TRAVEL APP(https://5130f2023-hazel.vercel.app/)

Team Members

KavyaSree_Soma@student.uml.edu	AkhilaSai_Kuncha@student.u ml.edu	Sahil_Munjuluri@student.uml.edu
Kavya Sree Soma	**Akhila Sai Kuncha**	**Sahil Munjuluri**

Overview: -
This is an open-source Travel website idea inspired by Airbnb idea which aims to completely change how people travel and experience the world. The platform strives to create remarkable journeys above and beyond the norm by providing the users nearest visiting places to their location in addition to the unique accommodations and authentic experiences.

Abstract:
The main goal of the project is to create a dynamic online platform based on MERN STACK technology that aims to transform travel experiences by offering a carefully chosen selection of unique accommodations and authentic activities. The goal is to provide users with personalized, memorable journeys that break away from the constraints of conventional travel platforms. The Travel App, envisioned as an open-source platform inspired by the idea of Airbnb, seeks to redefine the travel experience by addressing common limitations in conventional travel platforms. Focused on offering a curated selection of unique accommodations and genuine experiences, the project aims to diversify travel options, combat over tourism, and foster community engagement among like-minded travelers. Utilizing the MERN Stack technology, the app provides a seamless user experience with features such as personalized recommendations, local insights, and real-time booking.

Alternative Approaches Considered for Travel App Development

1.MEAN Stack:

Description: Similar to MERN, the MEAN stack involves MongoDB, Express.js, Angular, and Node.js. Choosing MEAN would mean using Angular for the frontend instead of React.
Considerations: While MEAN is a robust and popular stack, Angular might introduce a steeper learning curve compared to React. The choice between MERN and MEAN often depends on the development team's familiarity and preferences.

2. Traditional LAMP Stack:

Description: The LAMP stack includes Linux, Apache, MySQL, and PHP/Python/Perl. It's a classic choice for web development.
Considerations: LAMP has been a reliable stack for many years, but it might not provide the real-time capabilities and dynamic front-end experience offered by the MERN stack. It could be more suitable for static websites or less interactive applications.

3. Vue.js with Serverless Architecture:

Description: Combine Vue.js for the frontend with serverless architecture (e.g., AWS Lambda, Azure Functions) for the backend. Vue.js is known for its simplicity, and serverless architecture can offer scalability and cost-effectiveness.

Considerations: Vue.js is a progressive JavaScript framework known for its simplicity and ease of integration. While MEAN is a robust and popular stack, Vue.js differs in several aspects that should be carefully considered:

CHOOSEN METHODOLOGY AND JUSTIFICATION
Chosen Approach - MERN Stack:
Description:

Opting for the MERN (MongoDB, Express.js, React, Node.js) stack implies utilizing a set of technologies that seamlessly work together to build the Travel App.

Considerations:

Familiarity and Productivity:

The team's existing familiarity with MERN components, like MongoDB for the database, Express.js for the backend, React for the frontend, and Node.js as the runtime, is akin to having a well-practiced toolkit. This familiarity boosts productivity, as team members can leverage their existing skills to create a robust and efficient travel platform.

Dynamic and Interactive Frontend:

Choosing React for the frontend is like opting for LEGO blocks that offer dynamic and interactive building possibilities. This is crucial for creating an engaging user interface, allowing users to explore unique accommodations and experiences seamlessly.

Adaptability to Future Technologies:

MERN's modular nature, akin to a set of building blocks, facilitates easy adaptation to emerging technologies. This adaptability ensures that the Travel App can integrate new features or updates without undergoing a complete overhaul, keeping it technologically relevant in the future.

Progressive Web App (PWA):

Description: Opting for a Progressive Web App (PWA) approach involves leveraging web technologies to deliver an app-like experience. PWAs combine the best of web and mobile applications, providing users with a seamless and responsive experience across various devices.

Considerations:

Cross-Platform Compatibility:

A PWA is accessible through web browsers, making it compatible with various devices and platforms. This approach ensures a consistent user experience, whether users access the Travel App from a desktop, tablet, or mobile device, irrespective of the operating system.

