

GaganYatri - Software Documentation and User Manual

Table of Contents

Section	Page Number
1. Introduction	1
1.1. Overview	1
1.2. Key Features	1
2. Requirements	2
2.1. Functional Requirements	2
2.2. Non-Functional Requirements	2
2.3. System Constraints	2
3. Architecture and Design	3
3.1. System Architecture	3
3.2. Technology Stack	3
3.3. Database Schema	3
4. User Guides	4
4.1. End-User Guide	4
4.2. Administrator Guide	4
5. API Documentation	5
5.1. Base URL & Authentication	5

5.2. Core Endpoints	5
6. Deployment and Configuration	6
6.1. Frontend Deployment (Vercel)	6
6.2. Backend Deployment (Render)	6
6.3. Environment Variables	6
7. Maintenance and Support	7
7.1. Monitoring	7
7.2. Support Procedures	7
8. Troubleshooting	8
8.1. Common Issues & Solutions	8
8.2. FAQ	8
9. Appendices	9
9.1. Glossary	9
9.2. References	9

Introduction

1.1 Overview

GaganYatri is a modern, full-stack web application designed to streamline the process of discovering and booking travel experiences. It provides users with an intuitive interface to explore destinations, view detailed information, and manage their travel plans. The application follows a client-server architecture, with a responsive React.js frontend deployed on Vercel and a robust Node.js backend API deployed on Render.

1.2 Key Features

- **User Authentication:** Secure user registration and login system.
- **Space Adventure Exploration:** Browse a catalog of unique space destinations, including planets and moons like Mars, Europa, and Titan.
- **Detailed Destination Views:** Comprehensive pages for each destination, including descriptions, images, and key details.
- **Multi-step Booking Flow:** A guided, multi-stage booking process that allows users to select their adventure, preferred spacecraft, and launch site.
- **Passenger Details Management:** Easily add and manage information for multiple passengers in a single booking.
- **Booking Confirmation:** Receive a summary and confirmation of your booked space adventure.
- **Booking Management:** User dashboard to view, create, and cancel bookings.
- **Responsive Design:** Provides a seamless user experience across a variety of devices, including desktop and mobile.
- **RESTful API:** A well-defined backend API for all frontend operations and potential future integrations.

Requirements

2.1 Functional Requirements

1. The system shall allow users to register a new account.
2. The system shall allow authenticated users to log in.
3. The system shall allow users to browse a list of destinations.
4. The system shall allow users to view detailed information about a specific destination.
5. The system shall allow authenticated users to book a destination.
6. The system shall allow authenticated users to view and manage their list of bookings.
7. The system shall allow administrators to manage (Create, Read, Update, Delete) destinations and user bookings.

2.2 Non-Functional Requirements

- 1. Performance: The frontend should load in under 3 seconds on a standard 4G connection.
- 2. Availability: The application should aim for 99.5% uptime.
- 3. Usability: The user interface should be intuitive and require minimal training to use.
- 4. Security: User passwords must be hashed. API endpoints (except login/register) must be protected via authentication tokens.

2.3 System Constraints

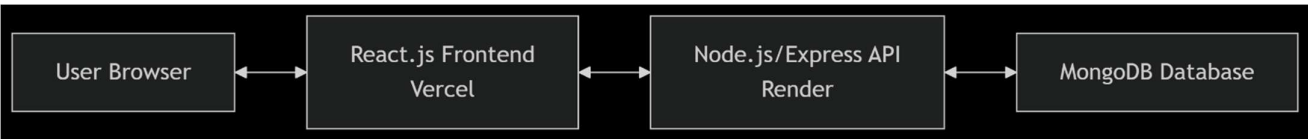
- * The frontend is built exclusively for modern web browsers.
- * The backend API is stateless and relies on JWT for session management.
- * The free-tier hosting services (Vercel, Render) may have cold start times, affecting initial load performance.

Architecture and Design

3.1 System Architecture

GaganYatri employs a 3-tier architecture:

- 1. Presentation Tier (Frontend): A Single-Page Application (SPA) built with React.js, hosted on Vercel. It handles the UI and user interactions.
- 2. Application Tier (Backend): A RESTful API built with Node.js and Express.js, hosted on Render. It processes business logic, handles authentication, and communicates with the database.
- 3. Data Tier: A MongoDB database which is hosted on MongoDB Atlas that persists all application data, including users, destinations, and bookings.



3.2 Technology Stack

Layer	Technology
Frontend	React.js, CSS, HTML, Vercel
Backend	Node.js, Express.js, JWT, Render
Database	MongoDB, Mongoose ODM
Other	Git, GitHub

3.3 Database Schema

Main collections and their key fields:

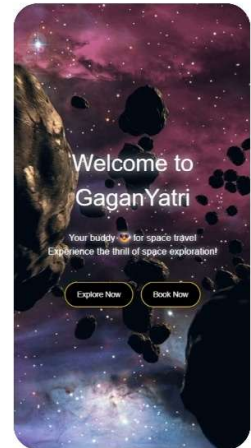
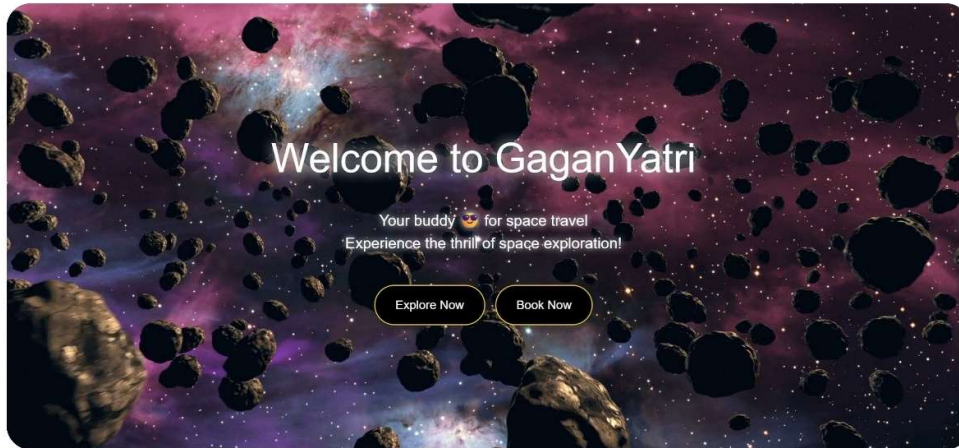
- users Collection
 - `_id` (ObjectId)
 - `username` (String)
 - `email` (String)
 - `password` (String - hashed)
 - `createdAt` (Date)
- destinations Collection
 - `_id` (ObjectId)
 - `name` (String)
 - `description` (String)
 - `location` (String)
 - `price` (Number)
 - `imageUrl` (String)
 - ...other fields
- bookings Collection
 - `_id` (ObjectId)
 - `userId` (ObjectId, ref: 'User')
 - `destinationId` (ObjectId, ref: 'Destination')
 - `bookingDate` (Date)
 - `status` (String) e.g., "confirmed", "cancelled"

User Guide

Welcome to GaganYatri

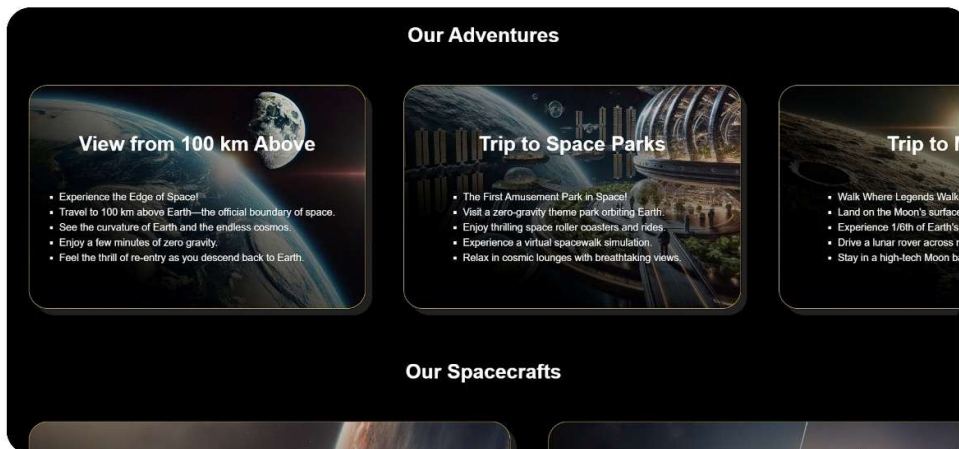
Welcome to the GaganYatri web application, your first step toward an unforgettable space adventure! The homepage is your launchpad, providing an overview of the thrilling destinations, advanced spacecraft, and global launch sites available for your journey.

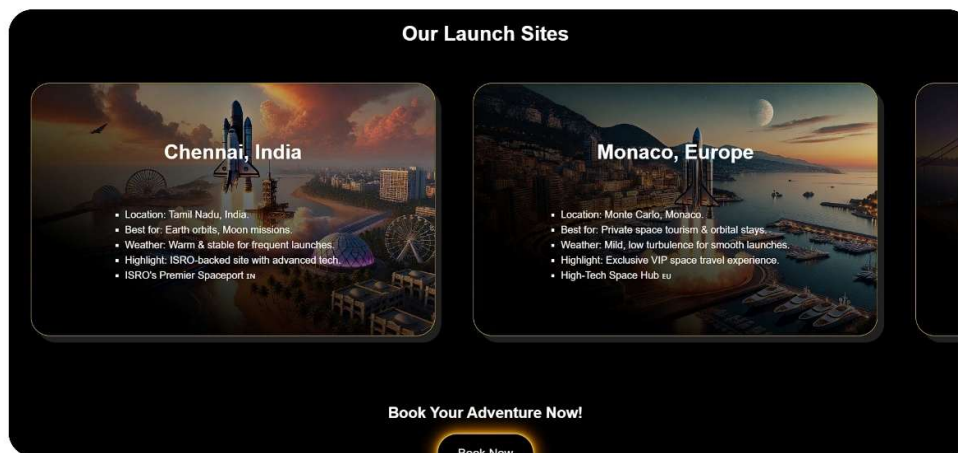
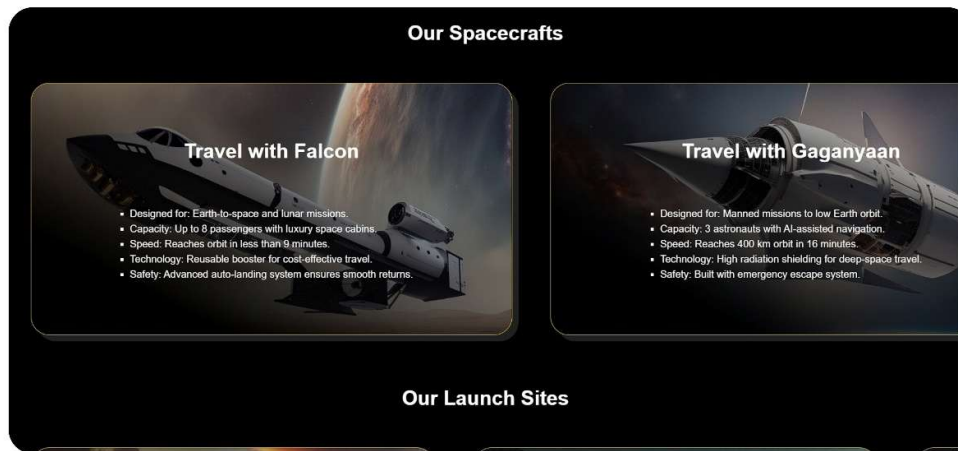
To begin exploring, simply scroll down to see the "Our Adventures" section.



