**Full Stack Development with MERN**

**1. Introduction**

**Project Title:** Flight Finder: Navigating Your Air Travel Options

**Team Members:**

B.S. Saniya Begam – Frontend Developer

Shabeena Nusrath A – Backend Developer

Devangam Pavithra – Database and API Integration

**2. Project Overview**

Purpose:

The purpose of the “Flight Finder” project is to simplify the flight search experience by providing users with real-time air travel options based on their preferences like source, destination, dates, and budget.

Features:

Search flights by source, destination, and date

Sort and filter by price, airline, and duration

Authentication for user login/register

View detailed flight information

Responsive and user-friendly UI

**3. Architecture**

Frontend:

Developed using React.js, the frontend handles user interaction, search inputs, and displays results in an intuitive manner.

Backend:

Built with Node.js and Express.js, the backend handles API requests, authentication, and business logic for retrieving and filtering flight data.

Database:

MongoDB is used to store user data, saved searches, and flight listings. Mongoose is used to define schema and interact with the database.

**4. Setup Instructions**

Prerequisites:

Node.js

MongoDB

Git

Installation:

1. Clone the repository

git clone <repository-url>

2. Navigate to client and install dependencies

cd client

npm install

3. Navigate to server and install dependencies

cd server

npm install

4. Create .env files and add necessary environment variables for MongoDB URI and JWT secret.

**5. Folder Structure**

Client:

/src

/components – Reusable components like SearchForm, FlightCard

/pages – Pages like Home, Login, Results

/services – API interaction logic

App.js – Main routing component

Server:

/routes – Contains route handlers for flights and authentication

/controllers – Business logic for each route

/models – Mongoose schemas

server.js – Entry point to the backend

**6. Running the Application**

Frontend:

cd client

npm start

Backend:

cd server

npm start

**7. API Documentation**

GET /api/flights?source=XXX&destination=YYY&date=ZZZ

Returns flight options matching criteria.

POST /api/register

Registers a new user.

POST /api/login

Authenticates a user and returns a JWT token.

GET /api/user/flights

Retrieves saved searches for the authenticated user.

**8. Authentication**

JWT (JSON Web Token) is used for secure authentication.

Tokens are stored in browser local storage.

Routes like saved flights are protected using middleware that verifies the JWT.

**9. User Interface :**

Screens include:

Search Page

Results Page

Login/Register Page

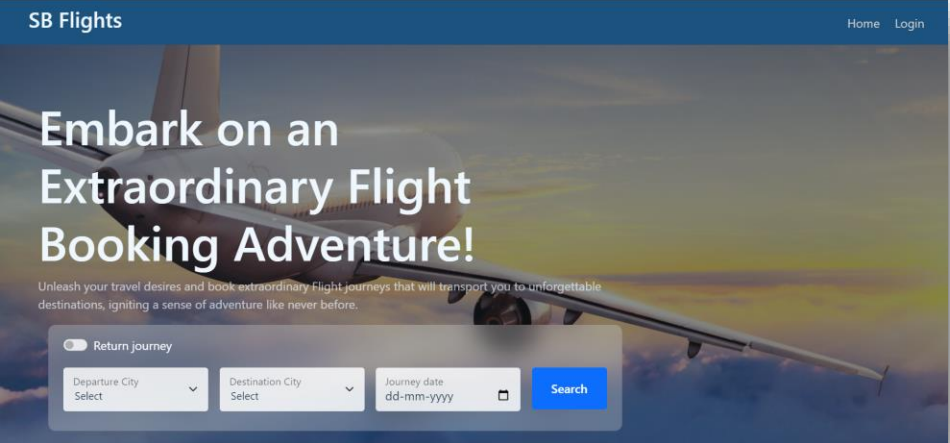
Flight Details Modal

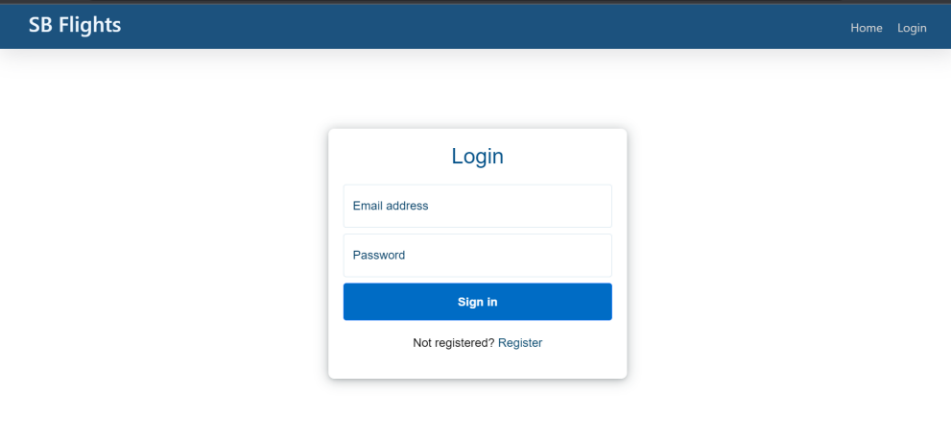
**10. Testing**

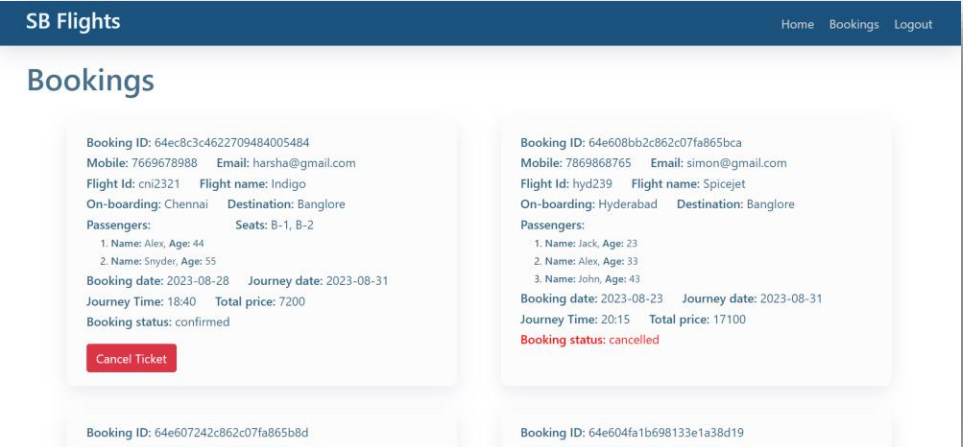
Manual testing was conducted on all major features.

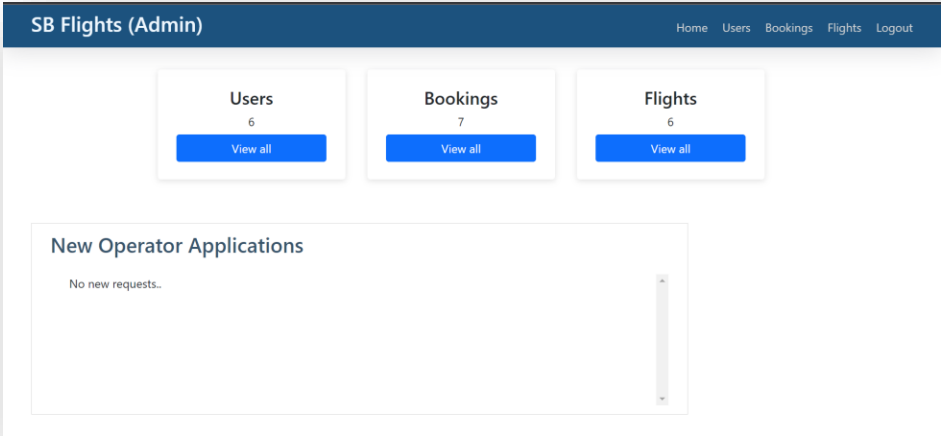
Unit tests can be added using Jest for backend and React Testing Library for frontend in future updates.

