MCA Semester – III

Front End Development Project

Name	**Jayatheertha S Rao**
USN	**231VMTR00128**
Elective	**Full Stack Development**
Date of Submission	**10 November 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

Jayatheertha S Rao

USN

231VMTR00128

Under the guidance of

Mr. Nikhil Karale

JAIN Online

DECLARATION

I, **Jayatheertha S Rao**, hereby declare that this Project Report has been prepared by me under the guidance of **Mr. Nikhil 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: Bengaluru

Date: 10-11-2024

Jayatheertha S Rao

USN: 231VMTR0012

CERTIFICATE

This is to certify that the Project report submitted by Mr. **Jayatheertha S Rao**, bearing University Seat Number, **231VMTR00128**, 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 fulfilment of Master of Computer Applications.

Place: Bengaluru

Date: 10-11-2024

Faculty Guide

Mr. Nikhil Karale

JAIN Online

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to JAIN University for its invaluable support in the successful completion of my project. I am especially grateful to the faculty for their guidance, expertise, and encouragement throughout the research process.

The knowledge gained during this project has significantly contributed to my academic and professional growth, and I am confident it will positively impact my future endeavors.

Thank you once again to JAIN University and its faculty for their generous support. I look forward to potential future collaborations that will further contribute to academic excellence and innovation.

Jayatheertha S Rao

USN: 231VMTR00128

EXECUTIVE SUMMARY

The purpose of this project is to develop **"TravelTub"**, a user-friendly online platform designed to simplify travel planning and booking. This platform offers travelers the ability to explore, compare, and book a variety of travel services, such as flights, hotels, and tour packages, all in one place. The goal is to provide a seamless, efficient experience for users looking to plan and book their trips.

The significance of this project lies in its ability to meet the growing demand for convenient and accessible travel solutions. By integrating multiple services into one platform, **"TravelTub"** enhances the ease and efficiency of the travel booking process, making it easier for users to find and book the best options based on their preferences.

The key outcomes of the project include the creation of an intuitive and secure website, a streamlined booking process, and the integration of a diverse range of travel services. The platform aims to improve the overall travel experience by offering a convenient, one-stop solution for modern travelers.

TABLE OF CONTENTS

1. Introduction

1.1 Background

Travel technology has significantly transformed the way people plan, book, and experience their journeys. Traditionally, travel planning involved physically visiting agencies or making phone calls to book flights, hotels, and activities. The advent of the internet introduced online travel booking platforms simplified the process by allowing users to book travel services from a single website. This technological advance made booking faster and more accessible.

However, while travel technology has evolved, existing platforms still present challenges. Fragmented services across multiple websites, outdated user interfaces, and lack of real-time data updates can make travel planning cumbersome. Moreover, security remains a concern for users providing sensitive information online. *TravelTub* seeks to address these issues by providing a unified, seamless platform that offers efficient, user-friendly travel booking.

1.2 Problem Statement

Despite the improvements in online travel booking, users still encounter several challenges:

- **Fragmentation**: Multiple platforms for booking different travel services create a disjointed experience.
- **Outdated User Interfaces**: Many booking platforms suffer from poor design, making navigation difficult and frustrating.
- **Lack of Real-Time Data**: Inaccurate or delayed updates on pricing and availability can lead to booking errors.
- **Security Concerns**: Handling sensitive data such as payment details remains a significant concern for users.

TravelTub addresses these challenges by offering a cohesive platform with an intuitive interface, real-time updates, and robust security features, ensuring a seamless booking experience.

1.3 Objectives of TravelTub

TravelTub aims to:

- **Create a User-Friendly Travel Booking Interface**: Simple, intuitive navigation to enhance the user experience.
- **Provide Responsive Design for Accessibility**: Optimized for desktop and mobile devices to ensure broad accessibility.
- **Ensure Real-Time Validation and Feedback for Secure, Error-Free Bookings**: Real-time data to minimize errors during booking.
- **Offer Additional Features such as Dynamic Content and Interactive UI Elements**: Users can select a package directly from the Packages section.

1.4 Technology Stack

The following technologies were chosen for their roles in creating a robust, efficient, and responsive travel booking platform:

- **HTML**: Provides the structural framework for the website's content.
- **CSS**: Handles styling, layout, and responsive design to ensure the website adapts to different devices.
- **JavaScript and jQuery**: Add interactivity and dynamic functionality, such as real-time content updates and form validation.
- **Bootstrap**: A responsive design framework that ensures a consistent, mobile-first experience across devices

2. Objectives

The primary objective of this project is to design and implement *TravelTub*, an intuitive and responsive travel booking platform. The website will offer users an easy way to explore destinations, view packages, and make secure travel bookings. The following are the specific goals and objectives for the project:

2.1 **Checkpoint - 1:** Website Layout and Initial Setup

Objective: Establish the foundation of the website, ensuring the core structure and layout are functional and visually appealing.

Tasks:

- Create the homepage, which serves as the primary entry point.
- Design a navigation bar with links to Home, Book, Packages, Services, Gallery, and About sections.
- Add a dropdown menu under the Packages section with options like United States, India, France, and Germany.
- Implement Login and Register buttons in the navigation bar.
- Design a background image with a "Welcome" message and dynamic country names, along with a "Book Now" button at the center.

2.2 **Checkpoint - 2:** Travel Booking Form and Package Gallery

Objective: Create a user-friendly travel booking interface, along with a display of various travel packages.

Tasks:

- Design a two-section row:
    - The left section will showcase an image related to the brand.
    - The right section will feature a booking form, where users can select a destination, the number of people, and dates. Validation will ensure the form is filled correctly.

- Develop a package gallery section displaying multiple travel packages, with images, descriptions, pricing, ratings, and a "Book Now" button.
- Implement a Services section that highlights various services such as Affordable Hotels, Food & Drinks, and Safety Guides.

2.3 **Checkpoint - 3:** Gallery and About Section

Objective: Add visual appeal and information about the brand and services, along with an interactive gallery and company details.

Tasks:

- Add a Gallery section to showcase images of travel destinations and experiences, with hover effects using CSS transforms (scale).
- Create an About Us section with two columns: one for a company image and the other for company details such as establishment and background.
- Implement a footer that includes brand name, social media links, and copyright information.

2.4 **Checkpoint - 4:** User Registration and Login

Objective: Develop a functional user registration and login system for account management.

Tasks:

- Create a registration page or modal with fields for full name, contact number, date of birth, email, password, and gender. Implement JavaScript validation to ensure all fields are correctly filled.
- Develop a login page or modal that accepts email and password inputs for returning users, with a link to the registration page for new users.
- Deploying the website on platforms like GitHub Pages or Netlify, making it publicly accessible.

3. Project Description

The *TravelTub* project is a user-centric travel booking platform designed to simplify the travel planning process. Focused on delivering a seamless user experience, the website offers intuitive navigation, attractive visuals, and interactive features. It allows users to explore destinations, view travel packages, and easily book trips online. The front-end development of *TravelTub* incorporates various technologies to enhance its responsiveness, functionality, and visual appeal. Below is a detailed breakdown of the features, functionality, and user experience for each checkpoint, based on the code structure.

