Project Report: FlightFinder – Navigating Your Air Travel Options

INTRODUCTION:

1. Project Overview

FlightFinder is a full-stack web application designed to streamline and enhance the flight booking experience. Built with the MERN stack (MongoDB, Express.js, React.js, and Node.js), it allows users to search, book, and manage flight tickets conveniently, while providing admin functionalities for managing flights and bookings.

Whether you're a frequent flyer or a casual traveller, FlightFinder ensures a seamless, fast, and secure way to plan your air travel.

---

2. Project Objective

- Simplify the flight booking process
- Provide real-time search and filter options
- Ensure secure and smooth user authentication
- Enable admin-level control over flights and bookings
- Create an intuitive and user-friendly UI

---

3. Key Features

For Users

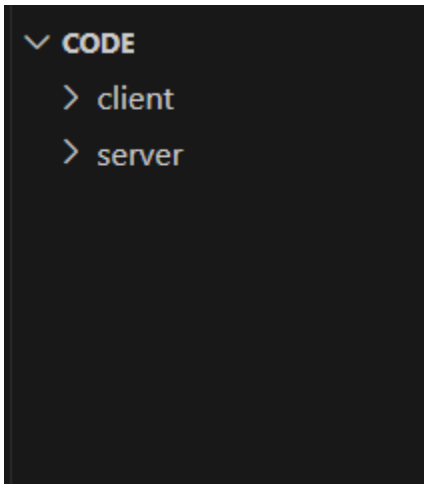
- Account registration and login
- Search flights by destination, date, time, and class
- Filter options: direct flights, preferred airline, etc.
- Book flights with real-time seat selection
- Secure payment simulation (no actual payment gateway)
- View, manage, and cancel bookings

For Admin

- Login with dedicated credentials
- Add, edit, and delete flights
- Monitor all bookings and users
- Dashboard for flight and booking analytics

---

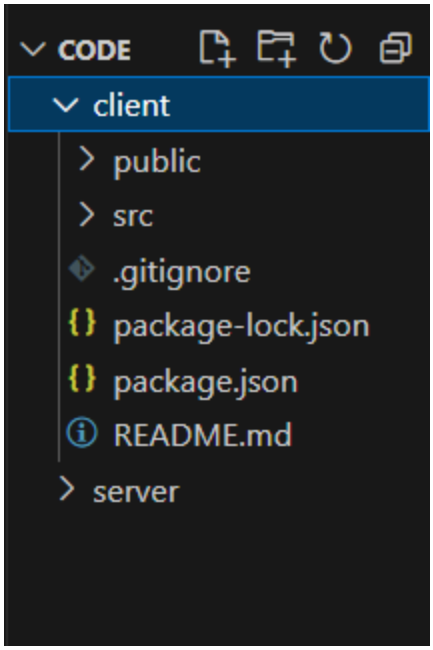
4. Technical Architecture

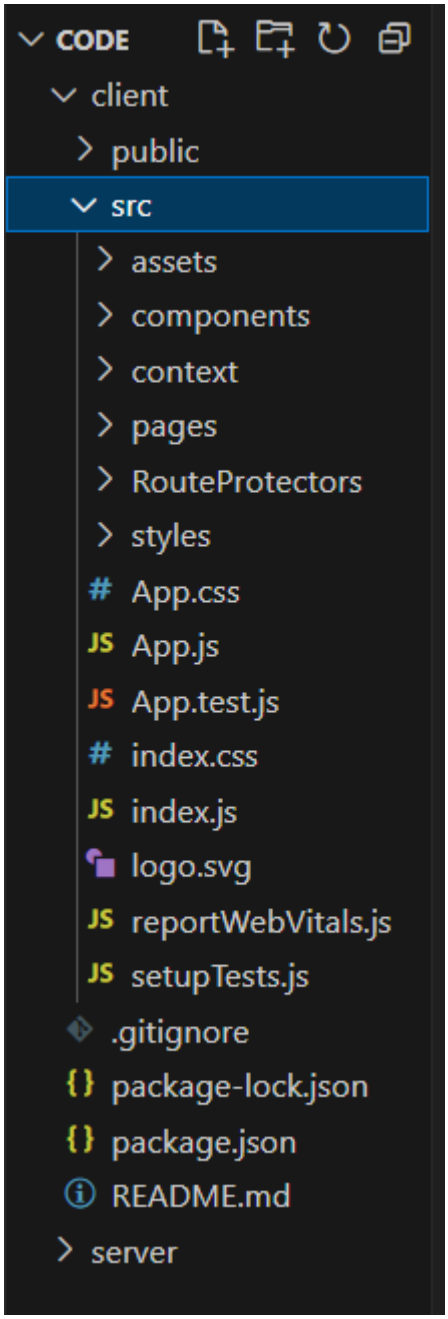


Frontend (React)

|-- User Interface (Login/Register, Search, Booking UI)

|-- Axios API Calls


CODE
client
public
src
.gitignore
package-lock.json
package.json
README.md
server

