Full Stack Development with MERN

# Project Documentation

Project Title: Flight Finder

Team Members:

* Hitanshu Yadav [22BCE10527]
* Mohammad Asif Raza [22BSA10190]
* Akshat Singh [22BCE10690]
* Mallika Manish Rajpal [22BCE10455]

# 2. Project Overview

## Purpose

The Flight Booking App aims to provide a modern, responsive, and user-friendly platform for booking flights. It streamlines the booking process by offering real-time search, secure payments, and clear pricing information. This MERN stack application caters to users seeking efficient flight comparisons and personalized booking experiences.

## Features

* Advanced flight search and filtering (destination, date, airline, price)
* Real-time comparison of flights
* Secure authentication and user account management
* Transparent and streamlined booking with integrated payment
* User dashboard for managing trips and history

# 3. Architecture

## Frontend

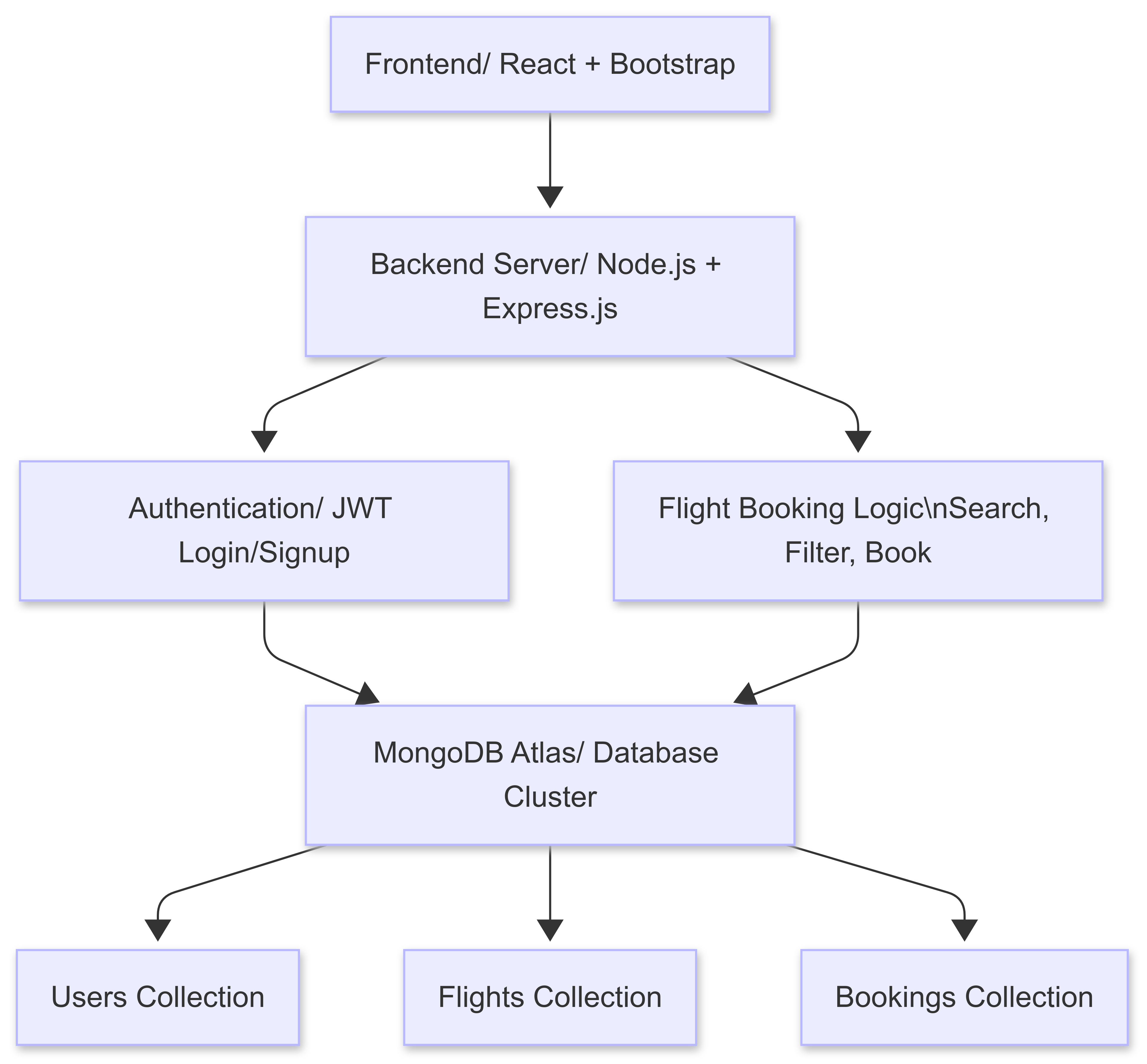
The frontend uses React for a modular, component-driven architecture. It leverages Bootstrap and custom CSS for responsiveness. API calls are made using Axios to fetch and display flight data.