Exploring Adventures, Spacecrafts, Launch sites

The homepage features a gallery of unique adventures, spacecrafts, launch sites each with a brief description. You can see destinations, vehicles, and launch pads available for service.





To start the booking process, you have two options:

- Click the "Book Now" button at the top of the page.
- Click the "Book Now" button at the bottom of the page.

Both buttons will take you to the first step of the booking flow.

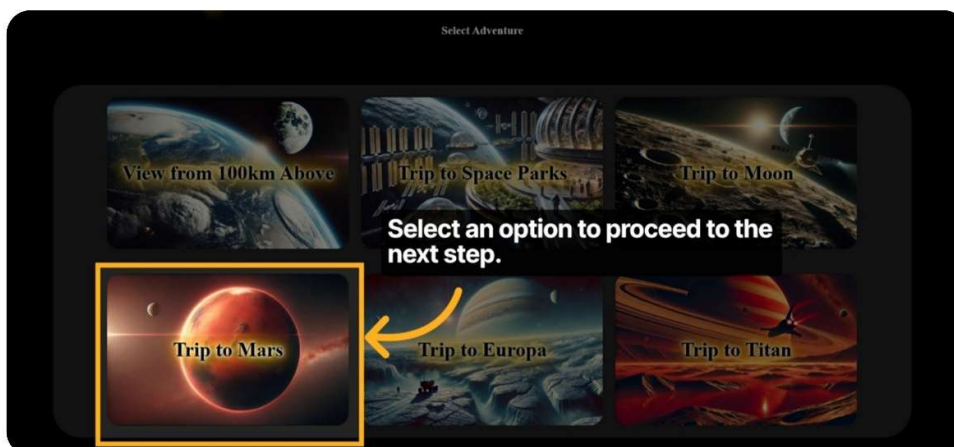
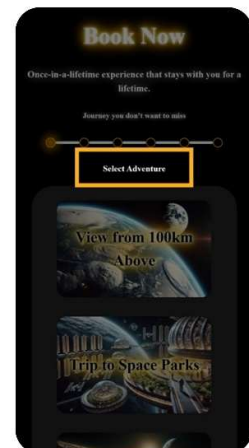
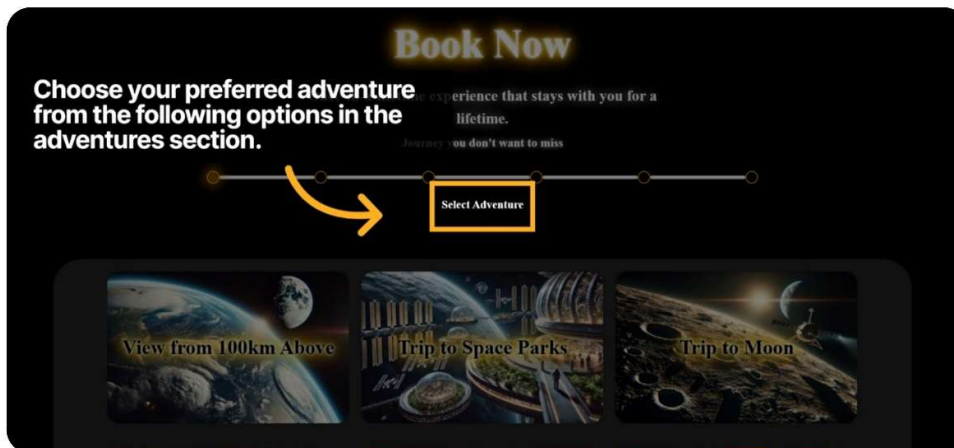
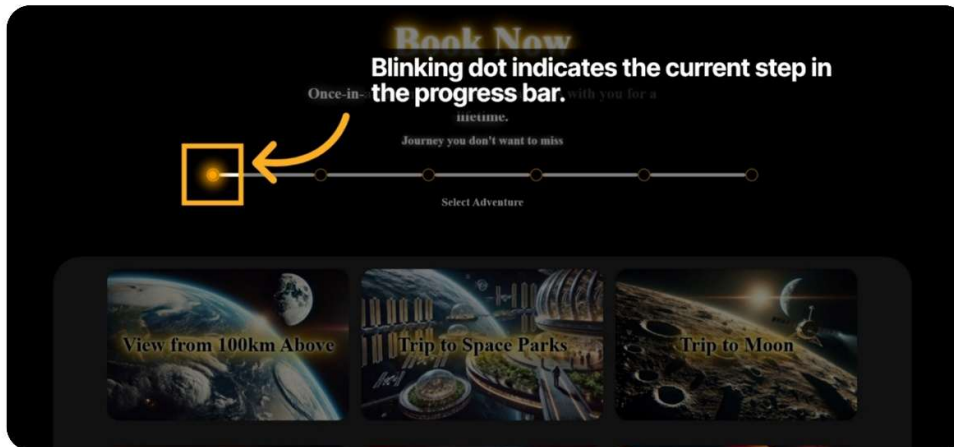
Booking Your Space Adventure