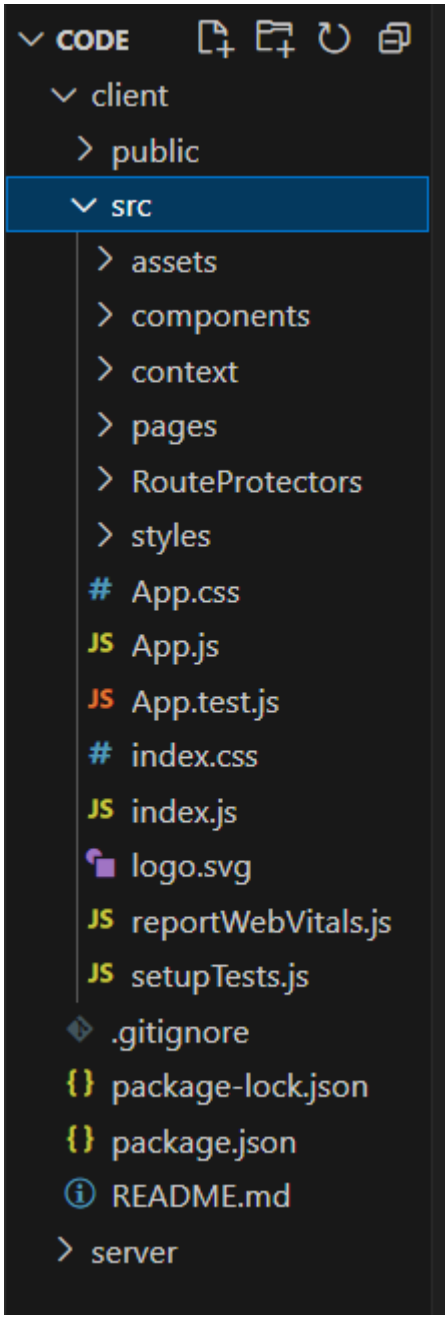


Backend (Node.js + Express)

|-- Routes: /users, /flights, /bookings, /admin

|-- Middleware: Authentication & Authorization

|-- Controllers: Business Logic for CRUD



Database (MongoDB)

|-- Collections: Users, Flights, Bookings

---

5. Scenario: Real-World Example

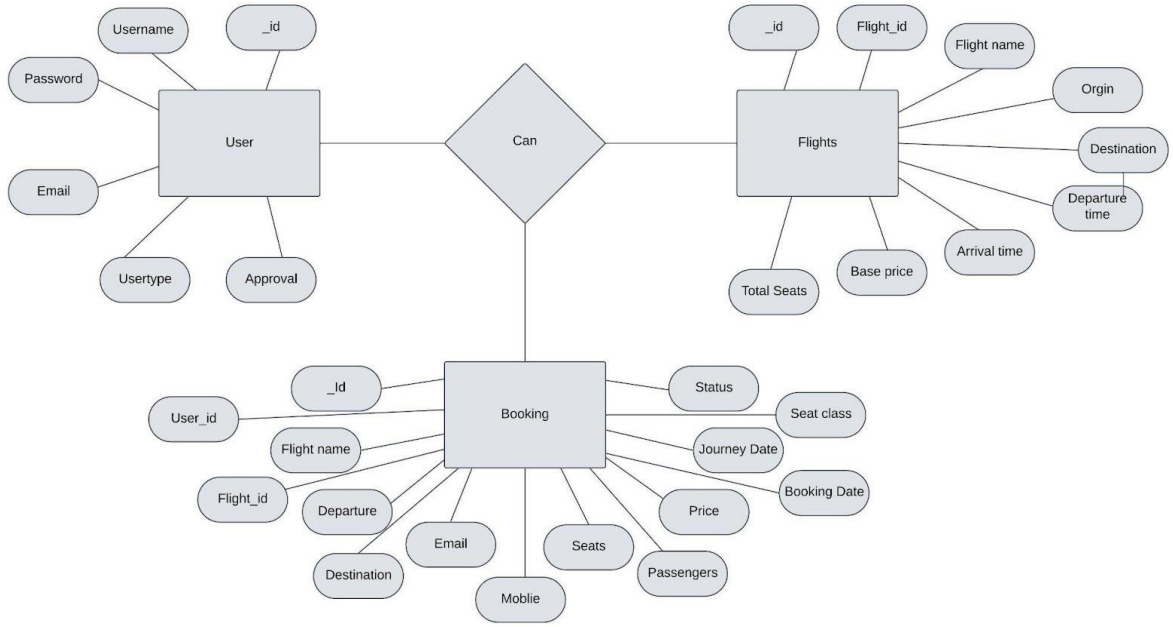
User:

- Books a business class flight from NYC to Paris
- Uses filters for direct flights and preferred airline
- Chooses a window seat with extra legroom
- Pays securely and receives booking confirmation instantly

This showcases the convenience and real-time assistance FlightFinder provides.

---

6. ER Diagram Summary



- **User:** Can make multiple bookings and payments
- **Booking:** Linked to one user and one flight
- **Flight:** Contains flight details, seat availability
- **Admin:** Controls backend activities including flight and booking management

