HACKATHON PHASE-I

COLLEGE CODE :6107

COLLEGE NAME : Government College of Engineering,Bargur

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID : 1c75d06878c1097aab048764debe2c9e

ROLL NO :2303610710422040

DATE :08.10.2025

TECHNOLOGY : Front End/Node JS

PROJECT NAME : Ticket Booking System with Payments

SUBMITTED BY:

Name :

Ranjana S (TL)
Varshini P
Sathya G
Varshini M

Mobile No: 6369600933

1. Project Overview & Objectives

Problem Statement

- Travelers face fragmented platforms for flight and hotel bookings, multiple payment portals, and lack of a unified dashboard to track upcoming trips. Manual searching, booking, and payment processes are time-consuming and error-prone.

- TravelMate addresses these challenges by providing a single, intuitive platform that integrates flight & hotel search, booking, and payments with a unified dashboard.

Key Features

- User authentication (JWT-based login & refresh tokens)

- Search and filter flights and hotels by date/location

- Booking flights and hotels with availability check

- Payment integration (mock or sandbox for demo)

- Dashboard displaying upcoming trips and past bookings

- Profile management with travel preferences

Expected Outcome

- A fully functional travel booking system suitable for hackathon demos

- End-to-end workflow: register → search → book → pay → dashboard update

- Seeded demo data for flights, hotels, and users for instant testing

- Extensible system for AI recommendations, maps, or real

payment integration

2. Technology Stack & Environment Setup

Backend

- Node.js + Express — RESTful API server for managing bookings, flights, hotels, payments
- MongoDB (Mongoose) — flexible schema to store users, bookings, flights, hotels, and payments
- JWT Authentication — access & refresh tokens for secure sessions

Frontend

- React — single-page application
- Tailwind CSS — responsive UI and consistent styling
- Recharts — dashboard charts and trend visualization

Tools

- Postman — API testing
- Git + GitHub — version control and collaboration
- Nodemon — development server for auto-reloading
- Heroku / Railway / Vercel — hosting backend and frontend

Environment Setup

- Clone repository

- Backend:

```
cd backend
npm install
cp .env.example .env
npm run seed  # populate demo flights/hotels/users
npm run dev
```

- Frontend:

```
cd frontend
npm install
npm start
```

3. API Design & Data Model

Planned REST Endpoints

Endpoint	Method	Description
/api/auth/register	POST	User registration
/api/auth/login	POST	User login
/api/auth/refresh	POST	Refresh access token
/api/auth/logout	POST	Revoke refresh token
/api/flights	GET	Search flights (from/to/date)
/api/hotels	GET	Search hotels (location)
/api/bookings	POST	Create a booking (flight/hotel)
/api/bookings	GET	List user bookings
/api/payments	POST	Make payment for a booking

Request / Response Example

POST /bookings


```
{
 "type": "flight",
 "itemId": "64f8f9dbe4b0a2a12345abcd",
 "nights": 0
}
```


Response


```
{
 "booking": {
  "_id": "64f90012e4b0a2a12345abcd",
  "userId": "64f8f8f4e4b0a2a12345abcd",
  "type": "flight",
  "itemId": "64f8f9dbe4b0a2a12345abcd",
  "totalPrice": 300,
  "status": "Pending"
 }
}
```


Database Schema

User: _id, name, email, passwordHash, preferences

Flight: _id, airline, from, to, departureDate, arrivalDate, price, seatsAvailable

Hotel: _id, name, location, availableRooms, pricePerNight

Booking: _id, userId, type, itemId, bookingDate, status, totalPrice

Payment: _id, bookingId, userId, amount, status, paymentDate

4. Front-End UI/UX Plan

Wireframes & Navigation Flow

Login / Register → authentication

Dashboard → shows upcoming trips + past bookings + charts

Flights Page → search, filter, book flights

Hotels Page → search, filter, book hotels

Bookings Page → view history & status

Profile Page → update personal info & preferences

Navigation Flow

Login/Register → Dashboard → Flights / Hotels → Booking → Payment → Dashboard

State Management Approach

React useState + useEffect for local state

API calls via Axios for server interactions

JWT stored in localStorage for authenticated requests

Optional: React Context or Redux for global state if project scales

5. Development & Deployment Plan

Team Roles

Frontend Developer: React UI, routing, responsive design

Backend Developer: Node.js APIs, database models, JWT auth, payment simulation

Full-stack / QA: Integration, testing, deployment, seed data

Git Workflow

Feature branches for each module (feature/auth, feature/flights)

