



HACKATHON PHASE-I

COLLEGE CODE :8203
COLLEGE NAME : A.V.C college of engineering
DEPARTMENT : Information Technology
STUDENT NM-ID : 4D2198FA90720AD8F834BBD4C69121C9
ROLL NO :23IT100
DATE :

TECHNOLOGY : Front End/Node JS

PROJECT NAME : Travel Booking System with Payments

SUBMITTED BY: (Team members name, mention TL)

Name : S.Sivarajaganapathi (TL)

Mobile No:7845102808

Team members:

K.siva 70CF1E8694019C2B720484EFFF6C1779

B.Senthamizhan 8CE2DC2CF6A1FBC9FDD415138FD7AD55

N.Selvaganapathy 7A4D88912130BA0A759424ABE7A0FD32

Travel Booking System with Payments

1. Project Overview & Objectives

The Travel Booking System with Payments is designed to provide users with a seamless and unified platform for planning, booking, and managing their travel experiences. In the modern travel industry, users often rely on multiple platforms to search for flights, hotels, and transport services, which leads to confusion and inefficiency. Our system aims to solve this problem by integrating all essential services into one platform.

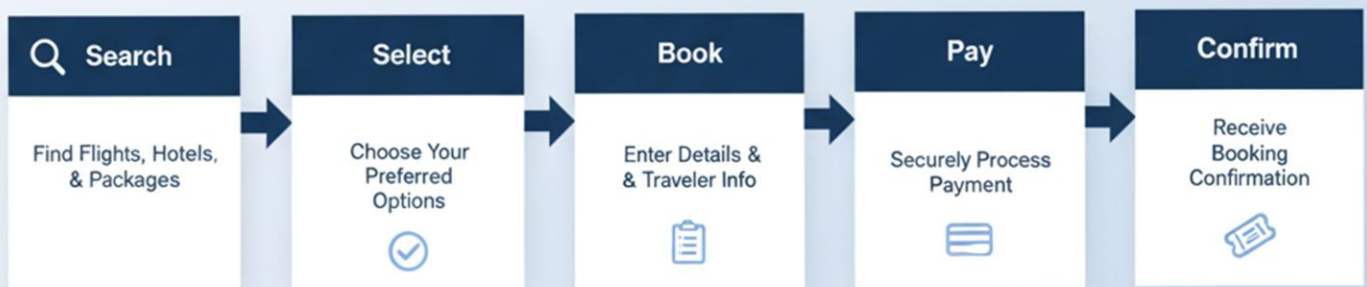
The main objective of this project is to create a convenient and secure environment where users can search, compare, and book various travel services with minimal effort. The platform allows users to manage their entire journey from start to finish — including booking flights, reserving hotels, and processing payments all in a single dashboard.

The key goals include:

- **Simplified travel management:** Enabling users to handle all bookings in one place.
- **Transparent pricing:** Providing accurate fare details with no hidden charges.
- **Secure payment system:** Using encryption and trusted payment gateways to protect user transactions.
- **User-friendly experience:** Offering a clean, intuitive interface that enhances usability.
- **Data-driven insights:** Using analytics to suggest personalized recommendations and optimize pricing.

The expected outcome is a reliable and scalable system that benefits both customers and service providers. For customers, it brings convenience and trust; for providers, it opens a larger customer base through digital integration. This project represents a step toward smarter, tech-enabled travel management where automation, security, and user satisfaction work together efficiently.

User Journey Flow: Search to Confirmation



2. Technology Stack & Environment Setup

The Travel Booking System will be developed using modern web technologies that ensure scalability, reliability, and easy maintenance.

Backend:

The server-side of the application will be built using **Node.js** with the **Express.js** framework. This combination provides fast and lightweight development with an efficient routing system. Express helps manage APIs, middleware, and authentication with minimal overhead.

Frontend:

For the front-end interface, a **React.js** framework will be used to provide a smooth, responsive, and interactive user experience. React's component-based architecture allows for reusable UI blocks and easy state management. The system's design will focus on accessibility, simplicity, and responsiveness across devices.

Database:

The data will be stored in **MongoDB**, a NoSQL database that handles flexible schemas and large-scale data efficiently. MongoDB is ideal for this project because travel data, such as user profiles, flight details, and payment histories, can vary in structure and volume.

Tools & Environment Setup:

- **Visual Studio Code (VS Code):** Development IDE for both frontend and backend.
- **Git & GitHub:** Version control and collaborative code management.
- **Postman:** For testing and debugging RESTful APIs.
- **npm & Node Environment:** To manage dependencies and run the server.
- **Stripe Sandbox:** For integrating and testing the payment gateway securely.

Together, this stack provides a robust foundation that is easy to maintain, scales well, and supports continuous integration and deployment. The development environment will ensure consistency across the team using .env configuration files to manage API keys and database credentials securely.

3. API Design & Data Model

The application's core functionality relies on a well-structured set of RESTful APIs that connect the front-end interface to the backend logic and database.

