

# **THE TRAVEL JOURNAL APP**

A Project Report

*Submitted by*

Sakthi Priya G (920422205089)

Hemavarshini K (9204222038)

Santhiya K (920422205092)

*In partial fulfilment for the award of the degree of*

**BACHELOR TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution – Affiliated to Anna University, Chennai)**

**K. VELLAKULAM, VIRUDHUNAGAR – 625 701**

**OCTOBER 2024**

**KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution – Affiliated to Anna University, Chennai)**  
**K. VELLAKULAM, VIRUDHUNAGAR – 625 701**

**BONAFIDE CERTIFICATE**

Certified that the project report “The Travel Journal App” is the bonafide work of Sakthi priya G (920422205089), Hemavarshini K (920422205038) and Santhiya K (920422205092) who carried ou the project work under my supervision.

Dr. E. Vakaimalar,  
HEAD OF THE DEPARTMENT,  
Associate Professor,  
Department of Information  
Technology,  
Kamaraj College of Engineering  
And technology  
K. Vellakulam, Virudhunagar.

Dr. R. Arthy  
SUPERVISOR,  
Assistant Professor,  
Department of Information  
Technology.  
Kamaraj College of Engineerning  
and Technology  
K. Vellakulam, Virudhunagar.

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **ABSTRACT**

The Travel Journal App is a user-friendly platform designed for travelers to document, edit, and share their experiences. It offers features for writing new journal entries, reviewing past ones, and making modifications. The app allows users to create detailed travel logs with titles, dates, descriptions, and photos, while the Read Journal serves as a personal library for revisiting memories. The Edit Journal feature ensures flexibility for refining past entries, and the Let's Travel page helps plan future trips. With its integrated journaling and travel planning tools, the app enhances the experience of capturing and organizing travel adventures.

## **ACKNOWLEDGEMENT**

I would like to thank to Dr. E. Vakaimalar, Head of the Department of Information Technology, for their encouragement and valuable insights throughout the development of the “The Travel Journal” project. Special thank to Dr. R. Arthy, my supervisor, whose guidance and expertise were instrumental in shaping the direction of this project. Their support helped me gain in-depth the knowledge of the MERN stack and its application. This project has been an invaluable learning experience, enhancing both my technical and problem-solving skills.

## LIST OF FIGURES

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
2.3.1	Flow diagram	4
3.1.1	Login Page	7
3.1.2	Travel Journal Page	7
3.1.3	Read Journal	8
3.1.4	Write Journal	8
3.1.5	Modify Journal	9
3.1.6	Edit the Journal	9
3.1.7	Delete the Entry	10
3.1.8	Let's Travel Page	10
3.1.9	Let's Travel	11
3.1.10	Booking Page	11
3.1.11	Booking saved image	12
3.1.12	MongoDB	12

## Table of Content

Chapter No.	TITLE	Page No.
	Abstract	(iii)
	List of figures	(iv)
<b>1.</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 HTML	1
	1.2 CSS	1
	1.3 Java Script	1
	1.4 MERN Stack	1
<b>2.</b>	<b>METHODOLOGY</b>	<b>4</b>
	2.1 Objective	4
	2.2 Problem Statement	4
	2.3 Block Diagram	4
	2.4 Module Explanation	5
<b>3.</b>	<b>Results and Discussion</b>	<b>7</b>
	3.1 Screenshots	7
<b>4.</b>	<b>Conclusion</b>	<b>14</b>
<b>5.</b>	<b>References</b>	<b>15</b>

## 1. INTRODUCTION

The **Travel Journal App** is built using modern web technologies to provide a seamless, interactive experience for users. The foundation of the app relies on three core front-end technologies: **HTML**, **CSS**, and **JavaScript**, along with the powerful **MERN stack** for full-stack development.

### 1.1 HTML (HyperText Markup Language):

HTML forms the backbone of the app's structure. It defines the content and layout of the pages, including elements such as headings, paragraphs, forms, and buttons. In the Travel Journal App, HTML is used to organize and display the main sections like the journal entries, submission forms, and travel planning features.

### 1.2 CSS (Cascading Style Sheets):

CSS is used to style the visual presentation of the app, making it attractive and user-friendly. Through CSS, the Travel Journal App's design is responsive and visually appealing across different devices. It defines colors, fonts, spacing, and the overall layout to create a cohesive look and feel for the app.

### 1.3 JavaScript:

JavaScript brings interactivity and functionality to the app. It enables dynamic behaviors such as form validation, journal entry edits, and real-time updates without reloading the page. In the Travel Journal App, JavaScript is crucial for handling user input, managing journal entries, and integrating with other parts of the app.

**1.4 MERN Stack    MongoDB:** A highly flexible, NoSQL database used to store journal entries and user data. It enables the Travel Journal App to scale efficiently, supporting large amounts of data without requiring rigid structures, which is ideal for managing diverse travel logs and user details.

