**HOTEL BOOKING SYSTEM**

BY

**M.C.K.AKHILESH -AP23110010203**

**K.ANAND – AP23110011544**

**P.ANISH – AP23110010059**

**V.BHUVANESH – AP23110010009**

SUMBITTED TO

**SYED ARSHAD**



**SRM University-AP, Neerukonda, Andhra Pradesh 522240**

**December 2025**

**HOTEL BOOKING SYSTEM(REACT)**

**Introduction:-**

The Hotel Booking System is a comprehensive web-based application developed to simplify and automate the hotel room reservation process. In today’s fast-paced digital world, customers expect quick, hassle-free, and reliable online booking platforms that provide real-time information about room availability, pricing, amenities, and hotel facilities. Traditional booking methods—such as calling the hotel or visiting physically—are time-consuming, prone to manual errors, and often lead to customer dissatisfaction. This project aims to overcome these limitations by offering a fully automated hotel reservation system that is accessible anytime, anywhere.

This system allows users to easily explore different room categories, read descriptions, view images, compare prices, and select check-in and check-out dates according to their travel schedules. Once users register or log in, they can proceed to book rooms securely through a streamlined booking workflow. The platform stores all booking details in a database, ensuring accurate record-keeping and reducing the workload of hotel staff. Additionally, the system provides an efficient way for hotel management to monitor reservations, manage room availability, update pricing, and maintain a smooth flow of operations.

The project also emphasizes modern web development practices by implementing a clean user interface, responsive design, and robust backend logic. By integrating technologies such as HTML, CSS, JavaScript, and backend frameworks (like PHP or Node.js), the system ensures both functional efficiency and visual appeal. The intuitive navigation helps users complete bookings without confusion, while the backend ensures data integrity and reliability.

Overall, the Hotel Booking System is designed not only to benefit customers by providing a seamless booking experience but also to enhance hotel operations through automation and digital management. This project serves as a practical real-world application of web development concepts, demonstrating how technology can improve traditional business processes and help organizations deliver better services.

**Scenario-Based Intro:-**

Imagine a customer planning a vacation and searching for a hotel room online. Instead of calling the hotel or visiting in person, they simply open the hotel's website, browse through the available rooms, check amenities like Wi-Fi, breakfast, or parking, and instantly book a room.

Simultaneously, the hotel management can view bookings, check room occupancy, and update room availability through an admin portal.

This **Hotel Booking System** replicates such a real-world scenario and provides a complete digital solution for hotel reservations.

**Target Audience:-**

**Hotel Customers**

* Travelers who want quick access to room details and smooth online booking.
* Guests interested in hotel facilities, pricing, and availability.

**Hotel Management**

* Admin staff responsible for managing bookings and room availability.
* Hotel owners who need automated reservation and customer management.

**Coding Enthusiasts & Students**

* Developers who want to understand a full-stack project.
* Students building web-based applications using modern web technologies.

**Project Goals and Objectives:-**

The system demonstrates complete coding fundamentals including:

* Authentication
* Form validation
* API communication
* Database operations
* UI/UX design

**Modern Tech Stack**

* Use of HTML, CSS, JavaScript for frontend
* Backend using PHP / Node.js / Laravel (based on your project)
* MySQL database integration
* Responsive UI compatible with all devices

**Seamless Booking Workflow**

* Customers can search rooms
* Select dates
* Confirm booking
* Receive confirmation

**Key Features:-**

**Live Code Editor (If project includes coding panels or dynamic interface)**

* System supports interactive components.
* Immediate reflection of UI changes (useful if React/JS used).

**Multiple Code Panels (If included)**

* Shows structured frontend, backend, and database modules separately.

**Collaboration Features**

* Allows team members (developers) to work on shared code using git or version control.

**Responsive Design Testing**

* Fully responsive layout using Bootstrap/Tailwind CSS.
* Works smoothly on desktops, mobiles, and tablets.

**Node.js and npm (If part of tech stack)**

* Used for environment setup.
* Handles dependency installation and application build tools.

**React.js (If used for UI)**

* Component-based structure for faster UI rendering.
* State management for handling bookings and user sessions.

