

SOFTWARE DOCUMENTATION

on

HOTEL BOOKING WEB APPLICATION

By Group 32

Adithya Jayadevan (BT20HCS005)

Anirudh Sharma (BT20HCS055)

Chetna Lamba (BT20HCS042)

Iftikar Asif (BT20HCS008)

Muskan Gupta (BT20HCS170)

Faculty Supervisors

Prof. Debashis Sengupta

Prof. Manish Hurkat

Web Application on the topic “Hotel Booking Application using MERN Stack”

Version	1.0
Document Status	Draft
Document Owner	Chetna Lamba
Designer	Muskan Gupta
Developers	Adithya Jayadevan , Anirudh Sharma , Chetna Lamba , Iftikar Asif , Muskan Gupta
Components	Admin (by the name admin1), API (Server side) and Client (client1)
Stakeholders	Hotel owners, Commoners

User Interaction and Design

Description	Login Screen	Activity Stream
-------------	--------------	-----------------

Prototype		
-----------	--	--

System Documentation

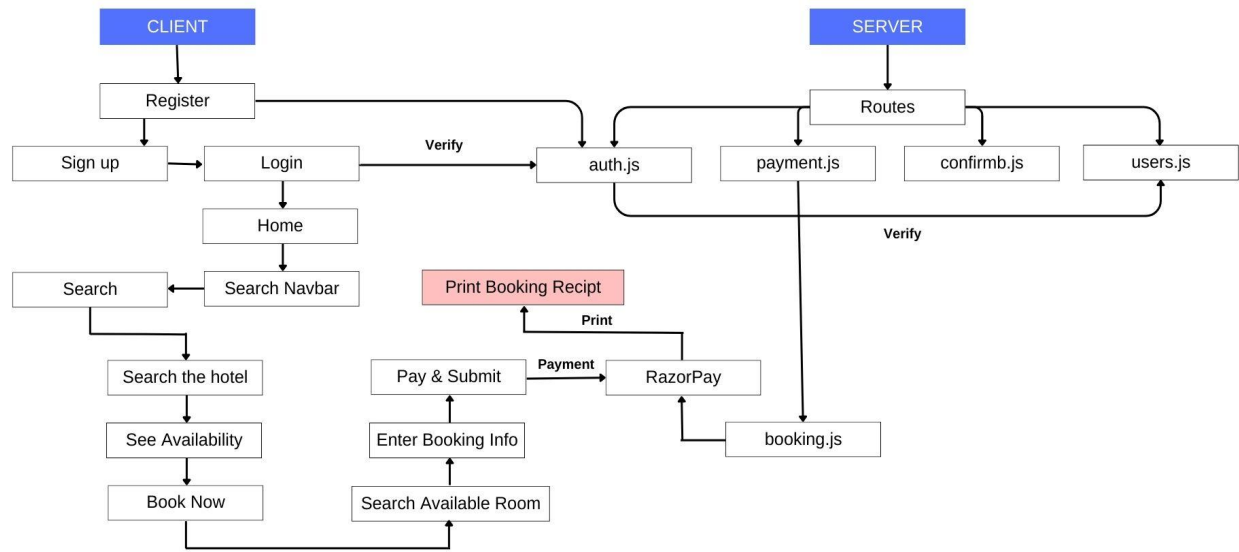
Pages	Working
How to Login/ Register?	<ul style="list-style-type: none"> ● React component represents the registration form for users ● Has input fields for the user's username, mail ID, phone, city, country, password and profile image. ● The component uses the Axios library to send HTTP requests to the server to handle the registration process. If the registration is successful, the user is redirected to the login page. ● The component has a handleChange function that updates the input fields' values whenever the user enters a new value. For login, the component renders an HTML form with two input fields for the user's username and password, respectively. When the user clicks on the login button, the handleClick function is called. This function sends an HTTP POST request to the server with the user's credentials, which is handled by the "/auth/login" route. ● The component also uses the useContext hook to access the AuthContext, which contains the loading and error states, and the dispatch function to update the states. ● The component uses the useState hook to manage the input fields' values and the file state for the image. ● The component also uses the useNavigate hook from the react-router-dom library to handle the routing. ● The component also has a handleClick function that handles the registration process. The function first uploads the user's image to Cloudinary using the Axios library and then sends a request to the server to register the user. If the registration is successful, the function updates the AuthContext's state and redirects the user to the login page. ● The component uses Fontawesome to display icons. ● The component also uses CSS to style the form and the page's layout. ● Server Side -> Routes -> Auth.js -> Post Request ->

	Register and Login Details
While booking the Hotel	<ul style="list-style-type: none"> • This is a React component called Hotel. It imports several modules and components such as Navbar, Header, MailList, Footer, Reserve, and some icons from Font Awesome. • It also imports some contexts such as SearchContext and AuthContext. The component uses useContext to access the values of the contexts. • The component fetches data from the server using useFetch hook and displays it. It also has some functions to handle sliders, dates, and user actions. When the user clicks on the "Reserve or Book Now!" button, it opens a reservation modal. • Server Side ->Routes -> Hotels.js (CRUD of Hotel) and Rooms.js -> Client Side information available dates and rooms
Payment Gateway	<ul style="list-style-type: none"> • There are several event handlers defined such as handleSelect, handleChange, handleClick, handlePayment, handleCopyClick, and Print. • Razorpay -> Payment Integration -> Testing Model -> Payment Success

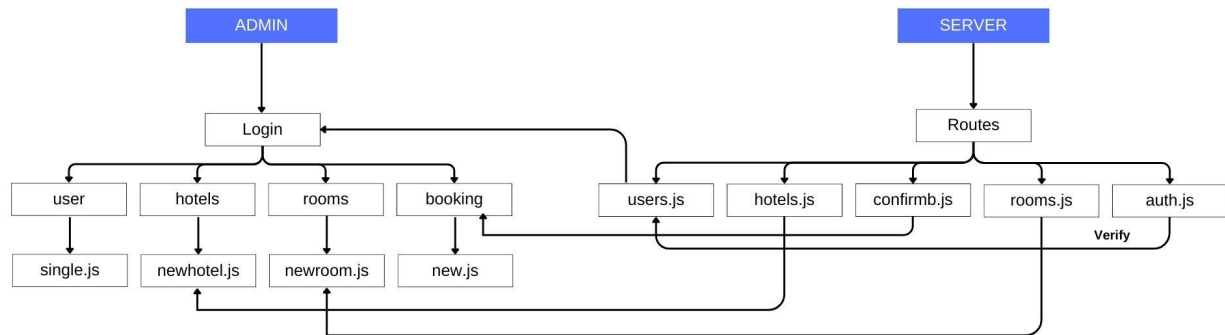
Milestones achieved

- ☐ Google Authentication
- ☐ Forgot password operational
- ☐ Loadrunner- multiple users are able to login and show available rooms at the same time
- ☐ CRUD operations on admin side
- ☐ Show availability/ unavailability of rooms
- ☐ Payment Gateway
- ☐ Print receipt after booking

Flowchart



Client and Server



Admin and Server

Future Aspects

- ☐ Adding more places and properties
- ☐ Adding more security features

- ☐ Making the web application more interactive by adding a chatbot
- ☐ Make the application more seamless by increasing its loading speed
- ☐ Getting a confirmation mail on user's registered mail ID