  
**S.P.Mandali’s**

**Ramnarain Ruia Autonomous College**

***(Affiliated with the University of Mumbai)***

**Field Project Documentation**

**Group 10**

|  |  |  |
| --- | --- | --- |
| ***Sr No*** | ***Name*** | ***Roll No*** |
| 1 | Adishree Khamgaonkar | 4518 |
| 2 | Anuja Kharat | 4519 |
| 3 | Darpan kondagekar | 4520 |
| 4 | Aastha Meher | 4525 |
| 5 | Kshiti Vartak | 4555 |
| 6 | Viraj Kamble | 4557 |
|  | | |

**Field Project Report on**

**Project Name: BHARATYATRA**

**Project Guide: Priyanka Vaddepalli**

**Project By: Adishree, Aastha, Anuja, Kshiti, Darpan, Viraj**

**SYBSC Computer Science 2024-25**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr no** | **Particulars** | **Page no** |
| 1 | Problem statement | 3 |
| 2 | Introduction | 3 |
| 3 | System analysis | 4 |
| 4 | System Design | 8 |
| 5 | Implementation | 11 |
| 6 | Results | 15 |
| 7 | Conclusion | 16 |
| 8 | Reference | 17 |

A] Problem Definition:

Tourists often struggle to find the best hotels and nearby attraction, leading to a less enjoyable travel experience. There’s too much information online which makes it hard to compare options and decide where to stay and what to visit. We need an easy to use app that help tourists quickly find the right hotel and discover nearby places of interest, making their trip planning simpler and more enjoyable

B] Introduction:

* Motivation:

Our motivation for building this app is to simplify the process of choosing from the vast amount of information available for tourism. By providing curated, personalized recommendations for nearby places local attractions and hotels, we aim to enhance the tourist experience, making it easier and more efficient for travellers to plan their visits and find the best accommodations.

* Problem statement:

Tourists struggle to find the best hotels and attractions due to overwhelming online information, making travel planning difficult. Our app simplifies this process by providing an easy-to-use platform to quickly discover and compare nearby places and accommodations.

* Purpose objective and goals:

**Enhance Tourist Experience**: Provide convenient, personalized recommendations for nearby attractions and accommodations.

**Promote Local Businesses:** Increase visibility and support for local hotels, restaurants, and attractions.

**Improve Accessibility:** Offer easy access to information, maps, and navigation for tourists.

**Ensure Safety and Comfort:** Provide reliable, verified listings and essential safety information.

**Cultural Exchange:** Offering information about cultural and historical significance of places to enrich tourists understanding and appreciation by highlighting local events and festivals to enhance the cultural experience.

**Encourage Sustainability:** Promote eco-friendly options and support local culture and traditions.

C] System Analysis:

* Existing Systems
* Scope and Limitations of Existing Systems:

Current travel websites although beneficial independently do not sufficiently offer an integrated approach to travel and tourism. Most of the recommendation systems are domain-specific, meaning they target a specific type, such as hotel bookings or flight reservations or even simple tours. However, they often fail to incorporate detailed transportation options or personal adaptable schedules that might the user favour. For instance, a website such as Google Maps will likely do a good job at identifying places of interest but has little data on how to get from one to the other. In the same way, a site such as TripAdvisor offers copious information and descriptions of the attractions but lacks a good connection between transport means or optimized suggested routes and schedules according to the client’s preferences.

* Notable features

**Reviews and Ratings:**

* Users can browse and contribute reviews and ratings for hotels, restaurants, attractions, and destinations worldwide.
* Detailed insights into service quality, cleanliness, location, and value for money.

**Travel Forums:**

* A space for users to ask questions and receive travel advice from locals or fellow travellers.
* Destination-specific discussions, including tips, recommendations, and trip reports.

**Hotel and Restaurant Search:**

* Search for hotels, restaurants, and attractions using filters for price, rating, type of cuisine, and amenities.
* Integrated booking options through partner websites.

**Photos and Videos:**

* Users can upload and browse photos and videos of hotels, restaurants, attractions, and experiences.
* Helps travelers get a visual idea of the destination.

**Things to Do:**

* A dedicated section that lists top attractions, tours, and activities at various destinations.
* Can book tours and activities directly through the platform.

**Trip Planning Tools:**

* Features like "Trip Planner" allow users to save and organize ideas for trips, including attractions, accommodations, and restaurants.
* Users can create custom itineraries and share them with others.

**Maps and Navigation:**

* Integrated maps to find places nearby your current location or planned destination.
* Includes distance and directions to restaurants, attractions, and accommodations.

**Flight Search:**

* Search and compare flight prices from various airlines and booking platforms.
* View flight details such as duration, layovers, and reviews of airlines.

**Traveller Photos and Experiences:**

* User-generated content allows travellers to view real experiences, not just curated photos.
* Reviews often include personal travel tips and hidden gems.