**HTML, CSS, and JavaScript**

* HTML → structure
* CSS → styling
* JavaScript → dynamic validations and functionality

**Version Control**

* Git/GitHub used to track changes, manage updates, and collaborate.

**Development Environment**

* VS Code as main IDE
* XAMPP/WAMP for PHP projects OR Node terminal for Node.js applications
* MySQL Workbench/phpMyAdmin for database

**PRE-REQUISITES:-**

Before running the **Hotel Booking System**, the following tools, software, and technical knowledge are required. Since your project is built using **React.js for frontend** and **JSON files for backend data storage**, the setup is simple and does not require MySQL, PHP, or server-side frameworks.

**Code Editor (VS Code Recommended)**

A powerful code editor is required for writing and managing your project files.

* **Visual Studio Code** is ideal because it supports React, JavaScript ES6+, JSX, and JSON handlers.
* It also provides useful extensions like *Prettier*, *ESLint*, and *Live Server*.

**Web Browser (Chrome / Firefox)**

A modern browser is needed to run and inspect your React application.

* **Google Chrome** is preferred due to strong developer tools (React Developer Tools extension).
* You can use Firefox/Edge for cross-browser testing.

**Node.js & npm (Required for React)**

React applications need Node.js to run the development server and build production files.

* **Node.js** provides the runtime environment.
* **npm** (Node Package Manager) installs all libraries and dependencies such as:
  + React
  + React Router
  + State management libraries
  + Axios/fetch for API or JSON requests

You cannot run a React project without Node.js.

**React.js Framework Knowledge**

Since your project is built with **React**, the developer should understand:

* Components (Functional/JSX)
* Props and State
* useState, useEffect Hooks
* Routing (React Router)
* Event handling
* Fecthing   
  This helps build an interactive and responsive user interface.

**JSON File Handling (Your Backend)**

Your project stores booking data, room details, and user details in JSON files.  
So developers should know:

* How to fetch JSON data in React
* How to update JSON (using simple write functions or local server)
* How to simulate backend operations using JSON  
  JSON acts like a *mini-database* in your project.

**Local Development Server (React Scripts)**

React automatically provides a development server using:

npm start

No Apache, PHP, or MySQL is required.

However, if JSON read/write needs special handling, you may use:

* **json-server** npm package
* Custom Node API (optional)
* Or static JSON files loaded directly

**Basic HTML, CSS, and JavaScript Skills**

Though React handles UI, fundamental skills are still necessary:

* HTML structure inside JSX
* CSS modules or global styles
* JavaScript ES6+ syntax (arrow functions, promises, async/await)

Your booking system relies heavily on JavaScript logic, form validations, and dynamic rendering.

**Version Control (Optional but Recommended)**

If you want to manage your project professionally, tools like:

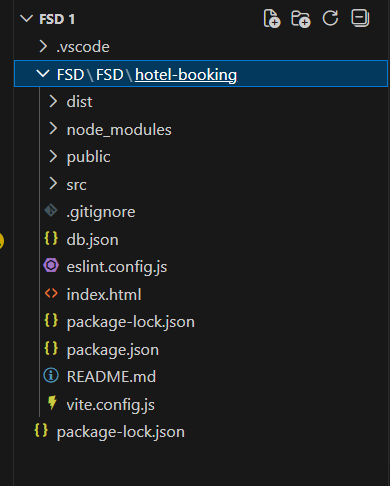
* **Git**
* **GitHub**  
  help track changes, create backups, and collaborate with team

**System Requirements**

To run React smoothly:

* Minimum **4GB RAM** (8GB recommended)
* Dual-core processor
* 5GB free disk space
* Stable internet for installing packages

**Project structure:**

****

The project contains:

* **Frontend (React):** All UI components, pages, and styling files.
* **Backend (JSON Server):** A simple JSON file acting as the database.
* **Assets:** Images, icons, and other media files.
* **Configuration Files:** Package.json, JSON server config, and environment files.