---

7. Prerequisites for Development

- Node.js & npm
- MongoDB (local or Atlas)
- React.js

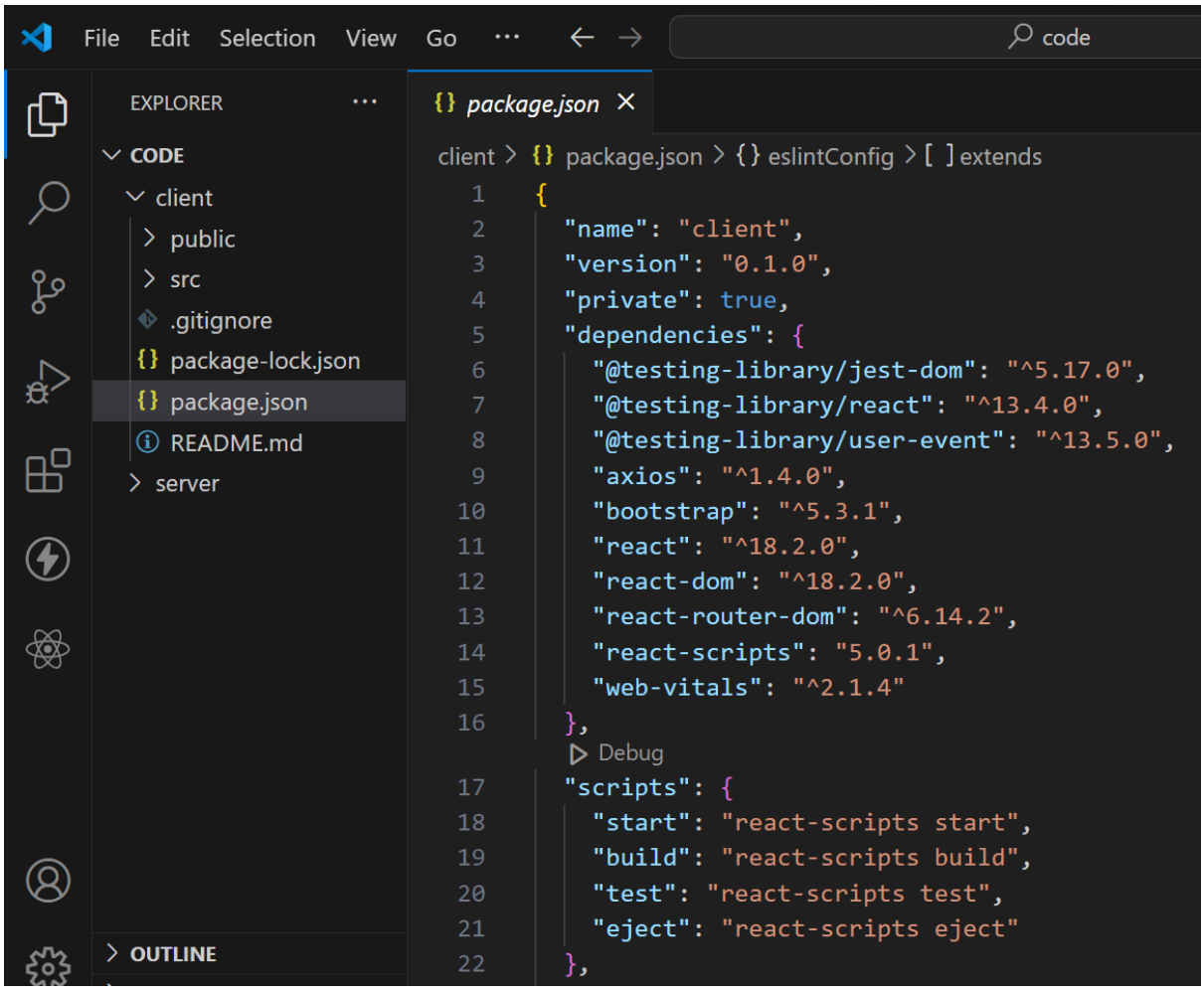
- Express.js
- Git & GitHub
- Visual Studio Code

Useful Links:

- Node.js: nodejs.org
- MongoDB Atlas: mongodb.com
- React: reactjs.org
- Git: [git-scm.com](git-scm.com)

