fields for "Email" and "Password", followed by a "Login" button. Below the button is a link to "Register here". At the bottom of the page, there is a dark footer bar with the hotel's name, "Quick Links" (Home, Our Rooms, My Bookings), "Contact Us" information (address, phone number, email), and "Follow Us" social media links.

Welcome Back

Email

Password

Login

Don't have an account? [Register here](#)

Blue River Hotel

Experience luxury by the ocean. Your perfect getaway awaits with stunning views, world-class amenities, and unforgettable memories.

Quick Links

- Home
- Our Rooms
- My Bookings

Contact Us

- 123 Ocean Drive, Goa 400001
- +91 98765 43210
- info@blueriverhotel.com

Follow Us

Stay connected with us on social media for updates and special offers.

f t i n

Register page:

The screenshot shows the register page of the Blue River Hotel website. At the top, there is a dark blue header bar with the hotel's logo and name "Blue River". To the right of the logo are navigation links for "Home", "Rooms", "Login", and "Register". The main content area has a white background with a central form titled "Create Account". The form contains fields for "Full Name", "Email", "Password", and "Confirm Password", followed by a "Register" button. Below the button is a link to "Login here".

Create Account

Full Name

Email

Password

Confirm Password

Register

Already have an account? [Login here](#)

Rooms:

Blue River

Home Rooms My Bookings Test User Logout

Our Rooms

Search rooms... All All Prices



₹16,600/night

Deluxe Ocean View
DELUXE

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

2 Guests 5 Amenities

View Details



₹29,050/night

Executive Suite
SUITE

Spacious suite with separate living area and premium amenities.

4 Guests 6 Amenities

View Details



₹9,960/night

Standard Room
STANDARD

Comfortable room for a relaxing stay. Great value.

2 Guests 3 Amenities

View Details

Room Bookings:

Blue River

Home Rooms My Bookings Test User Logout



Book This Room

Check-in Date

Check-out Date

Confirm Booking

Standard Room
Standard

Comfortable room for a relaxing stay. Great value.

Amenities

₹9,960
per night

Service Booking:



Private Beach Access

Enjoy exclusive access to our pristine private beach. Whether you want to soak up the sun, take a refreshing dip, or enjoy a romantic sunset walk, our private beach offers the perfect escape.

Professional staff available
 Premium facilities and equipment

Book Your Experience

Select Date
mm/dd/yyyy

Select Time Slot
Choose a time

Number of Guests
1

Special Requests (Optional)
Any special requirements or preferences...

Price per person	₹500
Guests	1
Total Price	₹500

11. Conclusion

The Blue River hotel booking system successfully demonstrates the development of a modern, responsive, and user-friendly hotel reservation platform using React.js and JSON Server. Through this project, we implemented essential features such as room browsing, real-time availability display, secure user authentication, service booking, and a structured backend for managing data. The use of reusable components, Context API for global state management, and RESTful API integration ensured scalability, maintainability, and smooth user interaction.

Working on this project allowed us to gain practical knowledge of full-stack development concepts, routing, state handling, and UI/UX design. The system simplifies the booking process for users while providing hotel administrators with an efficient way to handle room and service information. Overall, the project meets its objectives by offering a seamless booking experience, a clean interface, and a reliable mock backend, making it a strong foundation for future enhancements such as payment integration, advanced admin dashboards, and real hotel database connectivity.

12. References

React.js Official Documentation – <https://react.dev/>

React Router Documentation – <https://reactrouter.com/>

Tailwind CSS Documentation – <https://tailwindcss.com/>

JSON Server Guide – <https://github.com/typicode/json-server>

<https://www.booking.com>