**Booking and Deals:**

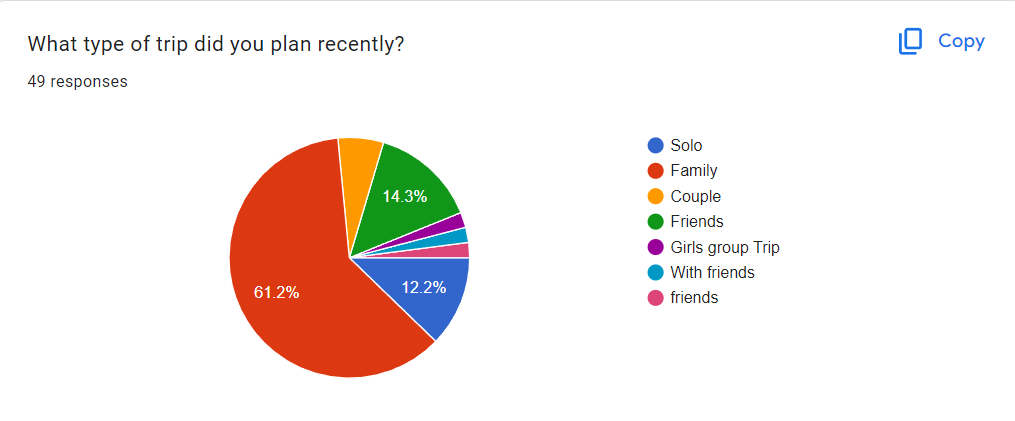
* Users can book hotels, tours and restaurants directly through the site or app.
* Special offers and discounts available from partners.

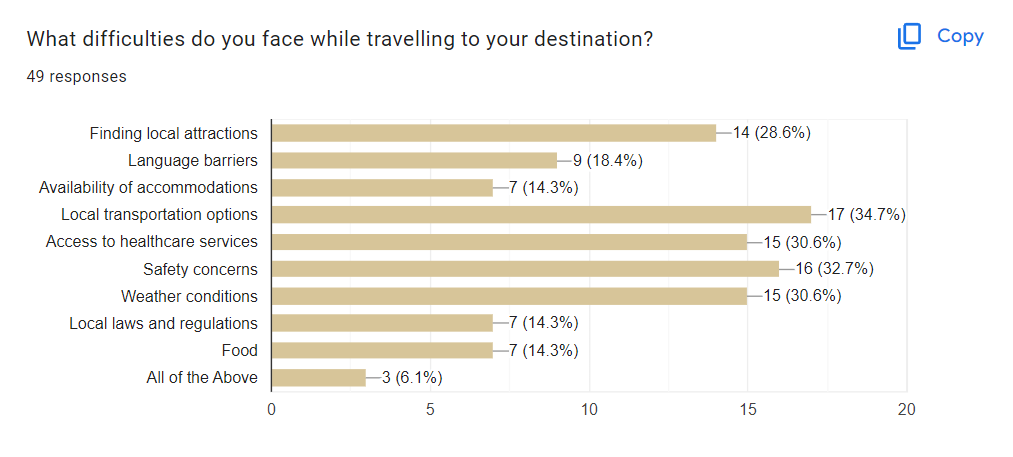
**Traveller Articles and Guides:**

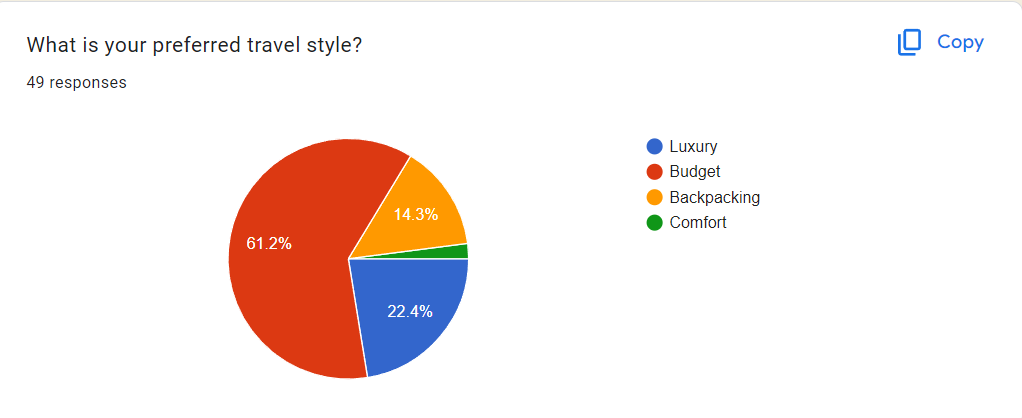
* Destination-specific articles and travel guides curated by experts and users.
* Tips on local culture, must-see spots, and travel safety.
* Data collection methods

