FlightFinder – Navigating Your Air Travel Options

**1. Introduction**

• **Project Title:** FlightFinder – Navigating Your Air Travel Options

• **Team Members:**

**Team Leader : Kiran Ratnala**

**Team member : Ashritha Ratnala**

**Team member : Malla Prudvi Vinay Hanuman Krishna**

**Team member : Manikanta Padala**

**2. Project Overview**

## • Purpose:

FlightFinder is a web-based flight booking system developed using the MERN stack. The purpose of this project is to provide a centralized platform for users to search, compare, and book flights easily and securely. The system aims to simplify travel planning by integrating real-time flight data handling, secure authentication, and payment processing into a single platform.

The project focuses on improving user experience, ensuring secure transactions, and maintaining scalable architecture.

## • Features:

* User Registration & Login
* Secure Authentication (JWT Based)
* Flight Search by Source, Destination & Date
* Flight Filtering (Price, Airline, Timing)
* Booking System
* Secure Payment Integration
* Booking Confirmation with Booking ID
* Booking History Management
* Admin Dashboard for Flight Management

**3. Architecture**

## • Frontend Architecture (React.js)

The frontend is developed using React.js. It follows a component-based architecture.

Main Components:

* Home Page Component
* Login & Registration Component
* Search Flights Component
* Flight Results Component
* Booking Page Component
* Payment Page Component
* User Dashboard Component
* Admin Dashboard Component

React Router is used for navigation between pages. Axios is used for API communication with the backend.

## • Backend Architecture (Node.js & Express.js)

The backend is built using Node.js and Express.js.

Main Backend Modules:

* Authentication Controller
* Flight Controller
* Booking Controller
* Payment Controller
* Admin Controller

The backend follows RESTful API architecture and handles business logic, request validation, and database operations.

Middleware Used:

* JWT Authentication Middleware
* Error Handling Middleware

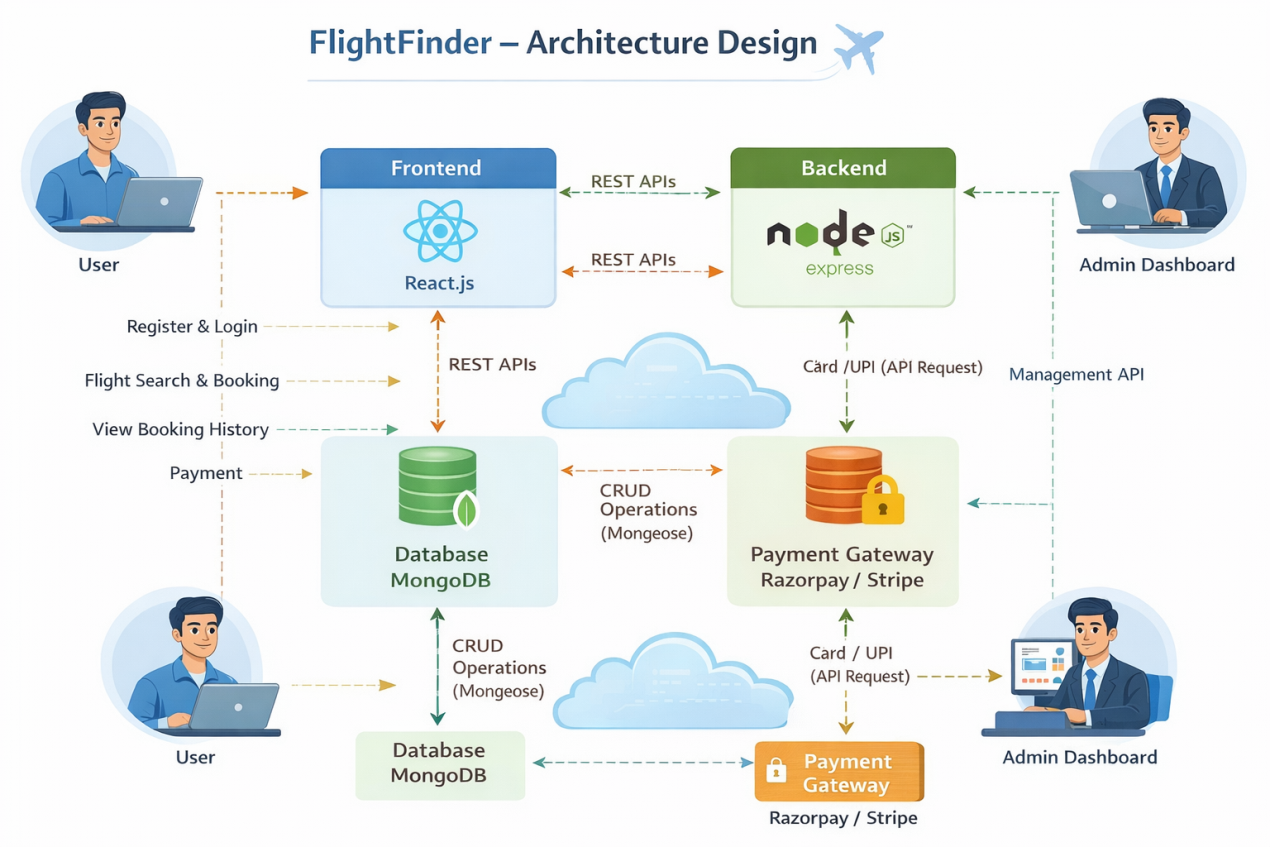
## • Database (MongoDB)

MongoDB is used as the NoSQL database.

Collections:

1. Users Collection
   * userId
   * name
   * email
   * password (hashed)
   * role
2. Flights Collection
   * flightId
   * airline
   * source
   * destination
   * departureTime
   * arrivalTime
   * price
3. Bookings Collection
   * bookingId
   * userId
   * flightId
   * paymentStatus
   * bookingDate

MongoDB Atlas can be used for cloud deployment.