**PROJECT FLOW:-**

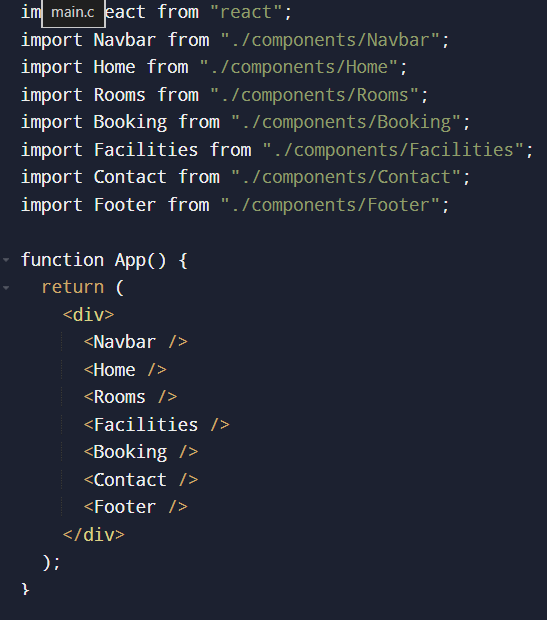
**Project demo:**

Before starting to work on this project, let’s see the demo.

Demolink: [DEMO LINK](https://github.com/AP23110010203/hotel-booking-system/blob/main/README.md)

Use the code in: [code in](https://github.com/AP23110010203/hotel-booking-system)

**App.js component**



**Code Description:-**

**Imports React :**

Required for creating and using React components.

**Imports all components :**

* Navbar
* Home
* Rooms
* Booking
* Facilities
* Contact
* Footer  
  These are different sections of your website

**Defines the App component :**

App is the main/root component of your entire project.

**Returns JSX :**

JSX is the syntax used to write HTML inside React.

**Navbar is shown at the top :**

This acts as the website’s navigation bar.

**Website sections are loaded one after another :**

* <Home /> → Home section
* <Rooms /> → Rooms section
* <Facilities /> → Facilities section
* <Booking /> → Booking form
* <Contact /> → Contact information

**Footer is shown at the bottom :**

Displays website footer details.

**App component is exported :**

export default App; makes the App component available to main.js or index.js to render the whole website.

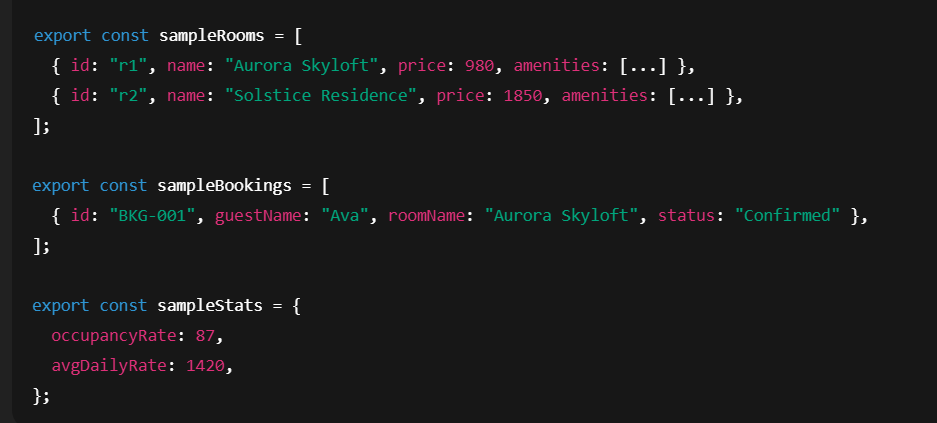
**Design UI components:**

* Create and style visual elements like buttons, forms, and sections.
* Ensure layout is user-friendly and responsive.

**Implement frontend logic:**

* Add interactivity to the UI (e.g., click actions, form validation).
* Connect components with data or state to make them functional.

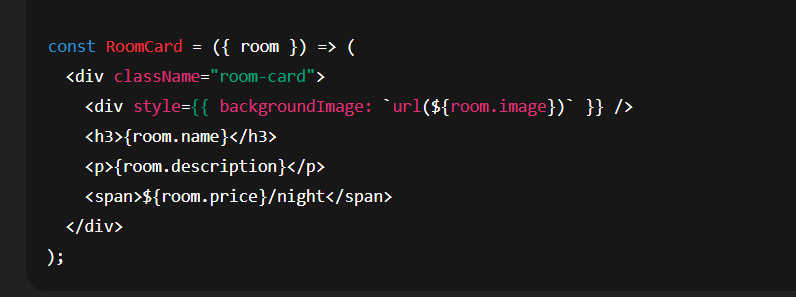
**Data  Provider:-**

****

**Code Description**

* Contains arrays of sample rooms, bookings, and hotel stats.
* Used as a dummy backend for testing.
* Helps UI load data without any API/database.
* Makes development faster and easier.

