

# Full Stack Development with MERN Project

## Documentation format

### 1. Introduction

- **Project Title:** flightfinder: navigating your air travel options
- **Team Members:**
  - **Team Leader :** Borranihanth
  - **Team member :** Battina Vishnu Vardhan
  - **Team member :** Bavireddy Vaishnavi
  - **Team member :** Bhargavi Induri

### 2. Project Overview

- **Purpose:** The Flight Booking App is a web-based system that enables users to search, view, and book flights between cities. The application supports three roles: users, flight operators, and administrators. It provides functionalities such as user authentication, role-based access, flight management, and ticket booking.
- **Features:**
  - User registration and login
  - Role-based access (user, admin, operator)
  - Flight search with optional return journey
  - Booking of flights with passenger details
  - Admin approval for flight operators
  - Management of flights and bookings

### • 3. Architecture

- **Frontend:** Developed using **React.js** with routing handled via react-router-dom. Bootstrap is used for UI components. Axios is used to interact with backend APIs.
- **Backend:** Developed using **Node.js** and **Express.js**. RESTful APIs are created to handle user management, flight operations, and booking logic.
- **Database:** **MongoDB** stores all application data, including user details, flight information, and bookings. The Mongoose library is used for

schema definitions and CRUD operations. MongoDB can be hosted locally for development .

## 4. Setup Instructions

- **Prerequisites:**

1. Install [Node.js and npm](#)
2. Install [MongoDB](#)
3. Install [Git](#)

- **Installation:**

**1.Clone the Repository:** Open a terminal or command prompt and run

- `git clone https://github.com/YourUsername/FlightFinders.git`
- `cd FlightFinders`

**2.Install Backend Dependencies:** Navigate to the server folder and install packages:

- `cd server`
- `npm install`

**3.Install Frontend Dependencies:** Navigate to the client folder and install React dependencies: `$ cd ../client`

- `cd ../client`
- `npm install`

## 5. Folder Structure

**Client:**

- 1.components` - Contains reusable components like navigation bars, buttons, and footers.
- 2.`/pages` - Houses major screens such as LandingPage, Authenticate, Bookings, Admin, etc.
- 3.`/RouteProtectors` - Controls access to routes based on user authentication and roles.
- 4.`/context` - Provides global state management (e.g., storing selected dates or user info).

- **Server:**

1. `index.js` - The main file where Express is configured, and the MongoDB connection is established.
2. `schemas.js` - Mongoose schemas defining the structure of users, flights, and bookings in the database.

## 6. Running the Application

- Provide commands to start the frontend and backend servers locally.
  - **Frontend:** cd client  
npm start
  - **Backend:** cd server  
npm start

## 7. API Documentation

- - `POST /register`: Registers a new user by taking username, email, password, and user type.
  - `POST /login`: Logs in an existing user by checking credentials.
  - `GET /fetch-flights`: Retrieves all available flights from the database.
  - `POST /book-ticket`: Books a ticket with flight ID, user ID, and passenger details.
  - `GET /fetch-users`: Admin-only endpoint to view all users.
  - `POST /add-flight`: Flight operators can add new flight listings.
  - `PUT /update-flight`: Modify an existing flight's details.
  - `POST /approve-operator`: Admin approves a flight operator.
  - `POST /reject-operator`: Admin rejects a flight operator.