3.1 **Checkpoint - 1:** Website Layout and Initial Setup

Features and Functionality:

- **Landing Page & Dynamic Text:** The homepage features a large background image that immediately captures the user's attention. A dynamic text Animation is included to rotate between destination names (like, "Paris", "London", "Rome", etc.), changing every 0.2 seconds, helping users quickly engage with popular destinations.
- **Navigation Bar:** The sticky navigation bar ensures users can easily access sections like Home, Book, Packages, Services, Gallery, and About, no matter where they are on the page. The brand logo is positioned on the left, while the navigation links are placed to the right, ensuring a clean and structured layout.
    - Dropdown for Packages: The "Packages" dropdown menu allows users to explore travel options by destination, offering a quick way to navigate the desired package details.
    - Login & Register Buttons: Positioned at the far right, these buttons let users log into their accounts or register a new one, making account management seamless.
- **Responsive Background & Text:** The background image is set to cover the full width and a reduced height of 90vh, creating an immersive experience. The welcoming message and rotating destination names are centered to maintain readability and engagement. A Book Now button prominently placed in the center encourages users to start their booking journey.

3.2 **Checkpoint - 2:** Travel Booking Form and Package Gallery

Features and Functionality:

- **Booking Form:**
    - The form is split into two halves. The left side displays an image to visually represent the section, while the right side contains the functional booking form. The form allows users to:
        - Select a destination from a dropdown list populated with options.
        - Choose the number of travelers.
        - Pick start and end dates, with validation to ensure the end date is always later than the start date. The dates are dynamically adjusted using JavaScript to ensure valid selections.
        - Provide a description of the trip, with character count monitoring and validation to ensure the description stays within acceptable limits.
        - The Book Now button triggers an alert to confirm the booking details if all fields are correctly filled.
- **Package Gallery:**
    - The gallery displays nine travel packages divided into three rows, each featuring:
        - Images representing the destination.
        - Place Names as prominent headings.
        - A Description that highlights key attractions of the location.
        - Price in USD for the package.
        - Ratings give users a sense of the quality of each package.
        - A Book Now Button that directs users to the booking page for that specific package.
- **Services Section:**
    - This section highlights services offered by *TravelTub*, including affordable hotels, food & drinks, and safety guides. Three additional custom services are also featured, reflecting the platform's unique offerings. These services are displayed as images for better understanding.

3.3 **Checkpoint – 3:** Gallery and About Section

Features and Functionality:

- **Interactive Gallery:**
    - The gallery section showcases images of popular destinations, with hover effects that scale the images slightly to create an interactive experience. This feature encourages users to explore the various destinations offered by *TravelTub* in a visually engaging way.
- **About Us Section:**
    - The About section is split into two halves:
        - The Left Half displays an image representing *TravelTub*.
        - The Right Half contains text representing details on how users can interact with us. This section helps build trust with users by considering ways of interaction beyond the platform.
- **Footer:**
    - The footer includes essential information, such as the brand name, social links and copyright details. The footer is simple and functional, ensuring users can easily navigate the company's social profiles or understand legal information.

3.4 **Checkpoint - 4:** User Registration and Login

Features and Functionality:

- **User Registration:**
    - Upon clicking the "Register" button, users are presented with a modal form that prompts them to enter personal details such as name, contact number, date of birth, email address, password, and gender. The form includes JavaScript validation to ensure that all fields are correctly filled before submission.
    - A Login Link is provided for users who already have an account, allowing for easy navigation between the registration and login processes.
- **User Login:**
    - The login process also uses a modal to allow users to log in using their email and password. The form includes HTML5 validation to ensure that all fields are correctly filled before submission.

- A Registration Link is provided for users who don’t have an account, allowing for easy navigation between the registration and login processes

- **Website Deployment:**

After completing the development, the website was deployed on GitHub Pages, making it publicly accessible.

4. Additional Features

The following additional features go beyond the core functionalities of *TravelTub*, enhancing the user experience and ensuring smooth interaction throughout the website. These features were designed with user convenience and engagement in mind, leveraging modern front-end techniques to create an intuitive and responsive platform.

4.1 Real-Time Character Count for Trip Description

- **Feature:** As users enter their trip description, the website provides real-time feedback on the number of characters typed, ensuring they stay within the allowed limit.
- **Implementation:** JavaScript tracks the user's input and updates the character count dynamically.
- **Benefit:** This feature prevents users from exceeding the character limit, improving data submission accuracy and ensuring users' input aligns with form requirements.

4.2 Responsive Form Validation

- **Feature:** Forms are validated in real-time, with immediate visual feedback on the correctness of user inputs (e.g., for email, password, phone number).
- **Implementation:** Regular expressions are used to validate form fields as users type, providing instant feedback and guiding them to correct any errors.
- **Benefit:** This minimizes submission errors, enhances user experience by providing clear instructions, and speeds up form completion, ensuring higher accuracy and user satisfaction.

4.3 User-Friendly Modals for Registration and Login

- **Feature:** Registration and login forms appear as modals, allowing users to sign up or log in without navigating away from the main page.
- **Implementation:** Modal dialogs are used to overlay forms on the current page, triggered by JavaScript when the user clicks the appropriate buttons.

- **Benefit:** This eliminates the need for page reloads, maintaining user focus and ensuring a seamless transition between different parts of the website. It streamlines the process, providing a non-disruptive way to collect user information.

4.4 Dynamic Package Selection

- **Feature:** The travel package gallery allows users to select a package that automatically updates the destination in the booking form.
- **Implementation:** JavaScript dynamically updates the selected destination in the booking form based on the package chosen from the gallery.
- **Benefit:** This reduces the number of actions required by the user, making the booking process faster and more intuitive by enabling a seamless transition from browsing to booking.

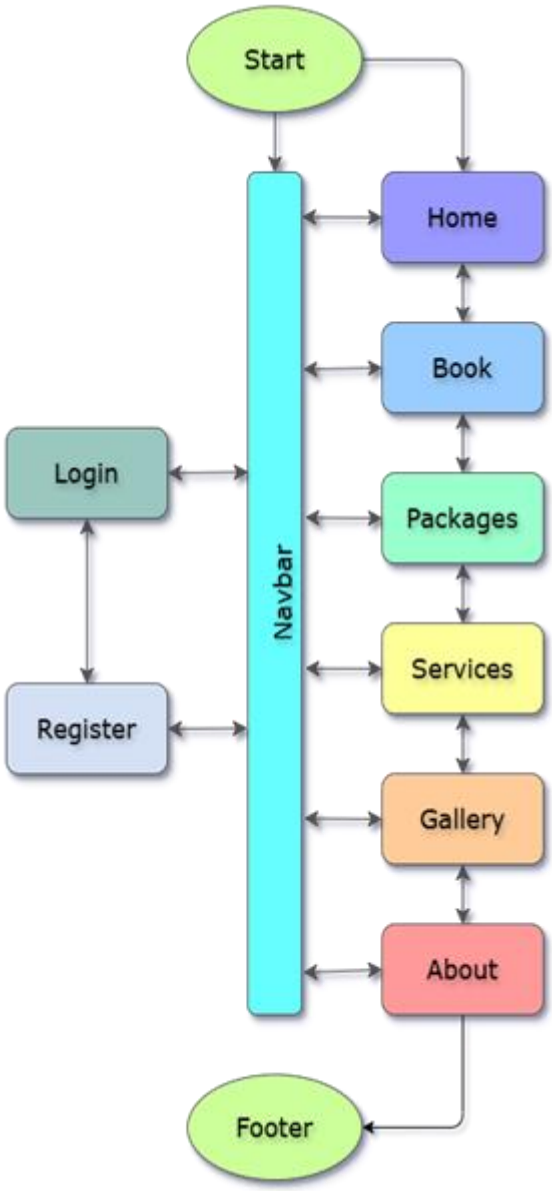
4.5 Date Picker with Date Validation

- **Feature:** A date picker is used to select trip start and end dates, with validation ensuring that the end date is always later than the start date.
- **Implementation:** JavaScript dynamically adjusts the available date ranges and enforces the date validation rule.
- **Benefit:** This prevents users from selecting invalid dates, enhancing the booking experience and minimizing errors.