The booking process is a simple, five-step journey that guides you through every decision. You can track your progress using the indicator at the top of the page.

Step 1: Select Adventure

First, you will be asked to select your desired space adventure. Choose from a variety of trips, including "View from 100km Above," "Trip to Mars," or "Trip to Europa." Simply click on the card for the adventure you want to book.

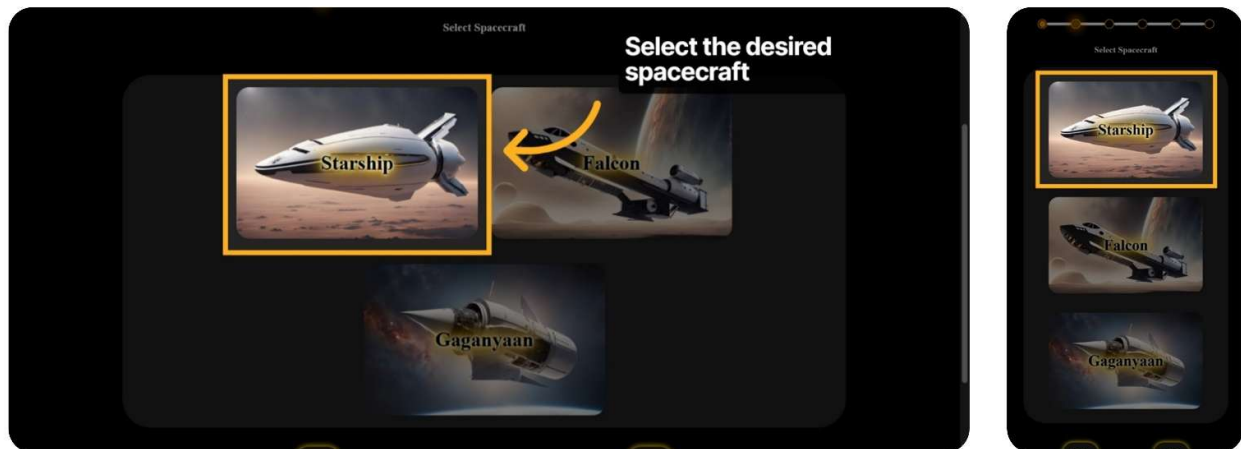
Once you have made your choice, click the Next button at the bottom of the page.



Step 2: Select Spacecraft

Next, you will choose the spacecraft for your journey. Options include the Gaganyaan, Starship, and Falcon. Each spacecraft is designed for different types of missions.

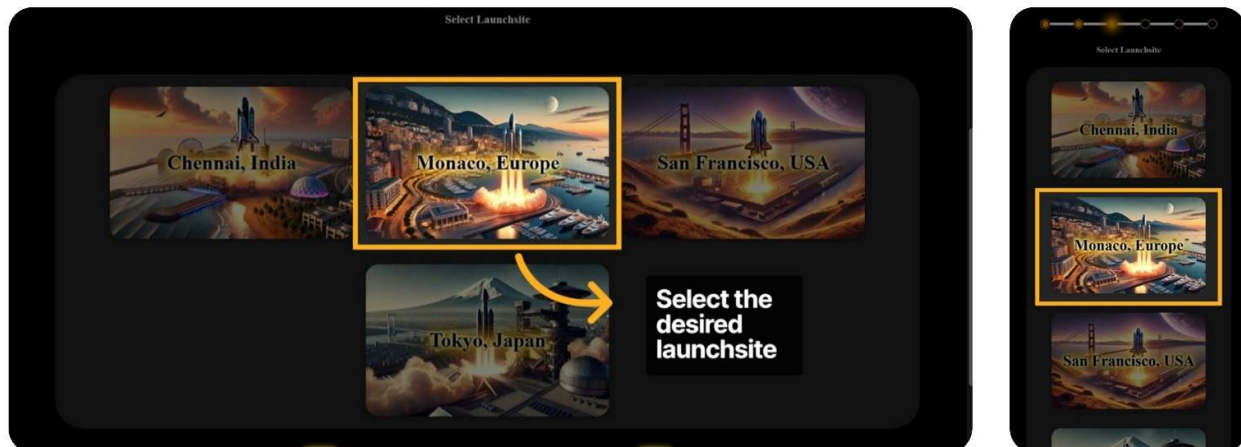
Click the card for the spacecraft you prefer, then click Next to proceed.

