**📘 FlightFinder Documentation**

**🚀 Overview**

**FlightFinder** is a modern flight booking web application designed to simplify and streamline the process of booking flight tickets. Built with a powerful tech stack (React, Node.js, Express, MongoDB), it offers users a smooth and secure experience from search to booking.

**📌 Features**

* User Authentication & Registration
* Real-time Flight Search and Filtering
* Class-wise Booking (Economy, Business, etc.)
* Interactive Seat Selection
* Secure Payments
* Admin Dashboard for Flight Management
* Booking History and E-Ticket Generation

**🛠️ Tech Stack**

| **Layer** | **Technology** |
| --- | --- |
| Frontend | React.js, Axios, Bootstrap |
| Backend | Node.js, Express.js |
| Database | MongoDB |
| Authentication | JWT (JSON Web Token) |
| Payment Integration | Razorpay/Stripe (optional) |

**🧭 Step-by-Step Workflow**

**✅ 1. User Registration & Login**

* User registers or logs in using a secure form.
* Authenticated using JWT.
* User session maintained for further steps.

**🔍 2. Flight Search**

* User enters:
  + Departure location
  + Destination
  + Departure & Return Dates
  + Number of Passengers
  + Class (Economy/Business)
* API fetches available flights based on query.
* Options shown with airline, price, duration, layovers, and time.

**🧰 3. Filter & Customize**

* User applies filters:
  + Direct flights only
  + Preferred airlines
  + Time preferences
* Sorted dynamically using filters.

**🎯 4. Select Flight & Seat**

* User selects a flight.
* App displays seat map for chosen class (e.g., Business).
* User selects available seat (highlighted).

**💳 5. Payment & Booking Confirmation**

* User provides payment details via secure gateway.
* Upon successful transaction:
  + Booking is saved in the database.
  + E-ticket and itinerary are generated.
  + Confirmation is shown.

**📋 6. Admin Panel**

* Admin login for managing:
  + Flights (Add/Edit/Delete)
  + Bookings overview
  + User management
* Dashboard with analytics and flight overview.

**🏗️ Technical Architecture**

plaintext

Copy code

┌────────────────────────────────┐

│ Frontend │

│ React.js Web App │

│ - Login/Register │

│ - Flight Search & Booking │

│ - Seat Selection │

│ - Booking Confirmation │

└────────────┬───────────────────┘

│ API Calls (Axios)

▼

┌────────────────────────────────┐

│ Backend │

│ Node.js + Express APIs │

│ - User Routes │

│ - Flight Routes │

│ - Booking Routes │

│ - Admin Routes & Auth │

└────────────┬───────────────────┘

│ Mongoose ODM

▼

┌────────────────────────────────┐

│ Database │

│ MongoDB │

│ - Users Collection │

│ - Flights Collection │

│ - Bookings Collection │

└────────────────────────────────┘

**🗂️ Project Structure**

**🌐 Client (React.js)**

pgsql

Copy code

client/

│

├── public/

│ └── index.html

│

├── src/

│ ├── components/

│ │ ├── Login.js

│ │ ├── Register.js

│ │ ├── FlightSearch.js

│ │ ├── FlightList.js

│ │ ├── SeatSelection.js

│ │ └── BookingConfirmation.js

│ │

│ ├── pages/

│ │ ├── Home.js

│ │ ├── BookingPage.js

│ │ └── AdminDashboard.js

│ │

│ ├── services/

│ │ └── api.js (Axios instance)

│ │

│ ├── App.js

│ └── index.js

│

└── package.json

**🖥️ Server (Node.js + Express)**

pgsql

Copy code

server/

│

├── controllers/

│ ├── authController.js

│ ├── flightController.js

│ ├── bookingController.js

│ └── adminController.js

│

├── models/

│ ├── User.js

│ ├── Flight.js

│ └── Booking.js

│

├── routes/

│ ├── authRoutes.js

│ ├── flightRoutes.js

│ ├── bookingRoutes.js

│ └── adminRoutes.js

│

├── middleware/

│ ├── authMiddleware.js

│ └── errorHandler.js

│

├── config/

│ └── db.js (MongoDB connection)

│

├── server.js

└── package.json

**🧪 Testing Workflow**

* Test each API with Postman.
* Validate login, search, filter, and booking.
* Test admin flight CRUD operations.
* Test payment flow (sandbox mode).

**🔐 Security**

* JWT-based auth for both User & Admin
* Passwords hashed using bcrypt
* Rate limiting and input validation
* Secure payment integration

**🔄 Future Enhancements**

* Email notifications for tickets
* QR Code for boarding pass
* Flight status tracking (Live API)
* PNR management
* Multi-language support