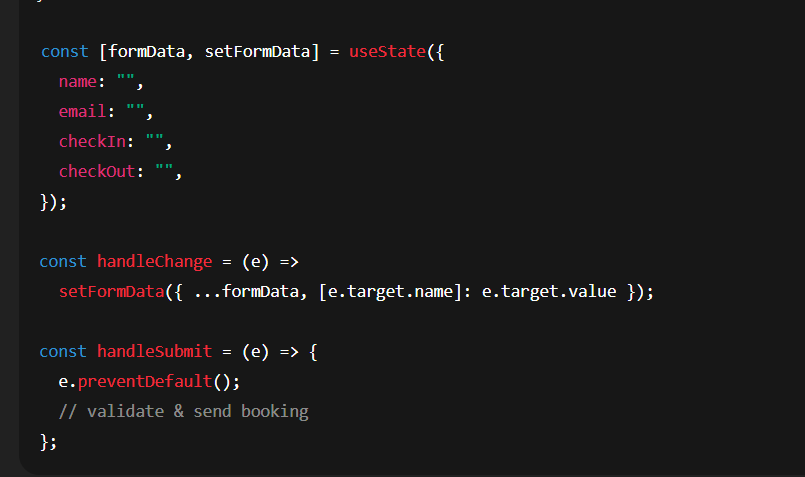
**Code Component:-**



**Code Description**

* Accepts room as a prop and shows room details.
* Used in homepage/rooms page to display each suite.
* Shows room image, title, description, and price.
* Pure UI component with no business logic.

**Code-Editor Component:-**

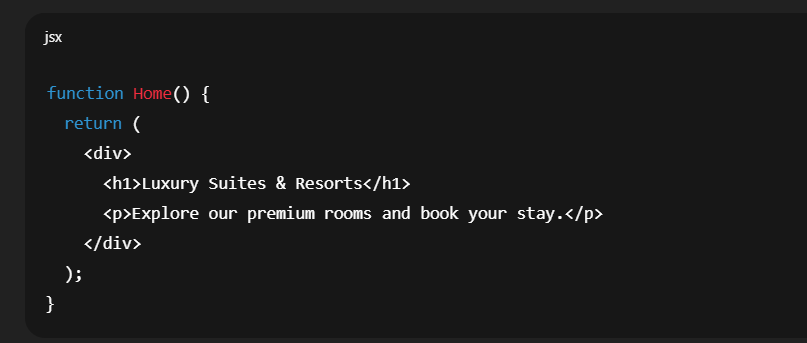
****

**Code Description**

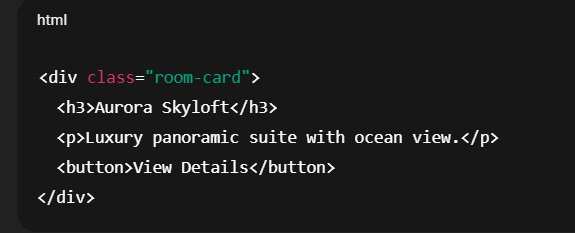
* Uses React state to store user-entered booking data.
* Contains input fields for name, email, dates, guests.
* Validates before submitting.
* Sends data to backend via createBooking() API call.

**Project Execution:**

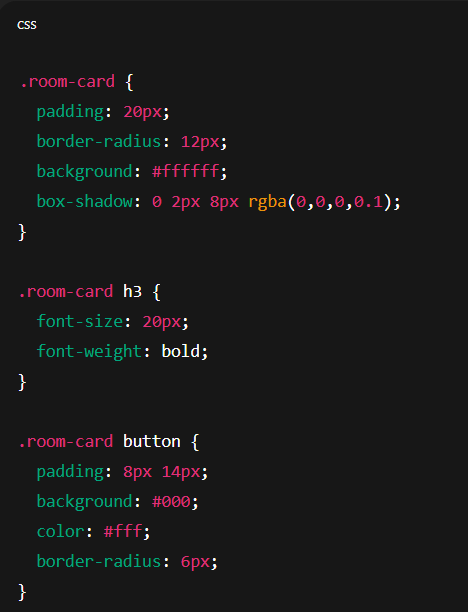
* After completing the code, run the react application by using the command “npm start” or “npm run dev” if you are using vite.js
* Here are some of the screenshots of the application.
* **Home Component :**