Offline Accessibility:

PWAs are designed to work seamlessly offline or in areas with a weak or intermittent internet connection. This is especially relevant for a Travel App, where users may find themselves in locations with limited connectivity. Offline accessibility enhances user engagement and satisfaction.

Engaging User Experience:

The app-like experience offered by PWAs, including features like push notifications and smooth transitions, enhances user engagement. Users can add the Travel App to their home screens without the need for app store downloads, providing a frictionless and immersive experience.

Front-end part:
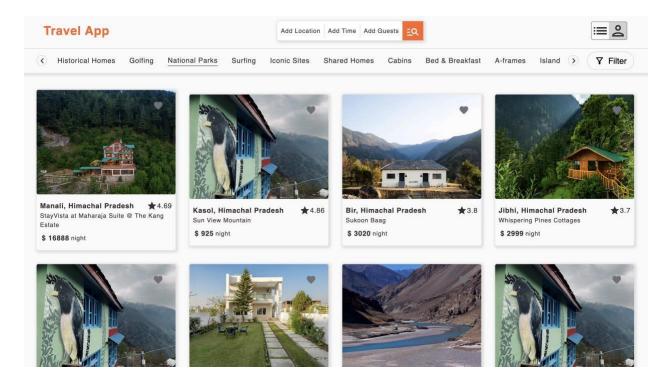**Instructions on How to Use the Application:**

Once you open the app: https://5130f2023-hazel.vercel.app/

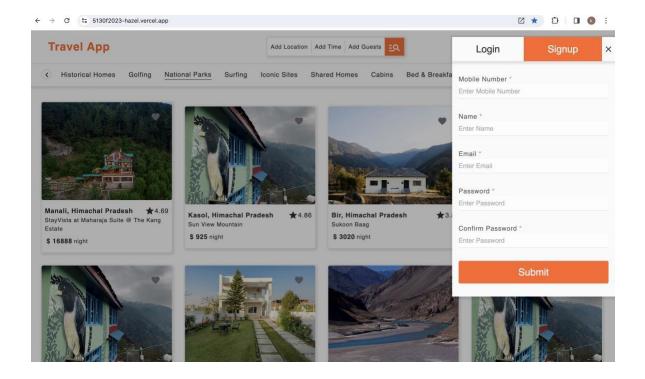We have built a progressive web app – we can even download the App.

We can select from the list of Categories i.e., National Park, Tiny Homes (Data has been added only for these two Categories)

For Suppose if we have selected the Category as National Park, we will be able to see all the places whose category is National Park
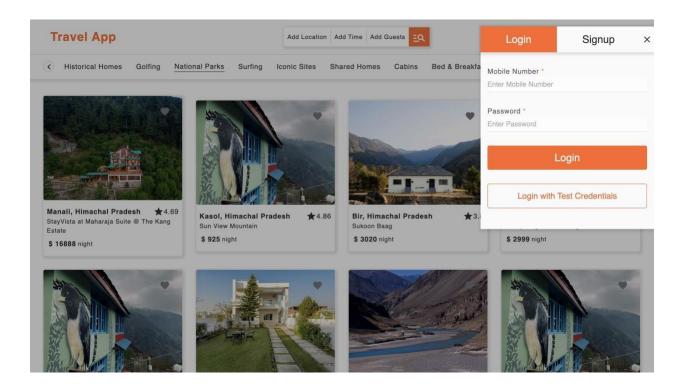