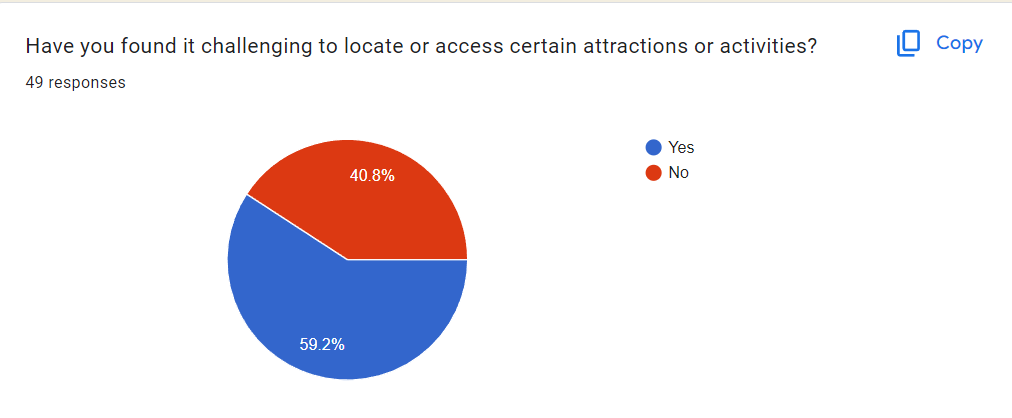
To ensure the content is relevant and helpful, we collected data through a Google Form survey, which allowed us to gather insights directly from travellers and locals. The survey covered various aspects, such as popular tourist spots, difficulties faced while travelling, preferred travel styles, and recommendations for accommodations. This data-driven approach ensures that our website meets the needs of tourists by offering tailored suggestions and up-to-date information for an enriching travel experience.

* Data Visualization









* Requirement analysis

**Functional Requirements:** We have designed our website to provide a user-friendly interface where users can easily input their location and receive personalized information about nearby attractions, travel modes, hotels, and suggested itineraries. The website includes interactive maps and filters, allowing users to explore various types of attractions such as historical sites, nature spots, and entertainment hubs. Users can also book hotels and travel services directly from the website and create and save customized itineraries. To further enhance user experience, we have integrated recommendation features that suggest attractions based on user preferences and previous selections.

**Performance Requirements:** To ensure a smooth and responsive experience, we focused on optimizing the website's performance. Our goal was to keep the loading time of the main pages under 3 seconds, even during peak traffic. We have implemented caching, optimized database queries, and minimized the use of heavy scripts to handle at least 1,000 simultaneous users without compromising performance. The search functionality and map interactions have been designed to provide instant results, typically within a second, ensuring a seamless and engaging user experience.

**Security Requirements:** Understanding the importance of data security, we have implemented robust security measures for our website. Secure login and user authentication protocols have been put in place to protect user accounts and sensitive data. All data transmissions, especially those involving personal information or payment details, are encrypted using HTTPS. We have also conducted thorough security testing to safeguard against vulnerabilities such as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). These measures ensure that our website provides a secure environment for all users, maintaining data integrity and privacy at all times.

D] System Design:

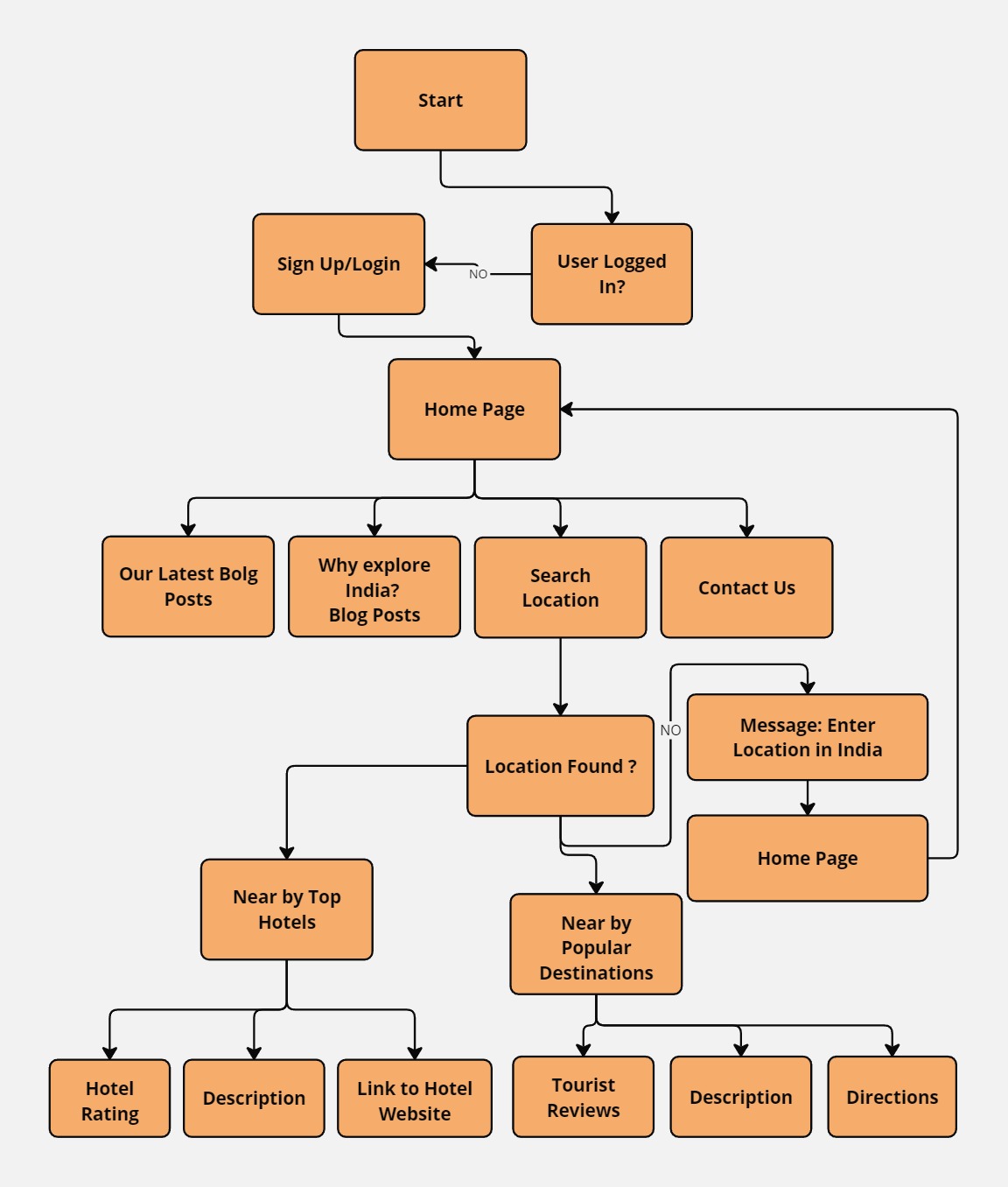
* Design Constraint

**Responsive Design:** Ensuring the website functions well on all device sizes (mobile, tablet, desktop) for an optimal user experience.

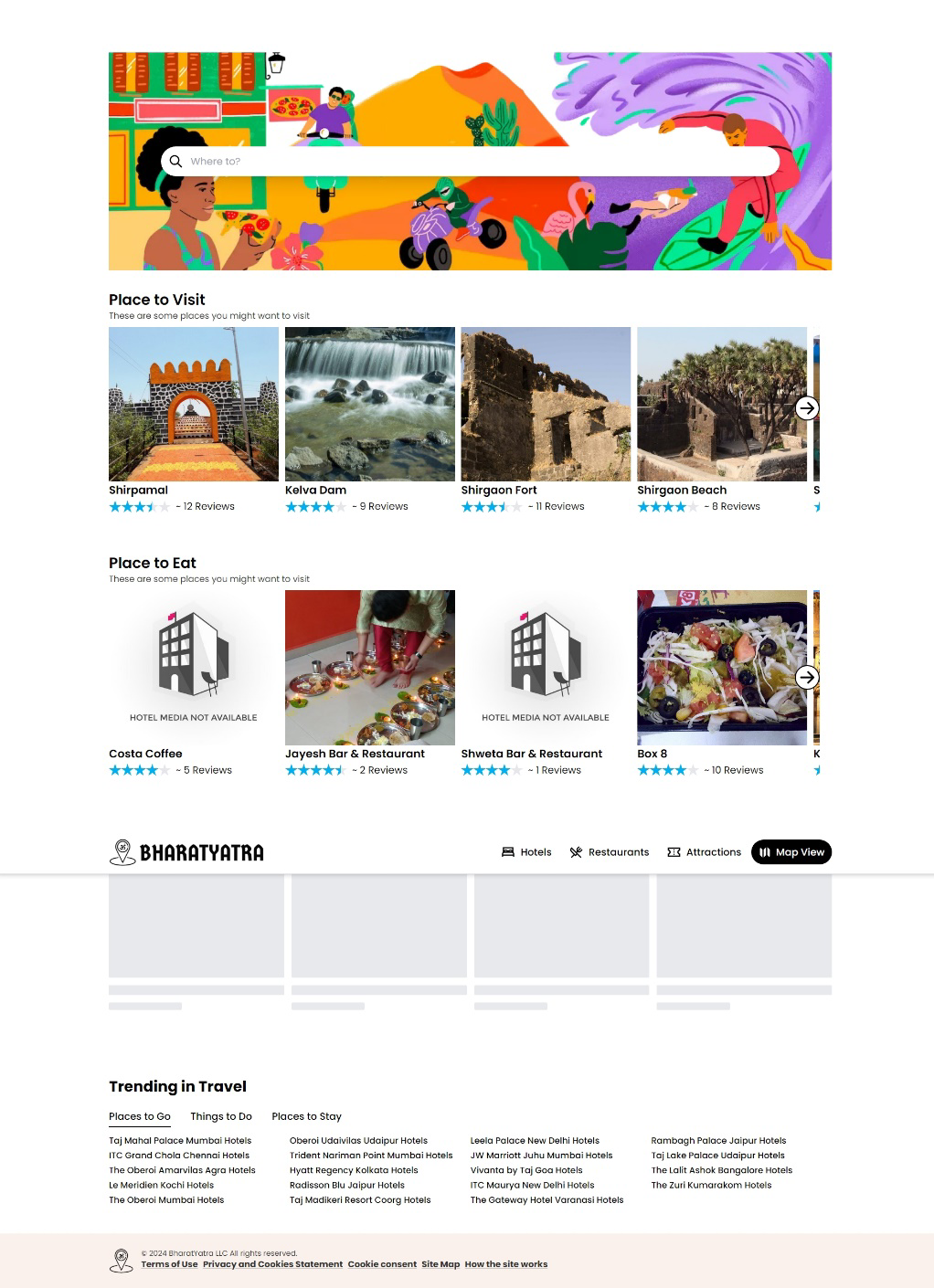
**Performance Optimization:** Balancing feature richness with fast load times, despite the inclusion of reviews, ratings, and local attractions.

