

MCA Semester – III
Front End Development Project

Name	Ravi Kumar
USN	231VMTR00265
Elective	FULL STACK DEVELOPMENT
Date of Submission	01/11/2024



Front End Development Project - Report

Project Report submitted to Jain Online (Deemed-to-be University)
as part of the course “Front End Development Project”

Master of Computer Applications

Submitted by

Ravi Kumar

USN

231VMTR00265

*Under the
guidance of*

Nikhil Eknathrao Karale

(Faculty-JAIN Online)

DECLARATION

I, *Ravi Kumar*, hereby declare that this Project Report has been prepared by me under the guidance of *Nikhil Eknathrao Karale*. I declare that this Project is towards the partial fulfilment of the credit requirement for the course “Front End Development Project,” which is part of the Master of Computer Applications degree given by Jain University, Bengaluru. I declare that the work done by me towards this Project is original in nature and is my own contribution.

Place: Bangalore

Date: 01/11/2024

Ravi Kumar

Name of the Student

USN: 231VMTR00265

CERTIFICATE

This is to certify that the Project report submitted by Mr. *Ravi Kumar* bearing (231VMTR00265) on the title “**Front End Development Project**” is a record of project work done by him during the academic year 2023-24 under my guidance and supervision in partial fulfillment of Master of Computer Applications.

Place: _____

Date: *Faculty Guide*

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Jain Online (Deemed-to-be University), my faculty guide *Nikhil Eknathrao Karale*, and other faculty members for their invaluable guidance and support. I am also thankful to my classmates and family for their encouragement and assistance.

Ravi Kumar

USN: 231VMTR00265

EXECUTIVE SUMMARY

Travel Buddy's website offers a seamless user experience with a clear focus on travel planning. The header and navigation bar provide intuitive access to key sections, such as Home, Book, Packages, Services, Gallery, and About Us.

The visually appealing hero section sets the tone with a compelling call to action, encouraging users to explore travel opportunities in the United States.

The booking form simplifies the process of planning a trip, allowing users to easily input essential details like destination, number of travellers, and travel dates.

The packages section showcases diverse travel options, inspiring exploration, and customization.

The services section highlights the company's commitment to customer satisfaction, offering a range of valuable services like affordable hotels, local cuisine experiences, and personalized itineraries.

The gallery section visually captivates users with stunning images, showcasing the beauty of various destinations and the exciting experiences offered by Travel Buddy.

The About Us section builds trust and credibility, highlighting the company's mission and values.

The footer provides essential information and contact options, ensuring easy access to important details.

The website's use of modals and scripts enhances user experience, making it a comprehensive and engaging platform for travel enthusiasts.

TABLE OF CONTENTS

Title	Page Nos.
Executive Summary	6
Introduction	8
Objectives	9
Project Description	12
Additional Features	18
Flowchart	19
Coding	20
Output	28
Validation	31
Conclusion	32
References	33

1. Introduction

Project Background:

The increasing popularity of travel and the desire for personalized, hassle-free experiences have led to a growing demand for innovative travel solutions. Traditional travel agencies often struggle to provide tailored recommendations and efficient booking processes.

Problem:

The primary problem this project aims to address is the lack of a comprehensive and personalized travel planning platform. Existing solutions may be limited in terms of customization, real-time information, and user experience.

Solution Overview:

Travel Buddy is a web-based platform designed to revolutionize the way people plan and book their trips. It offers a range of features, including:

- **Personalized Recommendations:** Leveraging user preferences and past travel history, the platform provides tailored suggestions for destinations, accommodations, and activities.
- **Real-time Information:** Access to up-to-date flight schedules, hotel availability, and local events.
- **Seamless Booking:** Integrated booking system for flights, hotels, and activities, streamlining the entire process.
- **Interactive Maps:** Visual representation of itineraries and points of interest.
- **Community Features:** User forums and reviews to facilitate knowledge sharing and social interaction.

Significance of Technology Choice:

- **React JS:** A powerful JavaScript library for building user interfaces, React enables the creation of dynamic and responsive web applications. It is ideal for complex UI components and efficient updates, ensuring a smooth user experience.
- **Swiper JS:** It enhanced the visual appeal and user engagement of the platform. It allows for smooth transitions between slides, touch gestures, and responsive design, making it a valuable tool for creating dynamic and captivating user interfaces.

2. Objectives

2.1 Checkpoint - 1

Objective 1: Create a Welcoming and Informative Homepage

- Design a visually appealing and user-friendly navigation bar:
 - Include essential navigation links: Home, Book, Packages, Services, Gallery, and About Us.
 - Implement a dropdown menu for the "Packages" link to offer country-specific options (United States, India, France, Germany).
 - Provide clear and concise labels for all navigation items.
- Develop an engaging hero section:
 - Display a high-quality background image to set the tone for the website.
 - Incorporate a prominent heading to introduce the brand and its offerings.
 - Utilize a dynamic text element to showcase popular destinations, rotating automatically.
 - Include a prominent "Book Now" button to encourage user interaction.
- Ensure optimal layout and responsiveness:
 - Align all elements within the hero section to create a visually balanced and aesthetically pleasing design.
 - Optimize the layout for various screen sizes to provide a consistent user experience across different devices.

2.2 Checkpoint - 2

Objective 2: Create a Comprehensive Booking and Package Section

- Develop a user-friendly booking form:
 - Implement a responsive form layout that adapts to different screen sizes.
 - Include the following form fields:
 - Destination: A dropdown list of popular destinations.
 - Number of Persons: A numerical input field.
 - Start Date: A date picker that only allows future dates.
 - End Date: A date picker that ensures the end date is after the start date.
 - Description: A text area with a minimum and maximum character limit.
 - Implement a "Book Now" button that triggers a success alert upon form submission.
- Design an attractive package gallery:
 - Create a visually appealing grid layout to display package cards.
 - Include the following information in each package card:
 - Image: A high-quality image of the destination.
 - Destination Name: A clear and concise heading.
 - Description: A brief overview of the destination's attractions.
 - Price: Displayed in USD currency.
 - Rating: A star rating system to indicate the popularity or quality of the package.
 - Book Now button: Linked to the booking form or a dedicated package page.
- Create a dedicated services section:
 - Highlight a variety of services offered by the company, such as:

- Affordable Hotels
- Food and Drink Recommendations
- Safety Guides
- Personalized Itineraries
- Visa Assistance
- Luggage Storage
- Use a visually appealing layout, such as cards or a slider, to showcase the services.
- Provide clear and concise descriptions for each service.