API Design:

The system will include APIs for user registration, login authentication, travel search, booking management, and payment processing. Each API will follow REST conventions, using standard HTTP methods like GET, POST, PUT, and DELETE.

For example:

- **POST /register:** Create a new user account.
- **POST /login:** Authenticate users and return an access token.
- **GET /flights:** Retrieve available flights based on user search criteria.
- **POST /book:** Create a new booking for flights or hotels.

- **POST /payment:** Process payments through the integrated gateway.
- **GET /bookings/:id:** Fetch user-specific booking details.

Each response will be returned in JSON format for easy parsing on the frontend.

Data Model:

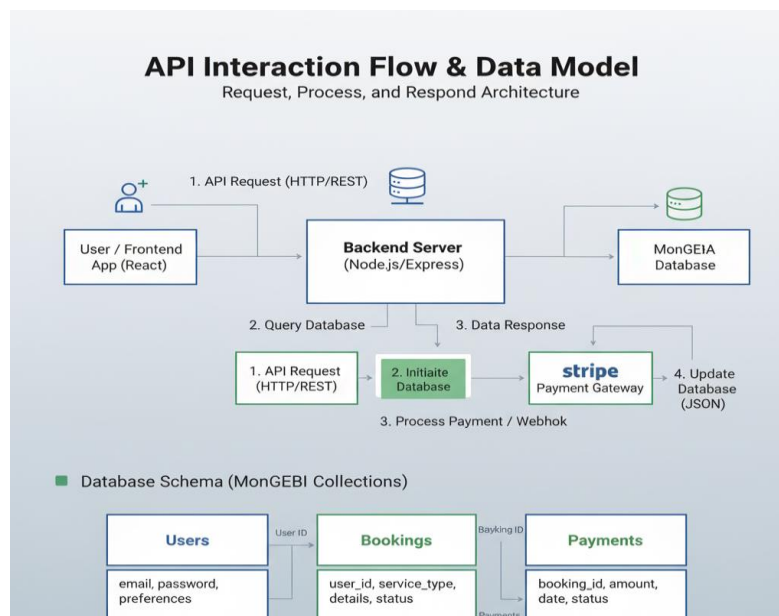
The database schema will include three main collections Users, Bookings, and Payments.

- **Users:** Contains user credentials, personal details, and saved preferences.
- **Bookings:** Stores flight, hotel, and transport booking information linked by user ID.
- **Payments:** Records transaction details, amount, payment status, and gateway response IDs.

Sample Schema (MongoDB):

```
{
  "user_id": "U101",
  "booking_id": "B2001",
  "service_type": "Flight",
  "source": "Chennai",
  "destination": "Delhi",
  "date": "2025-12-01",
  "status": "Confirmed",
  "payment_id": "P9876"
}
```

The API design ensures modularity and allows the system to be extended easily in the future for instance, to include travel insurance or ride-sharing integration. Proper validation and error handling will be implemented to maintain data integrity.



4. Front-End UI/UX Plan

The frontend interface is the most crucial element of the user experience. The goal is to create a clean, engaging, and responsive interface that feels natural and easy to navigate.

Wireframes & Navigation Flow:

The homepage will feature a unified search bar where users can enter destinations, dates, and preferences. The search results page will display available options with filters like price range, ratings, and travel time. The booking confirmation page will summarize details and guide users through secure payment.

The navigation flow will be designed for clarity and simplicity:

1. **Home Page → Search → Result Page → Booking Page → Payment → Confirmation**
2. Users can log in or sign up before completing their bookings.
3. A dashboard section will allow users to view and manage their previous bookings and payment histories.

Design Elements:

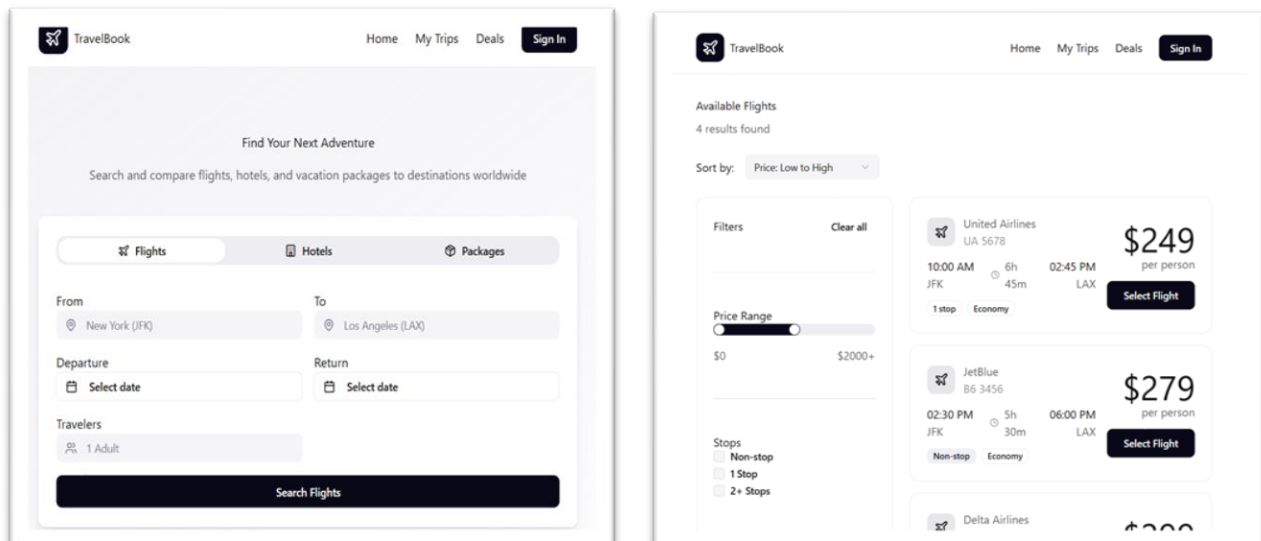
- **Color Scheme:** Calm and professional tones (blue and white) to reflect trust and reliability.
- **Typography:** Clean fonts like Roboto or Open Sans for readability.
- **Layout:** Grid-based responsive layout that adapts to mobile and desktop screens.
- **Icons & Visuals:** Use of standard icons to indicate flights, hotels, and payments, improving recognition and aesthetics.