## 8. Authentication

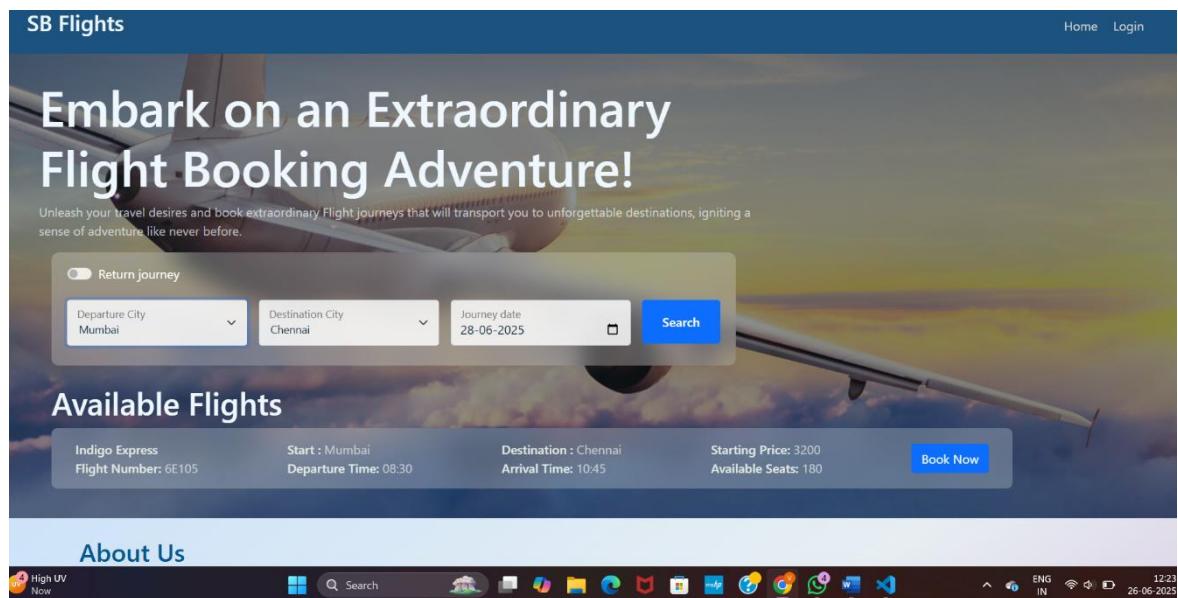
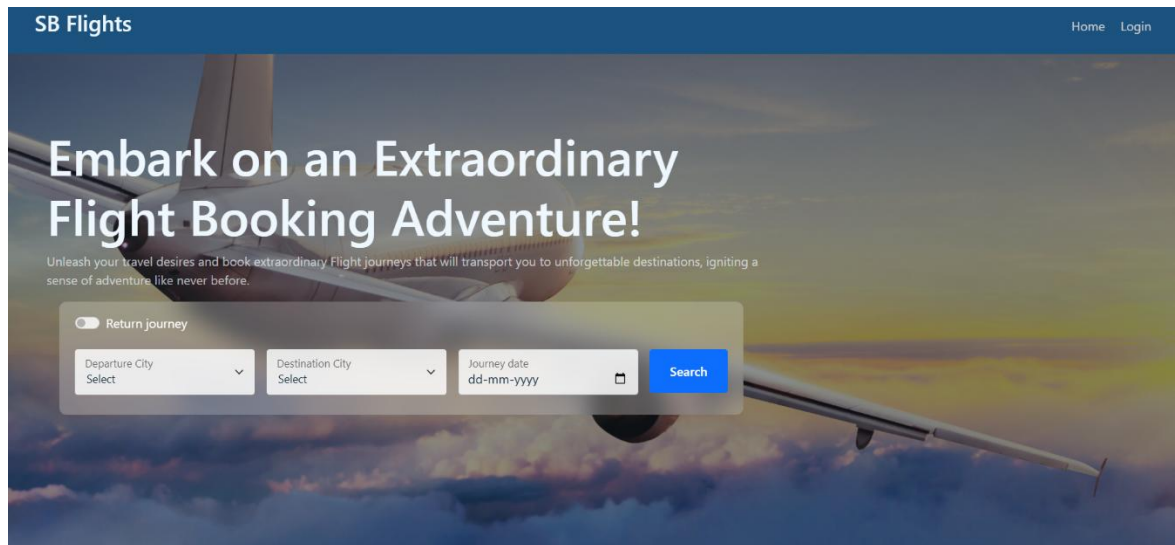
**User Registration:** Validates and saves new users with hashed passwords.