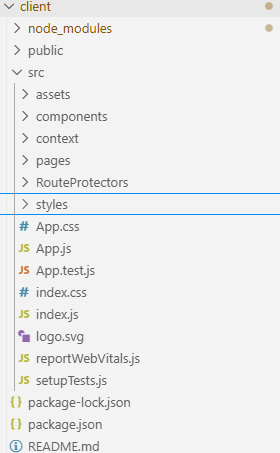
**4. Setup Instructions**

## • Prerequisites:

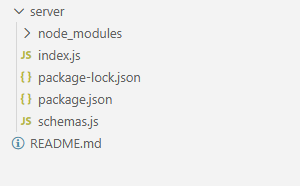
* Node.js installed
* MongoDB installed or MongoDB Atlas account
* npm package manager
* Git

**5. Folder Structure**

Client (React Frontend):



Server:



**6. Running the Application**

## Start Backend:

cd server  
node index.js

## Start Frontend:

cd client  
npm start

Frontend runs on: http://localhost:3000  
Backend runs on: http://localhost:5000

**7. API Documentation**

### 1. User Registration

* Method: POST
* Endpoint: /api/auth/register
* Body:

{  
 "name": "Siri",  
 "email": "siri@gmail.com",  
 "password": "123456"  
}

### 2. User Login

* Method: POST
* Endpoint: /api/auth/login

### 3. Get Flights

* Method: GET
* Endpoint: /api/flights

### 4. Book Flight

* Method: POST
* Endpoint: /api/bookings

**8. Authentication**

Athentication is handled using JWT (JSON Web Token).

Process:

* User logs in
* Server generates JWT token
* Token is stored in client
* Protected routes verify token
* Role-based authorization for admin

Passwords are hashed using bcrypt before storing in database.

**9. User Interface**

The UI is designed to be simple, clean, and responsive.

Pages Include:

* Home Page
* Search Results Page
* Booking Page
* Payment Page
* Dashboard
* Admin Panel

**10. Testing**

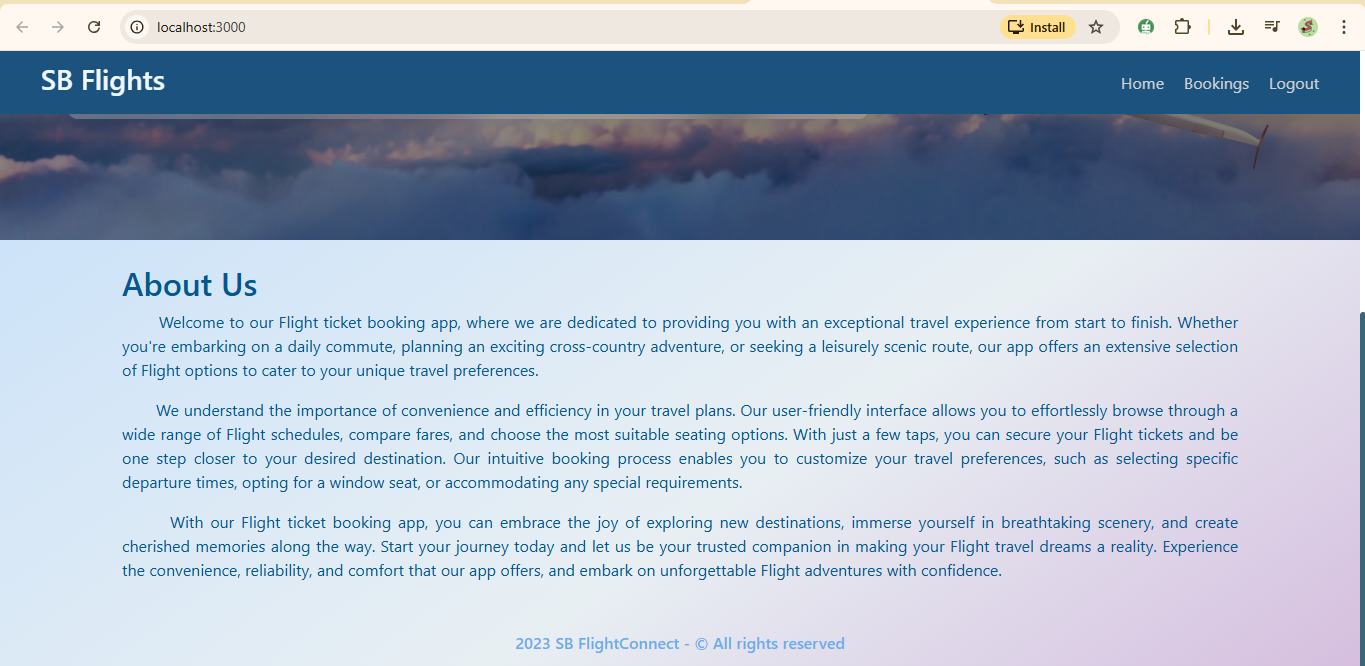
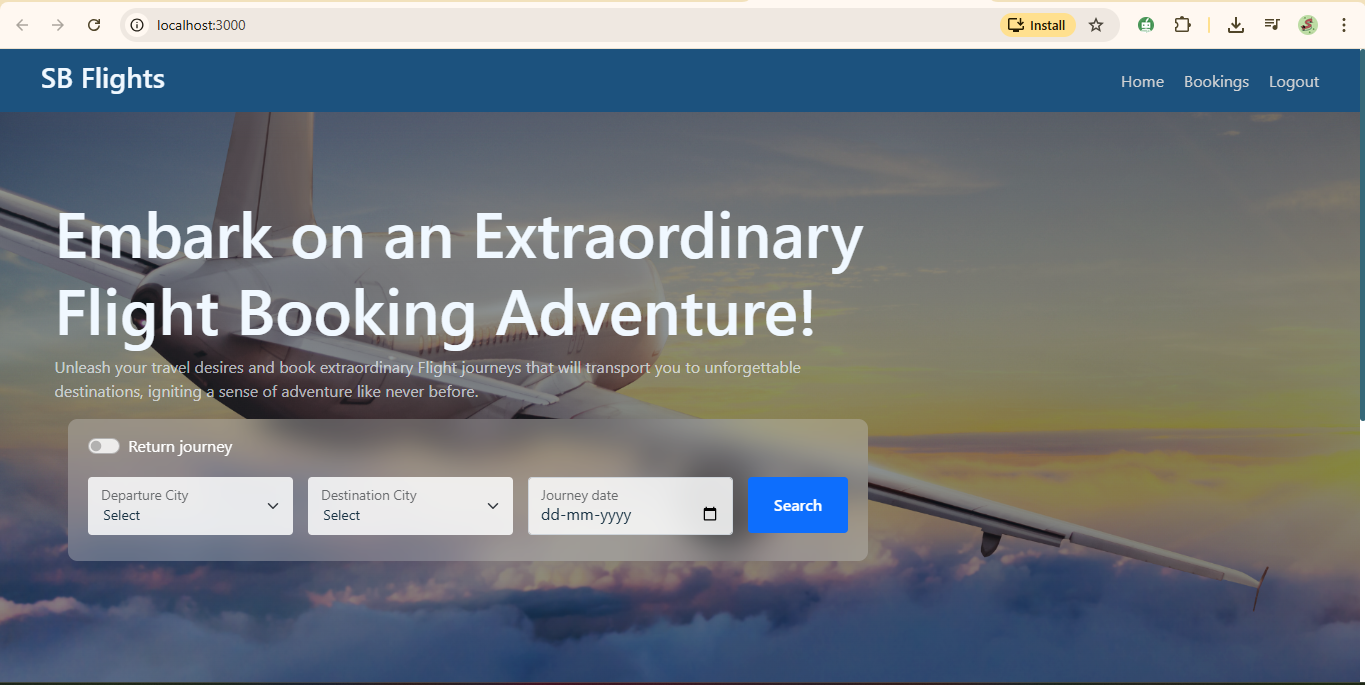
Testing includes:

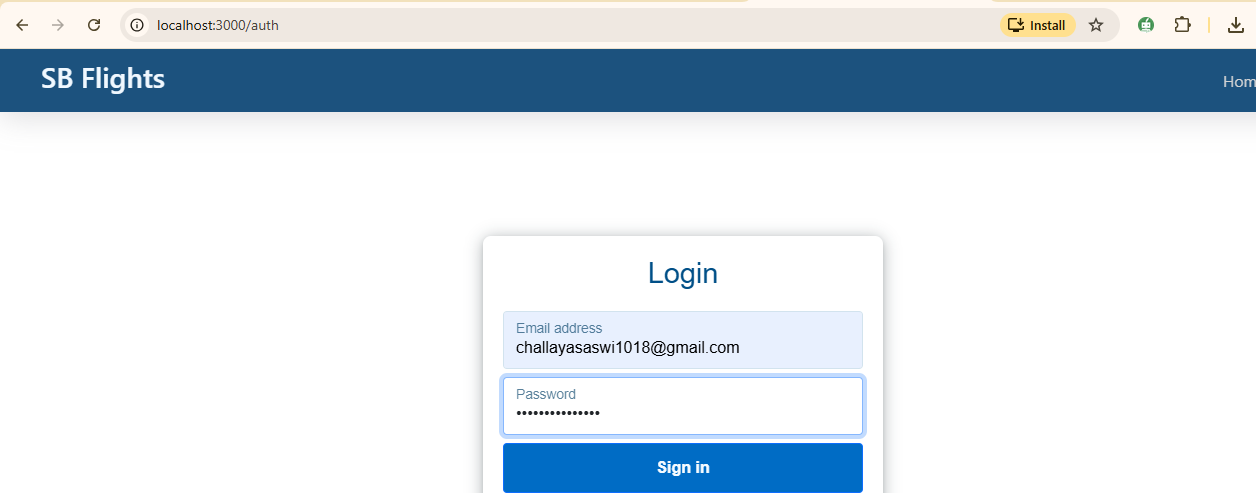
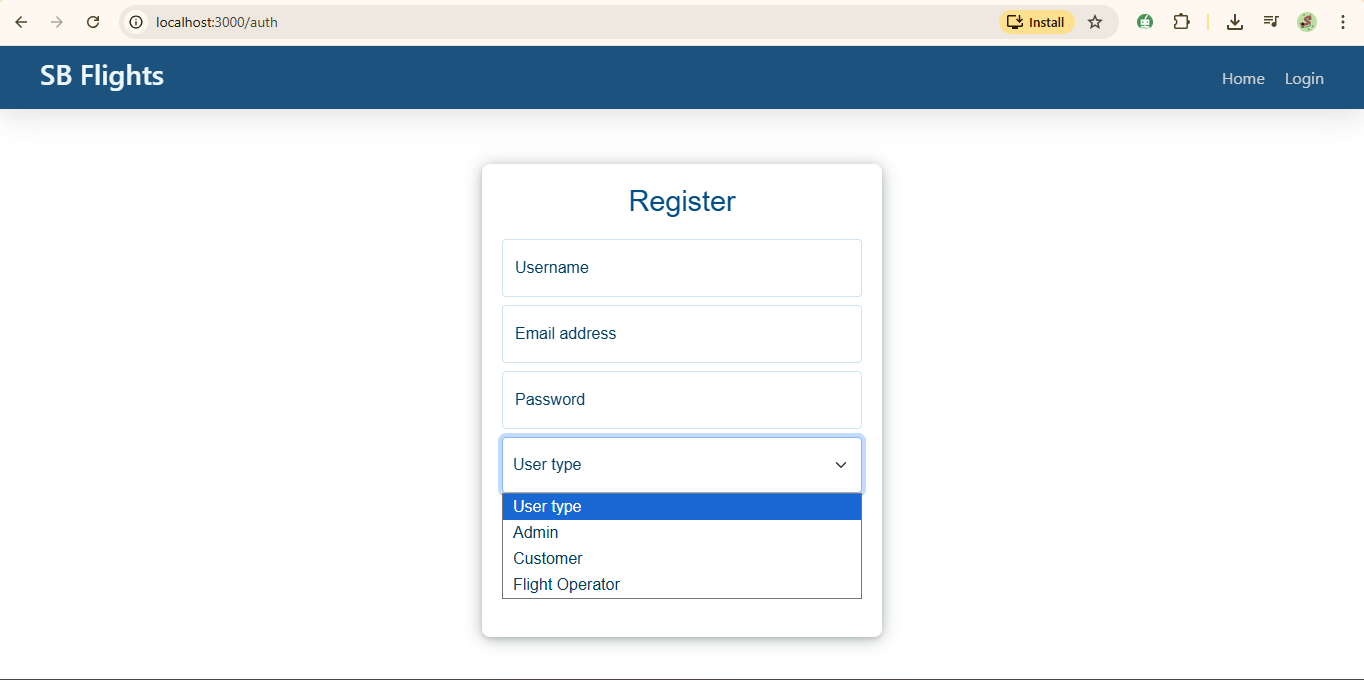
* Functional Testing (All modules)
* Authentication Testing
* API Testing using Postman
* Manual UI Testing
* Performance Testing (Response Time Check)