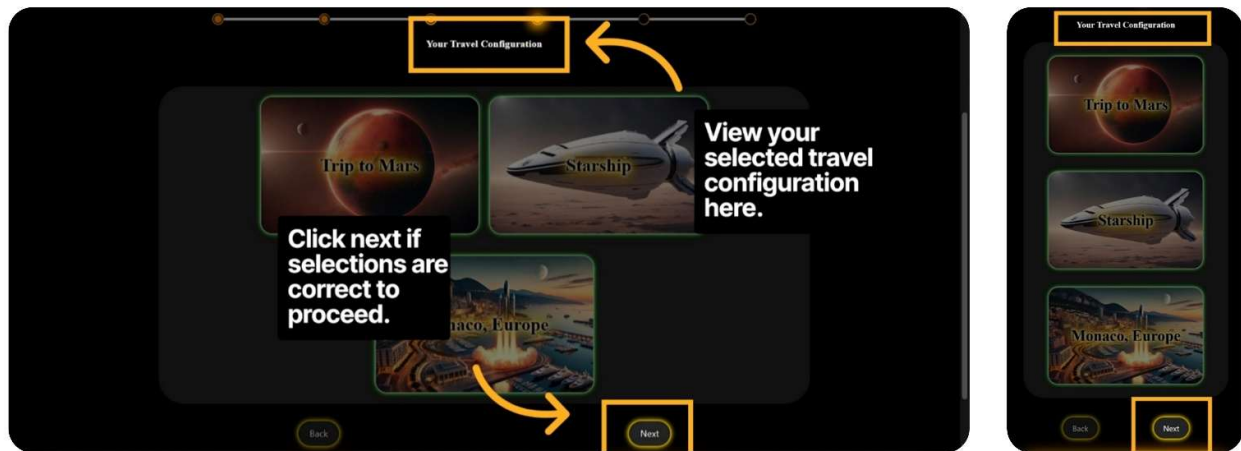


Step 3: Select Launch Site

Now, select your launch site. The application offers a few options to choose from, such as Chennai, India, Monaco, Europe, and San Francisco, USA.

Click on the location you want to launch from and then press Next.





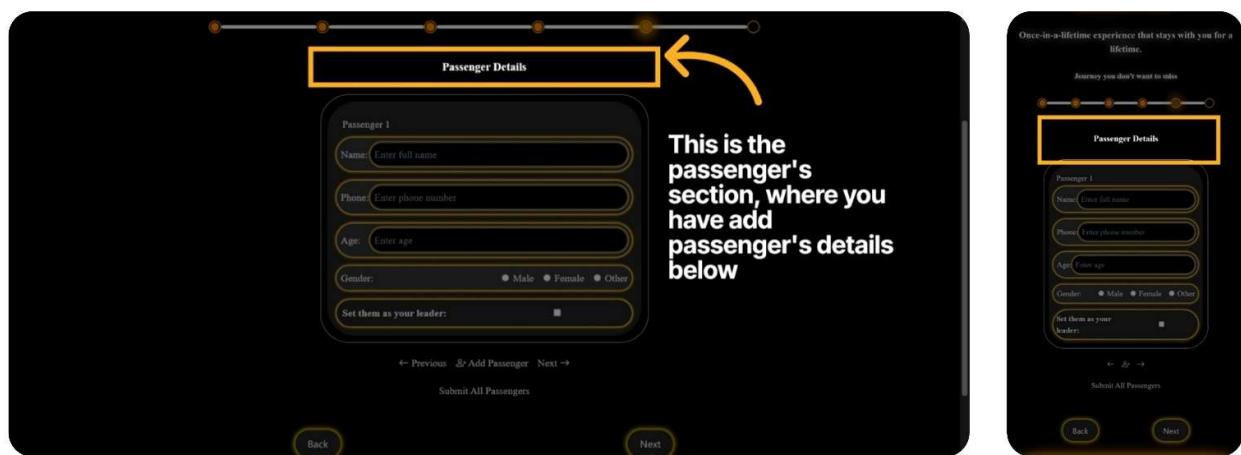
Step 4: Enter Passenger Details

This is where you'll enter the information for yourself and any other passengers. For each passenger, you need to provide their:

- Name
- Phone number
- Age
- Gender
- Email
- Address

You can also use the checkbox to "Set them as your leader." To add more passengers, click the Add Passenger button.

Once all passenger details are filled out, click Submit All Passengers. The application will verify that all required fields are complete before moving on. If any information is missing, you will receive a message.



This is where you can decide to make this person as the group leader for further communication

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☐

← Previous & Add Passenger Next →

Submit All Passengers

Back Next

Once-in-a-lifetime experience that stays with you for a lifetime.