**Express.js:** A minimal, fast, and robust web application framework built on Node.js. It manages the app's back-end logic, handling server-side routing and interactions with the database. Express simplifies connecting the front-end React components with the database, enabling smooth data flow.

**React.js:** A powerful front-end JavaScript library that allows for the creation of dynamic user interfaces. In the Travel Journal App, React's component-based architecture ensures that features like writing, editing, and reading journals are rendered efficiently and updated in real time without refreshing the entire page.

**Node.js:** A versatile JavaScript runtime environment that runs on the server side. It processes user requests, manages routes, and interacts with the MongoDB database, allowing for a smooth, non-blocking performance. Node.js ensures that the app handles multiple user interactions and back-end processes effectively, making it highly scalable.

### **How the MERN stack works?**

The **MERN stack**—which includes MongoDB, Express.js, React.js, and Node.js—powers the Travel Journal App, letting users create, edit, and share their travel stories. Here's how it all works together:

**React.js (Front-End):** Users interact with the app through React, writing, editing, and reviewing journal entries. Each part of the interface is a reusable component that updates dynamically without refreshing the page. It sends data to the server using HTTP requests.

**Express.js (Back-End):** Express handles incoming requests from React, routes them to the right endpoints (like saving or fetching journals), and ensures data is validated before heading to the database.

**Node.js (Back-End Runtime):** Running on Node.js, the server processes the requests from React and interacts with the database. It ensures multiple users can work smoothly at the same time.

**MongoDB (Database):** MongoDB stores all journal entries, user data, and travel info as documents, making it easy to manage, scale, and retrieve the data through CRUD operations.

### **How It All Comes Together:**

**React sends data to the server:** When a user creates or edits a journal entry, React gathers the input (like title, date, and description) and sends it to the back-end using HTTP requests via libraries like Axios or Fetch.

**Express processes the request:** Express.js on the server handles the incoming request, processes the data, and ensures it's valid. It then routes the request to the appropriate endpoint for further actions like saving the data.

**MongoDB stores the journal entry:** Express passes the journal data to MongoDB, where it's saved as a document in a collection. MongoDB's flexible schema stores journal entries and any additional information.

**Server responds, React updates the UI:** Once the journal entry is saved, the server sends a confirmation back to React. React then updates the UI, showing the new or edited entry without refreshing the entire page.