---

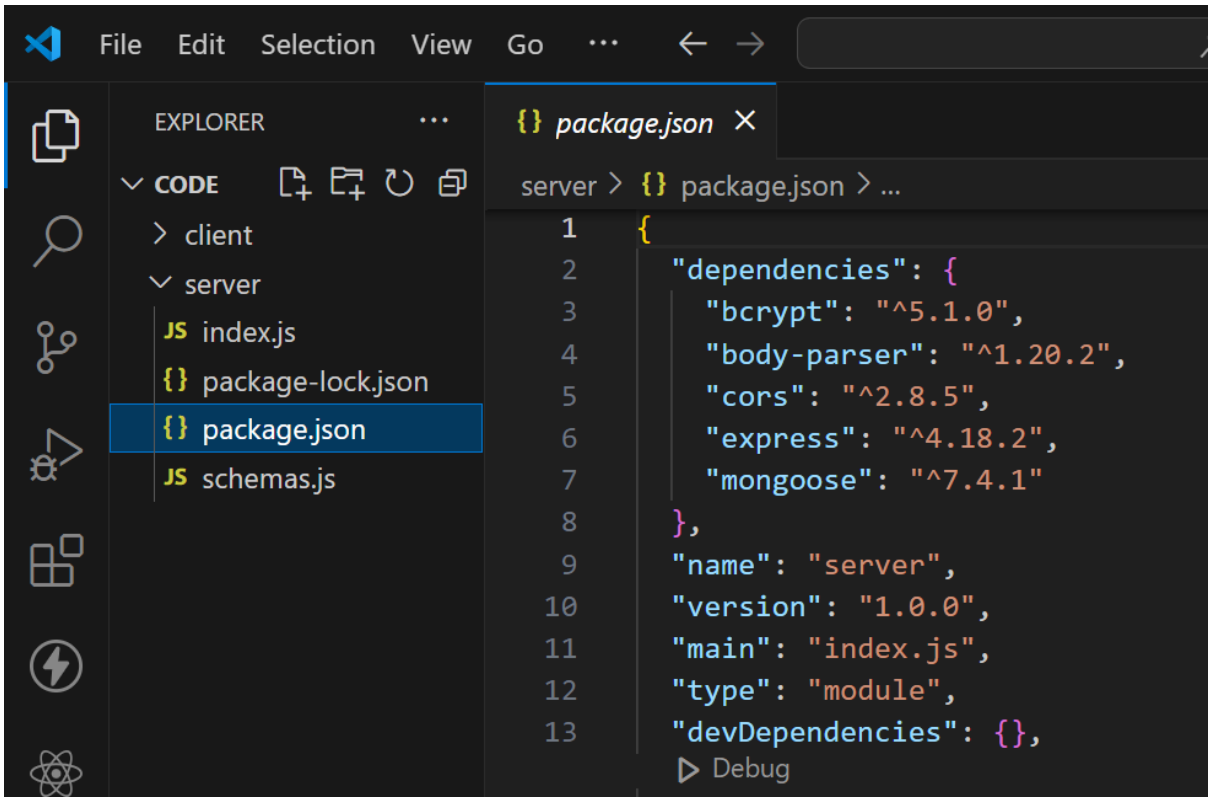
8. Project Setup



Clone the Repository

git clone https://github.com/harsha-vardhan-reddy-07/Flight-Booking-App-MERN

cd Flight-Booking-App-MERN



Install Dependencies

npm install

Start Development Server

npm run dev

Access

Visit: http://localhost:3000

---

9. Project Structure

Client Folder

- Built with React.js and Bootstrap
- Contains components for login, registration, search, booking, dashboard

Server Folder

- Built with Node.js and Express
- Contains API routes, models (Mongoose schemas), controllers

---

10. Application Flow

User Flow

1. Register/Login
2. Search Flights
3. Apply Filters
4. Book Seat
5. Confirm Booking
6. Cancel Booking (if needed)

Admin Flow

1. Login
2. Add New Flights
3. View All Flights & Bookings
4. Manage User Activity

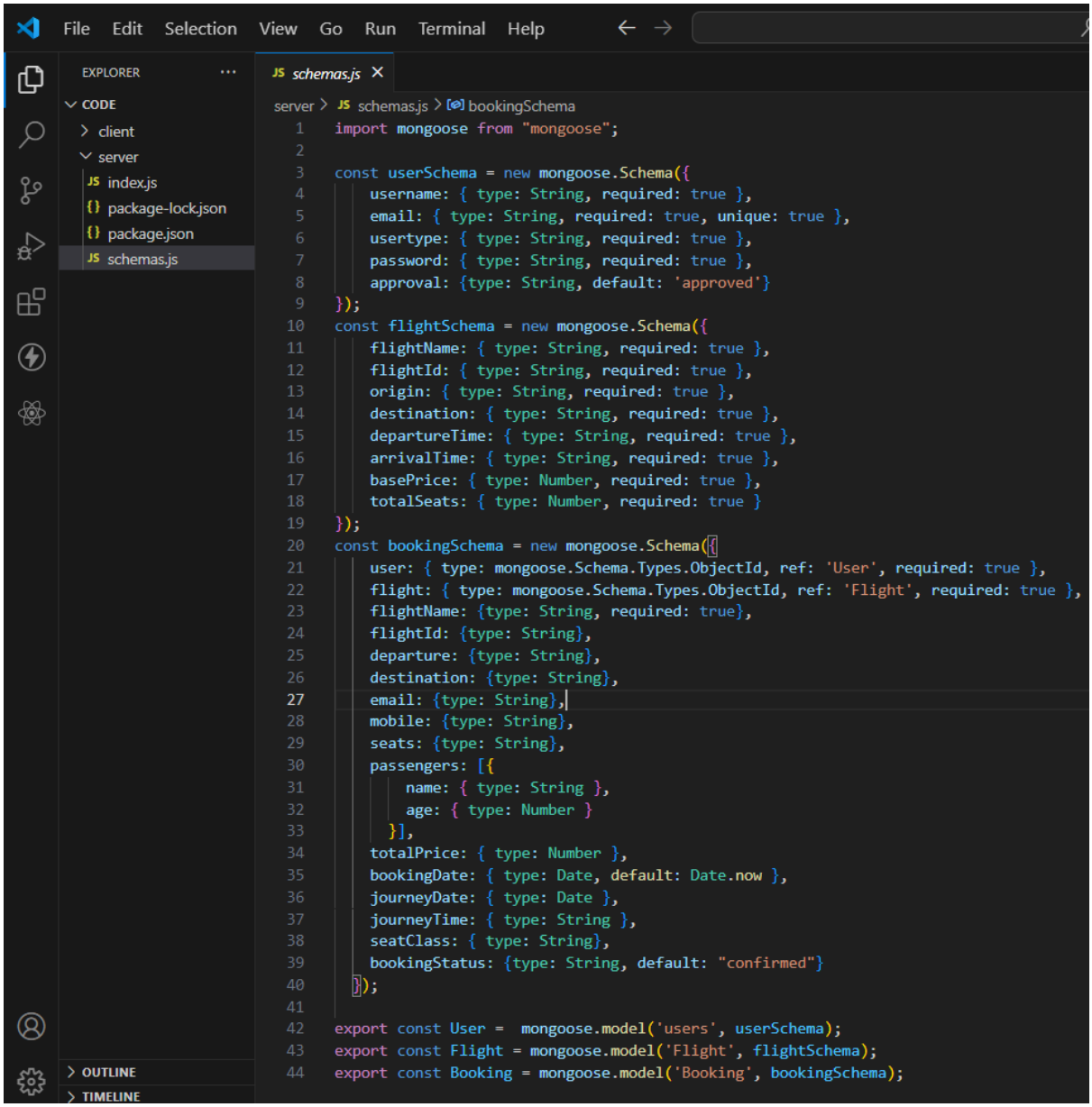
---

11. Backend Development Highlights

- MongoDB connected using Mongoose
- API endpoints: /api/flights, /api/bookings, /api/users
- User roles: User, Admin, Airline Operator
- Bcrypt used for password encryption
- Middleware handles route protection
- Error handling for all major APIs