Merge to main after review & testing

Use Pull Requests for collaboration

Testing Approach

Backend: Postman / Insomnia for endpoint testing

Frontend: Manual UI testing for flows (login, search, book, pay, dashboard)

Optional: Jest / React Testing Library for component testing

Hosting / Deployment Strategy

Frontend: Vercel for React SPA

Backend: Heroku / Railway / Dockerized Node.js server

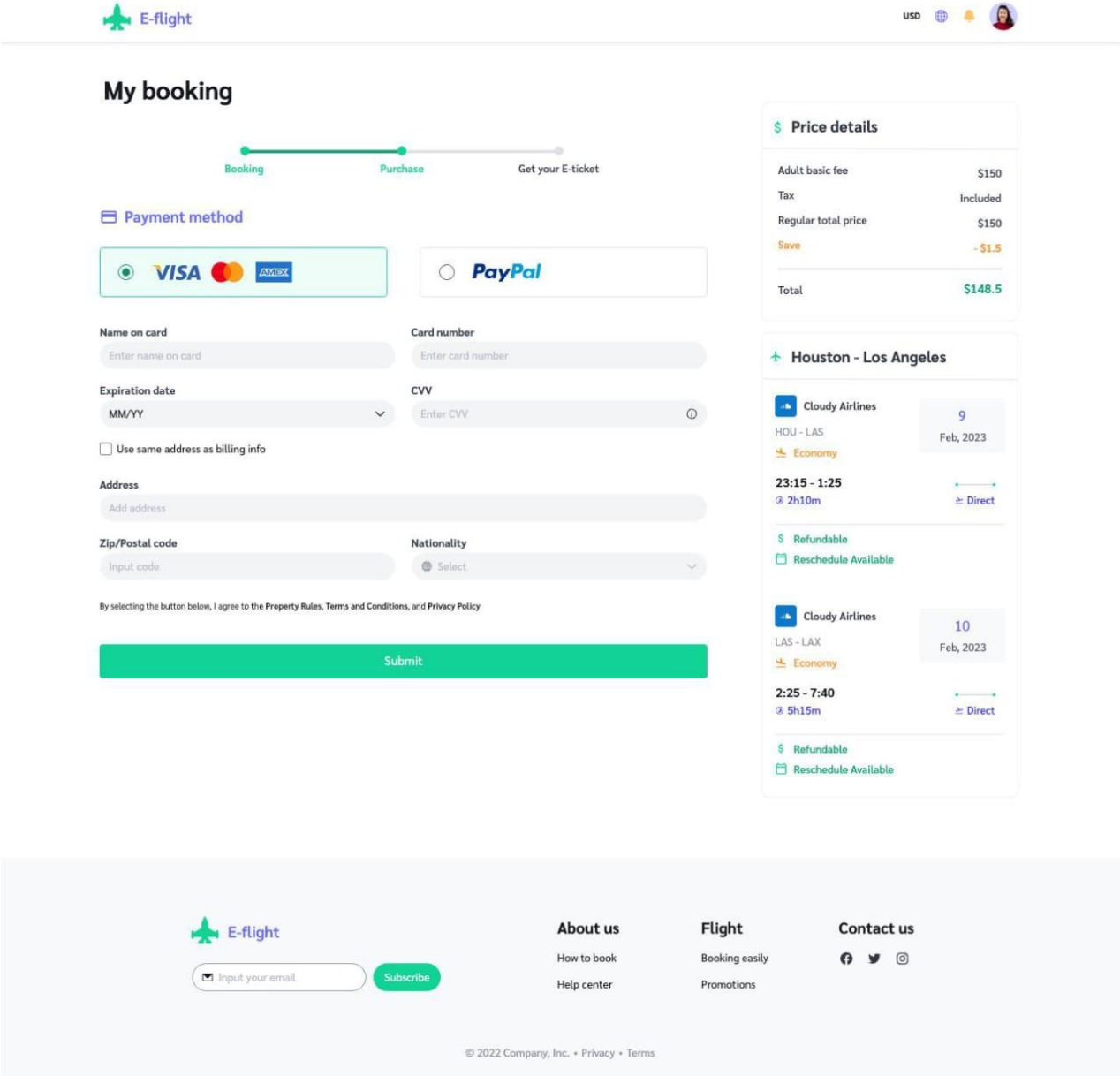
Database: MongoDB Atlas cloud instance

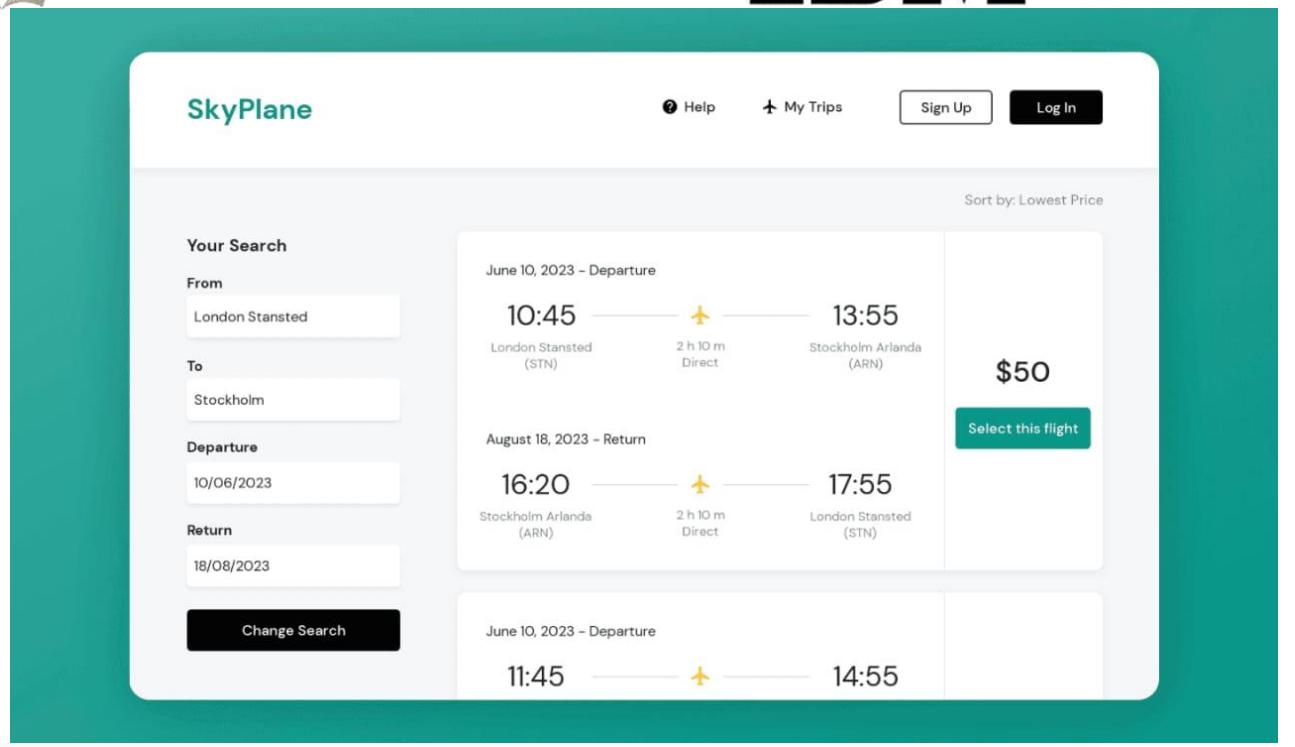
Environment Variables: .env for secrets and API keys