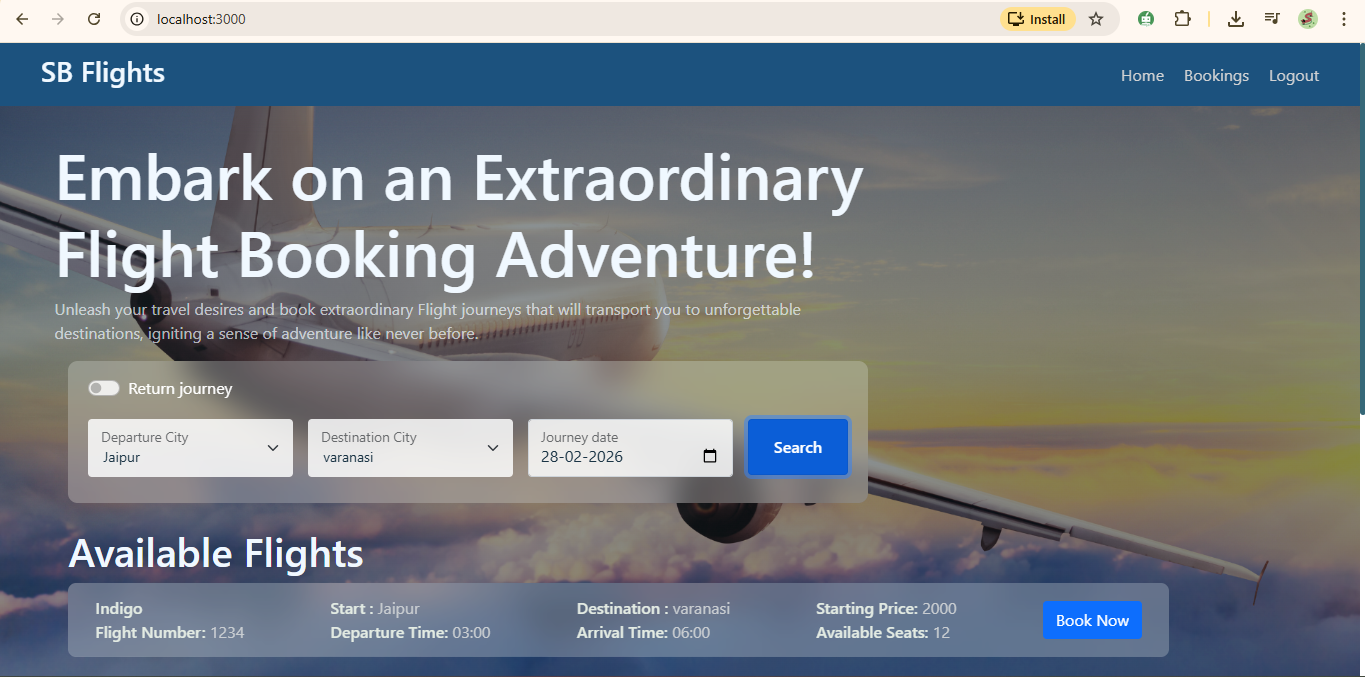
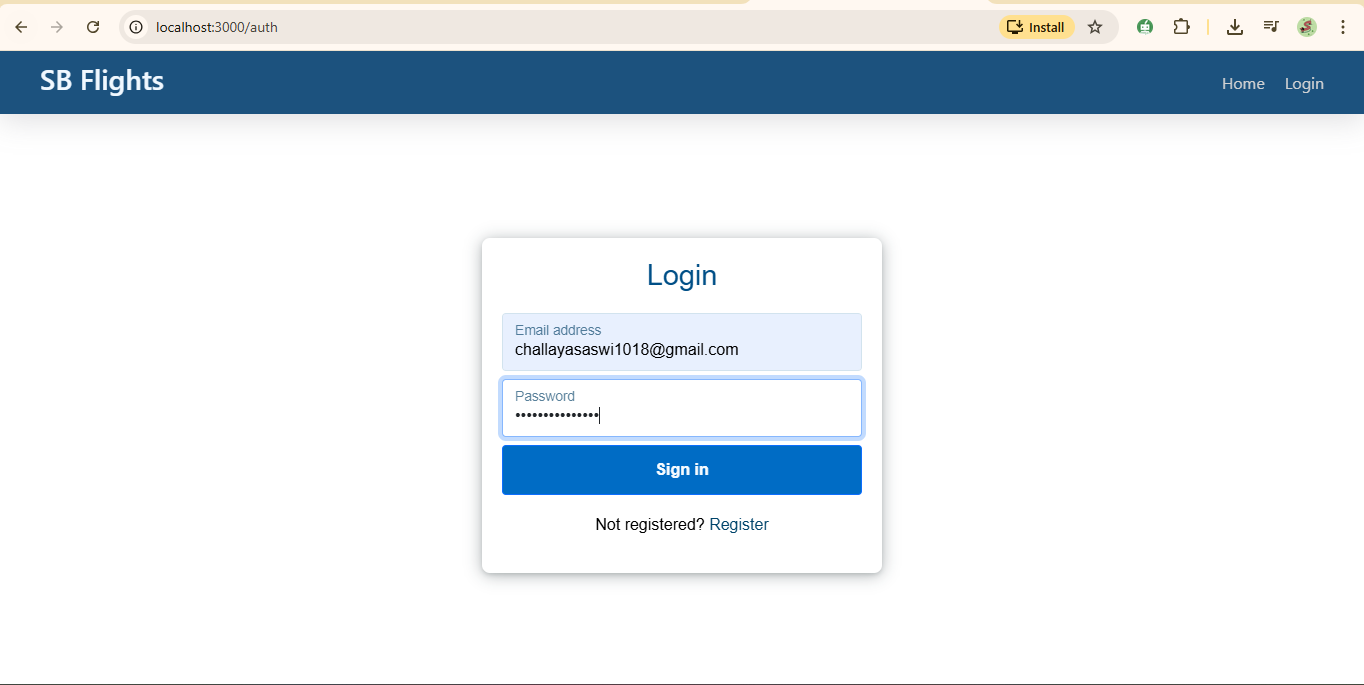
All major functionalities were tested and validated.