---

12. Database Design



Schemas

- User: name, email, password, role
- Flight: source, destination, date, time, class, seat count
- Booking: userID, flightID, seat number, status

Database Connection

```
const PORT = process.env.PORT || 6001;
mongoose.connect(process.env.MONGO_URL, {
    useNewUrlParser: true,
    useUnifiedTopology: true
}).then(()=>{


    server.listen(PORT, ()=>{
        console.log(`Running @ ${PORT}`);
    });


}).catch((err)=>{
    console.log("Error: ", err);
})
```

- Connected via MongoDB Atlas or Compass
- Handles CRUD operations on all collections

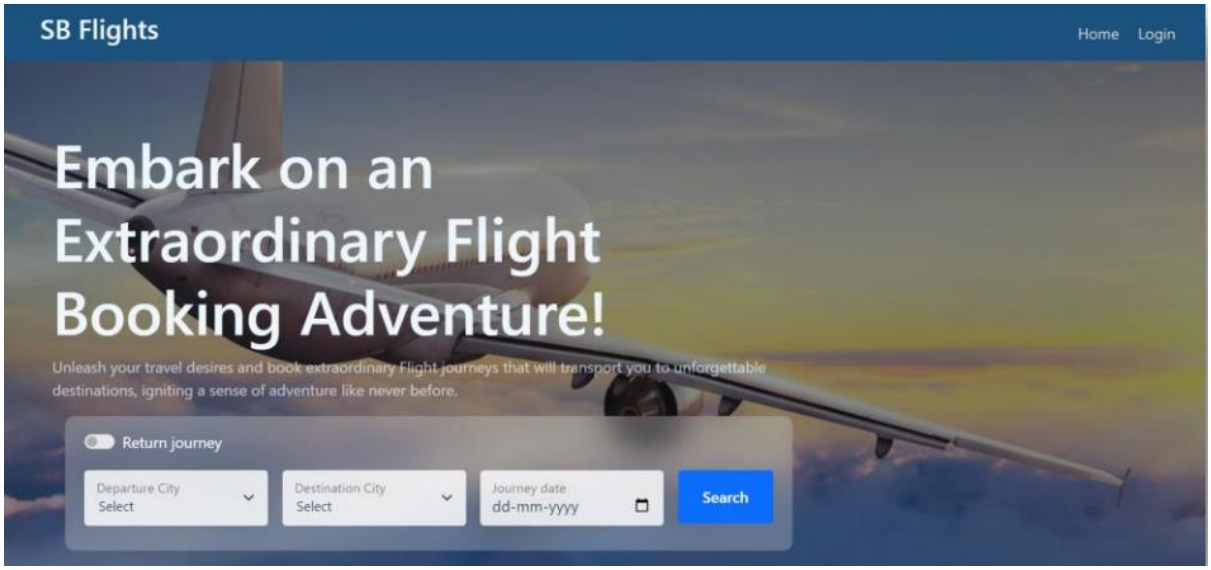
---

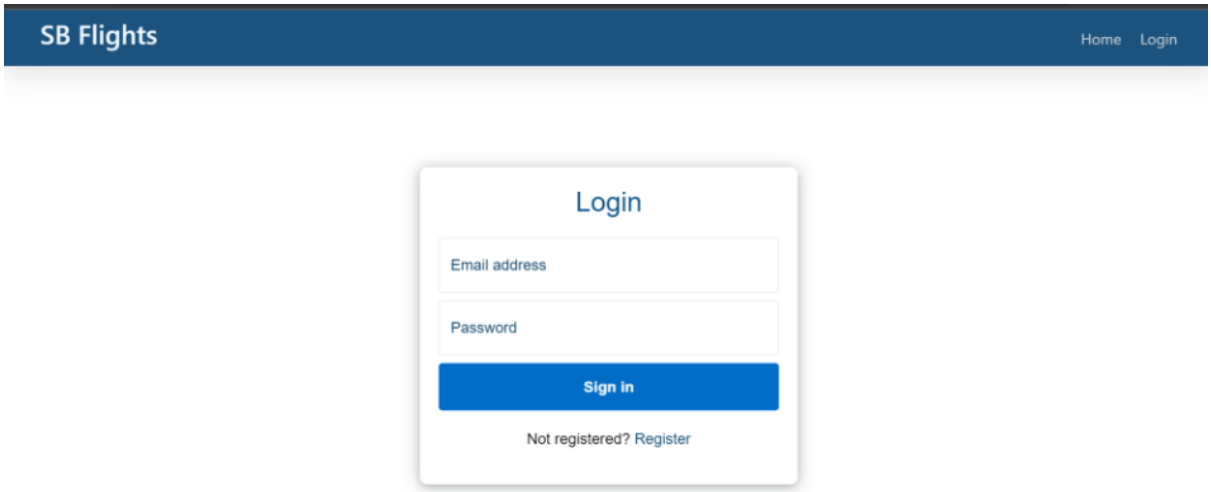
13. Frontend Development Highlights

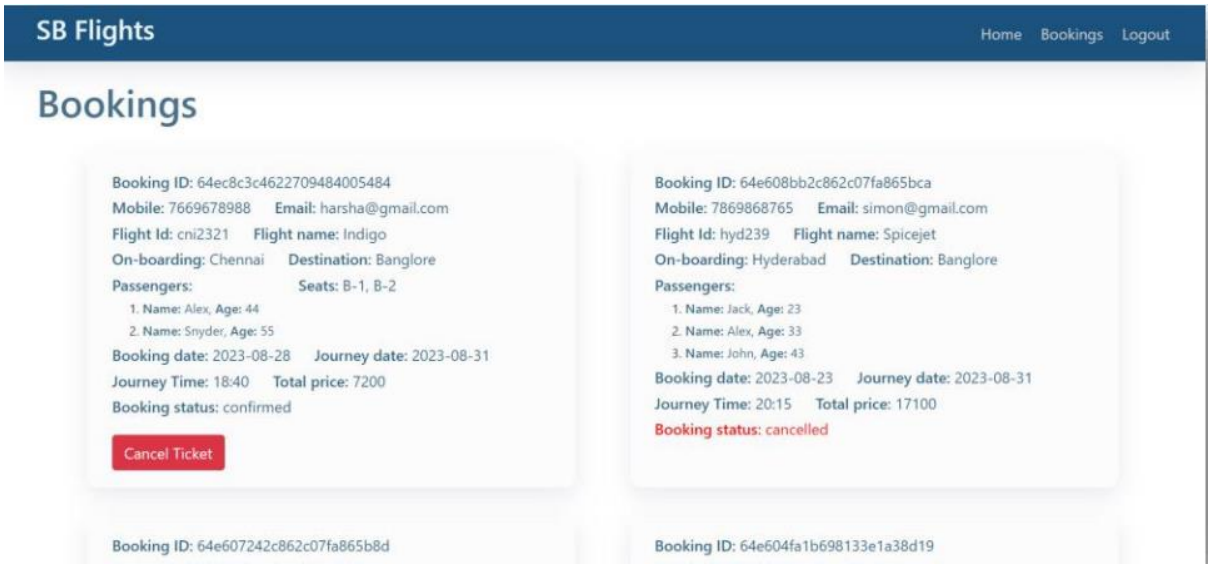
- React components for each page (Home, Bookings, Dashboard)
- Axios for API communication
- Modals used for booking flow
- Conditional rendering for different user roles
- Admin dashboard features full CRUD on flights