2.3 Checkpoint - 3

Objective 3: Create a Visually Appealing Gallery and Informative About Us Section

- Design an engaging gallery section:
 - Create a visually appealing gallery to showcase captivating travel moments.
 - Use a suitable layout, such as a grid or slider, to display images effectively.
 - Implement a hover effect that scales the images upon mouseover, enhancing user interaction.
 - Include a brief description or caption for each image to provide context.
- Develop an informative About Us section:
 - Create a two-column layout to balance text and visuals.
 - Display a relevant image or logo on one side.
 - On the other side, provide detailed information about the company, including:
 - Company history and mission
 - Core values and principles
 - Team members and their expertise
 - Contact information
 - Use clear and concise language to convey the company's message.
- Create a functional footer:
 - Include the company's brand name or logo.
 - Add social media icons (Instagram, Facebook, etc.) with inactive links.
 - Display a copyright notice with the appropriate message.
 - Provide essential information, such as contact details and terms of service.

2.4 Checkpoint - 4

Objective 4: Implement User Registration and Login Functionality

- Create a User Registration Form:
 - Design a responsive registration form with fields for:
 - Full Name
 - Contact Number
 - Date of Birth
 - Email Address
 - Password
 - Gender
 - Implement robust JavaScript validation to ensure data accuracy and security.
 - Include a "Register" button to submit the form.

- Provide a link to the login page for existing users.
- Create a User Login Form:
 - Design a simple login form with fields for:
 - Email Address
 - Password
 - Implement secure password hashing and salting.
 - Include a "Login" button to authenticate the user.
 - Provide a link to the registration page for new users.
- Implement Modal Functionality:
 - Use a modal component to display the registration and login forms without full page reloads.
 - Trigger the modal on button clicks or specific events.
 - Ensure smooth modal behavior, including proper opening, closing, and overlay effects.
- Deploy the Website:
 - Choose a suitable platform like GitHub Pages or Netlify.
 - Follow the platform's specific deployment instructions.
 - Configure the deployment settings to ensure proper hosting and accessibility.
 - Test the deployed website thoroughly to identify and fix any issues.

3. Project Description

3.1 Checkpoint - 1

1. Header and Navigation

The header of the Travel Buddy website establishes a welcoming and organized first impression. It features a navigation bar that allows users to easily access different sections of the site, such as Home, Book, Packages, Services, Gallery, and About Us. The use of a logo at the top left adds a branding element, while the menu includes dropdowns for quick access to country-specific travel packages. This design facilitates user interaction with the website and enhances navigation ease. Additionally, login and registration buttons are prominently displayed, catering to user engagement right from the outset. The use of icons for user-related actions adds a modern touch and encourages users to explore further.

Feature: Clear and Intuitive Navigation Bar

- **Functionality:** Implement a horizontal navigation bar with essential links: Home, Book, Packages, Services, Gallery, and About Us.
- **User Experience:** Easy access to key sections of the website, enhancing user experience.

2. Hero Section

The hero section serves as the focal point for welcoming visitors to Travel Buddy. It features a captivating headline that highlights the company's name, followed by a dynamic call to action that encourages users to explore travel opportunities in the United States. This section uses visually appealing text formatting, with important keywords highlighted, making it easy to draw attention. The "Book Now" button directly invites potential customers to engage with the booking process, seamlessly integrating visual elements with functionality. This section effectively sets the tone for the user's journey through the site, emphasizing the adventurous spirit of travel while promoting immediate action.

Feature: Eye-Catching Hero Section

- **Functionality:** Design a visually appealing hero section with a captivating background image, a prominent headline, and a dynamic destination slider.
- **User Experience:** Creates a strong first impression and encourages user exploration.

3.2 Checkpoint - 2

1. Booking Form

The "Book Now" section is crucial for user interaction, allowing visitors to begin the booking process directly on the website. This section includes a comprehensive form that collects essential travel information, such as destination, number of travellers, travel dates, and a description of their preferences. The structured layout, featuring clearly labelled fields, ensures a user-friendly experience. Each input field is required, which minimizes incomplete submissions and only future dates can be selected in start and end dates fields. The end date must be greater than the start date, this is achieved by dynamically setting the min value attribute in end date input field using JS. By incorporating an engaging image alongside the form, this section captures attention while providing a practical function, making it a vital part of the user's travel planning process. Overall, this section successfully combines usability with visual appeal, enhancing the overall user experience.

Feature: User-Friendly Booking Form

- **Functionality:** Develop a simple and intuitive booking form with fields for destination, number of travellers, dates, and preferences.
- **User Experience:** Streamlines the booking process, minimizing user effort.

2. Packages Section

In the Packages section, Travel Buddy emphasizes the diversity of travel options available to customers. The introductory text highlights the global reach of the company, instilling a sense of excitement and adventure. This section is designed to showcase various travel packages, although specific details about individual packages are yet to be populated, leaving room for visual and textual content that could highlight unique destinations and experiences. The overall aesthetic encourages exploration and entices users to find their ideal getaway, making it an essential part of the website's offering. By enticing customers to discover their dream destinations, this section plays a crucial role in generating interest and engagement.

Feature: Visually Appealing Package Cards

- **Functionality:** Design visually appealing cards to display package details, including destination images, names, descriptions, prices, ratings, and a "Book Now" button.
- **User Experience:** Enhances user engagement and helps users make informed decisions.