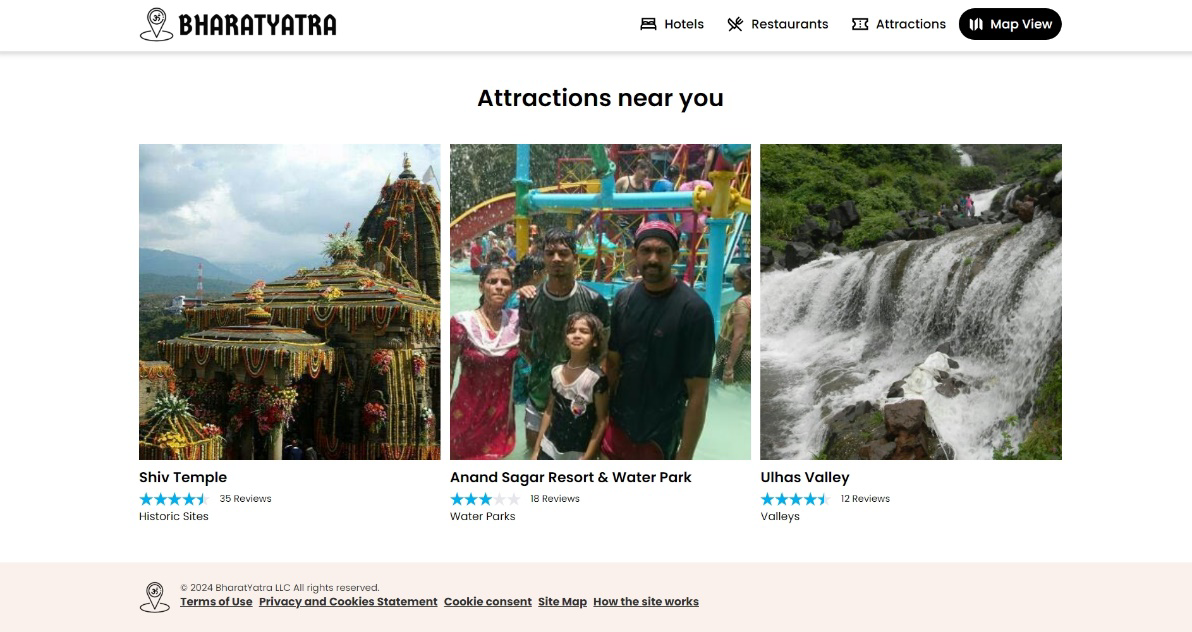
**User Interface Consistency**: Maintaining a uniform and intuitive design throughout the site, ensuring users can easily navigate between different sections like reviews, attractions, and itineraries.

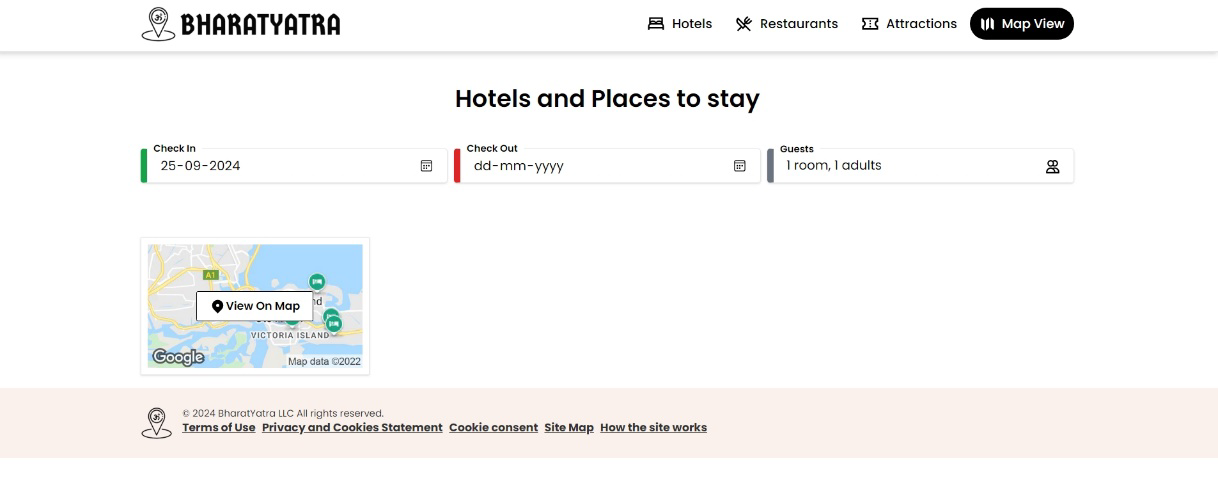
* System architecture diagram/System flow chart