State Management:

React's **Context API** or **Redux** will handle global states such as user authentication, search filters, and booking data. This ensures consistent data flow and avoids unnecessary re-renders.

User Experience Goals:

The platform should reduce the time and effort required to complete a booking. Every step should feel intuitive, and all important actions like “Pay Now” or “Cancel Booking” should be visible and clearly labeled. Accessibility features, such as keyboard navigation and screen reader support, will also be included.



1Travelers

2Review

3Pay

Traveler Information

+ Add Traveler

Traveler 1

First Name

John

Last Name

Doe

Email

john@example.com

Phone

+1 (555) 123-4567

Date of Birth

dd-mm-yyyy

Traveler Type

Adult (18+)

Back

Continue

TravelBook

HomeMy TripsDealsSign In

K → LAX

ited Airlines • UA 5678

Total Price

\$249

per person

Book Now

Flight Details

Departure

John F. Kennedy International
JFK

2025-11-15

10:00 AM

6h 45m
1 stop

Arrival

Los Angeles International
LAX

2025-11-15

02:45 PM

Cabin Class

Economy

Baggage Allowance

1 checked bag

Payment Method

Credit Card

Debit Card

PayPal

Card Number

1234 5678 9012 3456

Expiry Date

MM/YY

CVV

123

Billing Address

Street Address

123 Main St

City

New York

State

NY

ZIP Code

10001

Country

Select country

Back

Complete Booking

Order Summary

JFK → LAX

United Airlines • UA 5678

Price (1 person)

\$249.00

Taxes & Fees

\$29.88

Service Fee

\$29.99

Total

\$308.87

✓

2Review

3Pay

Review Your Booking

Booking Details

Order Summary

JFK → LAX

United Airlines • UA 5678

Price (1 person)

\$249.00

Taxes & Fees

\$29.88

Service Fee

\$29.99

Total

\$308.87

TravelBook

HomeMy TripsDealsSign In

✓

Booking Confirmed!

Your booking has been successfully completed. A confirmation email has been sent to your inbox.

Booking Reference

BKSHQGLJD

October 12, 2025 at 05:25 PM

Booking Details

JFK → LAX

United Airlines • UA 5678

2025-11-15

5. Development & Deployment Plan

The project development will follow an **Agile methodology**, ensuring iterative progress, feedback integration, and adaptability to changes. The work will be divided into sprints with clear deliverables and milestones.

Team Roles:

- **Frontend Developer:** Handles UI components and integration with backend APIs.
- **Backend Developer:** Builds and maintains server logic, routes, and database communication.
- **Database Administrator:** Manages schema design, indexing, and data security.
- **Tester/QA Engineer:** Conducts testing for functionality, usability, and performance.
- **Project Manager:** Oversees timeline, sprint management, and deployment coordination.

Git Workflow:

All team members will use GitHub for collaboration. Feature branches will be created for each module and merged only after successful testing through pull requests. Continuous integration tools like GitHub Actions can be added to automate testing on every commit.

Testing Approach:

- **Unit Testing:** Using Jest for testing API endpoints and UI components.
- **Integration Testing:** Ensuring end-to-end functionality between frontend and backend.
- **Load Testing:** Simulating multiple concurrent users to ensure performance stability.

Hosting & Deployment Strategy:

The application will be deployed using cloud-based services:

- **Backend:** Hosted on **Render** or **AWS EC2** with environment variables for API keys.
- **Frontend:** Deployed on **Vercel** or **Netlify** for continuous delivery.
- **Database:** **MongoDB Atlas** will host the database with backup and monitoring.

The deployment pipeline will ensure that updates can be rolled out without downtime. Regular versioning and documentation will maintain clarity for future upgrades.

In the end, the goal is to deliver a system that not only functions efficiently but also reflects professionalism and modern web development standards suitable for real-world travel industry applications.