3. Services Section

The Services section outlines the variety of offerings provided by Travel Buddy, emphasizing their commitment to customer satisfaction. Each service is presented through individual cards that include engaging icons and succinct descriptions, ensuring that information is both accessible and appealing. Services such as affordable hotels, local cuisine experiences, safety guides, luggage

assistance, personalized itineraries, and language translation support are all highlighted. This section is strategically designed to build trust and convey the company's dedication to making travel enjoyable and hassle-free. By framing their services as essential for travellers, Travel Buddy effectively positions itself as a comprehensive solution for vacation planning.

Feature: Informative Service Cards

- **Functionality:** Create cards to highlight various services offered by Travel Buddy, such as affordable hotels, food and drink recommendations, safety guides, and personalized itineraries.
- **User Experience:** Provides valuable information and assists users in planning their trips.

3.3 Checkpoint – 3

1. Gallery Section

The Gallery section serves to visually showcase the experiences that Travel Buddy offers, enhancing the emotional appeal of the website. It features a carousel of images representing different travel adventures and destinations, effectively capturing the essence of joyful travel moments. The inclusion of a swiper component allows for interactive engagement, letting users browse through images easily. Accompanying text reinforces the narrative of shared experiences, inviting potential customers to imagine themselves in similar scenarios. This section not only enhances the website's aesthetic value but also serves to inspire trust by demonstrating the kinds of memorable experiences customers can expect when booking through Travel Buddy.

Feature: Engaging Image Gallery

- **Functionality:** Develop a visually appealing gallery showcasing stunning travel images, with a hover effect for enhanced user interaction.
- **User Experience:** Creates a captivating visual experience and inspires users to explore.

2. About Us Section

The About Us section offers an insightful look into the philosophy and mission of Travel Buddy. Founded in 2020, the company emphasizes its commitment to crafting personalized travel experiences tailored to individual needs. The narrative encapsulates the essence of travel to create lasting memories rather than just visiting new places. This section builds credibility by showcasing the passion and expertise of the team behind Travel Buddy, reassuring potential customers that they are in capable hands. By inviting visitors to connect through contact information, this section not only informs but also fosters a sense of community and approachability, which is vital for customer engagement.

Feature: Company Information and Contact Details

- **Functionality:** Provide essential information about the company's history, mission, team, and contact details.
- **User Experience:** Builds trust and transparency with users.

3. Footer

The footer of the Travel Buddy website serves several essential functions, consolidating important information and links for user convenience. It prominently displays the company logo, reinforcing brand identity, and includes quick links to the Terms & Conditions and Privacy Policy, demonstrating transparency. Additionally, the footer contains social media icons that encourage

users to connect with Travel Buddy on various platforms, enhancing their engagement outside the website. The copyright statement at the end reinforces the legitimacy of the site while providing a professional finish. Overall, the footer effectively wraps up the website experience, ensuring that visitors have easy access to crucial information and contact options.

Feature: Informative Footer

- **Functionality:** It provides the company's social media contacts and information about Terms of Service and Privacy Policy links and Copyright information.
- **User Experience:** Connects users to US building trust and enhancing user experience and brand visibility.

3.4 Checkpoint - 4

1. Modals and Scripts

The website incorporates modals for login and registration, enhancing user experience by allowing visitors to access these features without navigating away from their current page. The design is user-friendly, with clear prompts and required fields ensuring that users can easily engage with the registration or login process. The presence of dynamic validations of input fields along the use of “debounce function” to validate ensures the commitment to providing a dynamic, responsive experience that enhances interactivity and without compromising the performance, throughout the site. The alert modals uses dynamic text/content injection which helps in code reusability and improves the performance of the site. These technical elements are vital for ensuring that the website operates smoothly, facilitating a seamless user journey. Overall, the use of modals and scripts underscores the website's modern design, catering to user preferences for convenience and efficiency.

Step 6: Seamless User Authentication

- **Feature:** User Registration and Login Forms
 - **Functionality:** Develop user-friendly registration and login forms with robust validation and security measures.
 - **User Experience:** Enables users to create accounts, personalize their experience, and access exclusive features.
- **Feature:** Alert and Form Modals
 - **Functionality:** Implement modal dialogs to display alerts/error/success messages and registration/login forms without full page reloads.
 - **User Experience:** Enhances user experience by reducing page load times and maintaining context.

4. Additional Features

Additional Feature: Enhanced User Experience with Partial State and User Authentication.

Why Implement Partial State and User Authentication?

By incorporating partial state and user authentication into the Travel Buddy project, we can significantly enhance the user experience and security of the platform.

Key Benefits:

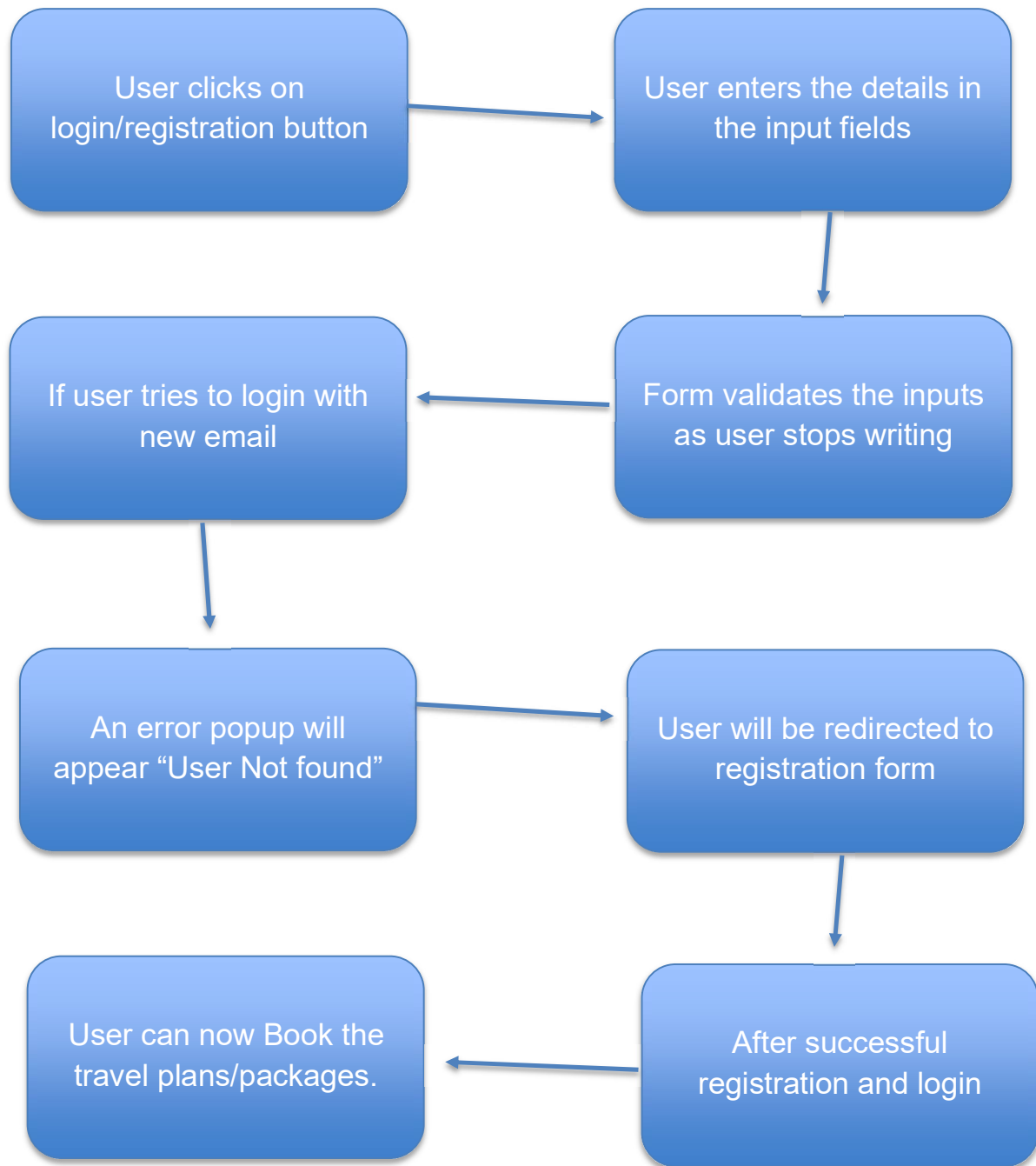
- **Personalized Experience:**
 - Users can save their preferences, itineraries, and booking history.
 - The platform can tailor recommendations based on past behaviours and preferences.
- **Enhanced Security:**
 - Protecting user data and preventing unauthorized access.
 - Implementing strong password hashing and salting techniques.
- **Seamless Booking Process:**
 - Streamlining the booking process by pre-filling forms with user information.
 - Enabling users to manage their bookings and cancellations.
- **Community Building:**
 - Facilitating user interaction through reviews, ratings, and social features.
 - Encouraging a sense of community among travellers.

How It Works:

1. **User Registration:**
 - A user can create an account by providing necessary information, such as name, email, password, and contact details.
2. **User Login:**
 - Registered users can log in to their accounts using their email and password.
3. **Partial State and Booking Restrictions:**
 - Users can browse packages and services, but they cannot book or save preferences unless they login.
 - When a user attempts to book a package, a login modal will appear.
 - If user enters a different email from the registered one then an error popup will show on screen saying “user not found” and redirects the user to login form in 3 secs.
4. **Personalized Experience:**
 - Once logged in, users can save their preferences, create personalized itineraries, and receive tailored recommendations.

By implementing these features, Travel Buddy can offer a more engaging and secure user experience, leading to increased user satisfaction and loyalty.

5. Flowchart



6. Coding

Technology Stack:

- React JS: A popular JavaScript library for building user interfaces, used to create dynamic and interactive components.
- HTML, CSS: The foundation of web development, used to structure and style the website.
- JavaScript: For handling client-side logic, DOM manipulation, and AJAX requests.
- Swiper JS: A JavaScript library for creating mobile-friendly touch sliders, used for image carousels and package sliders.

Code Structure:

1. Component-Based Architecture:

- Break down the UI into reusable components (e.g., Header, Footer, HeroSection, BookingForm, PackageCard, ServiceCard, Gallery, AboutUs).
- Organize components in a hierarchical structure to manage complexity.