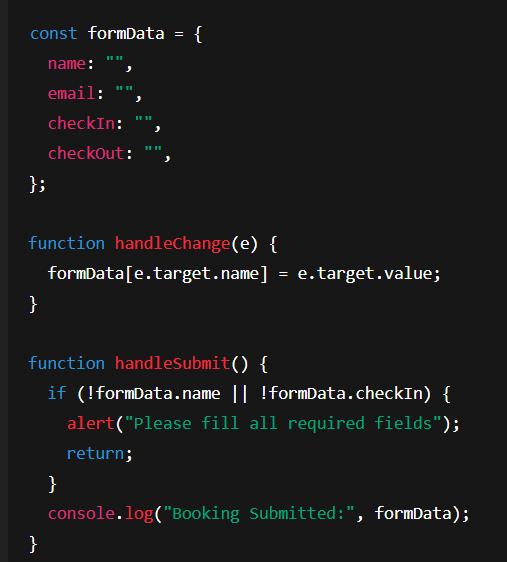
* **Html Code Component**



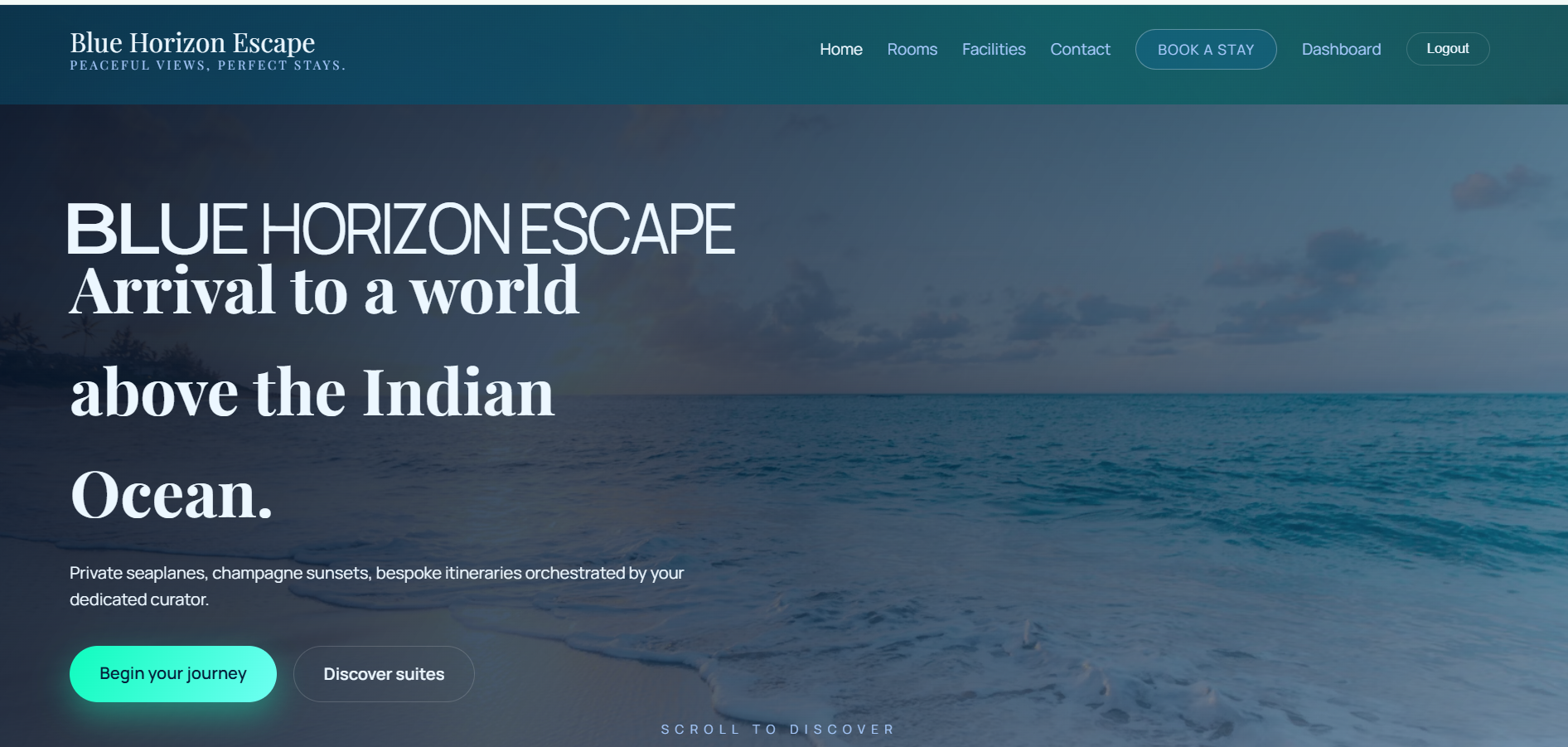