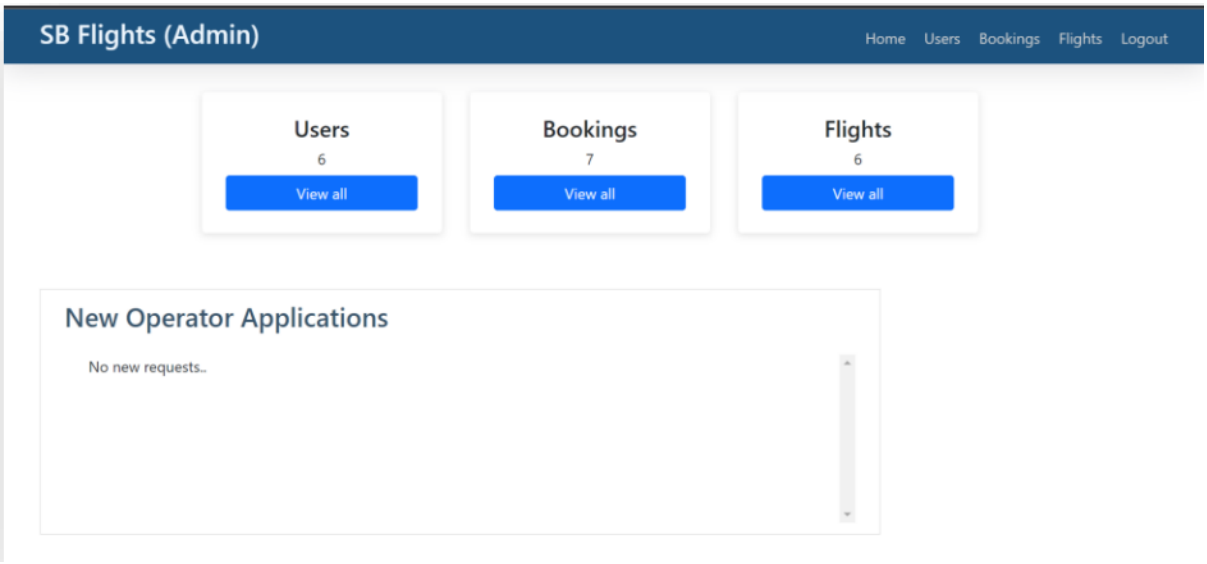
---

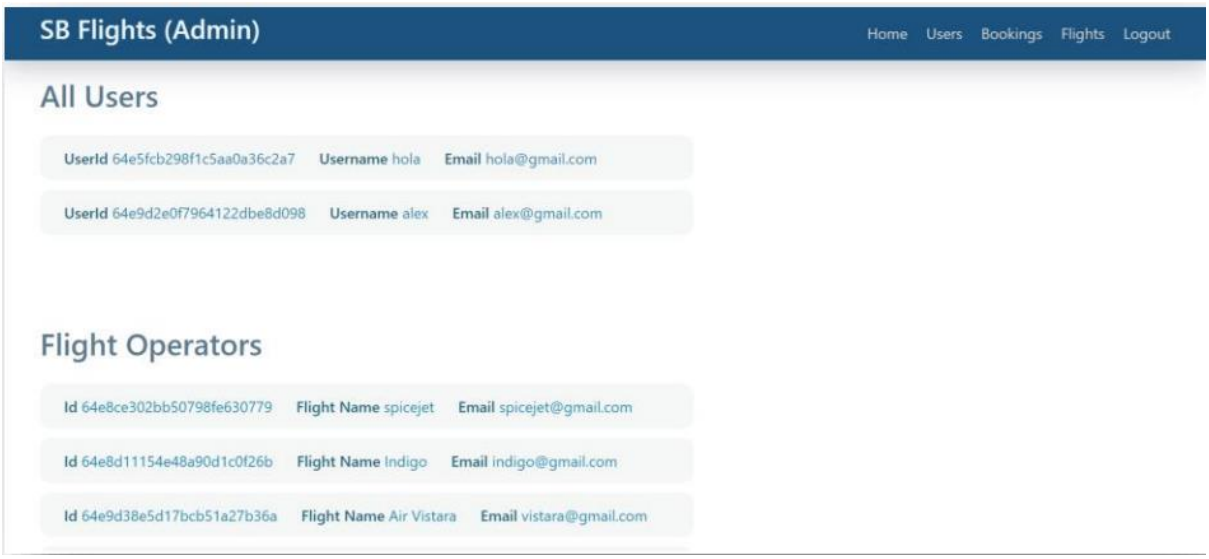
14. Demo and UI Screens

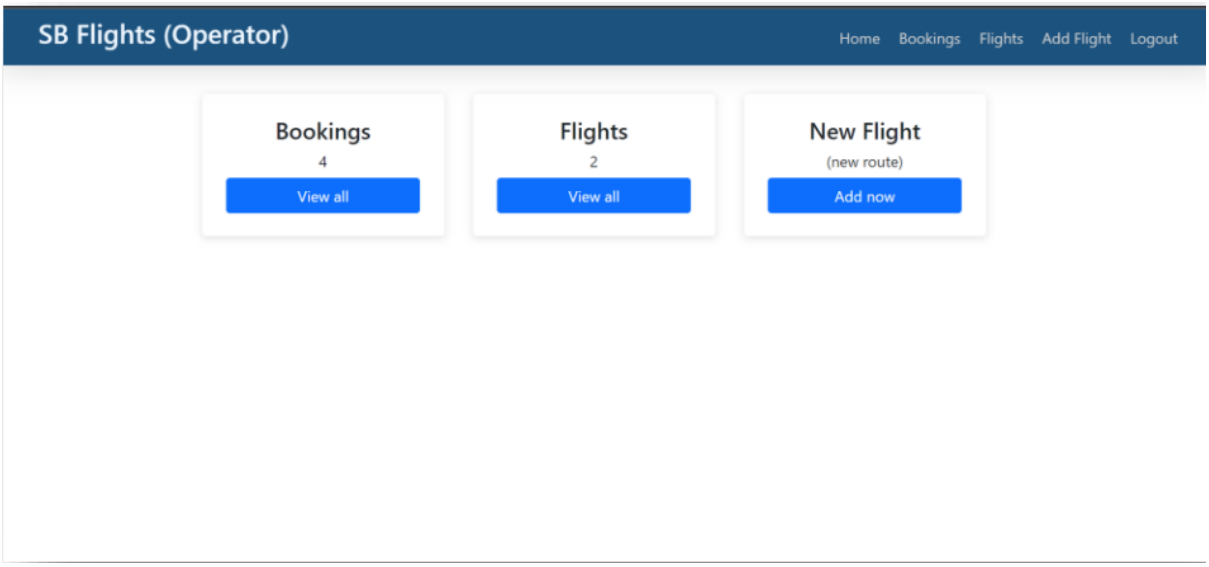

SB Flights
Home
Login
Embark on an Extraordinary Flight Booking Adventure!
Unleash your travel desires and book extraordinary Flight journeys that will