5. Flowchart

The flowchart outlines the logical flow and interaction within the *TravelTub* website, illustrating how users navigate and interact with various components.

1. **Start:** Users land on the homepage.
2. **Navbar:** Central to navigation, the navbar provides links to all main sections:
    - **Home:** Welcomes users and highlights popular destinations.
    - **Book:** Takes users to the booking form for trip details.
    - **Packages:** Offers a dropdown menu to explore travel packages by destination.
    - **Services:** Showcases available travel services.
    - **Gallery:** Displays images of travel destinations.
    - **About:** Shares information about the **company.**
3. **Login/Register:** Allows users to create an account or log in.
4. **Footer:** Provides access to additional resources across the website.



This structure ensures seamless navigation and easy access to all essential features.

6. Coding

The coding aspect of the *TravelTub* project focuses on the implementation of the front-end functionality using web technologies, primarily HTML, CSS, JavaScript, jQuery, and Bootstrap. These technologies were chosen for their ability to deliver a responsive, interactive, and dynamic user experience.

6.1 Head Section

1. **Meta Tags:**
   - `<meta charset="UTF-8">:` Ensures the webpage uses the UTF-8 character encoding for proper text rendering.
   - `<meta name="viewport" content="width=device-width, initial-scale=1.0">:` Ensures the site is responsive, adapting to various screen sizes.

2. **Favicon:**
   - `<link rel="icon" type="image/x-icon" href="./images/logo.png">:` Specifies the favicon for the site, improving brand visibility in browser tabs.

3. **Title:**
   - `<title>TravelTub</title>:` Defines the title of the webpage, displayed in the browser tab.

4. **External Stylesheets:**
   - **Bootstrap 4.6.2:** A popular CSS framework used to facilitate responsive layout and component design.
   - **Font Awesome**: A comprehensive icon set that enhances the visual experience of the website.
   - **Google Fonts (Exo 2):** Custom font to create a modern and clean typography, improving readability and aesthetic appeal.

5. **Custom Stylesheet:**

- `<link rel="stylesheet" href="styles/styles.css">`: Links to the custom CSS file for any additional styling unique to the *TravelTub* project.

```
* {
    font-family          : "Exo 2", sans-serif;
    font-optical-sizing: auto;
    font-weight          : 600;
    font-style           : normal;
    transition           : ease-in-out 0.5s;
}
```

1. Universal Selector (*):

- This global rule applies to all HTML elements, ensuring a consistent appearance across the entire website.

2. Font:

- `font-family: "Exo 2", sans-serif;` Utilizes the Exo 2 font for a contemporary, professional look. It serves as the primary typeface, falling back to a generic sans-serif font if unavailable.

3. Transition:

1. `transition: ease-in-out 0.5s;` Smoothens transitions on interactive elements like buttons or links, enhancing the user experience when elements change state.

4. Variables

- The `:root` selector defines global CSS custom properties (variables) to simplify styling across the entire website
- **Variables Defined:**

```
:root {
    --bg_light           : #abd7ebc0;
    --bg_dark            : #03489dc0;
    --bg_hover           : #03489d;
    --rating             : #ff9e2f;
    --text_shadow        : #000000;
    --text_light         : #ffffff;
}
```

6.2 Navigation Bar

The navbar is designed with Bootstrap and custom CSS, providing a responsive, user-friendly navigation experience.

1. HTML Structure:

The HTML structure consists of a navbar that houses links, a logo, a collapsible menu, and dropdown elements. The structure is built using Bootstrap 4 classes for responsiveness.

```
<nav class="navbar navbar-expand-lg navbar-dark bg_dark fixed-top" id="navbar">
    <a class="navbar-brand" href="#">
        <img src="./images/logo.png" width="38" height="30" class="d-inline-block align-top" alt="Logo" loading="eager">
        TravelTub
    </a>
    <button class="navbar-toggler ml-auto mr-0" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav">
            <a class="nav-link active" href="#home">Home<span class="sr-only">(current)</span></a>
            <a class="nav-link" href="#book">Book</a>
            <span class="dropdown" id="packages-dropdown">
                <a class="nav-link" href="#packages" role="button" data-toggle="dropdown" aria-expanded="false">
                    Packages<i class="fa-solid fa-angles-down"></i>
                </a>
                <div class="dropdown-menu">
                    <!—dropdown_items-->
```

```
            </div>
        </span>
        <a class="nav-link" href="#services">Services</a>
        <a class="nav-link" href="#gallery">Gallery</a>
        <a class="nav-link" href="#about">About</a>
    </div>
  </div>
  <!-- Login and Register buttons  -->
</nav>
```

- **Branding:** The logo is displayed using an image `<img>` inside an anchor tag `<a class="navbar-brand">.` This also includes the website name, "TravelTub".
- **Toggler Button:** The button, marked with the navbar-toggler class, ensures that the navbar collapses into a hamburger menu on small screen sizes. It uses the Bootstrap class navbar-toggler-icon to display the icon for toggling.
- **Navigation Links:** Links to the main sections of the page, such as Home, Book, Packages, Services, Gallery, and About, are placed inside the navbar-nav container. Each link uses the nav-link class for consistent styling.
- **Dropdown Menu:** The "Packages" link is part of a dropdown menu. It includes a list of travel destinations that expand when the user clicks on the dropdown icon (`<i class="fa-solid fa-angles-down">`).

2. CSS for Navbar Styling:

The navbar styling is heavily based on Bootstrap's built-in classes, with some custom styles added to ensure the dropdown menu appears on hover.

```
.dropdown:hover > .dropdown-menu {
  display: block;
}
```

- **Dropdown Hover Effect:** The `:hover` pseudo-class is used to make the dropdown menu appear when the user hovers over the "Packages" link. The menu is set to `display: block;` when hovered to ensure visibility.

3. Responsiveness and Accessibility:

The navbar is built to be responsive, ensuring the navigation adapts to various screen sizes. Key accessibility features are also included.

- **Responsive Navbar:** The navbar-expand-lg class makes the navbar expand on large screens and collapse on smaller ones (e.g., tablets or mobile devices). The collapsing functionality is controlled by Bootstrap's JavaScript, which toggles the visibility of the navbar links when the hamburger icon is clicked.
- **ARIA Attributes:** The aria-controls, aria-expanded, and aria-label attributes on the navbar-toggler button enhance accessibility for users relying on screen readers. These attributes help describe the button's function to assistive technologies, making the site more accessible.

```
<button class="navbar-toggler ml-auto mr-0" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
  <span class="navbar-toggler-icon"></span>
</button>
```

4. Active Link Indication:

To highlight the currently active page or section, the active class is applied to the corresponding navigation link. This is particularly useful for users to know their location on the site.

```
<a class="nav-link active" href="#home">Home<span class="sr-only">(current)</span></a>
```

- The active class is added to the Home link by default to indicate it as the current section.
- The `<span class="sr-only">(current)</span>` tag ensures screen readers announce the "current" page.

6.3 Login & Registration Modals

Login and Registration modals provide user authentication features, including login and registration forms, as well as client-side validation.

1. HTML Structure:

The modals consist of two parts:

- **Login Modal:** This modal allows users to log in by providing an email and password.

- **Register Modal:** This modal allows users to register by providing full name, date of birth, gender, email, phone, and password.