2. State Management:

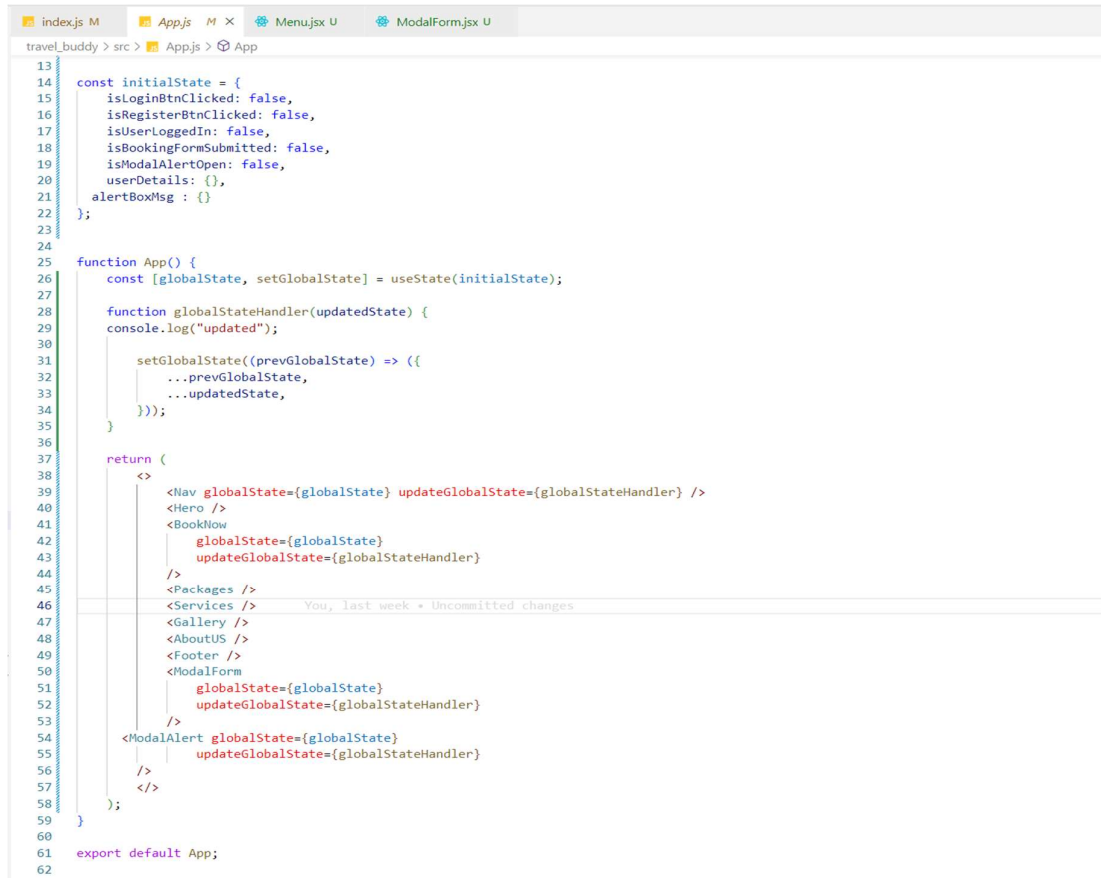
- Used React's state management to store and update component data (e.g., user login and registration states, user details, forms' and modals' active and inactive states).

3. Styling and Layout:

- Used CSS to style the website and ensure responsiveness across different devices.

Code Snippets:

App.js : Here, I have defined the global state object and created its update handler, both of which are passed to various components where state is getting used and updated.



```
13
14 const initialState = {
15   isLoginBtnClicked: false,
16   isRegisterBtnClicked: false,
17   isUserLoggedIn: false,
18   isBookingFormSubmitted: false,
19   isModalAlertOpen: false,
20   userDetails: {},
21   alertBoxMsg: {}
22 };
23
24
25 function App() {
26   const [globalState, setGlobalState] = useState(initialState);
27
28   function globalStateHandler(updatedState) {
29     console.log("updated");
30
31     setGlobalState((prevGlobalState) => ({
32       ...prevGlobalState,
33       ...updatedState,
34     }));
35   }
36
37   return (
38     <>
39       <Nav globalState={globalState} updateGlobalState={globalStateHandler} />
40       <Hero />
41       <BookNow
42         globalState={globalState}
43         updateGlobalState={globalStateHandler}
44       />
45       <Packages />
46       <Services /> You, last week * Uncommitted changes
47       <Gallery />
48       <AboutUS />
49       <Footer />
50       <ModalForm
51         globalState={globalState}
52         updateGlobalState={globalStateHandler}
53       />
54       <ModalAlert globalState={globalState}
55         updateGlobalState={globalStateHandler}
56       />
57     </>
58   );
59 }
60
61 export default App;
62
```

Here globalStateHandler function takes an argument which needs to be an object containing the variables of initialState.

Menu.jsx

Each component has its own local state which gets updated when global state is updated by using useEffect hook of React. Here in the menu.jsx hamburger menu and login/register button handlers are defined and globalState is being updated based on the user interaction with these buttons.

```

function Menu(props) {
  let localState = {
    isRegisterBtnClicked: false,
    isLoginBtnClicked: false,
  };

  const pkgMenu = ["United States", "India", "France", "Germany", "Japan"];
  let [open, setOpen] = useState("");
  let [userStatus, setUserStatus] = useState(false);
  let [logoutActive, setLogoutActive] = useState(false);
  function hamburgerClickHandler() {
    if (open === "") {
      setOpen("open");
    } else {
      setOpen("");
    }
  }

  function btnClickHandler(event) {
    let target = event.target.textContent;
    localState = {
      isRegisterBtnClicked: target.toLowerCase() === "register",
      isLoginBtnClicked: target.toLowerCase() === "login",
    };
    props.updateGlobalState(localState)
  }

  function logoutHandler(e) {
    e.preventDefault();
    setLogoutActive(false);
    props.updateGlobalState({isUserLoggedIn:false})
  }

  function showLogout(e) {
    e.preventDefault();
    if (logoutActive) {
      setLogoutActive(false);
    }else{
      setLogoutActive(true);
    }
  }

  useEffect(() => {
    setUserStatus(props.globalState.isUserLoggedIn);
  }, [props.globalState])
}

```

Hero.jsx

The major issue I faced in this section was to change the destination names every 200 ms, while it was so simple in javascript but react behaves differently with the timeout functions. As react uses virtual dom to update the changes on UI and that too once all changes are done, so set I have to

use useEffect hook with empty dependency array to force the react js to update the UI everytime setInterval is called without any need of dependency changes.

```
function Hero() {
  const [title, setTitle] = useState("United States");

  useEffect(() => {
    let i = 0;
    const timeoutId = setInterval(() => {
      i = i === travelPackages.length - 1 ? 0 : i + 1;
      setTitle(travelPackages[i].title);
    }, 200);
    return () => clearTimeout(timeoutId);
  }, []);

  return (
    <Section className="hero_section" id="Hero">
      <Container>
        <h1 className="hero_heading">
          Welcome to <span>Travel Budddy</span>
        </h1>
        <h1 className="hero_big_text">
          Visit <span id="hero_big_text">{title}</span>
        </h1>
        <a href="#Book_Now">
          <DefButton extraClass= "hero_btn">Book Now</DefButton>
        </a>
      </Container>
    </Section>
  );
}

export default Hero;
```