Journey you don't want to miss

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☐

← & →

Submit All Passengers

Back Next

This will open two additional fields.

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☒

Email:

Address:

← Previous & Add Passenger Next →

Submit All Passengers

Back Next

Journey you don't want to miss

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☒

Email:

Address:

← & →

Submit All Passengers

Back Next

Fill the form with accurate details to stay updated.

Passenger 1

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☒

Email:

Address:

← Previous & Add Passenger Next →

Submit All Passengers

Journey you don't want to miss

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☒

Email:

Address:

← & →

Submit All Passengers

Back Next

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☒ Male ☐ Female ☐ Other

Set them as your leader: ☒

Email:

Address:

← Previous **Add Passenger** Next →

Submit All Passengers

You can add multiple passengers

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☒ Male ☐ Female ☐ Other

Set them as your leader: ☒

Email:

Address:

← **Add** →

Submit All Passengers

Back Next

Passenger Details

Passenger 1

Name:

Phone:

Age:

Gender: ☒ Male ☐ Female ☐ Other

Set them as your leader: ☒

Email:

Address:

Passenger 2

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☐

← Previous **Add Passenger** Next →

This will add multiple passengers on display

Passenger Details

Passenger 2

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☐

← **Add** →

Submit All Passengers

Back Next

Passenger Details

Passenger 2

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☐

Passenger 3

Name:

Phone:

Age:

Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☐

← Previous **Add Passenger** Next →

Delete the desired passenger.

Passenger Details

Passenger 2

Name:

Phone:

Age:

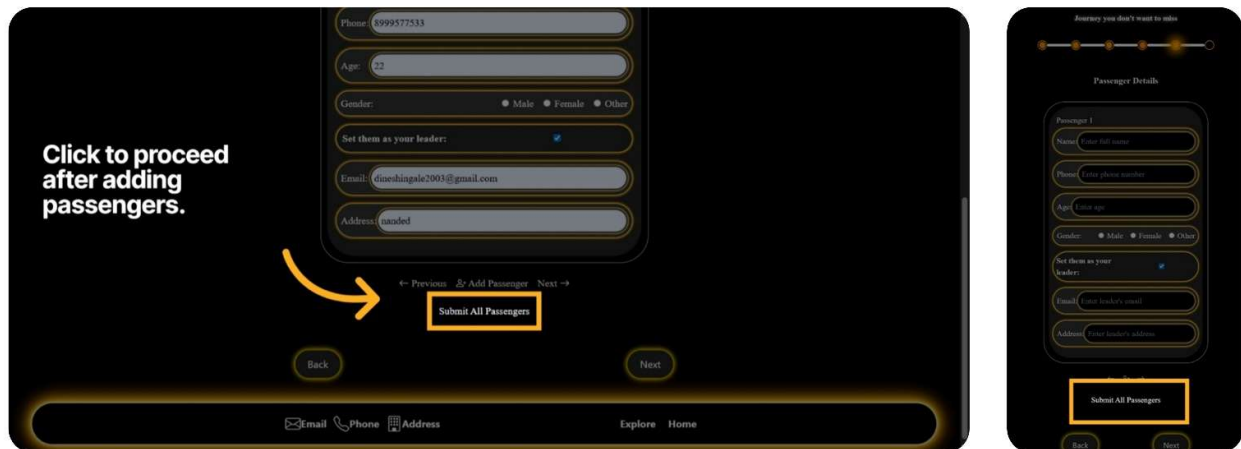
Gender: ☐ Male ☐ Female ☐ Other

Set them as your leader: ☐

← **Add** →

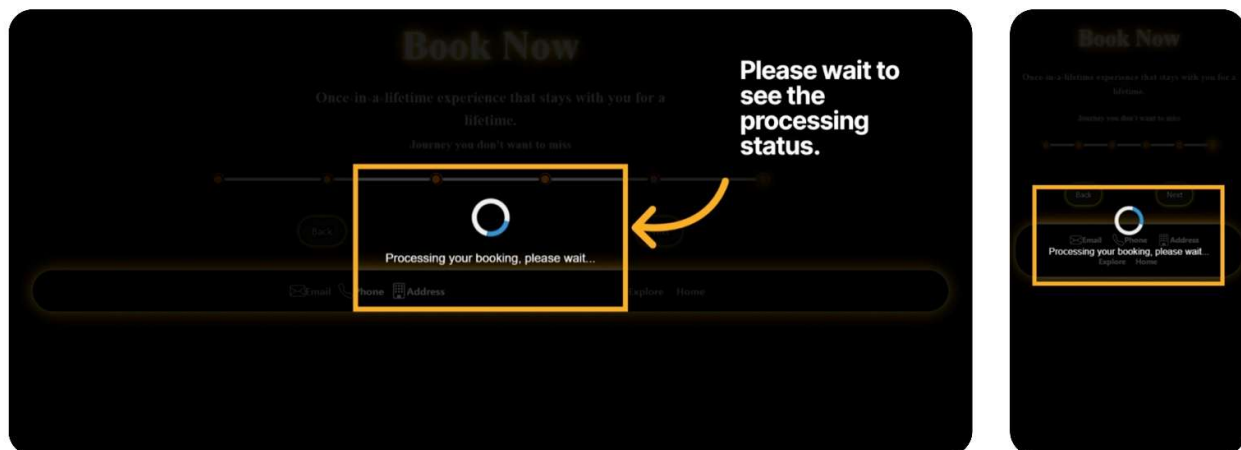
Submit All Passengers

Back Next



Step 5: Confirm Booking

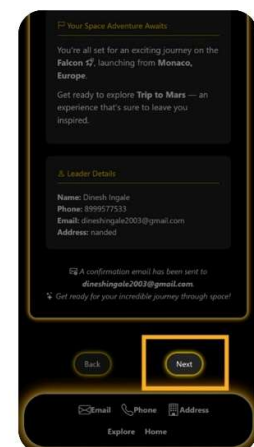
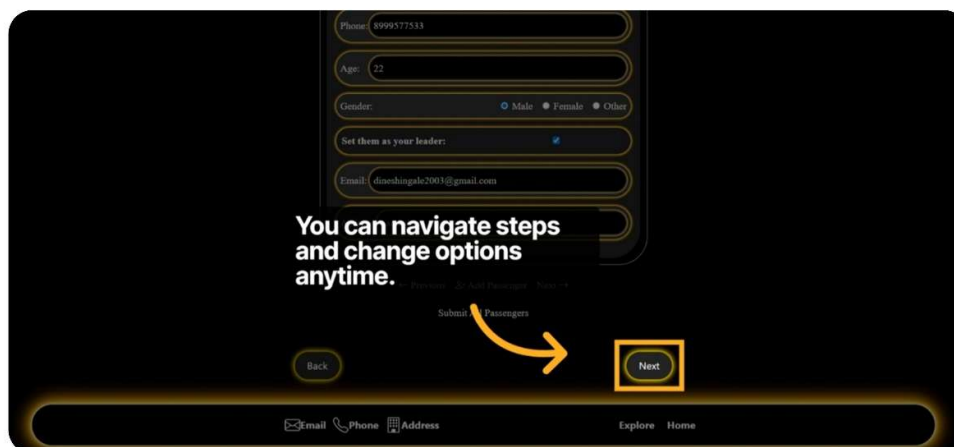
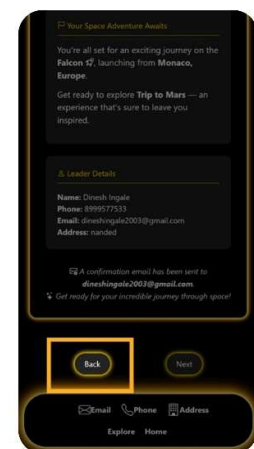
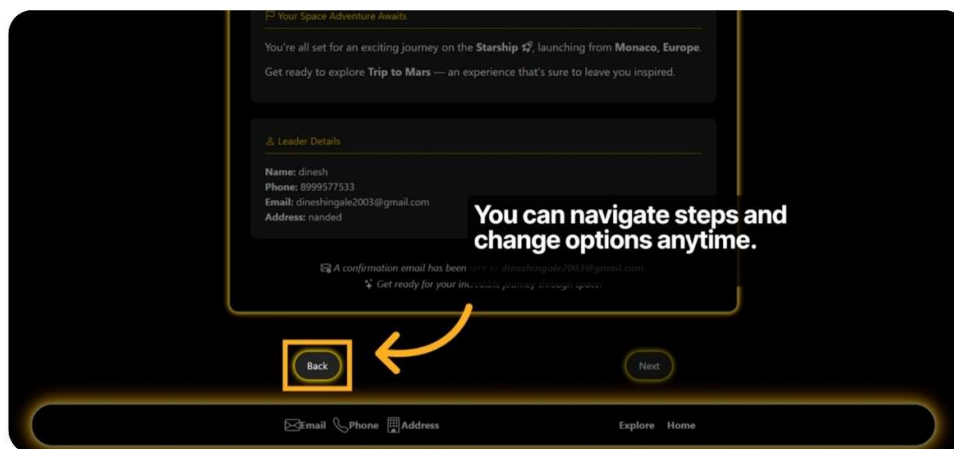
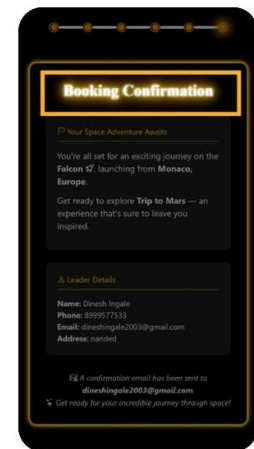
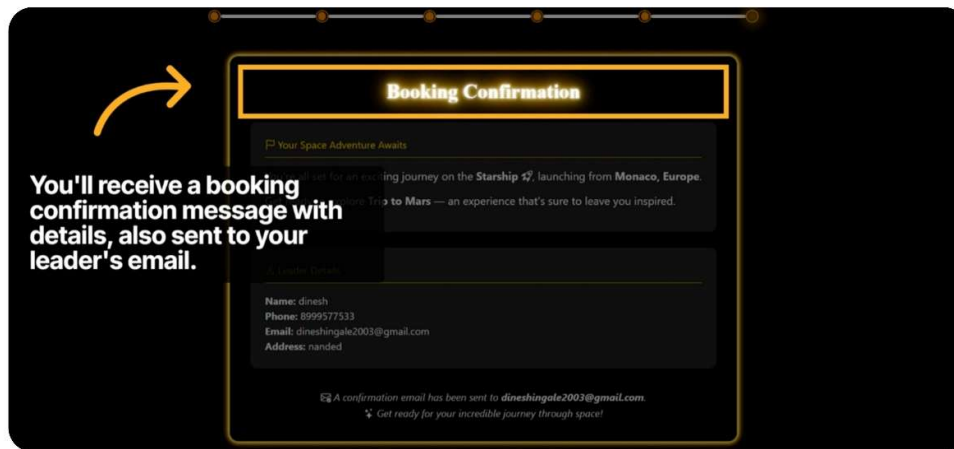
After successfully submitting your passenger details, the system will process your request. Please be patient, as this may take a moment.