**User Login:** Authenticates users and grants access based on their role.

**Security:** Passwords are encrypted using bcrypt, and sessions are managed via localStorage in the frontend.

**Protected Routes:** Only logged-in users or admins/operators can access specific pages like bookings or admin panels.

## 9. User Interface



SB Flights

Home Login

### Login

Email address  
jhos@gmail.com

Password  
\*\*\*\*

Sign in

Not registered? Register

## 10. Testing

**Manual Testing:** Each feature is tested manually using both browser actions and API tools like Postman.

**Validation:** Server-side validations for required fields.

**Console Logs:** Added in frontend and backend for easy debugging.

## 11. Screenshots or Demo

SB Flights

Home Bookings Logout

### Book ticket

Flight Name: Indigo Express Flight No: 6E105

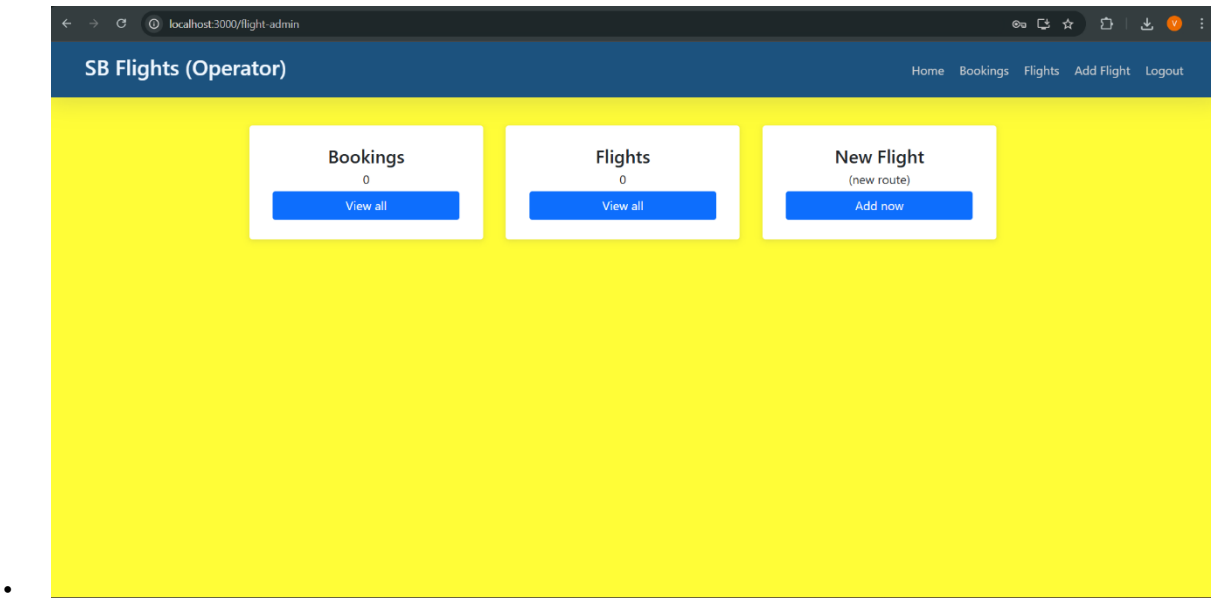