Login Modal HTML:

```
<button type="button" class="btn btn-sm btn-success" data-toggle="modal" data-target="#login">Login</button>
<div class="modal" id="login">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Login</h5>
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <div class="modal-body">
        <form>
          <input type="email" class="form-control" placeholder="Email" required>
          <input type="password" class="form-control" placeholder="Password" required>
          <button type="submit" class="btn btn-primary">Login</button>
        </form>
        <a href="#">Forgot password?</a>
        <span>New? <a data-toggle="modal" data-target="#register">Sign up</a></span>
      </div>
    </div>
  </div>
</div>
```

- **Login Button:** The button triggers the modal on click using the `data-toggle="modal"` and `data-target="#login"` attributes.
- **Form Fields:** Includes E-mail and Password fields with appropriate placeholders.
- **Remember Me:** A checkbox is provided for users who wish to stay logged in.
- **Sign Up Link:** Links to the Register Modal for new users to sign up.

Register Modal HTML:

```
<button type="button" class="btn btn-sm btn-secondary" data-toggle="modal" data-target="#register">Register</button>
<div class="modal" id="register">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">Register</h5>
        <button type="button" class="close" data-dismiss="modal">&times;</button>
      </div>
      <div class="modal-body">
        <form>
          <input type="text" class="form-control" placeholder="Full Name" required>
          <input type="date" class="form-control" required>
          <input type="email" class="form-control" placeholder="Email" required>
          <input type="text" class="form-control" placeholder="Contact" required>
          <input type="password" class="form-control" placeholder="Password" minlength="8" required>
          <button type="submit" class="btn btn-primary" disabled>Register</button>
        </form>
        <span>Already registered? <a data-toggle="modal" data-target="#login">Login</a></span>
      </div>
    </div>
  </div>
</div>
```

- **Register Button:** The "Register" button triggers the modal for the registration form.

- **Form Fields:** Includes fields like Full Name, Date of Birth, Gender, Email, Contact, and Password.
- **Terms and Conditions:** Users are required to accept the terms before they can register.
- **Existing User Link:** Provides an option to log in if the user already has an account.

2. **JavaScript for Date of Birth (DOB) and Validation:**

The script contains functions for validating the form fields and ensuring that they meet the specified requirements.

Date of Birth (DOB) Calculation:

The function setDOB calculates the age range for registration, ensuring that the user is between 13 and 120 years old.

```
const setDOB = () => {
  let tomSplit = dateIncrement().split('-');
  tomSplit[0] -= 13;
  dob.max = tomSplit.join('-');
  tomSplit[0] -= 120;
  dob.min = tomSplit.join('-');
}
```

- **DOB Max:** The maximum date is set to 13 years ago, ensuring that users are at least 13 years old.
- **DOB Min:** The minimum date is set to 120 years ago, ensuring that the user is not over 120 years old.

Form Validation:

The regValidate function ensures that all form fields are valid before allowing the user to submit the registration form.

```
const regValidate = () => {
  const register = document.querySelector('div#register form');
  const regBtn = document.querySelector("div#register button[type='submit']");
  const formGroupsReg = register.querySelectorAll('.form-control');
  const formGroup = Array.from(formGroupsReg);
  if(formGroup.every((el) => el.classList.contains("is-valid"))) {
```

```
        regBtn.disabled = false;
    } else {
        alert ("please fill and validate all fields");
        terms.checked = false;
        regBtn.disabled = true;
    }
}
```

- The function checks if all the form fields have the class is-valid (which indicates they have passed validation).
- If valid, the Register button is enabled; otherwise, it is disabled.

Real-Time Field Validation:

```
$(".modal .form-control").keyup(function() {
    const regex = {
        name       : /^[a-zA-Z\s]+$/,
        email      : /^([a-zA-Z0-9._%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,})$/,
        phone      : /^\d{10}$/,
        password : /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-
]).{8,15}$/
    }

    var id = $(this).attr("id").split("-")[1];
    if(regex[id].test($(this).val())) $(this).addClass("is-
valid").removeClass("is-invalid");
    else $(this).addClass("is-invalid").removeClass("is-valid");
});
```

- This validates each field as the user types in real-time by checking the input against the specified regular expressions.

6.4 Homepage

The Homepage consists of a Jumbotron section and CSS styling. It uses the CSS Keyframes to animate text (travel destinations) in a looping manner. Below is a breakdown of how the

code works:

HTML:

```
<div class="jumbotron-fluid d-flex" id="home">
   <div class="container text-center text-light align-self-center">
      <p class="display-5 pb-5">Welcome to <span>TravelTub</span></p>
      <p class="display-4 pb-5">Visit<br><span id="places"
class="places"></span></p>
      <p class="lead">
         <a class="btn bg_dark btn-lg text-light" href="#book" role="button">
         Book Now
         </a>
      </p>
   </div>
</div>
```

- **Structure:**
  - The `<div>` with class jumbotron-fluid creates a full-width, flexible jumbotron component with a background.
  - The `<div class="container text-center text-light align-self-center">` centers the content inside the jumbotron, ensuring the text is in the middle of the screen and light-colored for visibility.
  - The `<p>` tags contain introductory text: one for the welcome message and another for a rotating list of travel destinations that are updated using CSS animations.
  - The `<a>` tag creates a call-to-action button that links to a booking section on the page.

CSS:

```
body {
   background : var(--bg_light) url("../images/homeBg.jpg") no-repeat
center;
   background-size : cover;
   background-attachment : fixed;
}
```

```
@keyframes places {
    11.1% { content: "Rome";                color: #FF5500; }
    22.2% { content: "Switzerland";         color: #FFAA00; }
    33.3% { content: "London";          color: #FFFF00; }
    44.4% { content: "South Island";        color: #AAFF00; }
    55.5% { content: "Grand Canyon";    color: #00FF00; }
    66.6% { content: "Singapore";           color: #00AAFF; }
    77.7% { content: "Dubai";           color: #0055FF; }
    88.8% { content: "Sydney";              color: #AA00FF; }
}

.places:after {
    content     : "Paris";
    color       : #FF0000;
    animation   : places 1.8s infinite;
}
```

- **Background Styling:**
  - The body background is set with a fixed background image (homeBg.jpg), which remains fixed while the content scrolls.
  - The background-size: cover; ensures the image covers the entire viewport.
  - The background-attachment: fixed; creates a parallax effect where the background stays still while content scrolls.
- **Keyframe Animation:**
  - `@keyframes` places defines a custom animation to cycle through a list of travel destinations. Each destination appears in the `.places` element at specific time intervals (11.1%, 22.2%, etc.) and changes color.
  - The content property is used to change the text inside the `.places` span.
  - The color changes with each destination, creating a colorful effect as each location appears.
- **.places:after:**
  - This pseudo-element is used to insert content ("Paris") into the `.places` span.
  - The animation runs continuously, cycling through the list of destinations defined in `@keyframes places`.

6.5 Booking Section

HTML:

The HTML section defines the structure of the booking form within a container-fluid class. The content is divided into rows and columns for responsiveness, utilizing Bootstrap's grid system.

1. **Header and Title**:

```
<h2 class="col-12 text-center text-light display-5">Book</h2>
```

This header introduces the booking section with a large display font.

2. **Image Section**:

```
<div class="col-md-6 mx-auto my-3">
  <img src="./images/forms.jpg" alt="image" class="image-fluid"
  style="object-fit: cover" width="100%" height="450rem">
</div>
```

This column contains an image for visual appeal, responsive to screen size changes. It uses the `object-fit: cover` CSS style to ensure that the image maintains its aspect ratio and covers the designated area.

3. **Booking Form**: The main form is structured with several fields:
    - **Dropdown for Destination**: A select element where the user chooses their destination (e.g., Paris, Rome, London).
    - **Number of People**: A number input field for specifying the number of people traveling.
    - **Start and End Dates**: Date input fields to select the travel start and end dates, with validation to ensure the end date is after the start date.
    - **Description**: A `textarea` element where users can input travel details with a live character count.
    - **Buttons**: There are two buttons, one for submitting the form and another for clearing the form.