Once we Click on the menu, we will be able to see the Login and Signup Page.
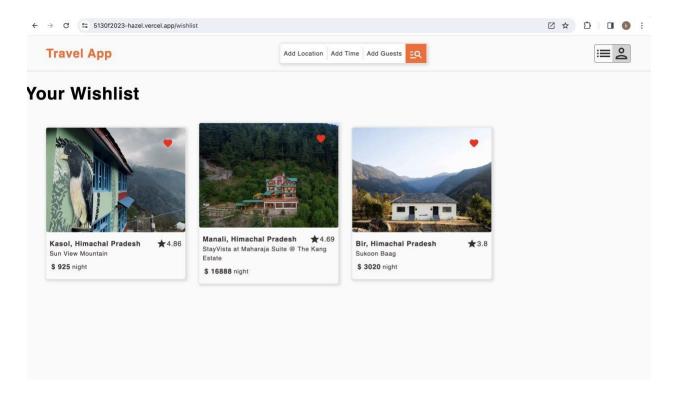One need to Signup with Mobile Number, Name, Email, Password (should contain capital Letters and length need to be greater that 8 characters)
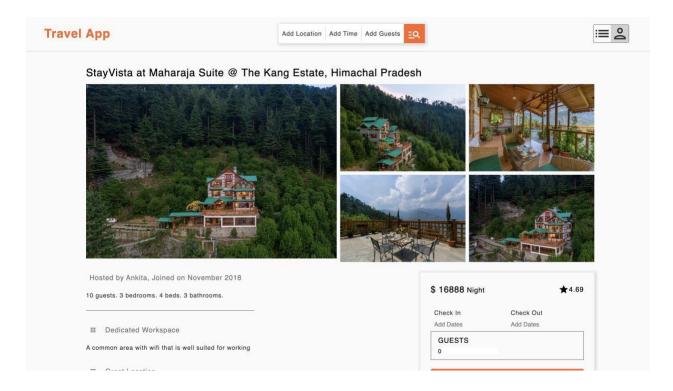
If we can Sign Up. Then using those credentials to Login. Or else we can you Login with Test Credentials button to sign up without any need for Sign Up.
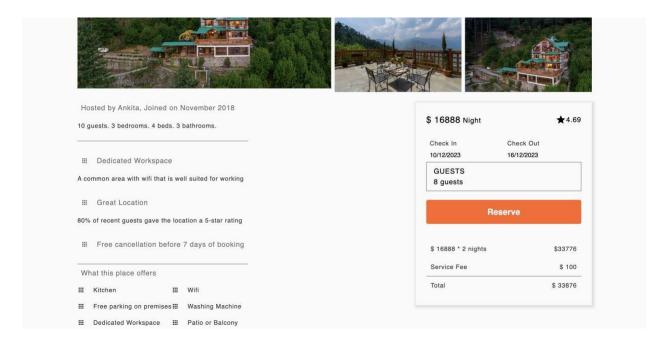
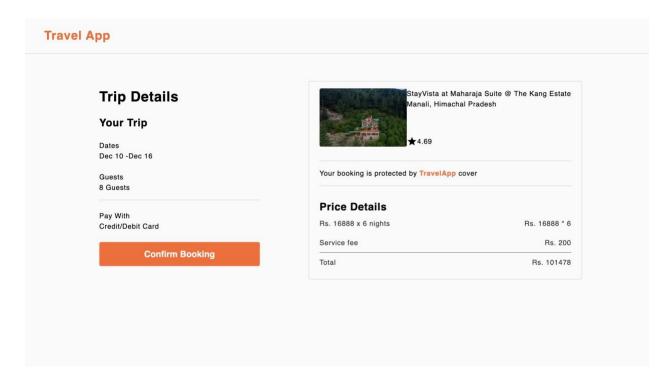Once we Logged in, we can add any hotels to the WishList,

Single Hotel Page:

Once we add CheckIn, Checkout dates and Number of guests we can hit on the Reserve button

Once we hit on the Reserve Button, we will be navigating to the Payment page where we can proceed with Booking.

Back-end Part:

API's Created:

https://real-goat-stole.cyclic.app/api/hotels/ - All the Hotels

https://real-goat-stole.cyclic.app/api/hotels/6546b3b21eed0aa6d33ed241 - Single Hotel Page

https://real-goat-stole.cyclic.app/api/category - Categories Page

APIs are used in React program to trigger the data.

Used Mongo dB to store the user details once he/she has logged In.