## 2. METHODOLOGY

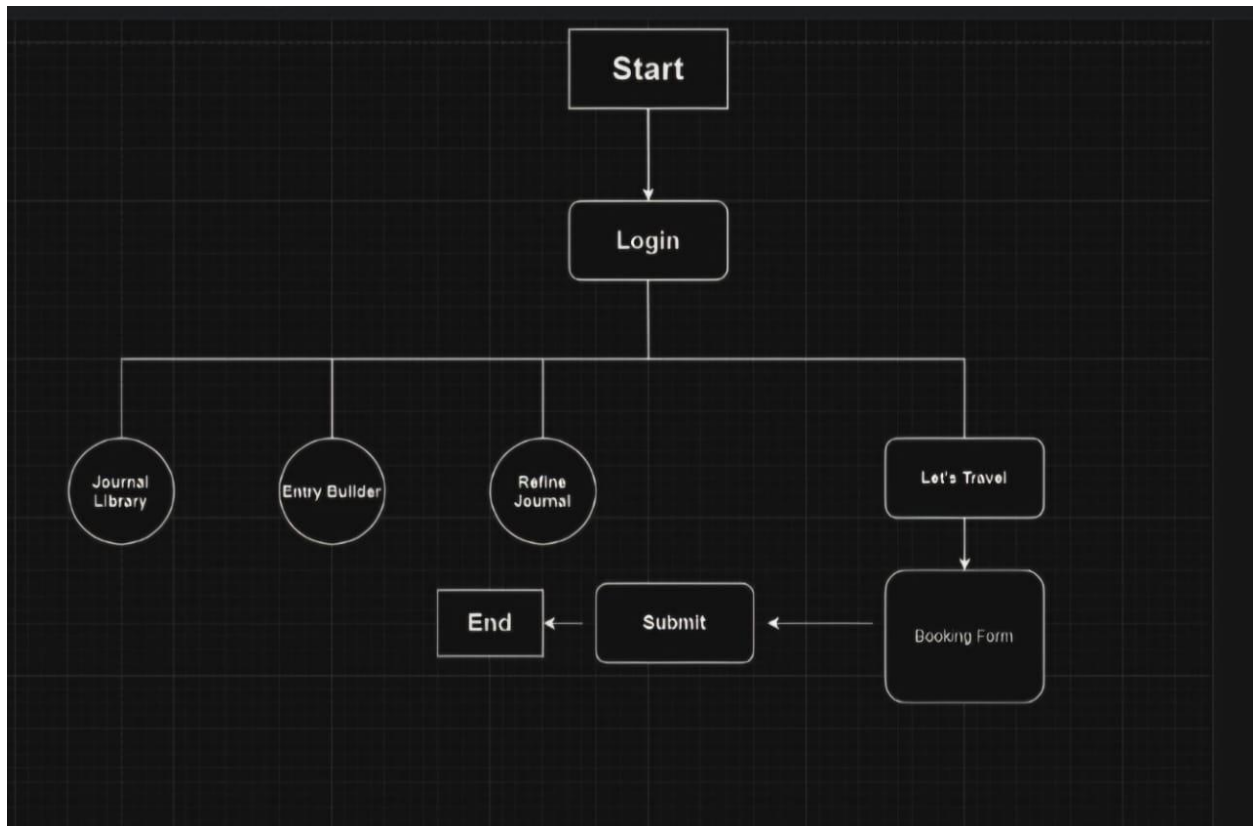
### 2.1 OBJECTIVE

The objective of the **Travel Journal App** is to provide travelers with a seamless platform to document, edit, and organize their travel experiences while integrating real-time trip planning, creating a personalized and interactive journaling experience.

### 2.2 PROBLEM STATEMENT

The problem addressed by the **Travel Journal App** is the lack of an organized, user-friendly platform for travelers to easily document, edit, and access their travel experiences while integrating essential trip planning tools, leading to fragmented and unstructured memories of their journeys.

### 2.3 BLOCK DIAGRAM



## 2.4 MODULE EXPLANATION

### User Interface (React Front-End):

- **Read Journal:** This component displays previously created journal entries in a well-organized format. Users can browse through their past trips, sorted by date or other relevant filters. Each journal entry shows key details like title, description, date, and location.
- **Edit Journal:** This module allows users to modify their existing journal entries. Users can update details such as the trip description, date, or attached photos. The edits are dynamically saved and reflected in the app without page reloads.
- **Write Journal:** Provides users with the functionality to create new journal entries. It includes input fields for the title, date, trip details, and a photo upload option. Once submitted, the journal entry is saved and displayed in the "Read Journal" section.

### Back-End (Booking Form):

- **Booking Form:** This module handles travel booking details. It allows users to input their name, source, destination, travel dates, and other booking-related details. Once the form is submitted, the data is processed by the back-end server (Node.js/Express), and the information is saved in the MongoDB database for future access.

### Back-End Logic (Express.js):

- Manages API routes and processes requests from the front-end. It validates incoming data, manages responses, and handles communication between React and MongoDB.

### Server-Side Logic (Node.js):

- Executes back-end operations, such as handling requests and managing database interactions. It ensures non-blocking, asynchronous processing of multiple requests.

**Database (MongoDB):**

- Stores booking informations. It offers flexibility in data storage, accommodating text, dates, and images.