BookingForm.jsx

In this file booking form dates selection and form submit events are handled. Here I have used the date object of the JS to calculate the current date and set the min value of the start date as current date and once user selected a start date then I calculated the min value of end date which would be a day after start date. In formsubmit handler function I am getting the data from the Form object of JS and to show the user success message, the globalState is updated.

```

6  function BookingForm(props) {
7    let today = new Date();
8    const dd = String(today.getDate()).padStart(2, "0");
9    const mm = String(today.getMonth() + 1).padStart(2, "0");
10   const yyyy = today.getFullYear();
11   today = yyyy + "-" + mm + "-" + dd;
12   const [startMinDate, setStartMinDate] = useState(today);
13   const [endMinDate, setEndMinDate] = useState(today);
14
15   function startDateChangeHandler(event) {
16     const nextDay = new Date(event.target.value);
17     nextDay.setDate(nextDay.getDate() + 1);
18     setEndMinDate(nextDay.toISOString().split("T")[0]);
19   }
20
21   const [selecSize, setSelecSize] = useState(5);
22
23   function selectionHandler(event) {
24     setSelecSize(1);
25     if (event.target[0].value === "") {
26       event.target.remove(0);
27     }
28   }
29
30   function userFormSubmit(e) {
31     e.preventDefault();
32     let userStatus = props.globalState.isUserLoggedIn;
33     if (!userStatus) {
34       return props.updateGlobalState({
35         isLoginBtnClicked : true
36       });
37     }
38     const formData = new FormData(e.target);
39     for (const [key, value] of formData.entries()) {
40       console.log(`${key}: ${value}`);
41     }
42     props.updateGlobalState({
43       alertBoxMsg: {
44         title: "!!!Hurray!!!",
45         subTitle: "Your booking is successful!",
46         img: "check.png",
47       },
48     });
49     e.target.reset();
50   }
51
52   return (
53     <div className="fs_form_wrap">
54       <form action="" id="bookingForm" onSubmit={userFormSubmit}>
55         <span>
56           <label htmlFor="dest">Where To:</label>
57           <select
58             name="dest"

```

ModalForm.jsx

It is the largest and most complex component of my react app, it handles the validation of form input, login and register form submits, showing the different alerts and popups after successful and unsuccessful events, opening and closing of the forms and modals, switching between login and register forms etc. Here are few sample snippets of the code.


```

function registerHandler(e) {
  e.preventDefault();
  const formData = new FormData(e.target);
  let details = {};
  for (const [key, value] of formData.entries()) {
    details[key] = value;
  }
  setGlobalState({
    userDetails: details,
    alertBoxMsg: {
      title: "Registration Successful!!",
      subTitle: "Please Login Now",
      img: "check.png",
    },
  });
  setTimeout(() => {
    setGlobalState({
      isLoginBtnClicked: false,
      isRegisterBtnClicked: false,
      alertBoxMsg: {},
    });
  }, 3000);
}

function formSwitchHandler(event) {
  event.preventDefault();
  resetForms();
  let target = event.target.textContent;
  let isLogin = target === "Login";
  let isRegister = target === "Register";
  setGlobalState({
    isLoginBtnClicked: isLogin,
    isRegisterBtnClicked: isRegister,
  });
}

function cancelBtnHandler(event) {
  event?.preventDefault();
  resetForms();
  let isLogin = false;
  let isRegister = false;
  setGlobalState({
    isLoginBtnClicked: isLogin,
    isRegisterBtnClicked: isRegister,
  });
}

```

```

90     function loginHandler(e) {
91         e.preventDefault();
92         let formData = new FormData(e.target);
93         let userDetails = {};
94         for (const [key, value] of formData.entries()) {
95             userDetails[key] = value;
96         }
97
98         let hasUser = props.globalState.userDetails;
99         let isUserRegistered = Object.keys(hasUser).length > 0;
100         if (isUserRegistered) {
101             if (hasUser.password === userDetails.password) {
102                 console.log("login successful");
103                 setGlobalState({
104                     isUserLoggedIn: true,
105                     isLoginBtnClicked: false,
106                     isRegisterBtnClicked: false,
107                     alertBoxMsg: {
108                         title: "Lets' Go",
109                         subTitle: "Login Successful",
110                         img: "check.png",
111                     },
112                 });
113             } else {
114                 console.log("incorrect password");
115                 setGlobalState({
116                     alertBoxMsg: {
117                         title: "Error",
118                         subTitle: "Invalid Password",
119                         img: "delete.png",
120                         class: "danger",
121                     },
122                 });
123             }
124         } else {
125             console.log("user not found, please register");
126             setGlobalState({
127                 alertBoxMsg: {
128                     title: "User Not Found",
129                     subTitle: "Please Register",
130                     img: "delete.png",
131                     class: "danger",
132                 },
133             });
134             setTimeout(() => {
135                 resetForms();
136                 setGlobalState({
137                     isLoginBtnClicked: false,
138                     isRegisterBtnClicked: true,
139                     alertBoxMsg: {},
140                 });
141             }, 3000);
142         }
143     }