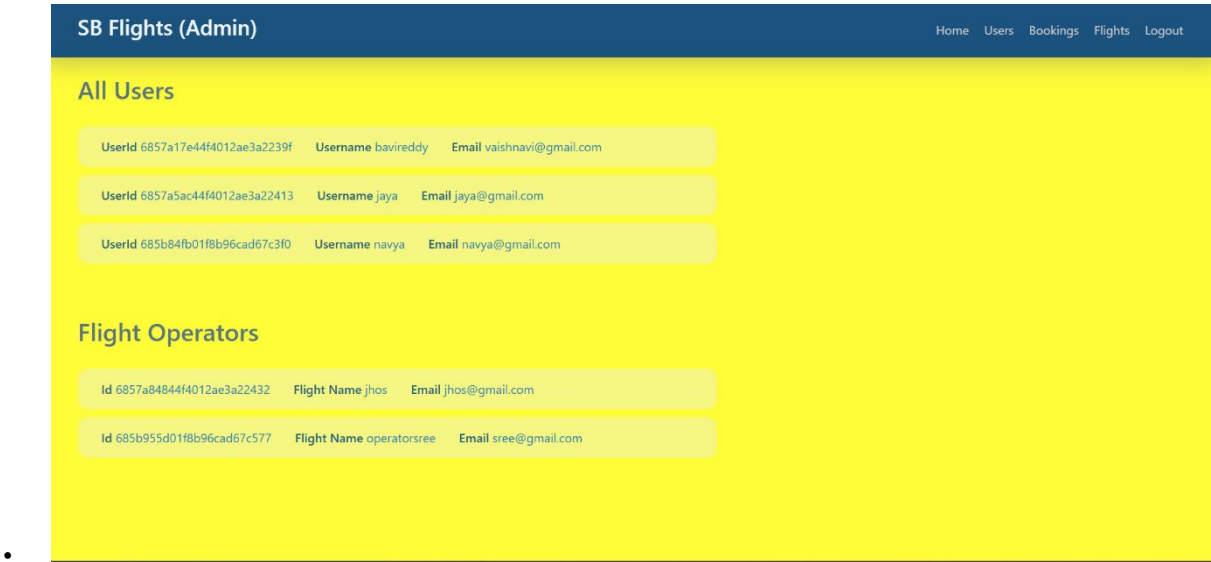
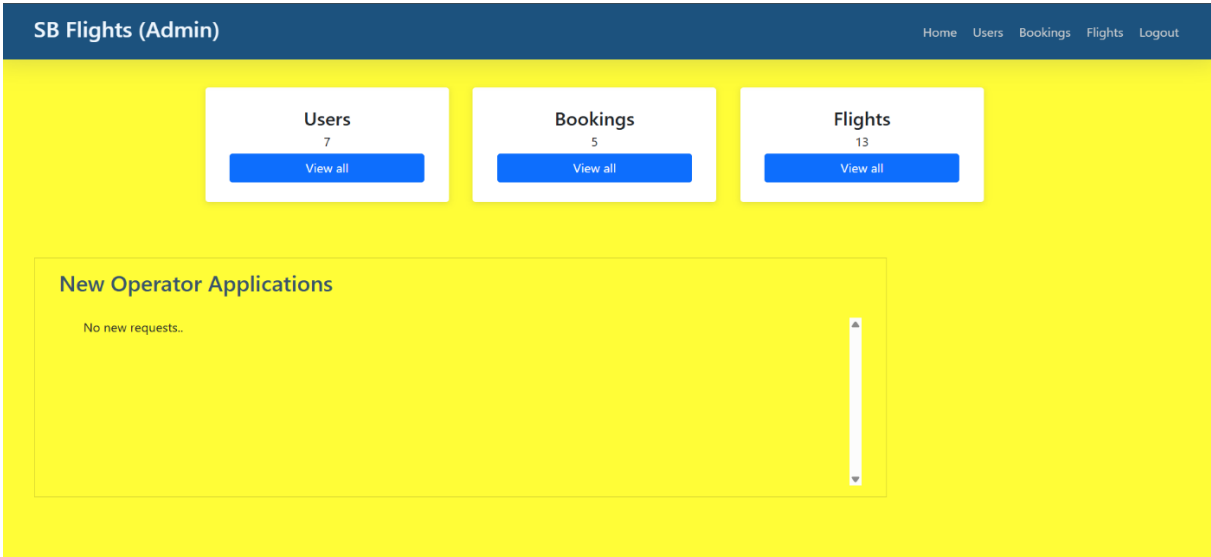
Base price: 3200

Email Mobile

No of passengers 0 Journey date 28-06-2025 Seat Class Select

Total price: 0

Book now



Demo link: <https://drive.google.com/file/d/19mbpjTRWXBSkBS65RN-x7Nhy36Pw90TP/view?usp=sharing>

**Known Issues** The seat selection is automatic and may not reflect real-time availability.

- No confirmation email or SMS is sent after booking.
- No JWT token is implemented for secure session management.

### 13. Future Enhancements

- **JWT Authentication:** Secure user sessions with tokens.
- **Payment Integration:** Add Razorpay or Stripe for real payments.
- **Email Notifications:** Send booking confirmation emails.
- **Real-time Updates:** WebSocket-based seat locking.
- **Testing Suite:** Add unit and integration tests using Jest/Mocha.
- **UI Improvements:** Enhance responsiveness and design.