

JourneyNest

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:

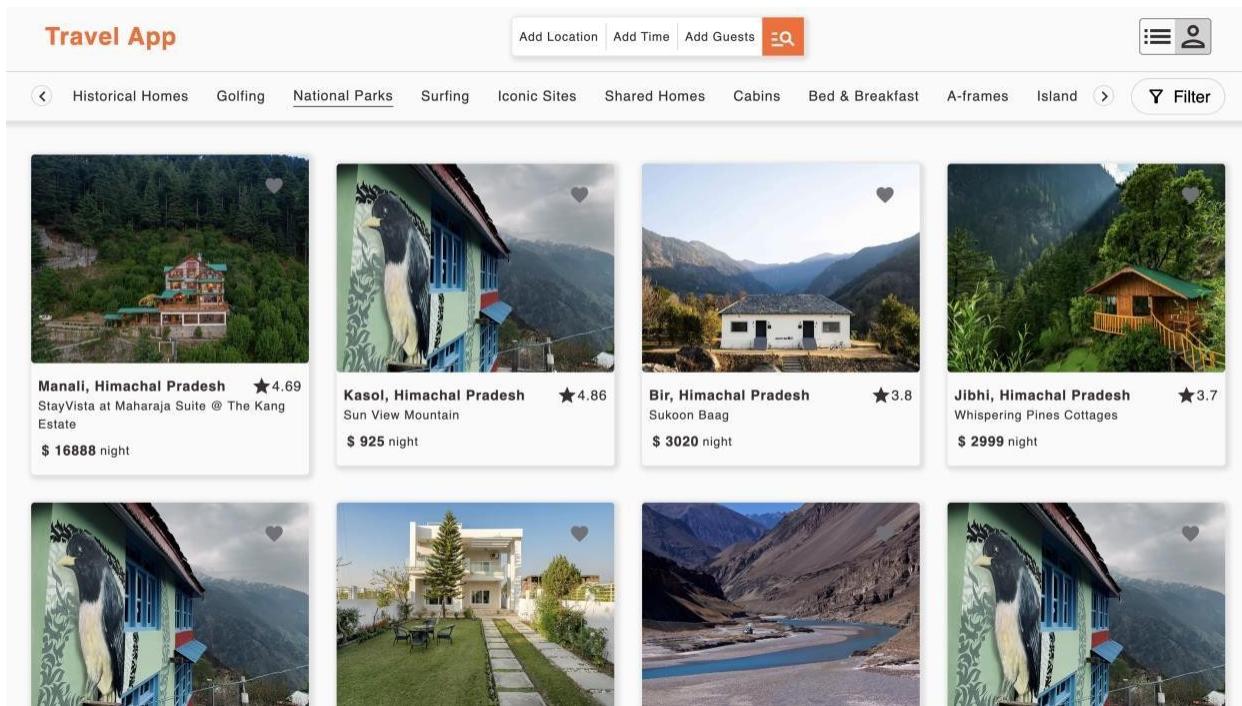
The main goal of the project is to create a dynamic online platform based on MERN STACK technology that aims to transform travel experiences by offering a carefully chosen selection of unique accommodations and authentic activities. The goal is to provide users with personalized, memorable journeys that 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.

User Interface Walkthrough:

Below is the screenshot of the home page of our Travel App. The home page serves as the entry point for users, showcasing the main features, categories, and navigation options. It is designed to provide a user-friendly interface, offering a seamless browsing experience for exploring accommodations and travel categories.

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.

If user don't have an account, User need to Sign up with Mobile Number, Name, Email, Password (should contain capital Letters and length need to be greater than 8 characters).

A modal window is open on the right side of the screen, containing fields for login and signup:

- Login**
- Signup**
- Mobile Number ***
Enter Mobile Number
- Name ***
Enter Name
- Email ***
Enter Email
- Password ***
Enter Password
- Confirm Password ***
Enter Password

At the bottom right of the modal is a large orange **Submit** button.

Once the sign up is done, it redirects to login with the valid credentials.

The screenshot shows the homepage of a travel application. At the top, there is a navigation bar with links for "Historical Homes", "Golfing", "National Parks", "Surfing", "Iconic Sites", "Shared Homes", "Cabins", "Bed & Breakfast", "Add Location", "Add Time", "Add Guests", and a search icon. On the right side of the header are "Login" and "Signup" buttons, along with a close button (X). Below the header, there are three hotel cards:

- Manali, Himachal Pradesh** ★4.69
StayVista at Maharaja Suite @ The Kang Estate
\$ 16888 night
- Kasol, Himachal Pradesh** ★4.86
Sun View Mountain
\$ 925 night
- Bir, Himachal Pradesh** ★3.8
Sukoon Baag
\$ 3020 night

A modal window is open on the right side, prompting the user to enter their mobile number and password. It also contains a "Login" button and a "Login with Test Credentials" button.