### 3. RESULTS AND DISCUSSION

#### 3.1 SCREENSHOTS

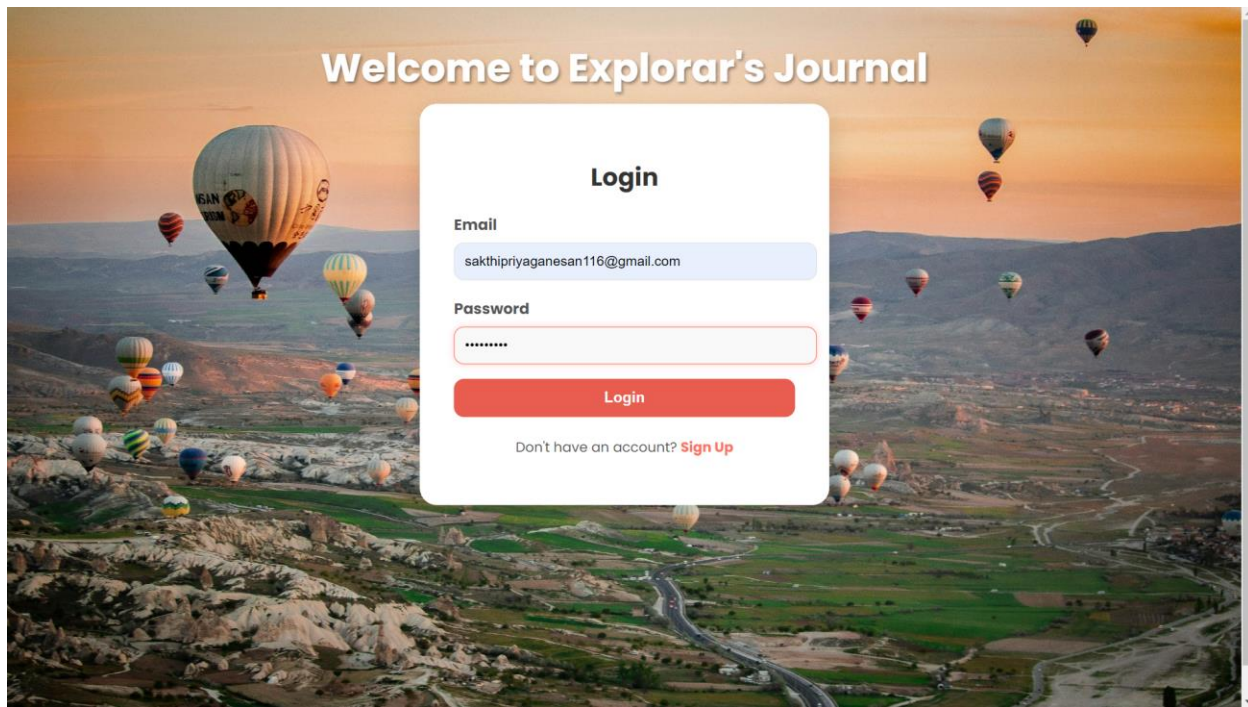


figure 3.1.1 Login Page

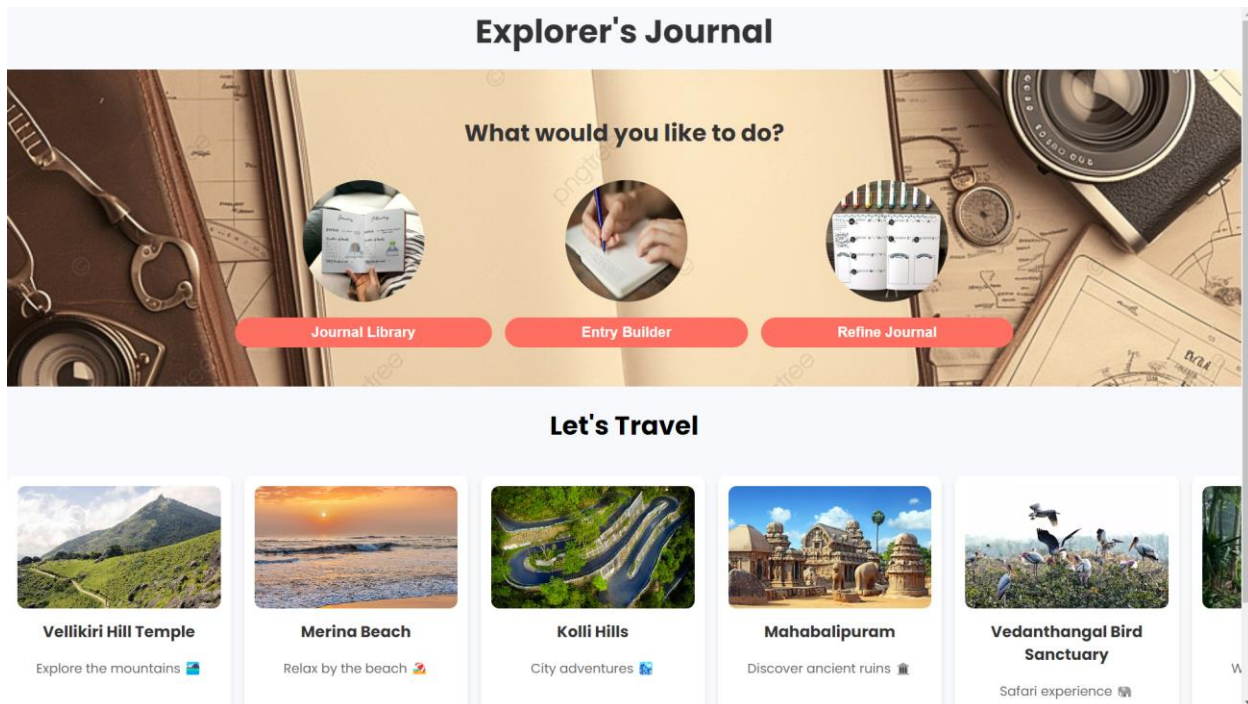


Figure 3.1.2 Travel Journal Page (After login, Travel Journal page will be open. We can modify, write and read the journal.)



Figure 3.1.3 ReadJournal.(We can view the journal)