## Backend

Node.js with Express.js handles server-side logic including routing, authentication, and booking transactions. It interacts with MongoDB using Mongoose ORM.

## Database

MongoDB stores user profiles, flight information, and booking records. Collections include users, flights, and bookings. Each booking references both user and flight documents.

**System Architecture Diagram:**  
  


*Figure 3.1*

# 4. Setup Instructions

## Prerequisites

* Node.js >= 16
* MongoDB >= 4
* npm or yarn
* Postman for testing (optional)

## Installation

1. Clone the repository  
2. Navigate to `/client` and run `npm install`  
3. Navigate to `/server` and run `npm install`  
4. Set environment variables in `.env` file (MONGO\_URI, JWT\_SECRET, etc.)  
5. Run `npm start` in both client and server directories

# 5. Folder Structure

## Client

`/components`: React UI components  
`/pages`: Page-level views  
`/services`: API service layer  
`/styles`: CSS files and theme

## Server

`/routes`: API endpoints  
`/controllers`: Business logic for endpoints  
`/models`: Mongoose schemas  
`/middleware`: Authentication, error handling

# 6. Running the Application

* Frontend: `cd client && npm start`
* Backend: `cd server && npm start`

# 7. API Documentation

\*\*Auth APIs\*\*

* `POST /api/auth/register` - Register a new user
* `POST /api/auth/login` - Login user and return JWT

\*\*Flight APIs\*\*

* `GET /api/flights` - Get list of available flights
* `GET /api/flights/:id` - Get flight details

\*\*Booking APIs\*\*

* `POST /api/bookings` - Book a selected flight
* `GET /api/bookings/user/:id` - Get bookings for a user