o User interfaces







E] Implementation details

o Source code

import React, { useState, useEffect } from 'react';

import { getPlacesByBounds, getPlacesByLatLng, getTransportOptionsByLatLng} from "../api";

import axios from "axios";

export const MainContext = React.createContext();

export const MainContextProvider = ({ children }) => {

const [places, setPlaces] = useState();

const [filteredPlaces, setFilteredPlaces] = useState();

const [coordinates, setCoordinates] = useState({});

const [bounds, setBounds] = useState({});

const [rating, setRating] = useState(0);

const [type, setType] = useState('restaurants');

const [isLoading, setIsLoading] = useState(false);

const [restaurants, setRestaurants] = useState();

const [hotels, setHotels] = useState();

const [attractions, setAttractions] = useState();

const [transportOptions, setTransportOptions] = useState();

// Get Current User Location

useEffect(() => {

// Getting the current position corrdinates from browsers naviagtor sensor

navigator.geolocation.getCurrentPosition(({ coords: {latitude, longitude} }) => {

// setting coordinates latitude and longitude to the state

setCoordinates({lat: latitude, lng: longitude})

})

}, [])

// Get Places for Map View

useEffect(() => {

let source = axios.CancelToken.source();

// Setting loading state to true while data is being fetched

setIsLoading(true);

// If bounds state value of southwest - 'sw' and northeast 'ne' is available then the try-catch block is fired

if (bounds.sw && bounds.ne) {

try {

// Calling on the getPlacesByBounds endpoint passing in the type (hotels || attractions || restaurant), bounds and 'source' for error handling and effect cleanup

getPlacesByBounds(type, bounds.sw, bounds.ne, source)

.then(data => {

// Response 'data' is ready and set to the places state

setPlaces(data?.filter(place => place.name))

// Loading state set back to false - to stop loading, after data is fetched

setIsLoading(false);

})

console.log('All set! ', bounds.sw, bounds.ne);

} catch (error) {

console.error(error)

}

}

// Effect Cleanup

return () => {

source.cancel();

}

}, [type, bounds])

// Get Places for Homepage

useEffect(() => {

let source = axios.CancelToken.source();

// Setting loading state to true while data is being fetched

setIsLoading(true);

// if coordinates state value latitude 'lat' and longitude 'lng' is found, the try-catch block is fired

if (coordinates.lat && coordinates.lng) {

try {

// Calling on getPlacesByLatLng for 'restaurants' type, passing in parameter for 'limits' & 'min\_rating'; and 'source' for error handling and effect cleanup

getPlacesByLatLng('restaurants', coordinates.lat, coordinates.lng, { limit: 20, min\_rating: 4 }, source)

.then(data => {

// Response 'data' received and set to restaurants state filtering out data without 'name' property, 'location\_id' === 0

setRestaurants(data?.filter(restaurant => restaurant.name && restaurant.location\_id != 0))

});

// Calling on getPlacesByLatLng for 'attractions' type, passing in parameter for 'limits' & 'min\_rating'; and 'source' for error handling and effect cleanup

getPlacesByLatLng('attractions', coordinates.lat, coordinates.lng, { limit: 20, min\_rating: 4 }, source)

.then(data => {

// Response 'data' received and set to attractions state filtering out data without 'name' property, 'location\_id' === 0

setAttractions(data?.filter(attraction => attraction.name && attraction.location\_id != 0 && attraction.rating > 0))

});

// Calling on getPlacesByLatLng for 'restaurants' type, passing in parameter for 'limits' & 'min\_rating'; and 'source' for error handling and effect cleanup

getPlacesByLatLng('hotels', coordinates.lat, coordinates.lng, { limit: 20, min\_rating: 4 }, source)

.then(data => {

// Response 'data' received and set to hotels state filtering out data without 'name' property, 'location\_id' === 0

setHotels(data?.filter(hotel => hotel.name && hotel.location\_id != 0 && hotel.rating > 0))

});

} catch (error) {

console.error(error)

}

}

// Effect Cleanup

return () => {

source.cancel()

}

}, [coordinates]);

// Get Filtered Places by Rating

useEffect(() => {

// Places filter by rating for Map view

// Set new filteredPlaces on change of 'rating' state

// filter in only data with 'rating' proper greater than or equal to the selcted rating value

setFilteredPlaces(places?.filter(place => Number(place.rating) >= rating))