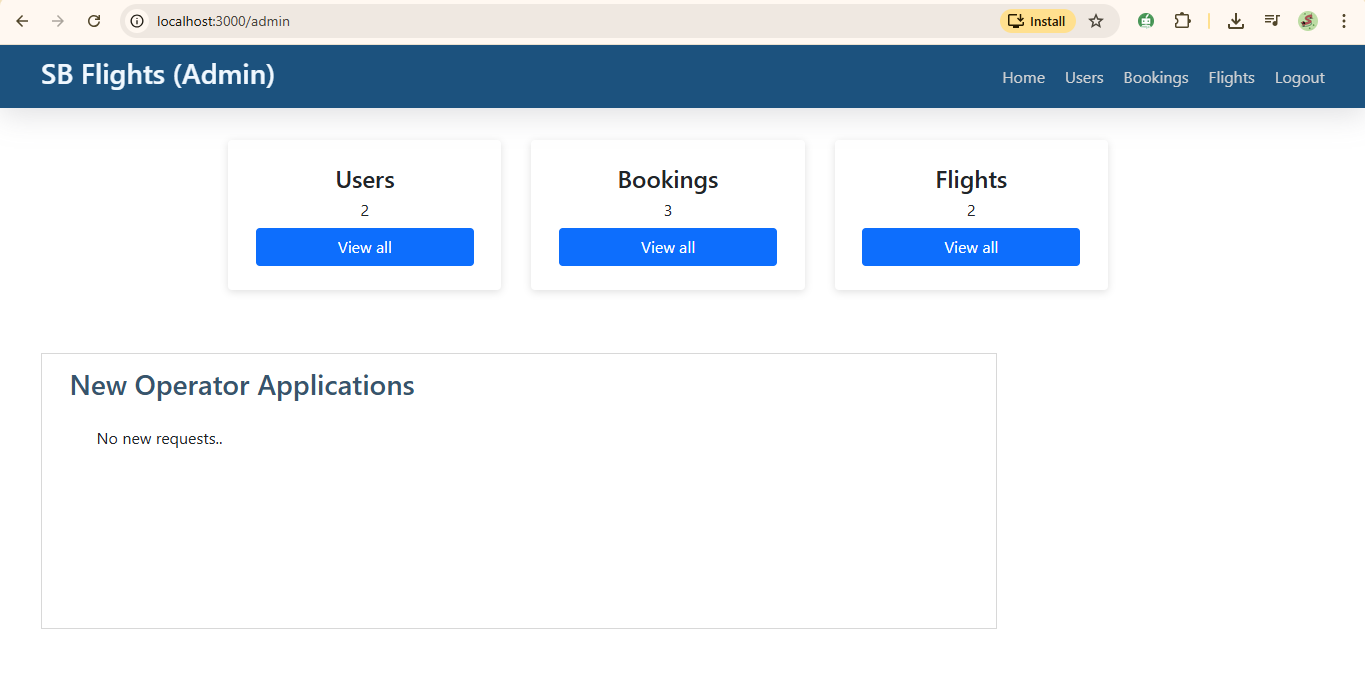
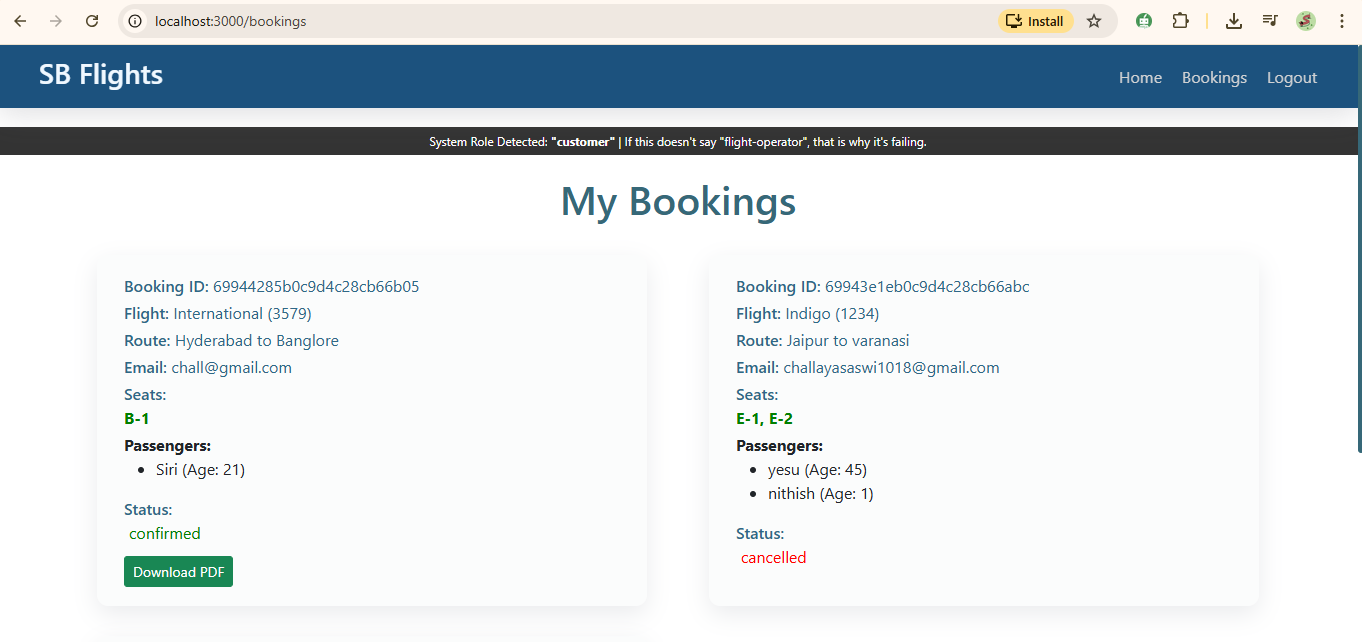
**11. Screenshots or Demo**