Once we logged in, we can also add any hotels to the wishlist.

The screenshot shows the "Your Wishlist" page. At the top, there is a header with the "Travel App" logo, "Add Location", "Add Time", "Add Guests", and a search icon. To the right of the header is a user profile icon. Below the header, the title "Your Wishlist" is displayed. Three hotel cards are listed, each with a red heart icon indicating it is added to the wishlist:

- Kasol, Himachal Pradesh** ★4.86
Sun View Mountain
\$ 925 night
- Manali, Himachal Pradesh** ★4.69
StayVista at Maharaja Suite @ The Kang Estate
\$ 16888 night
- Bir, Himachal Pradesh** ★3.8
Sukoon Baag
\$ 3020 night

Once, we select the hotel, we can view the description and give required details and can reserve the hotel.

The screenshot shows a hotel listing on the Travel App. At the top, there are buttons for "Add Location", "Add Time", "Add Guests", and a search icon. On the right, there are icons for a menu and user profile. The main title is "StayVista at Maharaja Suite @ The Kang Estate, Himachal Pradesh". Below the title are five images: a large exterior view of the two-story wooden building, a closer view of the building, an interior view of a balcony with a sofa and a view of the mountains, a view of the mountains under a cloudy sky, and another exterior view of the building from a different angle. Below the images, it says "Hosted by Ankita, Joined on November 2018", "10 guests. 3 bedrooms. 4 beds. 3 bathrooms.", and "Dedicated Workspace". A common area with wifi that is well suited for working is also mentioned. To the right, the price is listed as "\$ 16888 Night" with a rating of "★ 4.69". There are fields for "Check In" and "Check Out" with "Add Dates" buttons, and a "GUESTS" field set to "0".

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

The screenshot shows the "Trip Details" and "Price Details" sections of the Travel App. The "Trip Details" section includes "Your Trip" (Dates: Dec 10 - Dec 16, Guests: 8 Guests), "Pay With" (Credit/Debit Card), and a large orange "Confirm Booking" button. The "Price Details" section shows the total cost breakdown: Rs. 16888 * 6 nights (Rs. 101478), Service fee (Rs. 200), and Total (Rs. 101478). To the right, there is a thumbnail image of the hotel and the text "StayVista at Maharaja Suite @ The Kang Estate Manali, Himachal Pradesh" with a rating of "★ 4.69". It also states "Your booking is protected by TravelApp cover".

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

The screenshot shows the MongoDB Atlas interface for the TravelAppDB database. The left sidebar lists various services and databases, with 'TravelAppDB' selected. Under 'TravelAppDB', the 'users' collection is highlighted. The main panel shows two documents from the 'users' collection:

```
_id: ObjectId('657567a0b2ba41011bbe99d2')
username: "sambasivarao"
number: 83414811300
email: "sivarao@gmail.com"
password: "U2FsdGVkX1+iNVNbKytPz2+BggVASTuR7JUAb3YL/gU="
createdAt: 2023-12-10T07:24:16.150+00:00
updatedAt: 2023-12-10T07:24:16.150+00:00
__v: 0

_id: ObjectId('6575e04588fabd10ab52255b')
username: "rectx"
number: 9876543220
email: "tr@gmail.com"
password: "U2FsdGVkX1/sH2C8dVEQ9m4qNhw3oYDvHa3rJM4s(Y="
createdAt: 2023-12-10T15:59:01.059+00:00
updatedAt: 2023-12-10T15:59:01.059+00:00
__v: 0
```

The interface includes a search bar, storage statistics (STORAGE SIZE: 36KB, LOGICAL DATA SIZE: 1.05KB, TOTAL DOCUMENTS: 6, INDEXES TOTAL SIZE: 36KB), and various navigation and filtering tools.

Issues Faced:

- Time Management:** Balancing frontend development, backend integration, and testing within the project timeline was challenging.
- Technical Hurdles:**
 - Implementing the MERN stack required resolving integration issues between React, Node.js, and MongoDB.
 - Debugging API errors, such as delays in fetching data for categories and hotels, required extensive effort.
- Testing Limitations:** End-to-end testing for features like reservation workflows and category navigation was time-intensive, leading to minor bugs in some functionalities.

Lessons Learned:

- Technology Mastery:** Improved understanding of the MERN stack, including efficient data fetching, state management in React, and optimizing MongoDB queries.
- Problem-Solving:** Debugging issues with API performance and frontend-backend communication enhanced troubleshooting skills.
- Importance of User Experience:** Designing intuitive workflows and validating user input helped enhance the usability.

Remaining Bugs:

- **Wishlist Synchronization:** Adding hotels to the wishlist occasionally lags, requiring additional synchronization logic.
- **Error Handling:** Some edge cases, such as invalid API inputs or server errors, result in generic error messages instead of detailed feedback.
- **Category Filtering:** When switching between categories, previously selected filters sometimes persist, causing inconsistent results on the screen.