* **Css Code Component**

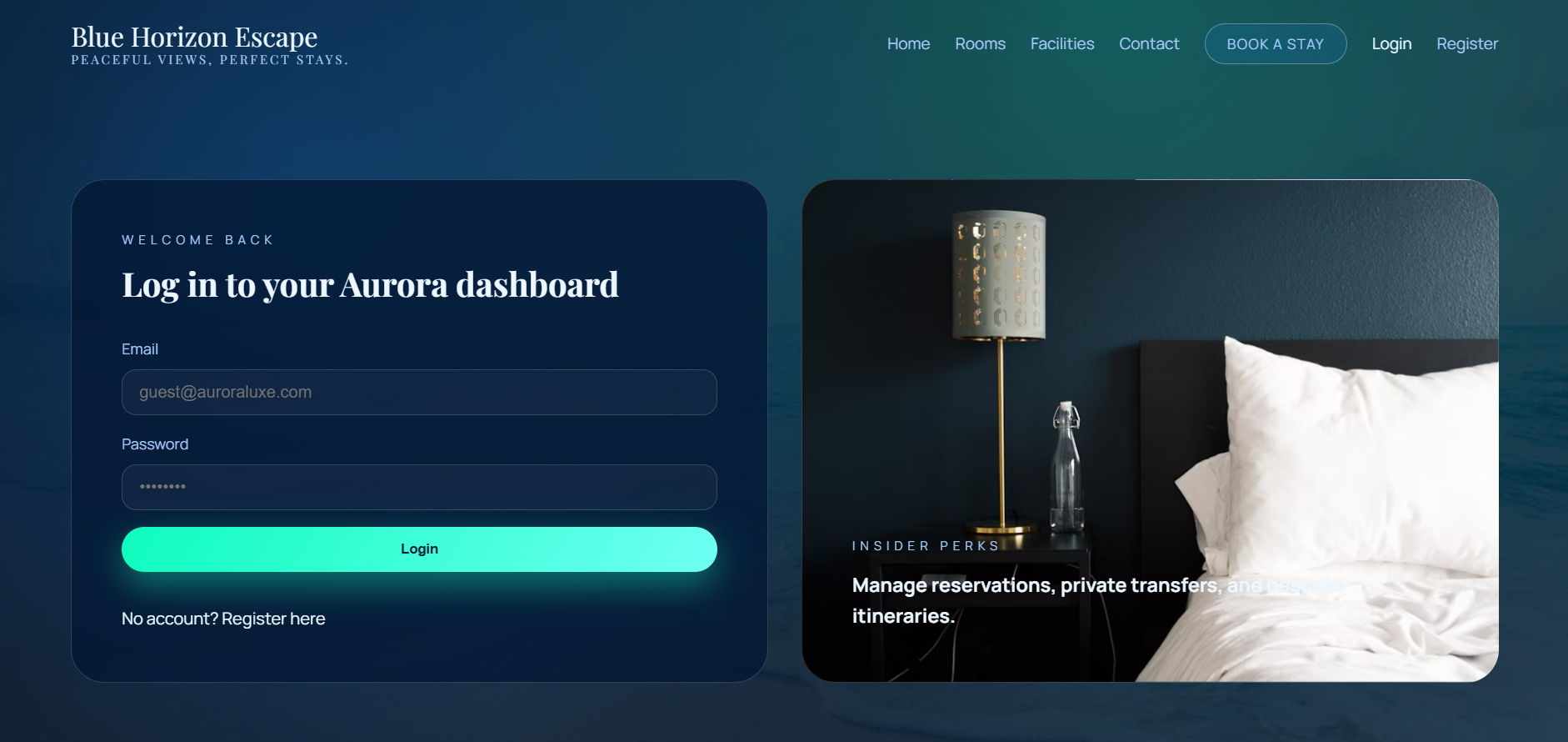


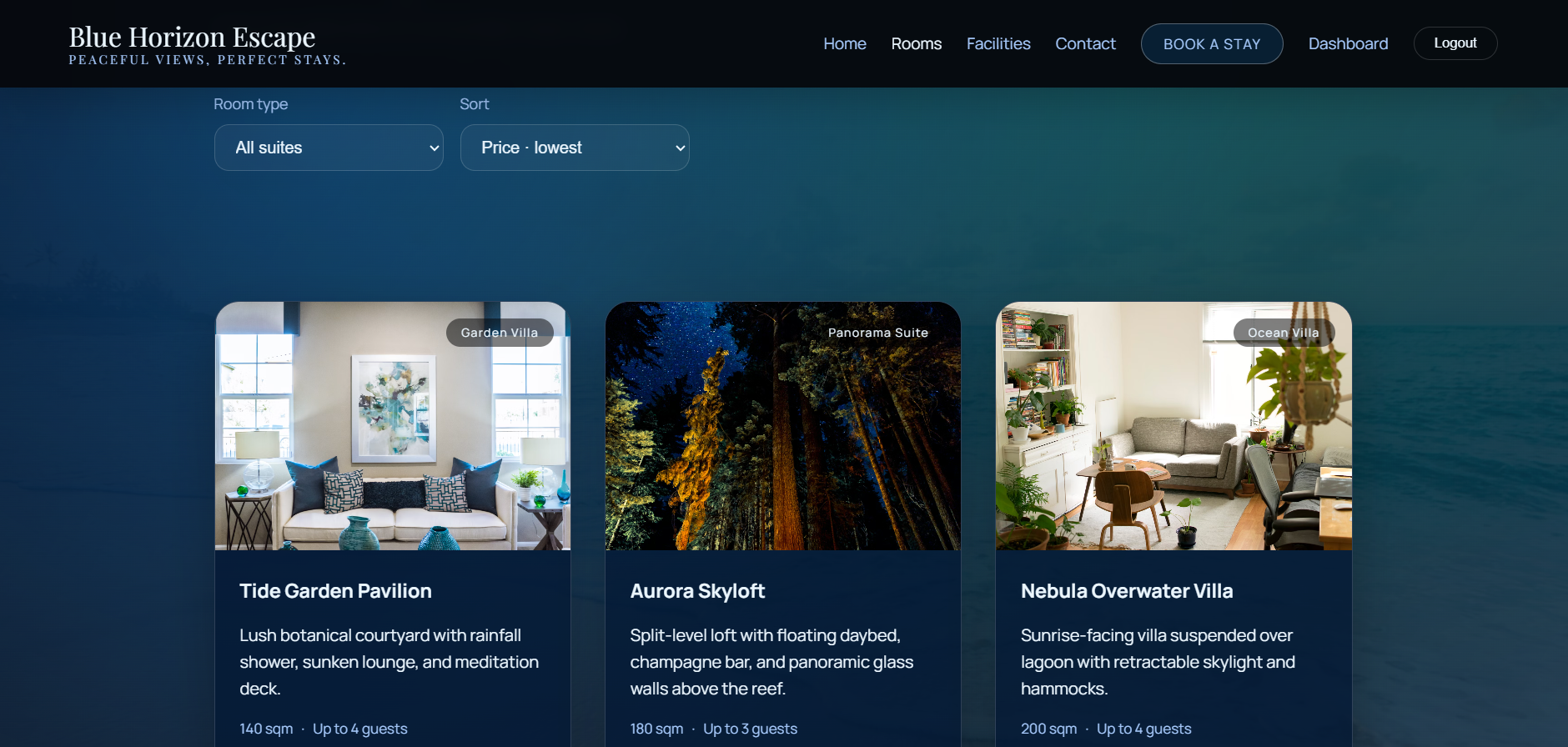