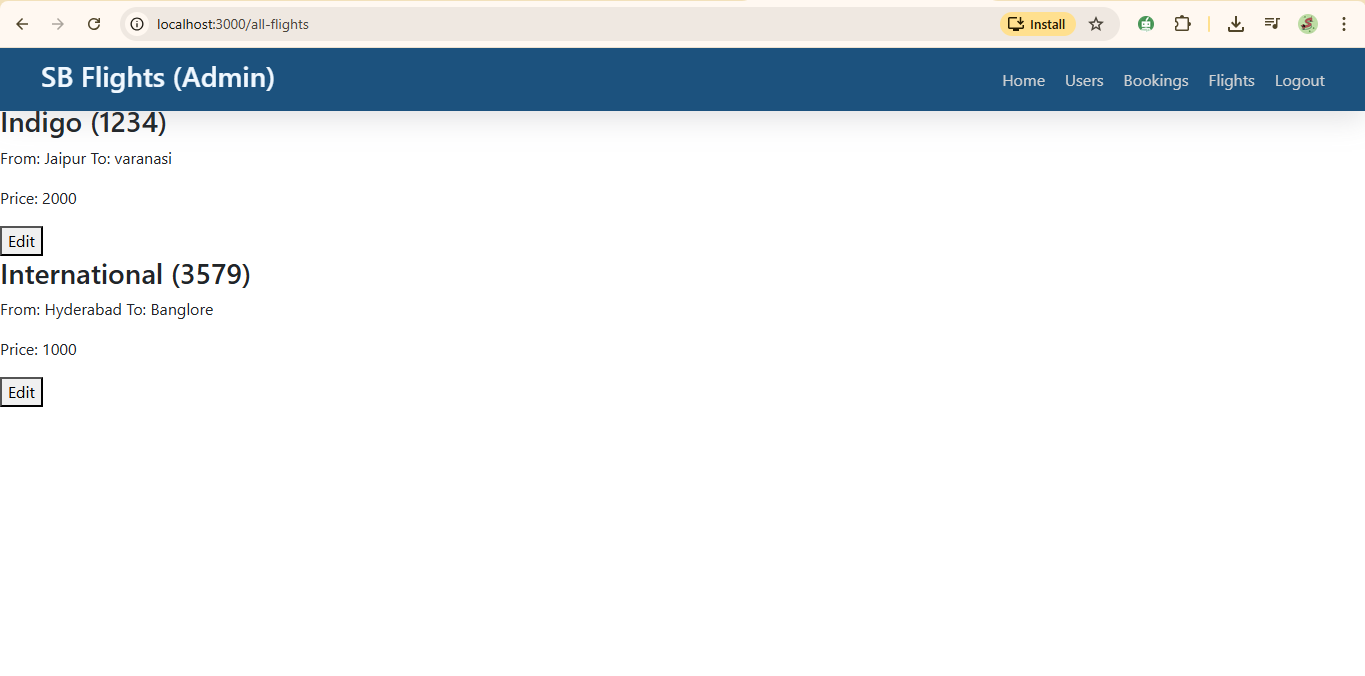
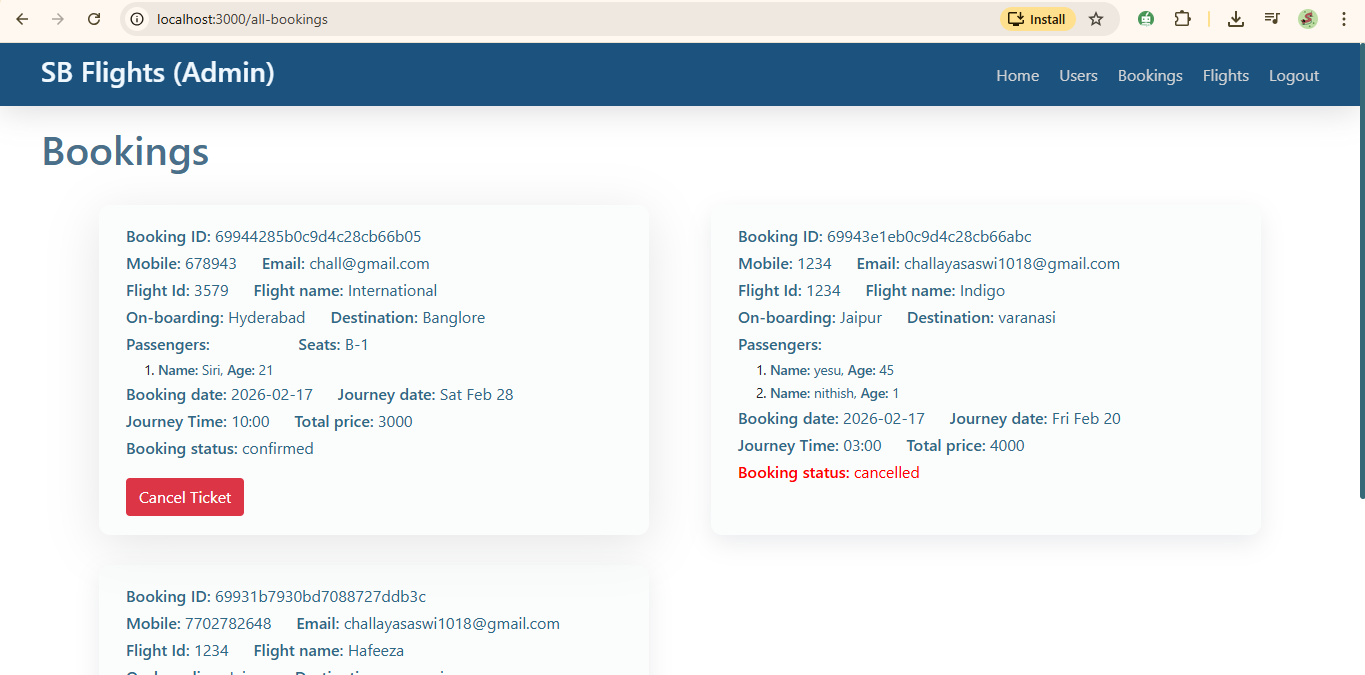
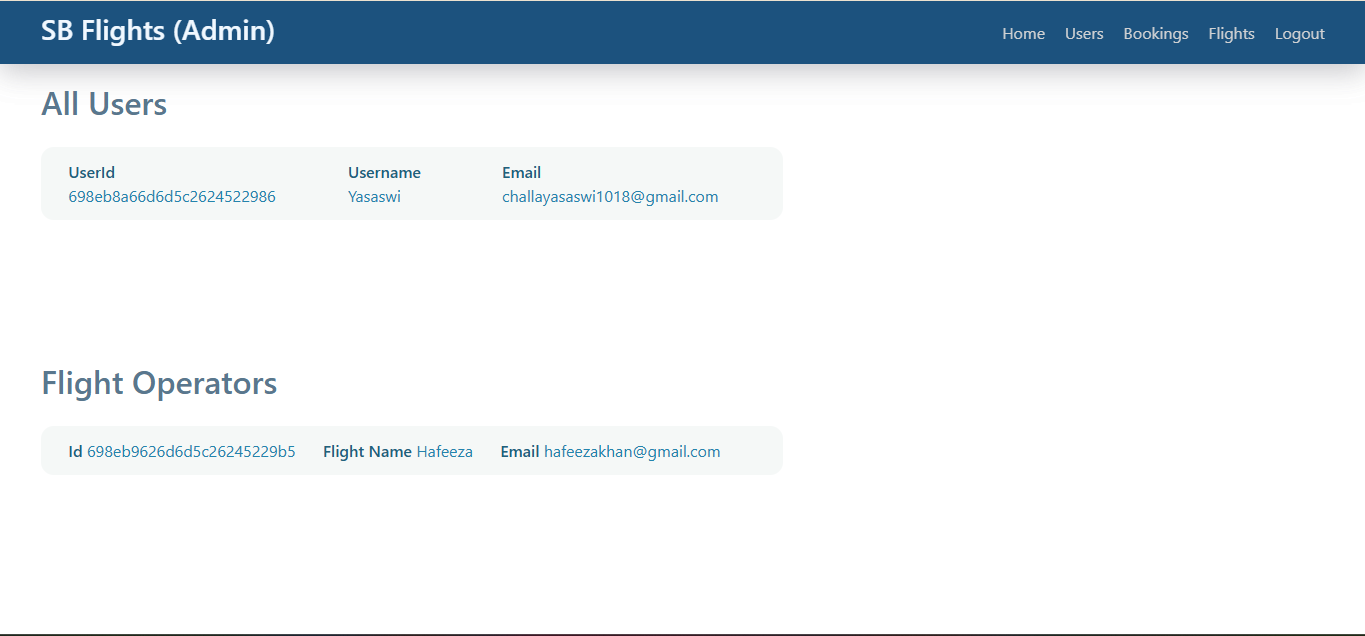
GitHub Repository:  
<https://github.com/challayasaswi/Flightfinder-Navigating-your-Air-Travel-options>