JavaScript:

1. **Dynamic Date Handling**:

```
const dateIncrement = (d) => {
    const date = d ? new Date(d) : new Date();
    date.setDate(date.getDate() + 1);
    const day = date.getDate();
    const month = date.getMonth() + 1;
    return `${date.getFullYear()}-${(month > 9) ? month : "0" +
String(month)}-${(day > 9) ? day : "0" + String(day)}`;
}
```

The dateIncrement function increments a given date by one day. This is used to ensure the minimum valid start and end dates for the travel booking.

2. **Setting Minimum Dates**:

```
const setStart = () => startDate.min = dateIncrement();
const setEnd = () => {
    endDate.min = dateIncrement(startDate.value);
    endDate.disabled = false;
}
```

The setStart function ensures that the minimum value for the start date input is set to today's date. The setEnd function ensures the end date cannot be earlier than the start date.

3. **Form Submission**:

```
const formSubmit = (e) => {
    e.preventDefault();
    const dialog = document.getElementById('formModal');
    dialog.showModal();
    document.getElementById('ok').onclick = () => dialog.close(); }
```

The formSubmit function prevents the default form submission (which would cause a page reload) and instead displays a modal to inform the user that their booking is successful.

4. **Live Character Count**:

```
const setCount = () => textChar.innerText = description.value.length;
```

This code sets up a live character counter for the description input field. It updates the

displayed count as the user types in the description field.

5. **Event Listeners**:

```
startDate.onclick   = setStart;
```

- The `startDate.onclick` event ensures that the `setStart` function is called when the user clicks the start date input field.

```
startDate.oninput   = setEnd;
```

- The `startDate.oninput` event triggers `setEnd` to dynamically enable the end date input once a start date is selected.

```
description.onkeyup = setCount;
```

- The `description.onkeyup` event triggers the `setCount` function, updating the character count display as the user types in the description.

6.6 Package Section

The Packages section of the *TravelTub* project showcases different travel destinations in a card format.

1. **HTML Structure:**

- The `container-fluid bg_dark` provides a full-width section with a dark background (`bg_dark` class), which contrasts well with the light-colored text.
- Each package is presented within a Bootstrap grid system (`row` and `col` classes) to ensure responsiveness, with the package cards adjusting to different screen sizes (small, medium, and large).
- **Each destination card includes:**
    - An image (`card-img-top`), displaying a representative picture of the destination.
    - A title (`card-title`) for the destination name.
    - A short list of key attractions (`card-text`) for each destination, providing an overview of popular sites and experiences.
    - Price details in a visually separated span section.
    - A star rating displayed through icons (`fa-solid fa-star` and `fa-star-half-stroke`) to indicate the popularity or quality.
    - A "Book Now" button that links to the booking section.

2. **CSS Styling:**

- The following CSS hides the scrollbar for a cleaner look:

```
.card-text::-webkit-scrollbar {
    display: none;
}
```

- Custom styling is used to ensure text readability and visual appeal, with adjustments for image and card layout.
- The `overflow-y: scroll` style enables the attractions list to scroll within a fixed height for uniformity.

3. **JavaScript Functionality:**

```
$("#packages a.btn").click(function() {
    var id = $(this).closest("div[id]").attr("id");
    $("#book select option[value=" + id + "]").attr("selected", "true");
    setTimeout(function() { $("#book #numberOfPeople").focus(); }, 500);
});
```

- The click function for the "Book Now" button retrieves the selected destination's id and pre-selects it in the booking form's dropdown.
- A slight delay (`setTimeout`) helps focus on the "Number of People" field, guiding users smoothly through the booking flow.

6.7 Services Section

The "Services" carousel within the *TravelTub* project is a visually dynamic section that leverages Bootstrap's carousel component to effectively display the diverse range of services offered by the platform.

Section Container:

- A `container-fluid bg_light div` with `text-left` alignment and `m-auto` for centering.
- Contains a header (`h2`) titled "Services," centered and styled with a larger font and light color for contrast.

Carousel:

- **Carousel Container:**
    - The Bootstrap carousel (carousel slide) is set inside a container with a defined maximum width (w-75 and max-width: 700px) to ensure it doesn't stretch too wide on large screens.
    - data-ride="carousel" enables auto-slide functionality.
    - Carousel has an id (carouselservices), which is referenced by controls and indicators.
- **Indicators:**
    - Carousel indicators (carousel-indicators) enable direct navigation between slides.
    - Each <li> represents a slide, with the active class applied to the first slide.
- **Carousel Slides:**
    - carousel-inner serves as the wrapper for individual slides. Each slide is a carousel-item with the first slide marked active.
    - Slides display images (img) that showcase different services (e.g., hotel, food, cabs). Each image is centrally aligned (d-block m-auto w-75) for a clean, balanced look.

Controls:

- Prev/Next controls (carousel-control-prev and carousel-control-next) allow manual navigation, with accessible labels (sr-only) for screen readers.

6.8 Gallery

The Gallery section of the *TravelTub* project utilizes a Bootstrap carousel to display images of popular travel destinations in a visually engaging way. The section is designed for responsiveness and interactivity, offering users immersive browsing experience.

1. HTML Structure:

The Gallery section is housed within a container-fluid to ensure it spans the full width of the viewport while maintaining responsive behavior across devices.

```
<div class="container-fluid m-auto bg_dark" id="gallery">
```

```
    <h2 class="text-center display-5 text-light">Gallery</h2>
    <div id="carouselgallery" class="carousel slide" data-ride="carousel">
        <div class="carousel-inner p-3 m-auto">
            <!-- Carousel Items -->
        </div>
    </div>
    <!-- Carousel Controls -->
    <div class="container-fluid text-right">
        <a class="btn btn-primary" href="#carouselgallery" role="button" data-slide="prev">
            <i class="fa fa-arrow-left"></i>
        </a>
        <a class="btn btn-primary btn-lg" href="#carouselgallery" role="button" data-slide="next">
            <i class="fa fa-arrow-right"></i>
        </a>
    </div>
</div>
```

2. Responsive Layout:

The images within each carousel item are displayed using the Bootstrap grid system, which ensures the gallery is responsive on all screen sizes. The use of `col-sm-12 col-md-6 col-lg-4` ensures that images adapt to the screen size, displaying in a single column on small screens, two columns on medium screens, and three columns on large screens.

```
<div class="carousel-item active">
    <div class="row">
        <div class="col-sm-12 col-md-6 col-lg-4 m-auto pb-3">
            <img class="img-fluid rounded" alt="Gallery-image" src="./images/destinations/south-island.jpg">
        </div>
        <!-- More images here -->
    </div>
</div>
```

3**. CSS Styling:**

The gallery images have been enhanced with a hover effect to provide interactivity. The following CSS scales up the images slightly when hovered over, along with a smooth transition.

```
#gallery img:hover {
   transform: scale(1.05);  /* Slight zoom effect */
   transition: transform 0.3s ease-in-out;  /* Smooth transition */
}
```

This effect adds a subtle visual cue for the user, encouraging interaction with the images. The transition ensures that the zoom effect appears smooth, enhancing the overall user experience.

6.9 About Us Section

The About Us section gives an overview of the company, showcasing the logo, offices, and contact details, encouraging user interaction.