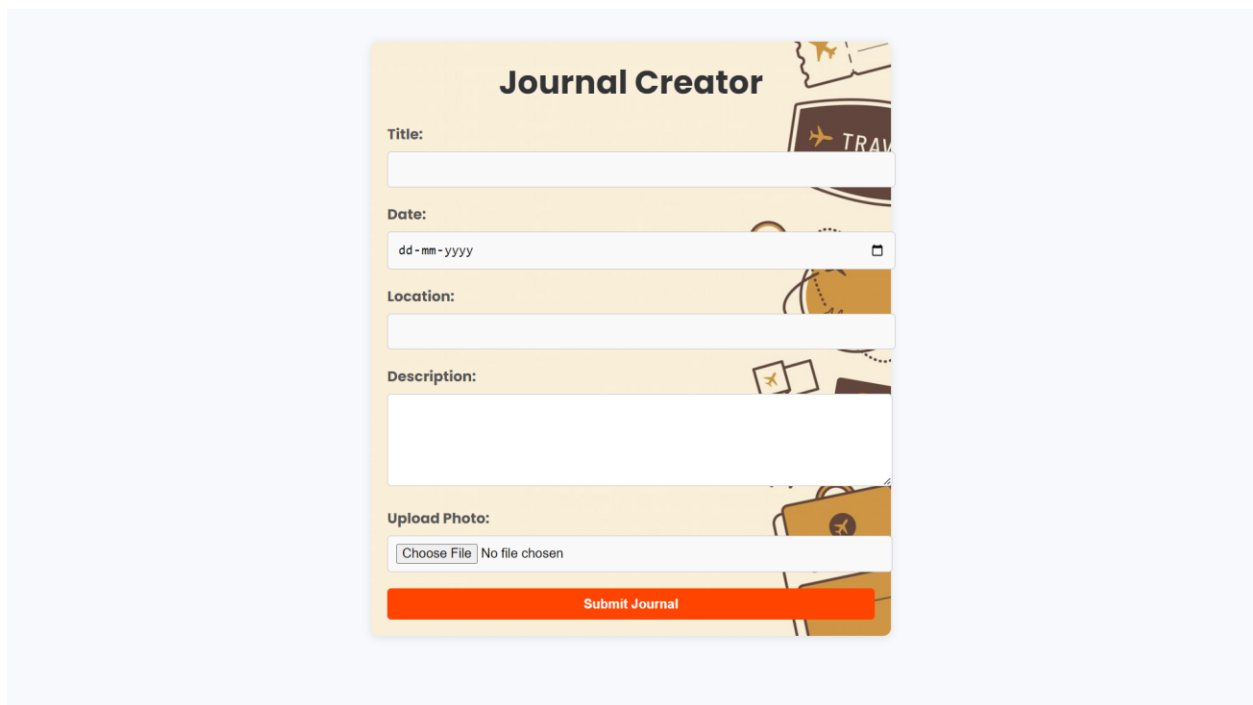


Figure 3.1.4 Write Journal (We can Entry the new journals.)