}, [rating])

return (

// Passing State value through main context to children for access

<MainContext.Provider value={{ places, setPlaces, coordinates, setCoordinates, bounds, setBounds, rating, setRating, type, setType, isLoading, setIsLoading, filteredPlaces, attractions, restaurants, hotels }}>

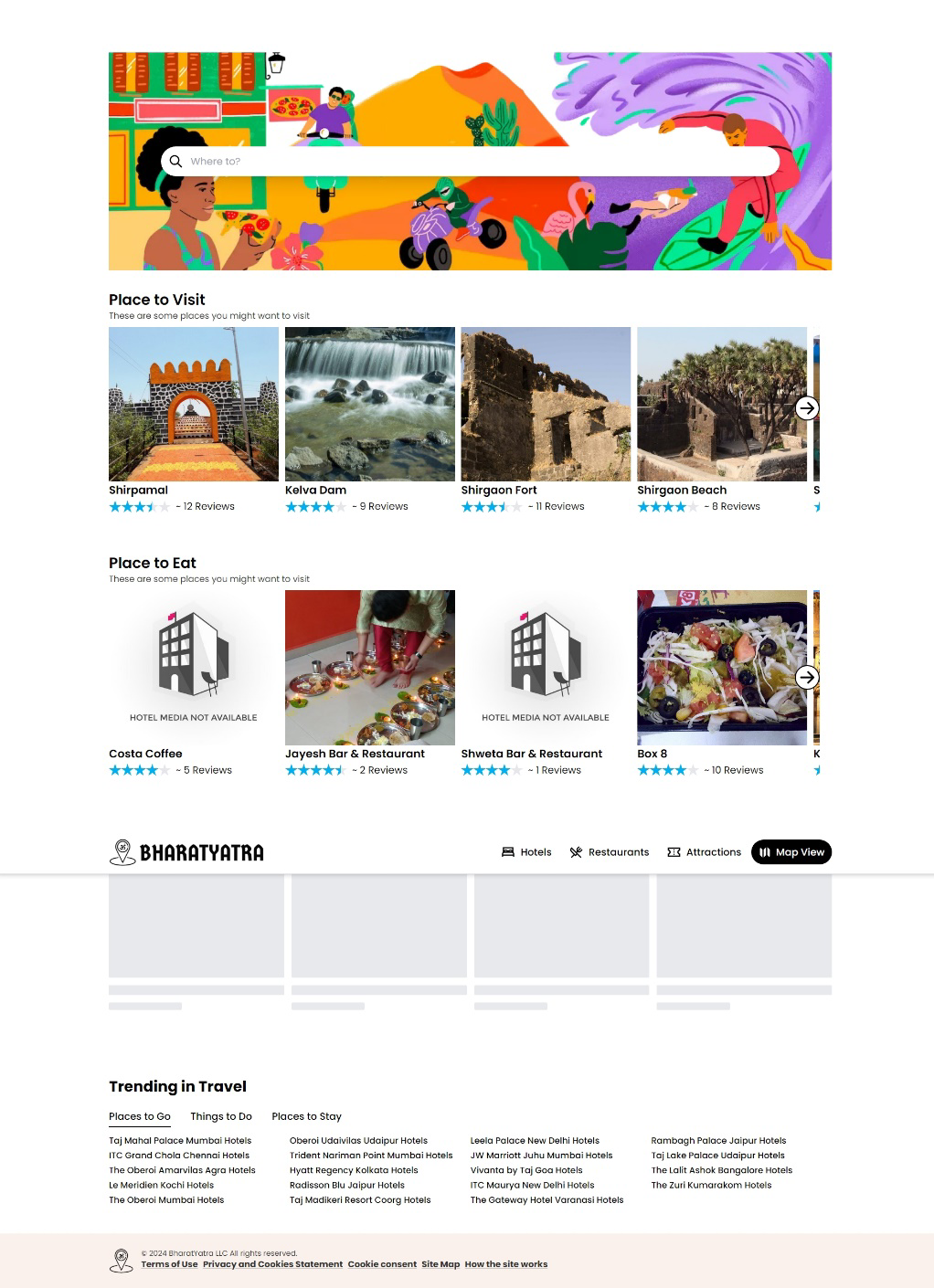
{ children }

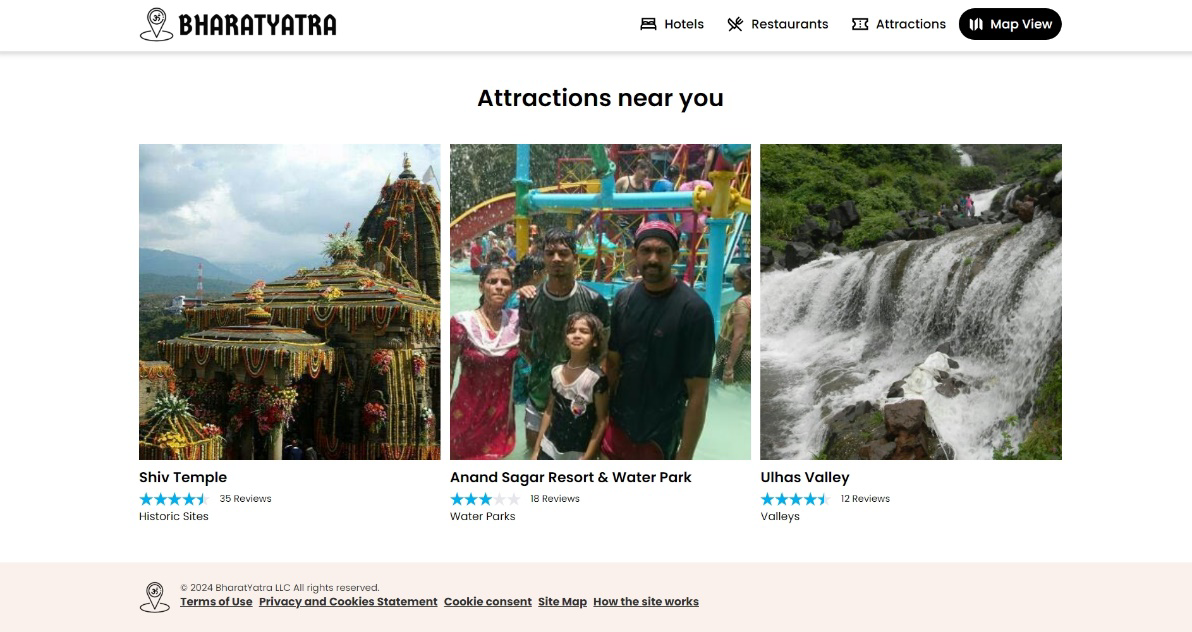
</MainContext.Provider>

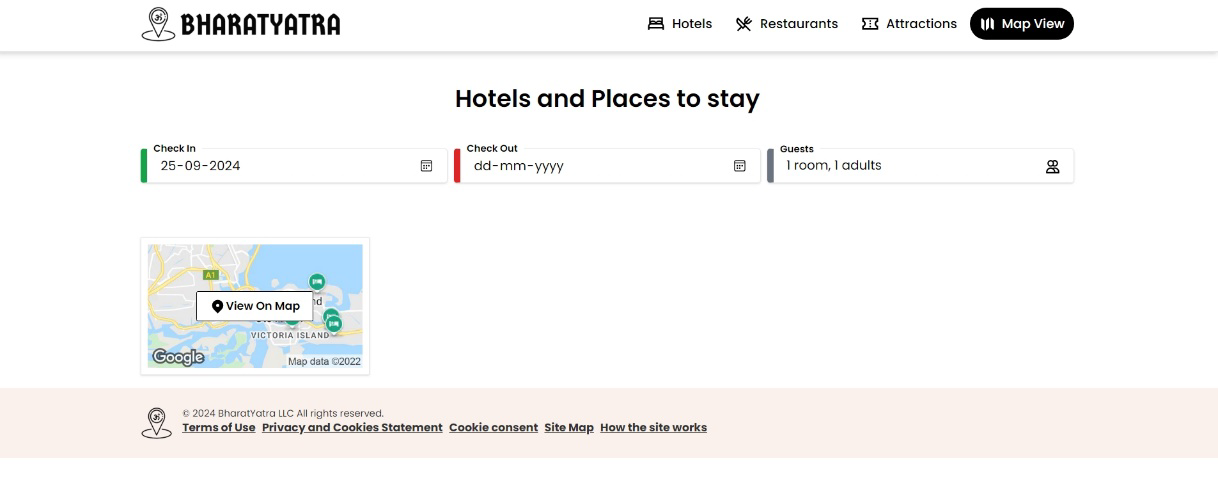
)

}

F] Outputs and Reports Testing







G] Conclusion and Recommendations

In conclusion, our project addresses a common challenge that tourists face—finding the best hotels and nearby attractions amidst an overwhelming amount of online information. Our website aims to simplify this process by providing an easy-to-use platform where tourists can quickly discover and compare hotels, local attractions, and things to do in their destination. By integrating features like reviews, ratings, and detailed itineraries, we make trip planning more straightforward and enjoyable. With a focus on user-friendly design and efficient information delivery, our website helps tourists make informed decisions, ultimately enhancing their travel experience.

H] Future Scope

The future scope of our travel website project offers several exciting opportunities for growth and improvement. One potential area is the integration of AI-driven recommendation systems, which can provide personalized suggestions for hotels, attractions, and activities based on user preferences and past behaviour. Finally, developing a mobile app version of the platform would cater to tourists who prefer on-the-go travel planning, making it even more convenient and accessible.

I] Bibliography and References (Follow APA format)

<https://github.com/babblebey/travelAdvisor>