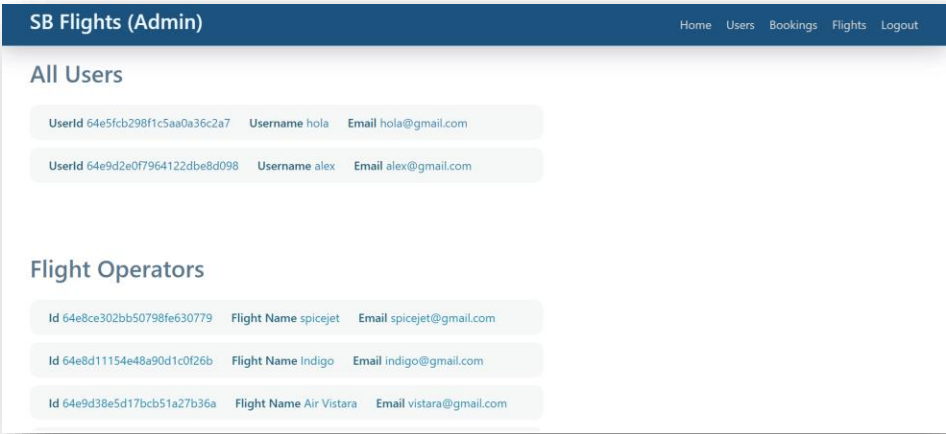
**11. Screenshots or Demo**

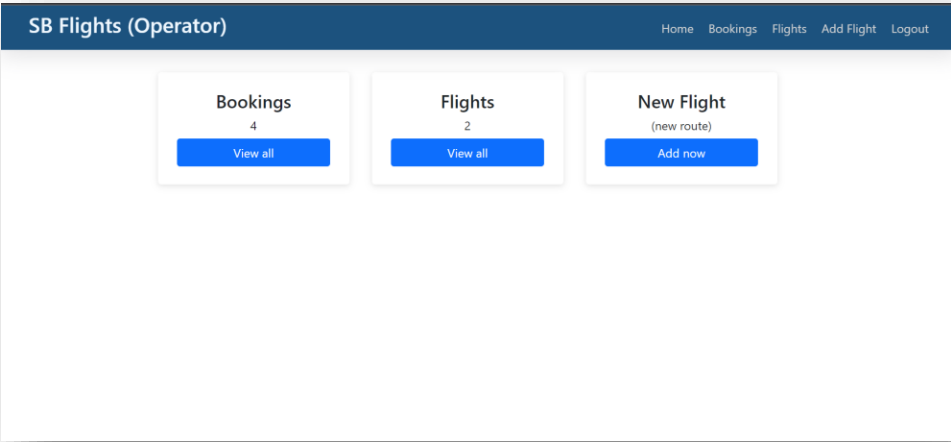
****

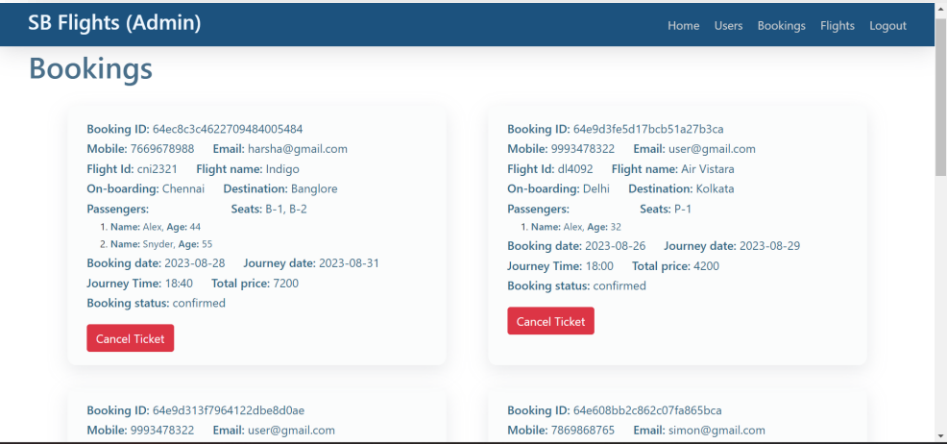
****

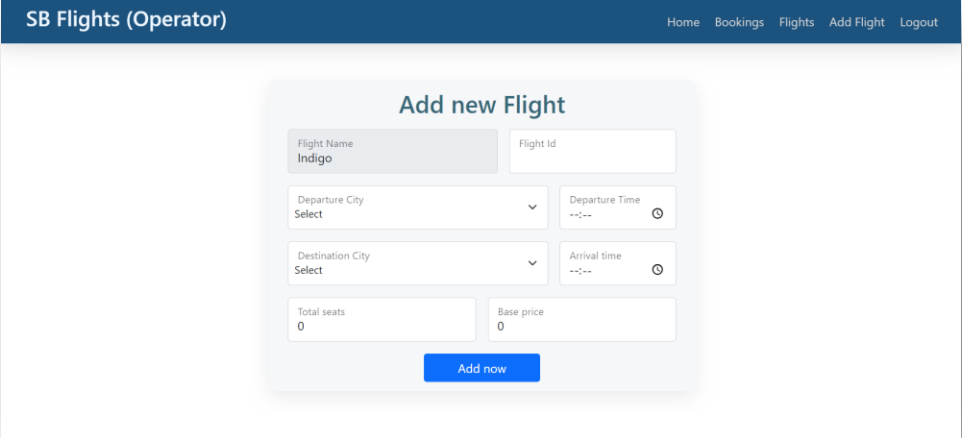
****

****

****

****

****

****

**The demo of the app is available at:**

<https://drive.google.com/file/d/1Q0XwKtAz7EkaKNJv3_gbo6mZE9nfuBTK/view?usp=sharing>

**12. Known Issues**

Real-time price updates are not yet integrated.

No payment gateway or booking system currently implemented.

**13. Future Enhancements**

Add flight booking and payment integration

Implement notification for flight price drops

Add international flights and multi-city support