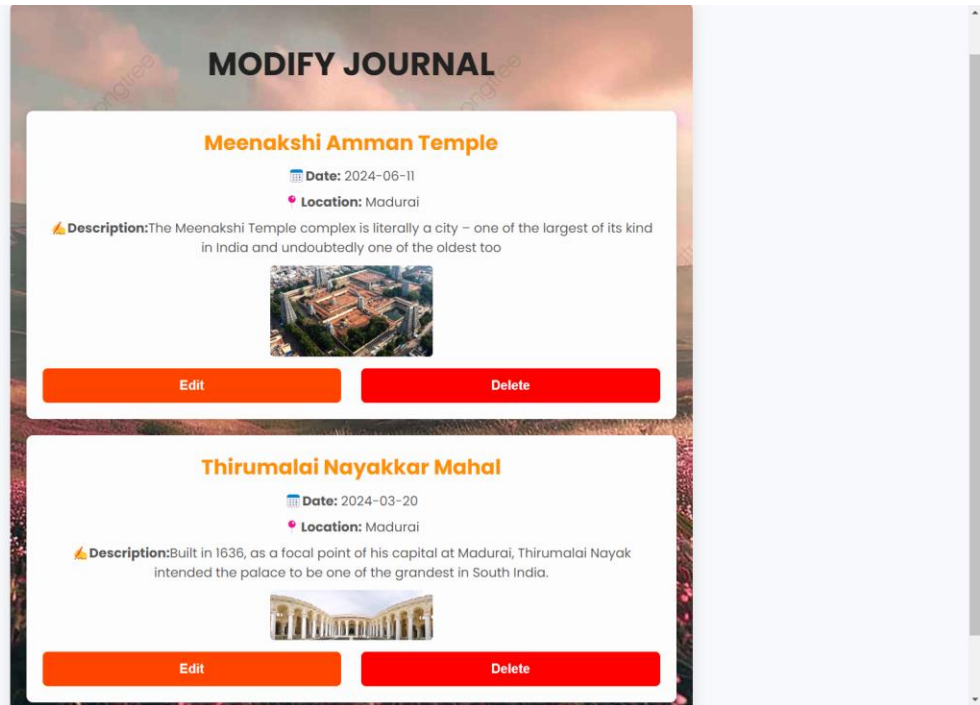


figure 3.1.5 Modify journal

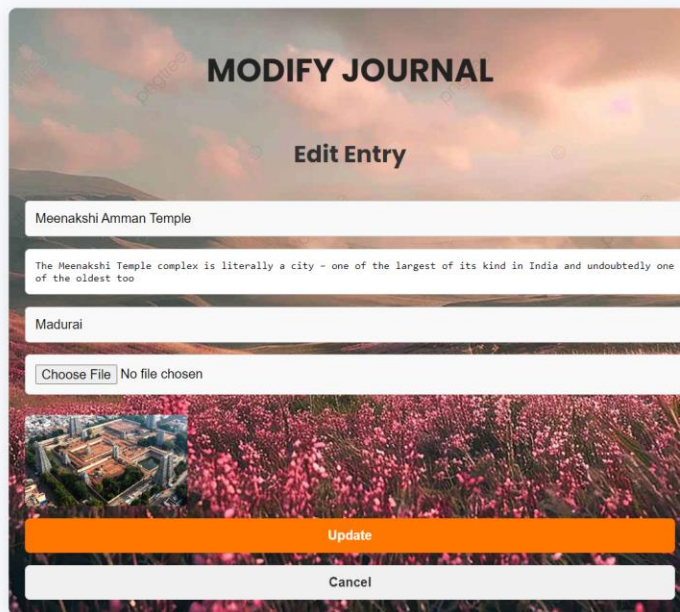


figure 3.1.6 Edit the Entry



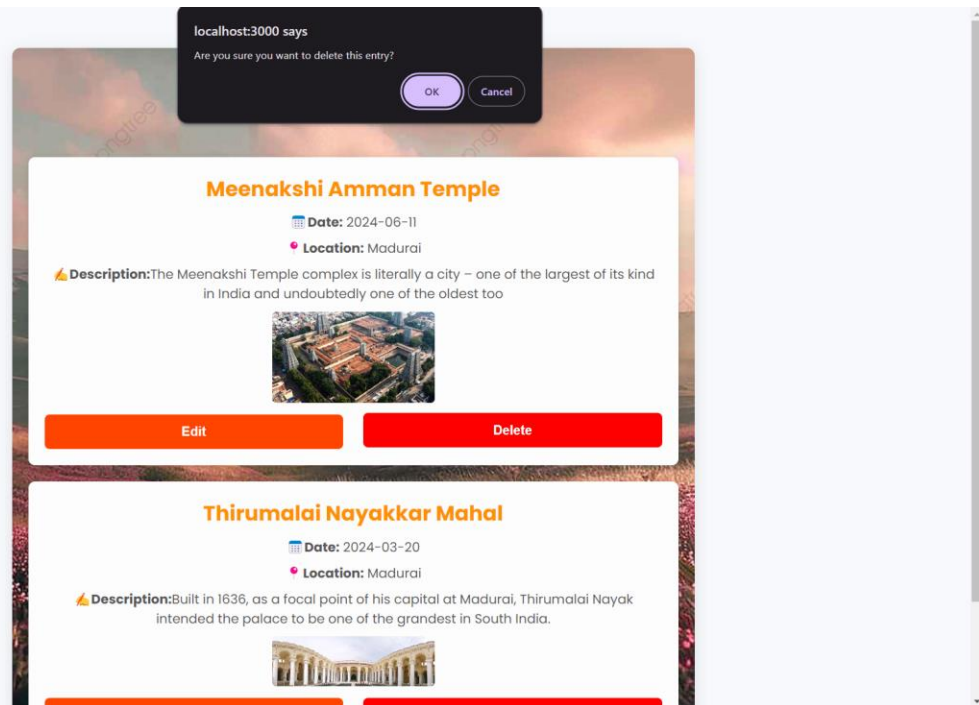


figure 3.1.7 Detele the journal

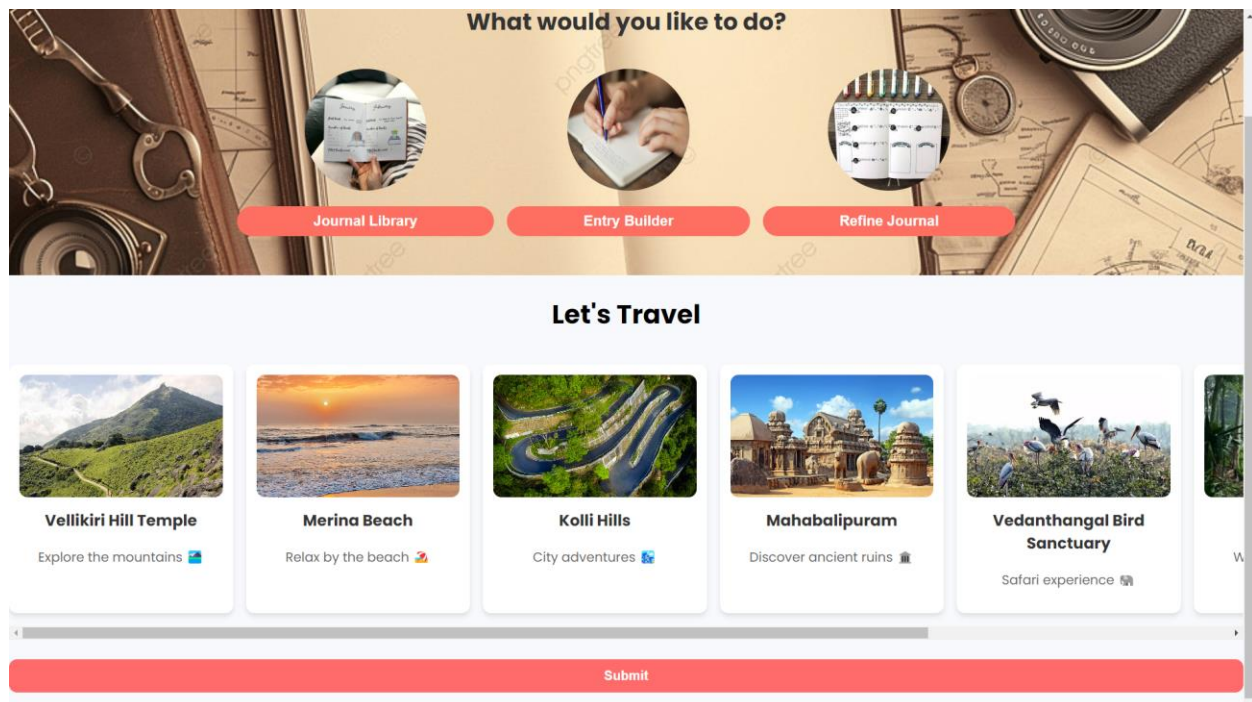


Figure 3.1.8 Let's Travel page (We can choose the tourist place.)

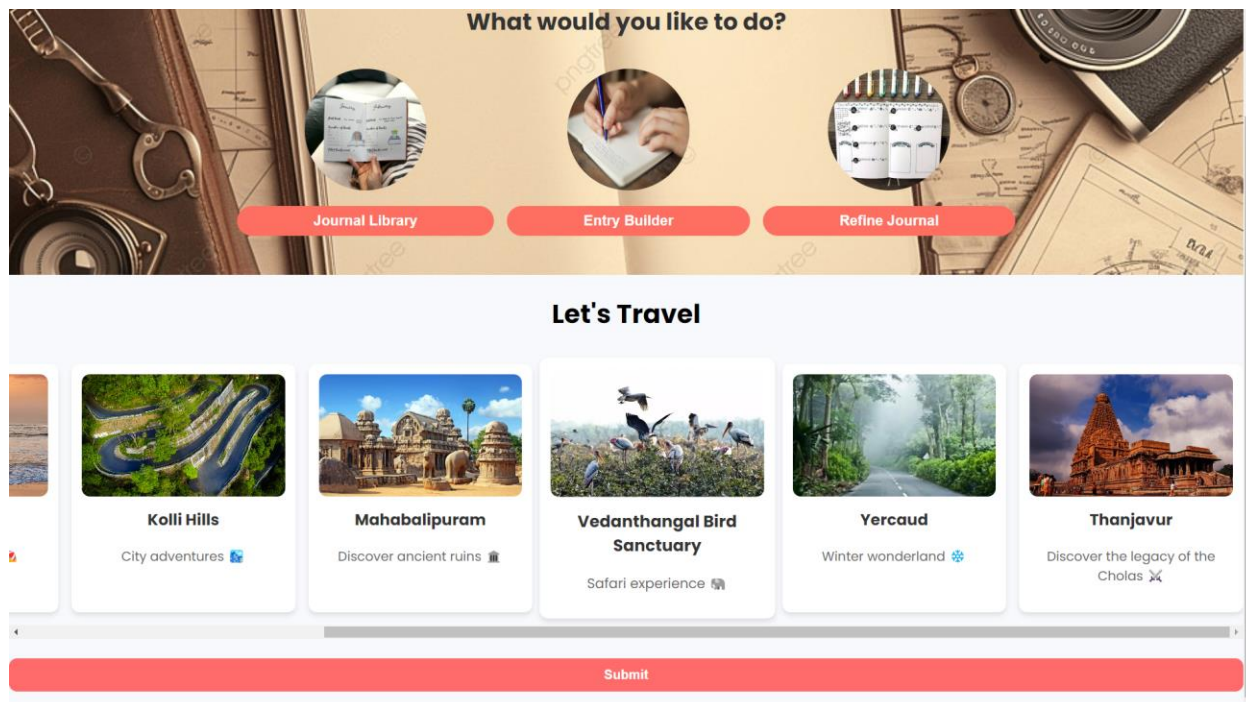


Figure 3.1.9 Let's Travel

### Plan Your Next Trip

Name:

Source:

Destination:

Start Date:

End Date:

Cost of this Place:

₹1300

Submit Booking

Figure 3.1.10 Booking page.



### Plan Your Next Trip

Booking saved successfully!

**Name:**

**Source:**

**Destination:**

**Start Date:**

**End Date:**

**Cost of this Place:**

Submit Booking

Figure 3.1.11 Booking saved image.