HTML Structure:

```
<div class="row container-fluid text-center m-auto pb-3" id="about">
   <h2 class="text-center display-5 text-light col-12">About Us</h2>

   <!-- Logo -->
   <div class="col-md-6 m-auto">
      <img src="./images/logo.png" alt="Logo" class="w-50" style="max-
width: 350px;">
   </div>

   <!-- Information -->
   <div class="col-md-6 text-left m-auto bg_light p-3 rounded">
      <div class="d-flex flex-column flex-md-row">
         <div class="col-md-6">
            <h3>Our Offices</h3>
            <p>Located globally to assist in planning your perfect
```

```
vacation.</p>
            </div>
            <div class="col-md-6">
                <h3>Write to Us</h3>
                <p>For inquiries, feedback, or suggestions, email us at:</p>
                <a
href="mailto:contact@traveltub.com">contact@traveltub.com</a>
            </div>
        </div>
    </div>
</div>
```

Responsive Design:

- On large screens, content is displayed side-by-side (flex-md-row).
- On smaller screens, content stacks vertically for better readability.

CSS Styling:

- **Logo:** The w-50 class scales the logo, while max-width: 350px ensures it doesn't grow too large.
- **Background:** Light background (bg_light), padding (p-3), and rounded corners (rounded) for modern styling.
- **Text Alignment:** text-left for information sections and text-center for the heading.

Flexbox Layout:

Uses d-flex and flex-column for vertical alignment on small screens and flex-md-row for horizontal alignment on larger screens.

6.10 Footer

The Footer section offers key links for user engagement, branding, and social media interaction, making it an essential part of the website's layout.

Key Components:

1. **Logo and Copyright:**

- Contains the website's logo with a small copyright notice and "All Rights Reserved" text.
- The logo image is responsive and uses the `loading="lazy"` attribute to optimize loading performance.

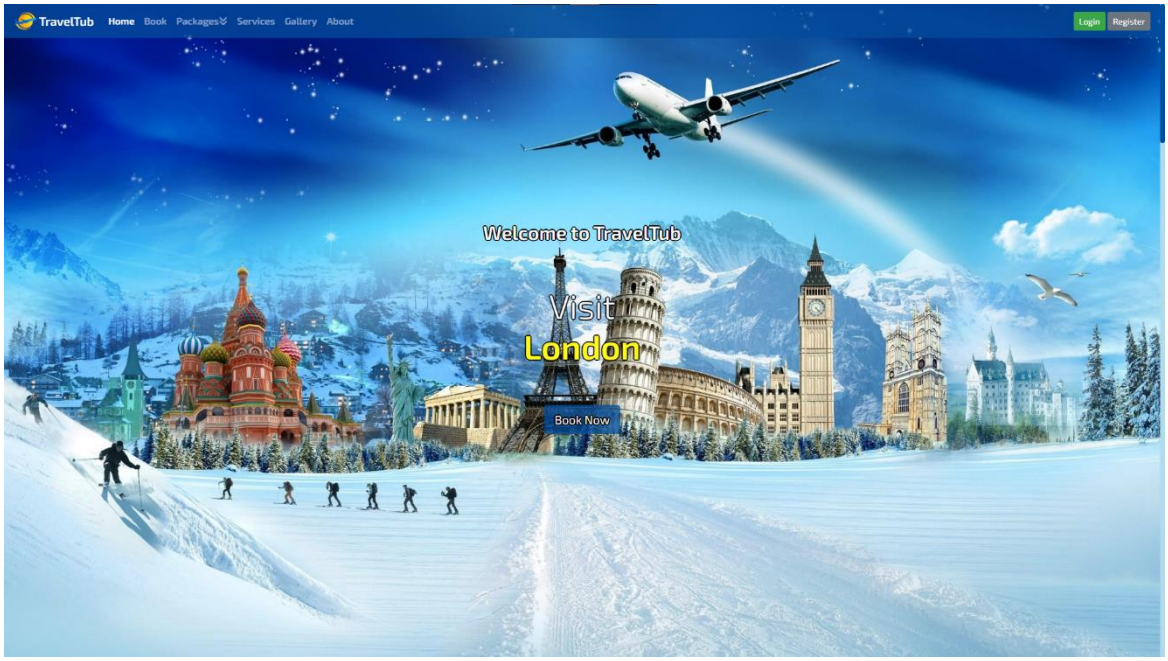
2. **Find Us Section:**
   - Displays a link to the website's GitHub repository with a GitHub icon for users to find the project's code.
   - The `text-nowrap` ensures the text and icons don't break into multiple lines.
3. **Follow Us Section:**
   - Links to the website's social media accounts (Twitter and Instagram) using font-awesome icons.
   - The `nav-link` class is used for consistent styling with other navigation elements.

Layout:

- The layout uses Bootstrap's flexbox utility classes:
  - `d-flex` creates a flexible container, aligning elements horizontally.
  - `flex-wrap` ensures that the footer elements will wrap appropriately on smaller screens.
  - `justify-content-between` spaces out the sections evenly across the width of the footer.
  - `justify-content-center` and `justify-content-end` are used to position content within their respective columns.