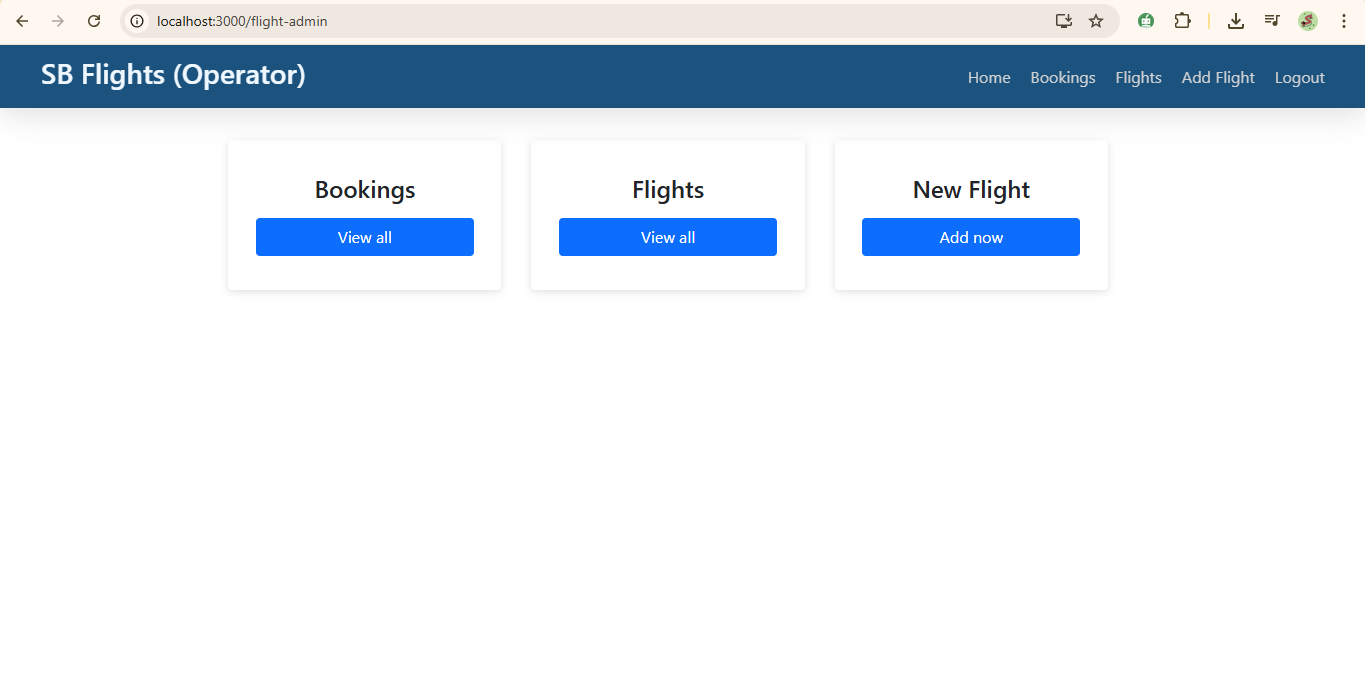
****

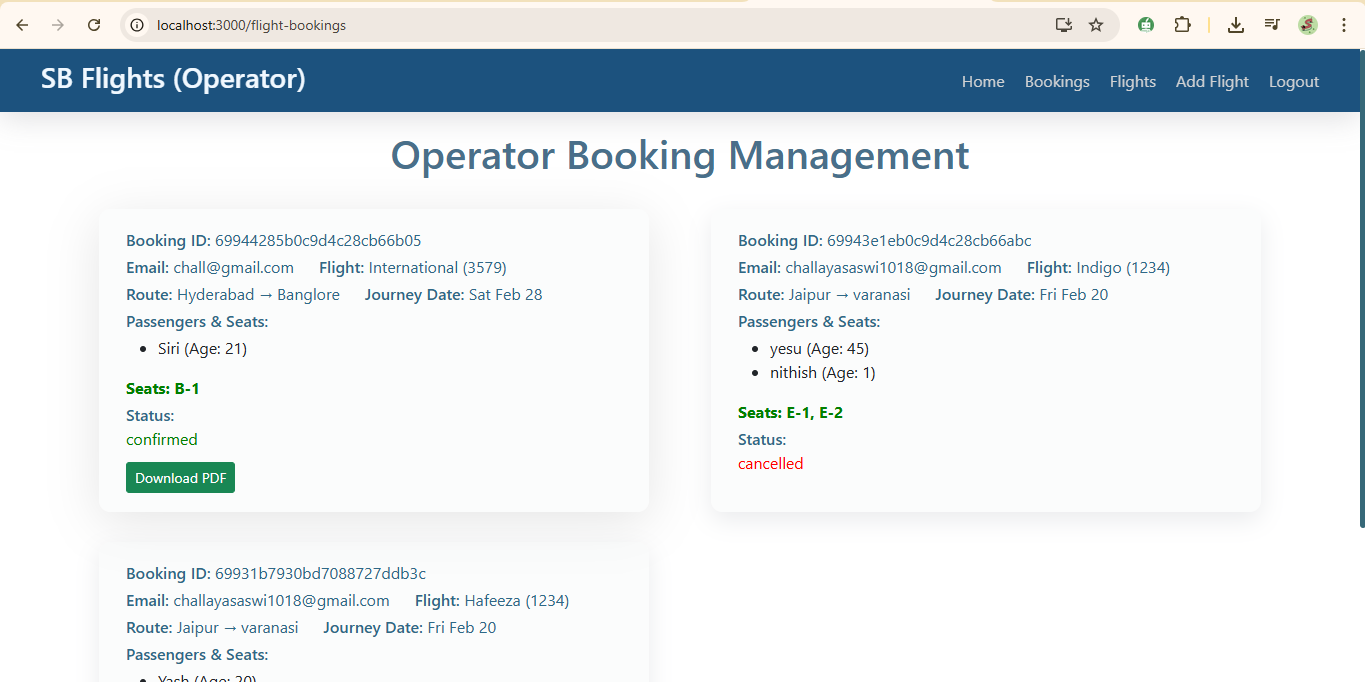
****

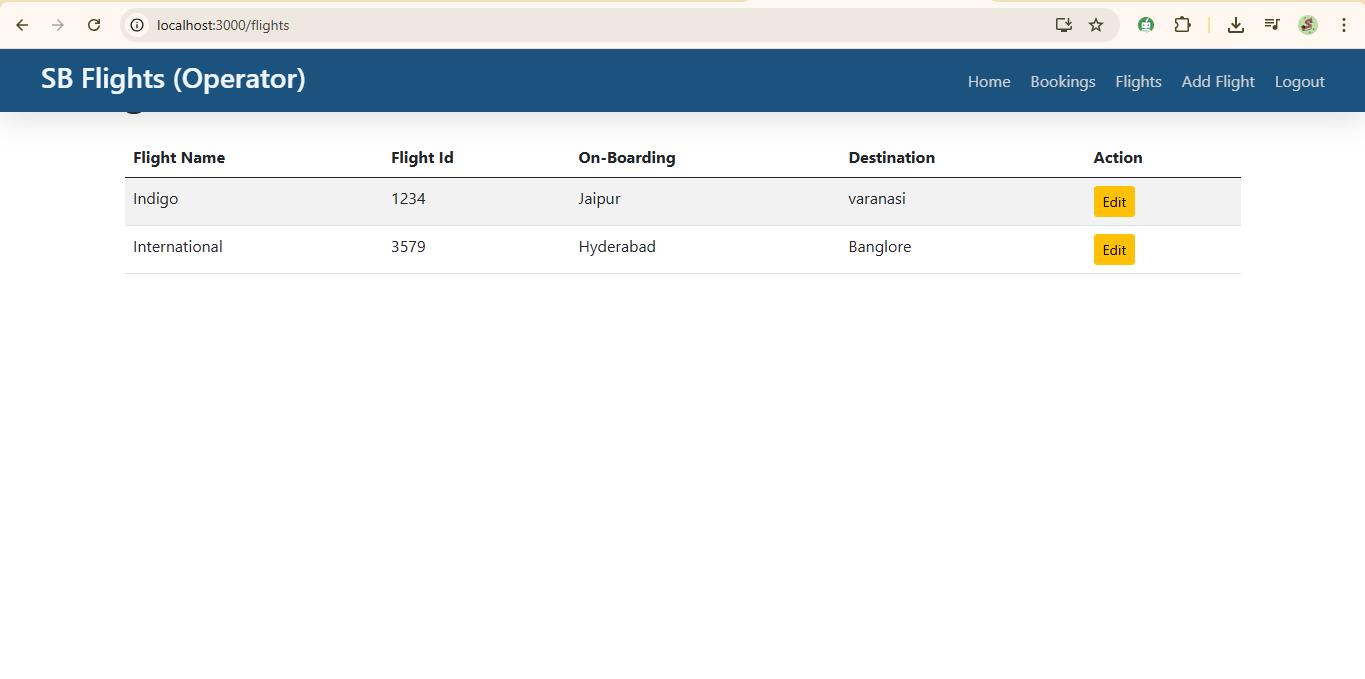
****

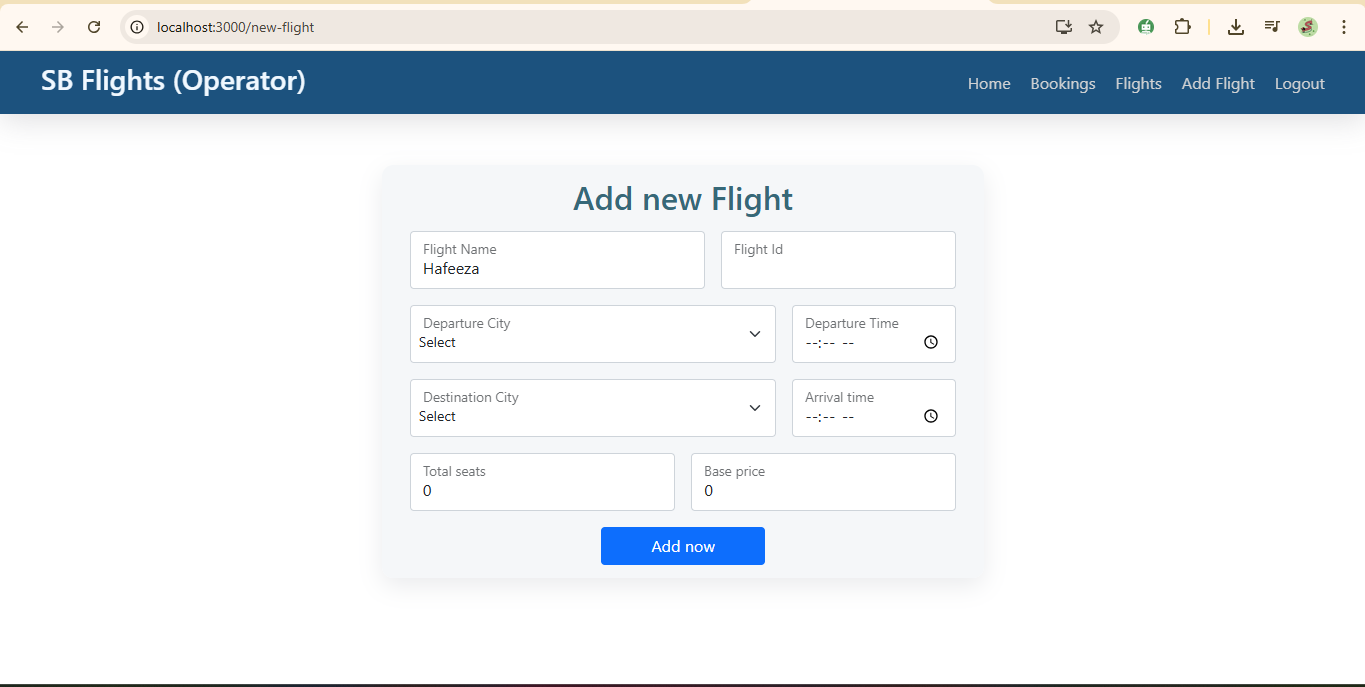
****

****

****

****

****

****

**12. Known Issues**

 Depends on third-party payment API availability

 Limited offline functionality

 Basic UI can be improved further

**13. Future Enhancements**

 AI-Based Flight Recommendations

 Price Alert Notifications

 Mobile Application

 Multi-language Support

 Hotel & Cab Booking Integration

 Advanced Analytics Dashboard