transport you to unforgettable destinations, igniting a sense of adventure like never before.
Return journey
Departure City
Select
Destination City
Select
Journey date
dd-mm-yyyy
Search

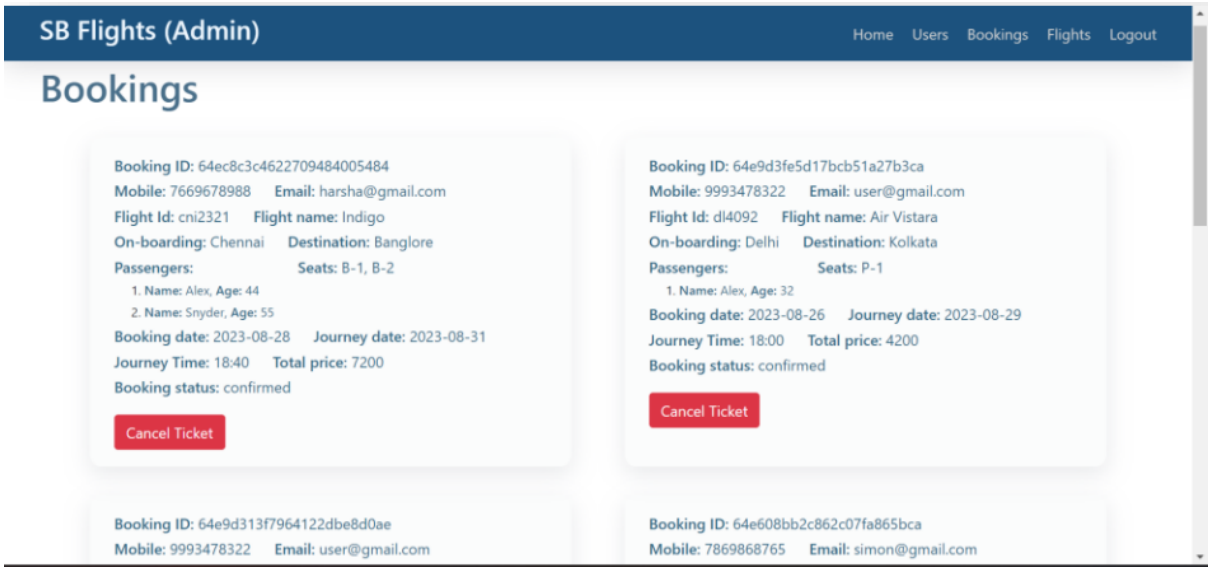

SB Flights
Home
Login
Login
Email address
Password
Sign in
Not registered? Register