```

```

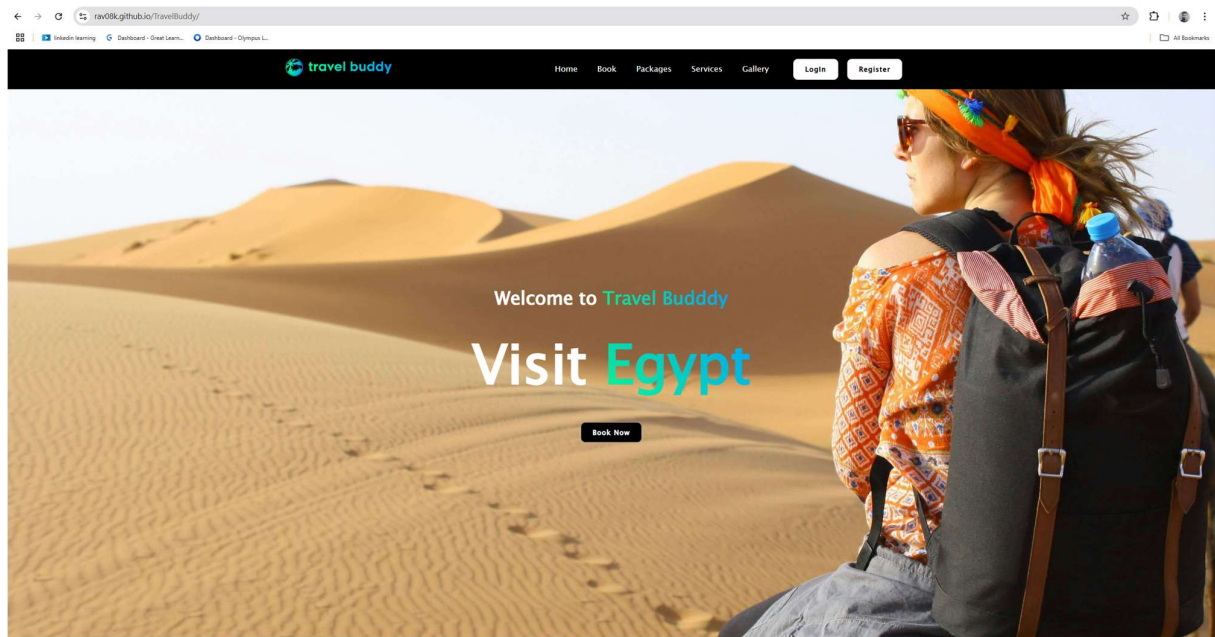
19 function debounce(func, delay) {
20     let timeoutId;
21
22     return function () {
23         clearTimeout(timeoutId);
24         timeoutId = setTimeout(() => {
25             func.apply(this, arguments);
26         }, delay);
27     };
28 }
29
30 function isValidInp(inp, target) {
31     if (!inp) {
32         return "Field cannot be empty";
33     }
34     const passwordPattern =
35         /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*~]).{8,15}$/;
36     const emailPattern = /^[a-zA-Z0-9._%~]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
37     const mobilePattern = /^\d{10}$/;
38     const namePattern = /^[a-zA-Z\s]+$/;
39     switch (target) {
40         case "fullName":
41             return namePattern.test(inp) ? true : "Invalid Name";
42         case "email":
43             return emailPattern.test(inp) ? true : "Invalid Email";
44         case "password":
45             return passwordPattern.test(inp)
46                 ? true
47                 : "Invalid Password : It must contain an UpperCase, LowerCase, Symbol and Digit";
48         case "contact":
49             return mobilePattern.test(inp) ? true : "Invalid Number";
50         default:
51             break;
52     }
53 }
54
55 function formInputHandler(event) {
56     let inpEle = event.target;
57     let errorMsgEle = inpEle.nextElementSibling;
58     let inpMsg = isValidInp(inpEle.value, inpEle.name);
59     if (inpMsg === true) {
60         errorMsgEle.classList.remove("active");
61         inpEle.classList.replace(inpEle.className, "good");
62     } else {
63         errorMsgEle.classList.add("active");
64         errorMsgEle.textContent = inpMsg;
65         inpEle.classList.replace(inpEle.className, "error");
66     }
67 }
68

```

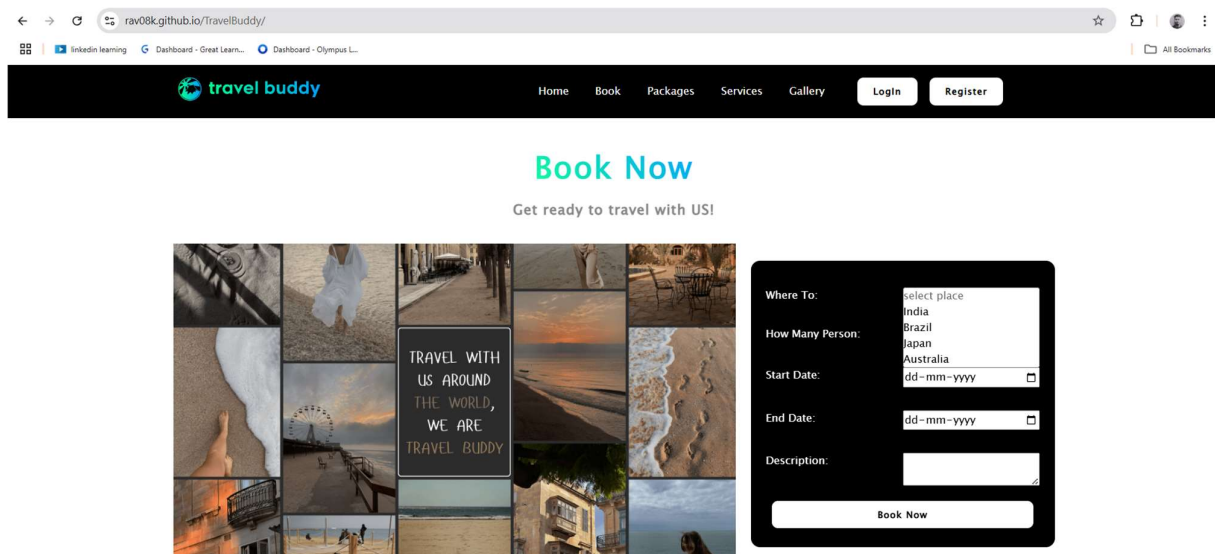
7. Output

Here are some sample outputs images of the site.


Homepage: It includes navigation bar and hero section.



Book Now Section: It contains booking form.




Packages Section: It shows the various packages offered by the company.



[Home](#) [Book](#) [Packages](#) [Services](#) [Gallery](#) [Login](#) [Register](#)

Choose Your Dream Place

We have our footprints all around the World!