* **Javascript Component**

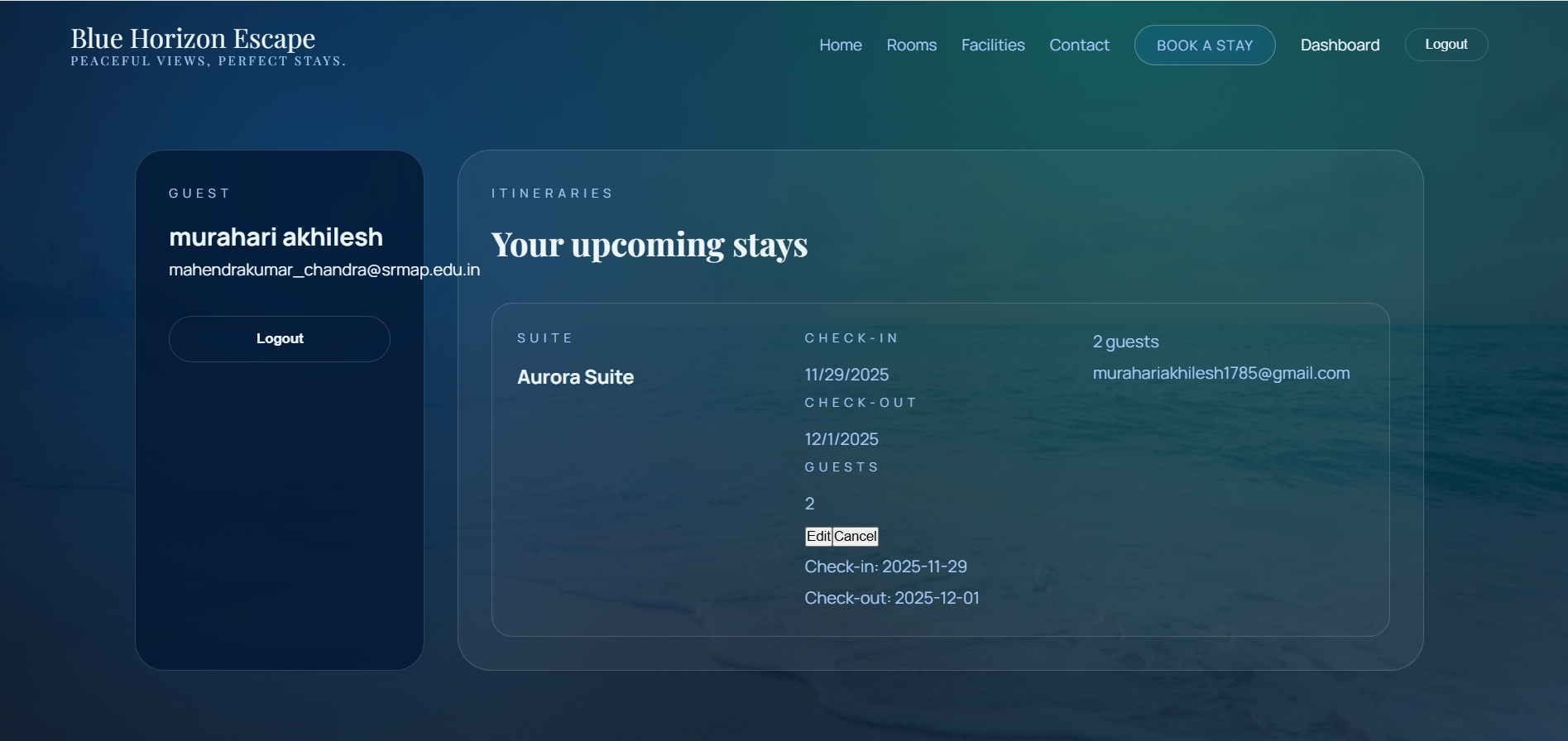


* **OutPut**







****

**Project Demo :**  [DEMO LINK](https://github.com/AP23110010203/hotel-booking-system/blob/main/README.md)