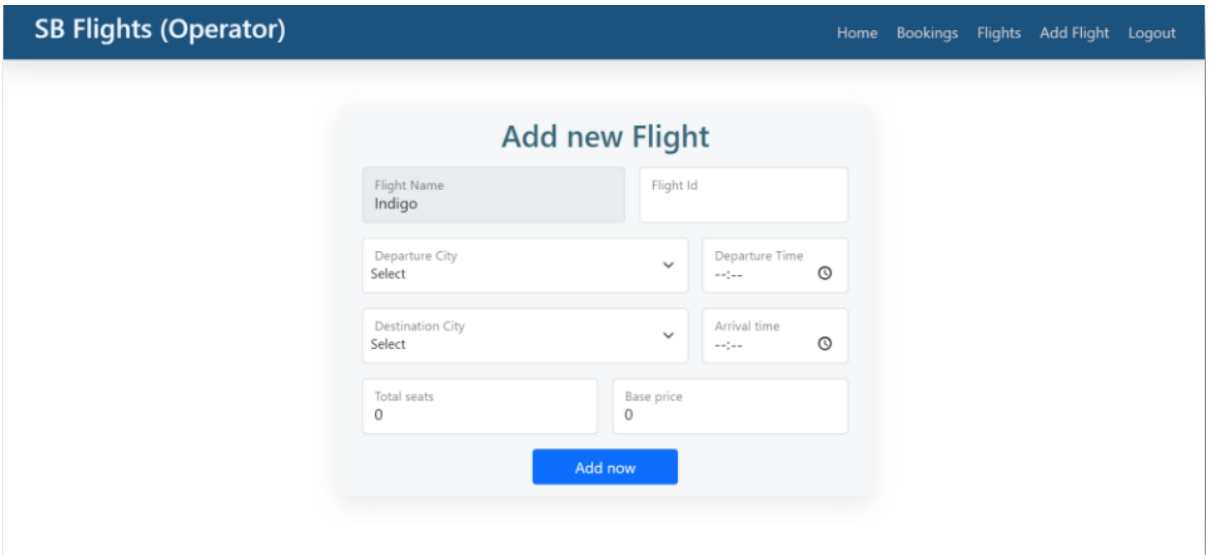

SB Flights
Home Bookings Logout
Bookings
Booking ID: 64ec8c3c4622709484005484
Mobile: 7669678988 Email: harsha@gmail.com
Flight Id: cni2321 Flight name: Indigo
On-boarding: Chennai Destination: Banglore
Passengers: Seats: B-1, B-2
1. Name: Alex, Age: 44
2. Name: Snyder, Age: 55
Booking date: 2023-08-28 Journey date: 2023-08-31
Journey Time: 18:40 Total price: 7200
Booking status: confirmed
Cancel Ticket
Booking ID: 64e608bb2c862c07fa865bca
Mobile: 7869868765 Email: simon@gmail.com
Flight Id: hyd239 Flight name: Spicejet
On-boarding: Hyderabad Destination: Banglore
Passengers:
1. Name: Jack, Age: 23
2. Name: Alex, Age: 33
3. Name: John, Age: 43
Booking date: 2023-08-23 Journey date: 2023-08-31
Journey Time: 20:15 Total price: 17100
Booking status: cancelled
Booking ID: 64e607242c862c07fa865b8d
Booking ID: 64e604fa1b698133e1a38d19


SB Flights (Admin)
Home Users Bookings Flights Logout
Users
6
View all
Bookings
7
View all
Flights
6
View all
New Operator Applications
No new requests..


SB Flights (Admin)
Home Users Bookings Flights Logout
All Users
UserId 64e5fcb298f1c5aa0a36c2a7 Username hola Email hola@gmail.com
UserId 64e9d2e0f7964122dbe8d098 Username alex Email alex@gmail.com
Flight Operators
Id 64e8ce302bb50798fe630779 Flight Name spicejet Email spicejet@gmail.com
Id 64e8d11154e48a90d1c0f26b Flight Name Indigo Email indigo@gmail.com
Id 64e9d38e5d17bcb51a27b36a Flight Name Air Vistara Email vistara@gmail.com


SB Flights (Operator)
Home Bookings Flights Add Flight Logout
Bookings
4
View all
Flights
2
View all
New Flight
(new route)
Add now





🔗 **Demo Link**
[FlightFinder App Demo]

Screenshots Covered

- Landing Page
- Authentication
- Flight Search & Booking
- Admin Dashboard
- User & Booking Management

- Add/Edit Flights
- View All Users & Bookings

---

15. Conclusion

FlightFinder successfully demonstrates how a modern web application can simplify complex travel booking processes. It highlights full-stack development best practices, from secure authentication to real-time CRUD operations with a clean UI. Built with scalability in mind, this app can be further extended with payment integrations, live flight APIs, and user notifications.