India

Experience the rich culture, diverse landscapes, and historic sites like the Taj Mahal and Jaipur forts.

Price: \$1500/- Only* for 2 person

User Ratings: 4.3 ★★★★★

[Book Now](#)




Brazil

Discover the vibrant culture of Rio de Janeiro, the Amazon rainforest, and stunning beaches in Bahia.

Price: \$2000/- Only* for 2 person

User Ratings: 4.5 ★★★★★

[Book Now](#)






Japan

Experience a blend of different culture, from ancient temples in Kyoto to neon lights in Tokyo.


Price: \$3000/- Only* for 2 person

User Ratings: 4.9 ★★★★★

[Book Now](#)




About Us Section:



[Home](#) [Book](#) [Packages](#) [Services](#) [Gallery](#) [Login](#) [Register](#)

About Us



About TravelBuddy


Founded in 2020, TravelBuddy has quickly become a trusted name in the travel industry. Our mission is to create unforgettable travel experiences for our clients, ensuring every trip is tailored to meet their unique needs and preferences.

At TravelBuddy, we believe that travel is more than just visiting new places; it's about creating memories that last a lifetime. Our team of experienced travel consultants is passionate about exploring the world and sharing that passion with our clients.

We offer a wide range of services, including personalized tour packages, adventure trips, family vacations, and corporate travel solutions. Whether you're looking for a relaxing beach holiday or an exhilarating mountain trek, TravelBuddy has got you covered.

Join us in discovering the beauty of the world, and let us take care of the details while you enjoy the journey!

Contact Us: For inquiries, reach us at info@travelbuddy.com or call us at (123) 456-7890.



Login Form:

The screenshot shows the Travel Buddy website with a login form overlay. The background image features a person with a backpack in a desert setting. The website's navigation bar includes links for Home, Book, Packages, Services, Gallery, Login, and Register. The login form is titled "Login" and contains the following fields and buttons:

- Email input field
- Password input field
- Login button (black)
- Cancel button (white)
- Link: Don't have an account? [Register](#)

Register Form:

The screenshot shows the Travel Buddy website with a register form overlay. The background image features a person with a backpack in a desert setting. The website's navigation bar includes links for Home, Book, Packages, Services, Gallery, Login, and Register. The register form is titled "Register" and contains the following fields and buttons:

- Full Name input field
- Email ID input field
- Password input field
- Contact input field
- Register button (black)
- Cancel button (red)
- Link: Already have an account? [Login](#)

8. Validation

Validation: Ensuring Quality and Performance

To ensure the quality and performance of the front-end, I've employed the following methods:

Manual Testing:

- **Visual Inspection:** Visually inspecting the website for any layout or design issues.
- **Functional Testing:** Testing the functionality of each feature, such as navigation, form submissions, and button clicks.
- **Usability Testing:** Testing the website's usability by observing how users interact with it.

Browser Compatibility Testing:

- Testing the website on different browsers (Chrome, Firefox, Safari, Edge) and devices (desktop, tablet, mobile) to ensure consistent performance and appearance.

Responsive Design Testing:

- Ensuring the website adapts to different screen sizes and resolutions.

By combining these methods, I've strived to create a high-quality, user-friendly, and performant website.

9. Conclusion

The Travel Buddy project has successfully achieved its primary objectives of creating a user-friendly and visually appealing travel booking platform. Key features such as the intuitive navigation bar, engaging hero section, comprehensive package and service information, immersive image gallery, and informative About Us section have contributed to a positive user experience.

The implementation of partial state and user authentication has further enhanced the platform's functionality and security, allowing for personalized experiences and secure transactions.

Key Takeaways and Results:

- **Successful Front-End Development:** The front-end development was completed within the given timeline and budget, delivering a high-quality, responsive, and user-friendly website.
- **Effective Use of Technology:** The project leveraged a combination of technologies, including React, HTML, CSS, JavaScript, and Bootstrap, to create a modern and dynamic web application.
- **Strong Focus on User Experience:** The website is designed to be intuitive and easy to navigate, with a focus on clear and concise information.
- **Robust Testing and Validation:** Rigorous testing and validation processes ensured the quality and performance of the website.

Future Improvements:

- **Advanced Features:** Explore advanced features like AI-powered recommendations, virtual reality tours, and real-time chat support.
- **Mobile Optimization:** Further optimize the website for mobile devices, including touch-friendly interactions and faster load times.
- **Accessibility:** Enhance accessibility by implementing WCAG guidelines and using assistive technologies.
- **Performance Optimization:** Continuously monitor and optimize website performance to ensure optimal loading times and responsiveness.
- **Security Enhancements:** Stay up-to-date with the latest security best practices to protect user data and prevent cyberattacks.

By addressing these areas, Travel Buddy can continue to evolve and provide an exceptional travel booking experience.

References

Resources, Frameworks, Libraries, and Tools

Front-End Development:

- React JS: A JavaScript library for building user interfaces. <https://react.dev/>
- HTML, CSS: The foundation of web development.
- JavaScript: For handling client-side logic and DOM manipulation.
- Swiper JS: A JavaScript library for creating mobile-friendly touch sliders. <https://swiperjs.com/>
- Font Awesome: An icon library for web fonts and SVGs. <https://fontawesome.com/>

Additional Resources:

- W3Schools: A web development tutorial site. <https://www.w3schools.com/>
- GitHub: A version control platform. <https://github.com/>
- GitHub Pages: A static web hosting service. <https://pages.github.com/>
- Wikipedia: A free encyclopaedia. <https://www.wikipedia.org/>
- MDN Web Docs: Web developer documentation. <https://developer.mozilla.org/en-US/>
- GeeksforGeeks: A computer science portal. <https://www.geeksforgeeks.org/>
- Pexels: A free stock photo website. <https://www.pexels.com/>
- Flaticon: A free icon resource. <https://www.flaticon.com/>

By leveraging these resources and tools, we were able to effectively develop and deploy the Travel Buddy project.