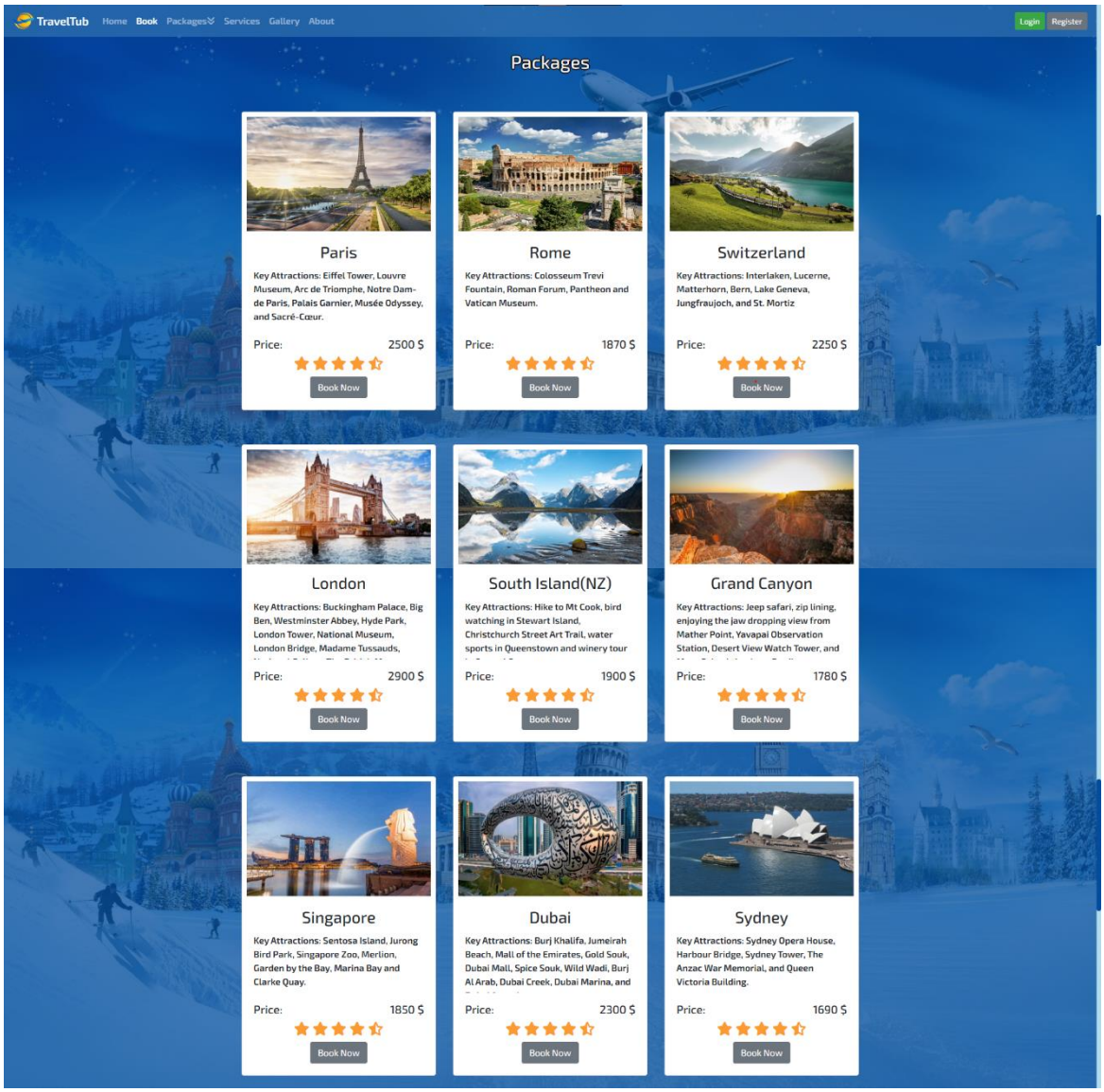
7. Output

7.1 Navigation Bar & Home

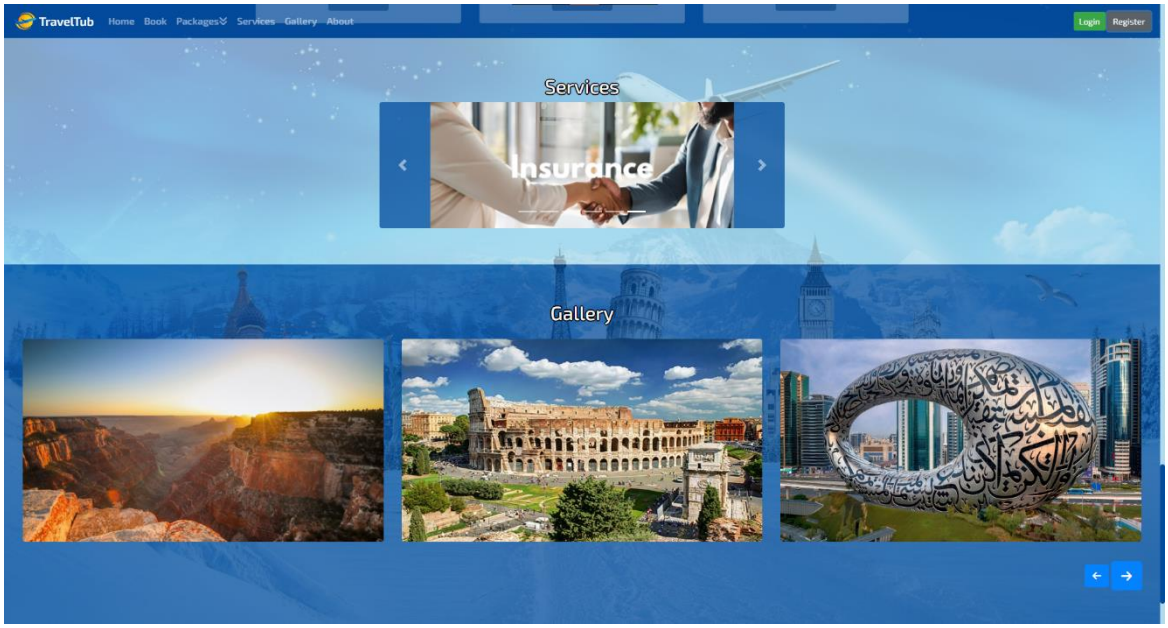


7.2 Booking Section

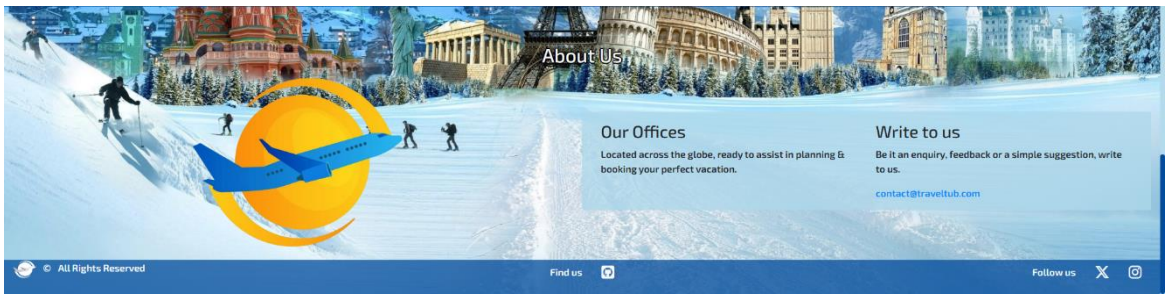
7.3 Packages Section



7.4 Services & Gallery



7.5 About Us & Footer



8. Validation

The *TravelTub* project meets all the objectives outlined in the project description through a combination of functional features and structural validations. The goal was to ensure that all functionalities work smoothly and that the webpage structure adheres to the specified requirements. Below, we describe the various aspects of the validation process:

8.1 Meeting Webpage Structure Requirements

One of the primary objectives of the project was to build a responsive, user-friendly webpage that aligns with the design and content goals. The structure of the webpage has been designed with the following requirements in mind:

- **Responsive Design:** The webpage is fully responsive, thanks to the use of Bootstrap 4. The layout adapts seamlessly across various screen sizes, from mobile phones to large desktop displays.
- **Navigation Bar:** A clear and intuitive navigation bar is implemented for easy access to key sections of the website, including Home, Login, Registration, and About Us.
- **Content Layout:** The content is structured with appropriate grid layouts and cards, ensuring readability and accessibility. For instance, the homepage displays essential information in a visually appealing manner.
- **Forms and Modals**: The login and registration forms are designed in modals that are triggered by buttons. This helps keep the page clean and ensures that users can quickly access the forms without navigating away from the homepage.

```
<!-- Modal for Login -->
<button type="button" class="btn btn-success" data-toggle="modal"
data-target="#loginModal">Login</button>
<!-- Modal for Registration -->
<button type="button" class="btn btn-secondary" data-
toggle="modal" data-target="#registerModal">Register</button>
```

These structural elements ensure that the webpage is organized and follows a user-friendly flow. The design and functionality align with the objectives of providing a seamless user experience.

8.2 Form Validation for Login and Registration

The login and registration forms are crucial to the user experience, and validation was implemented to ensure that these forms only accept valid data. Real-time validation is performed using regular expressions, ensuring that user input is immediately checked for correctness.

Login Form Validation (Email & Password)

```
$(".modal .form-control").keyup(function() {
    const regex = {
        email: /^([a-zA-Z0-9._%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,})$/,
        password: /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-
]).{8,15}$/
    }

    var id = $(this).attr("id").split("-")[1];
    if(regex[id].test($(this).val())) $(this).addClass("is-
valid").removeClass("is-invalid");
    else $(this).addClass("is-invalid").removeClass("is-valid");
});
```

In this code, the email and password fields are validated to ensure that users enter data in the correct format before they are allowed to submit the form. This guarantees that only valid information is processed.

Registration Form Validation (Name, Email, Phone, Password)

```
$(".modal .form-control").keyup(function() {
    const regex = {
        name: /^[a-zA-Z\s]+$/,
        email: /^([a-zA-Z0-9._%-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,})$/,
        phone: /^\d{10}$/,
        password: /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-
]).{8,15}$/
    }
```

```
    var id = $(this).attr("id").split("-")[1];
    if(regex[id].test($(this).val())) $(this).addClass("is-
valid").removeClass("is-invalid");
    else $(this).addClass("is-invalid").removeClass("is-valid");
});
```

For the registration form, fields such as Full Name, Email, Phone, and Password are validated. This ensures users can only submit forms with valid and properly formatted data.