Booking Confirmation

Once your booking is confirmed, you will see a summary page. This page provides a final overview of your space adventure, including your destination, launch site, and the details of all passengers. You will also be notified that a confirmation has been sent to the email address of the leader you selected.

Congratulations, your space adventure is booked!



4.2 Administrator Guide

Administrator functions are typically managed directly via the database or through protected API endpoints not exposed in the standard UI.

1. Managing Destinations: Use a database management tool (like MongoDB Compass) or a dedicated admin panel (if built) to perform CRUD operations on the destinations collection.
2. Viewing All Bookings: Query the bookings collection directly from the database.

API Documentation

5.1 Base URL & Authentication

- * Base URL: <https://gaganyatri-server.onrender.com/api>
- * Authentication: Most endpoints require a JWT token to be sent in the HTTP request header.
 - * Header: Authorization: Bearer <your_jwt_token_here>
 - * Obtain the token by calling the /auth/login or /auth/register endpoints.

5.2 Core Endpoints

Method	Endpoint	Description	Auth Required	Request Body
POST	/auth/register	Register a new user	No	{ "username", "email", "password" }
POST	/auth/login	Login a user	No	{ "email", "password" }
GET	/destinations	Get a list of all destinations	No	-
GET	/destinations/:id	Get details of a destination	No	-
GET	/bookings	Get all bookings for the user	Yes	-
POST	/bookings	Create a new booking	Yes	{ "destinationId", "bookingDate" }
DELETE	/bookings/:id	Cancel a specific booking	Yes	-

Deployment and Configuration

6.1 Frontend Deployment (Vercel)

1. Push code to the main branch of the connected GitHub repository.
2. Vercel automatically detects changes and triggers a new build and deployment.
3. The live site is updated at gaganyatri.vercel.app.

6.2 Backend Deployment (Render)

1. The backend is connected to its GitHub repository.
2. Render automatically deploys new commits to the main branch.
3. The live API is available at gaganyatri-server.onrender.com.

6.3 Environment Variables

- * Backend (Render) Variables:
 - * MONGODB_URI: The connection string for the MongoDB database.
 - * JWT_SECRET: A secret key for signing JSON Web Tokens.
 - * PORT: The port the server should listen on (e.g., 5000).
-

Maintenance and Support

7.1 Monitoring

- * Vercel: Provides analytics on frontend traffic and performance.
- * Render: Provides basic metrics and logs for the backend web service.
- * MongoDB Atlas: Provides database performance monitoring and alerting.

7.2 Support Procedures

- * Users: For support, users should contact the application owner via the contact information listed on the website or GitHub repository.
 - * Developers: Issues and bugs can be reported by creating a new "Issue" on the project's GitHub repository.
-

Troubleshooting

8.1 Common Issues & Solutions

- * Issue: "Failed to fetch" or network errors on the frontend.
 - * Solution: Check if the backend API (gaganyatri-server.onrender.com) is up and running. It may be spinning down due to inactivity on the free tier.
- * Issue: "Invalid JWT token" or "Authentication failed".
 - * Solution: The user's login session may have expired. Log out and log back in to generate a new token.

* Issue: Frontend looks broken or styles are missing.

* Solution: Hard refresh the page (Ctrl+F5 or Cmd+Shift+R) to clear the cached CSS/JS files.

8.2 FAQ

Q: Is my payment information secure?

A: This application is a demo. It does not currently process real payments, so no sensitive financial data is handled.

Q: I forgot my password. How can I reset it?

A: A password reset feature is not currently implemented in this version. Please contact support.

Q: Can I contribute to this project?

A: Yes! The code is open source on GitHub. Please fork the repository and submit a Pull Request with your changes.

Appendices

9.1 Glossary

- API (Application Programming Interface): A set of rules that allows programs to talk to each other.
- JWT (JSON Web Token): A compact, URL-safe means of representing claims to be transferred between two parties.
- CRUD: Acronym for Create, Read, Update, Delete - the four basic functions of persistent storage.
- Endpoint: A specific URL within an API that represents a specific resource or function.

9.2 References

- Frontend and Backend GitHub Repo: <https://github.com/dineshingale>
- Live Application: <https://gaganyatri.vercel.app>
- Live API: <https://gaganyatri-server.onrender.com>