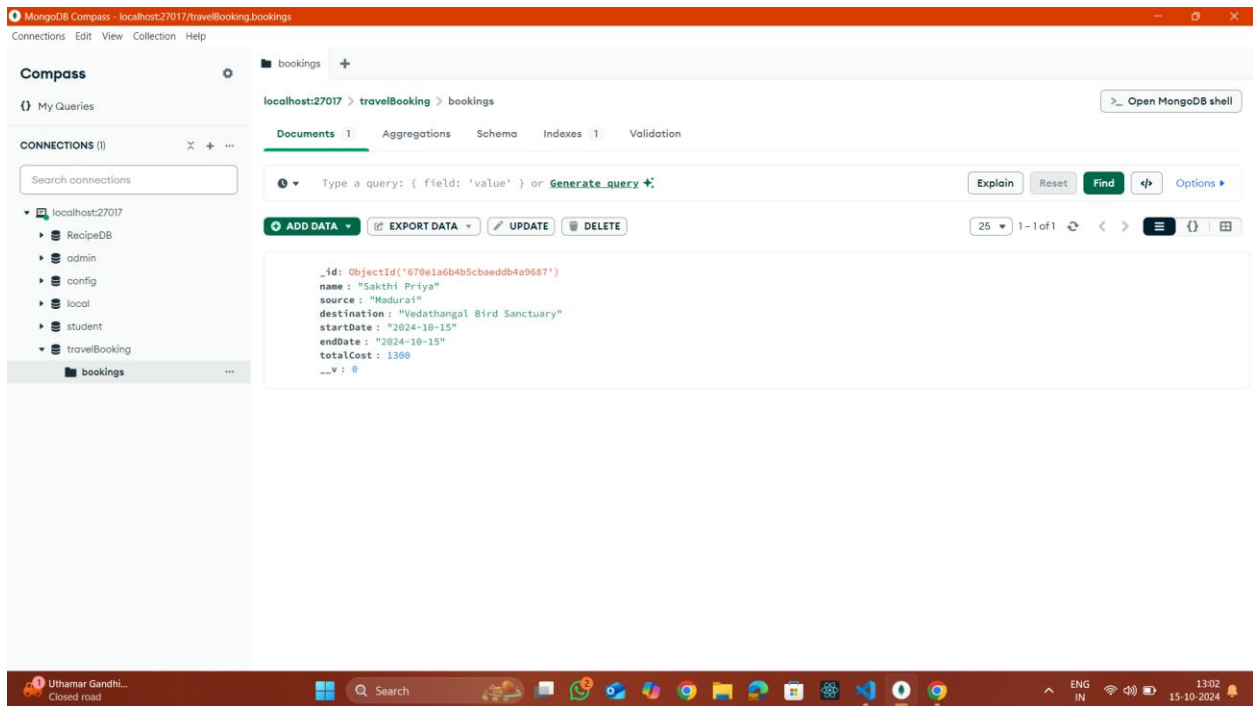


Figure 3.1.12 MongoDB (It displayed the storage datas)

## **DISCUSSION**

The Travel Journal App successfully enables users to create, edit, and view their travel experiences with an intuitive interface. The front-end, built with React, ensures a seamless user experience through dynamic updates without page reloads. The back-end, powered by Node.js and Express.js, efficiently handles requests and ensures secure storage of journal entries in MongoDB. The app's integration of essential features like journaling, trip planning, and booking forms provides a cohesive platform for organizing travel memories and future plans, demonstrating its effectiveness in addressing user needs for a travel documentation tool.

## CONCLUSION

In conclusion, the travel journal app provides a user-friendly platform for documenting travel experiences in an engaging and interactive way. With features like journal entry submission, photo uploads, and a simple interface for reading, editing, and managing entries, the app helps users capture every detail of their trips. This not only enhances reflection on past travels but also aids in planning future ones, making it a valuable tool for all travelers.

The app's simplicity and ease of use stand out. Travelers can quickly record experiences, add photos, and organize memories effortlessly. The smooth combination of text and images helps users create a lasting record of their adventures. Even those with minimal tech skills can easily navigate and manage their journals, thanks to the app's intuitive design.

Additionally, the app bridges traditional travel diaries and modern digital convenience. Unlike handwritten notes, the app allows users to edit entries, upload photos, and organize their travels by date or location, making it easier to revisit memories.

Photo uploads add a personal touch, allowing users to visually relive their adventures alongside their written accounts. The app also makes it easy to transition between reflecting on past journeys and planning new ones, all within one platform.

Ultimately, the travel journal app is more than just a tool for capturing memories—it serves as a digital companion for travelers, enhancing the experience from beginning to end. Whether it's the first adventure or the hundredth, the app offers an interactive way to preserve and share stories, making sure every adventure is remembered. It provides a well-organized, visually appealing solution for preserving life's memorable moments.

## REFERENCES

A. Smith, "Design and Development of a Travel Journal App for Documenting and Sharing Travel Experiences," *International Journal of Software Engineering and Applications*, vol. 10, no. 2, pp. 45-52, May 2023. [Online]. Available: <https://www.ijsea.com/article/travel-journal-app>. [Accessed: Oct. 16, 2024].