8.3 Modal Validation

The Login and Registration modals are validated to function correctly. They open when triggered by their respective buttons and close when the close button or "Sign up" link is clicked. This modal functionality ensures that the user flow is not interrupted, and users can easily switch between login and registration without leaving the current page.

```
<!-- Login Modal -->
<div class="modal" id="loginModal">
    <div class="modal-dialog">
        <div class="modal-content">
            <!-- Modal content here -->
        </div>
    </div>
</div>
```

This modal structure adheres to Bootstrap standards and ensures that the user interface remains clean and intuitive.

8.4 Date of Birth Validation

For the Registration Form, the Date of Birth field is restricted to ensure the user is within an acceptable age range. The user must be at least 13 years old to register.

```
const setDOB = () => {
    let tomSplit = dateIncrement().split('-');
    tomSplit[0] -= 13;  // Minimum age: 13
    dob.max = tomSplit.join('-');
```

```
    tomSplit[0] -= 120; // Maximum age: 120
    dob.min = tomSplit.join('-');
}

dob.onclick = setDOB;
```

This code dynamically sets the maximum and minimum date values for the Date of Birth field, enforcing an age restriction of 13 to 120 years.

8.5 Terms and Conditions Validation

The registration form includes a Terms and Conditions checkbox, which must be checked before the user can submit the form. The Register button is disabled until the user agrees to the terms, ensuring compliance.

```
const regValidate = () => {
    const register = document.querySelector('div#register form');
    const regBtn = document.querySelector("div#register
button[type='submit']");
    const terms = document.getElementById("terms");

    if (terms.checked) {
        regBtn.disabled = false;
    } else {
        regBtn.disabled = true;
    }
}

terms.onclick = regValidate;
```

This code ensures that users must agree to the terms before proceeding with registration.

8.6 Performance and User Experience

The user experience is enhanced by providing immediate feedback on form fields using real-time validation. By using Bootstrap's classes (like `.is-valid` and `.is-invalid`), users can

easily see if they have entered correct or incorrect data, ensuring a smooth and guided interaction. Additionally, the use of modals for login and registration keeps the interface uncluttered while maintaining accessibility.

8.7 Final Submission Validation

The system ensures that forms are not submitted until all fields are valid. If the user tries to submit a form with invalid or incomplete data, a message prompts them to correct the issues, providing clear and actionable feedback.

9. Conclusion

9.1 Project Success

The *TravelTub* project has been a successful implementation of front-end web development, meeting its objectives and providing a robust and user-friendly web interface. The project was designed with the goal of offering users an intuitive platform for managing travel-related information, including login and registration functionalities, along with a responsive and visually appealing layout.

Key successes include:

- **Responsive Web Design:** The site has been optimized for both mobile and desktop use, ensuring users across various devices can easily interact with the website. This was accomplished using Bootstrap 4, ensuring a seamless experience for users, regardless of their screen size.
- **User Interaction and Experience:** The project incorporates real-time form validation for both the Login and Registration modals, ensuring that users input valid data. The modals are intuitive and do not disrupt the user flow, contributing to a clean, functional user interface.
- **Functional Features:** Features such as the Date of Birth validation, Terms and Conditions acceptance, and the use of modals for authentication and registration contribute to the overall usability and security of the webpage, improving user trust and the user experience.

The combination of these features demonstrates that the project fulfills the major requirements and provides users with a functional and responsive web page.

9.2 Lessons Learned and Future Improvements

While the project has been successful, there are several areas where improvements could be made to further enhance both the functionality and performance:

1. **Integration with Back-End Services:**
    - While the front-end implementation meets the functional requirements,

integrating the website with a back-end service would allow the website to save user data, such as login credentials and registration information. This would improve overall interactivity and make the platform more dynamic and personalized for the users.

- **Future improvement:** Incorporating API calls for authentication, user profile management, and travel data would elevate the platform's capability.

2. **Enhanced User Interface:**
   - While the current design is clean and responsive, there is room for further enhancing the UI by adding animations, transitions, and interactive elements to make the website even more engaging.
   - **Future improvement:** Adding CSS animations for form submissions, button hovers, and modal transitions could improve the website's visual appeal and interactivity.

3. **Accessibility Improvements:**
   - While the project has used semantic HTML and some aria roles to improve accessibility, there could be further steps to ensure the website is fully accessible for users with disabilities.
   - **Future improvement:** Implementing more comprehensive ARIA (Accessible Rich Internet Applications) attributes, including keyboard navigability and screen reader-friendly features, would enhance accessibility.

4. **Cross-Browser Compatibility:**
   - Ensuring that the website performs consistently across different browsers (Chrome, Firefox, Edge, Safari) is critical for wide accessibility. Minor issues might arise with browser-specific quirks, which can affect layout or functionality.
   - **Future improvement:** Conducting cross-browser testing or similar validations would help ensure consistent behavior across browsers.

5. **Security Enhancements:**
   - As the website involves user authentication, security is crucial. While the front-end handles form validation well, additional layers of security could be added to

protect against malicious input, such as cross-site scripting (XSS) or SQL injection (if back-end integration is added).

- **Future improvement:** Implementing input sanitization and strong password policies (e.g., password hashing) on the server side would improve security.

9.3 Key Takeaways

The *TravelTub* project highlights the importance of responsive design, real-time validation, and a user-friendly interface in front-end development. The key takeaways from the project are:

- **Responsive Layout:** A mobile-first approach ensures a wide user base across various devices.
- **Real-time Form Validation:** Form validation, including checks for valid email, password strength, and date of birth, enhances data integrity and user experience.
- **Modular Design:** The use of Bootstrap and modals contributed to a clean structure, while maintaining ease of access to critical features like login and registration.

9.4 Conclusion

In conclusion, *TravelTub* successfully met its objectives, providing a robust front-end that is both functional and easy to use. By focusing on core principles such as responsive design, validation, and interactive features, the project delivers a solid user experience. Going forward, incorporating back-end integration, improving UI/UX with animations, and ensuring accessibility and security will make the platform even more reliable and user-friendly, further aligning it with industry standards.

References

1. **Bootstrap 4**

   Bootstrap was used for responsive layout, modals, and UI components.

   - Reference:

     Bootstrap. (2020). *Bootstrap 4 Documentation*. https://getbootstrap.com/docs/4.5/

2. **jQuery**

   jQuery was used for DOM manipulation and event handling, including form validation and modal behavior.

   - Reference:

     jQuery Foundation. (2020). *jQuery*. https://jquery.com/

3. **JavaScript**

   JavaScript was used for dynamic functionality, including handling form validation, modal toggling, and managing user input.

   - Reference:

     Mozilla Contributors. (2020). *JavaScript*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/JavaScript

4. **CSS**

   Custom CSS, including CSS variables, was used for styling the webpage's layout, colors, and components.

   - Reference:

     W3C. (2020). *CSS: Cascading Style Sheets*. https://www.w3.org/Style/CSS/

5. **Font Awesome**

   Font Awesome was used for icons, enhancing the webpage's user interface and experience.

   - Reference:

     Fonticons, Inc. (2020). *Font Awesome*. https://fontawesome.com/

6. **Google Fonts**

   Google Fonts were used to import and use modern web fonts.

   - Reference:

     Google. (2020). *Google Fonts*. https://fonts.google.com/