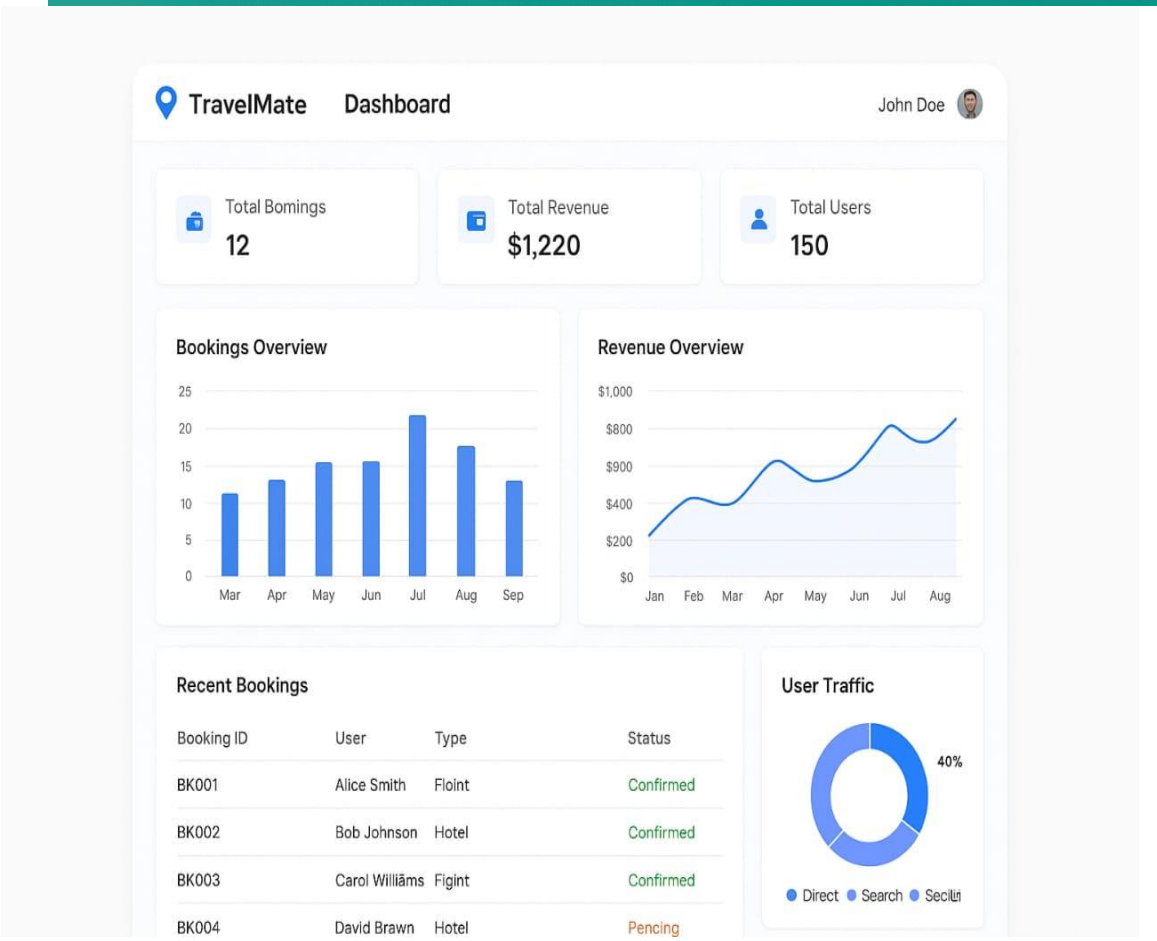
Deployment ensures demo-ready instance accessible to judges

Repository Link: https://github.com/vinithaselladurai/Ticket-booking-system-with-payments/tree/main

SCREENSHOTS:




SkyPlane
Help
My Trips
Sign Up
Log In
Sort by: Lowest Price
Your Search
From
London Stansted
To
Stockholm
Departure
10/06/2023
Return
18/08/2023
Change Search
June 10, 2023 - Departure
10:45
London Stansted (STN)
2 h 10 m Direct
13:55
Stockholm Arlanda (ARN)
$50
Select this flight
August 18, 2023 - Return
16:20
Stockholm Arlanda (ARN)
2 h 10 m Direct
17:55
London Stansted (STN)
June 10, 2023 - Departure
11:45
14:55


TravelMate
Dashboard
John Doe
Total Bomings
12
Total Revenue
$1,220
Total Users
150
Bookings Overview
25
20
15
10
5
0
Mar
Apr
May
Jun
Jul
Aug
Sep
Revenue Overview
$1,000
$800
$900
$400
$200
$0
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Recent Bookings
Booking ID
User
Type
Status
BK001
Alice Smith
Floint
Confirmed
BK002
Bob Johnson
Hotel
Confirmed
BK003
Carol Williāms
Figint
Confirmed
BK004
David Brawn
Hotel
Pencing
User Traffic
40%
Direct
Search