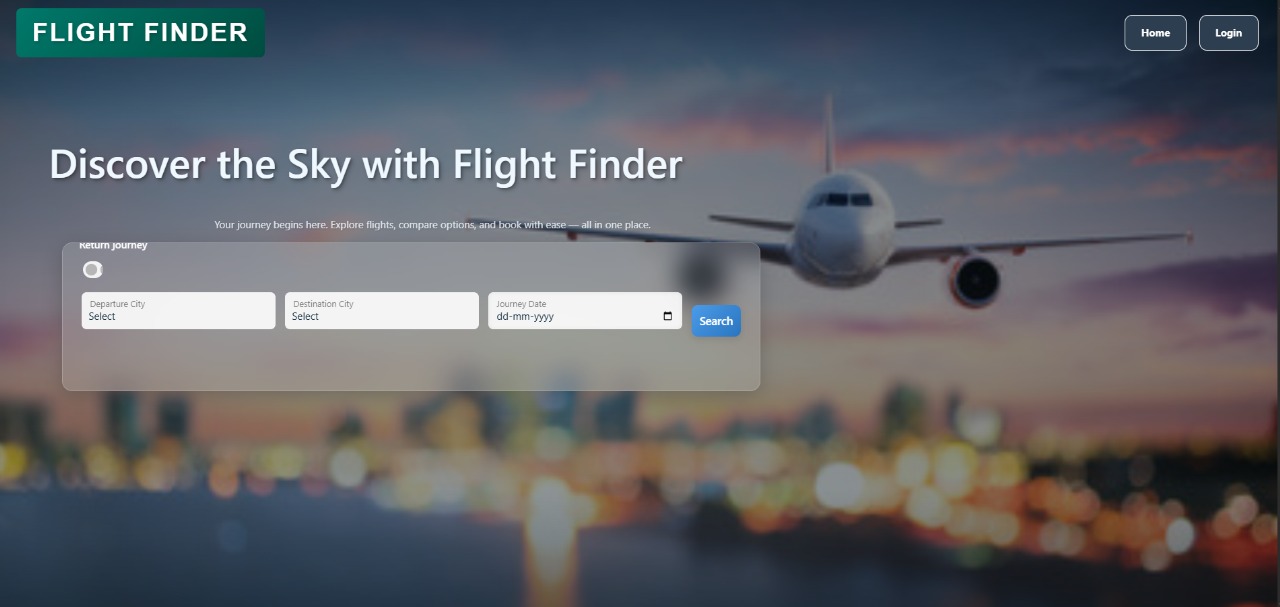
# 8. Authentication

Authentication is handled using JSON Web Tokens (JWT). Upon login, a token is issued and stored on the client. Protected routes use middleware to verify the token and authorize access. Role-based access (user/admin) can be integrated.

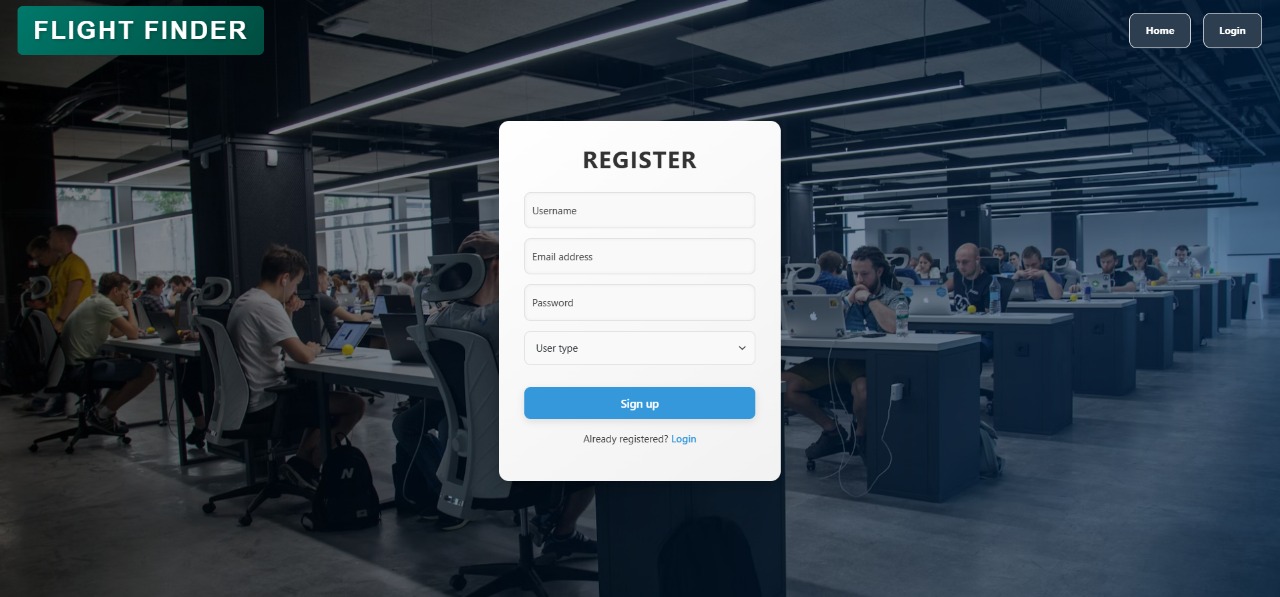
# 9. User Interface

Clean, responsive UI with the following sections:

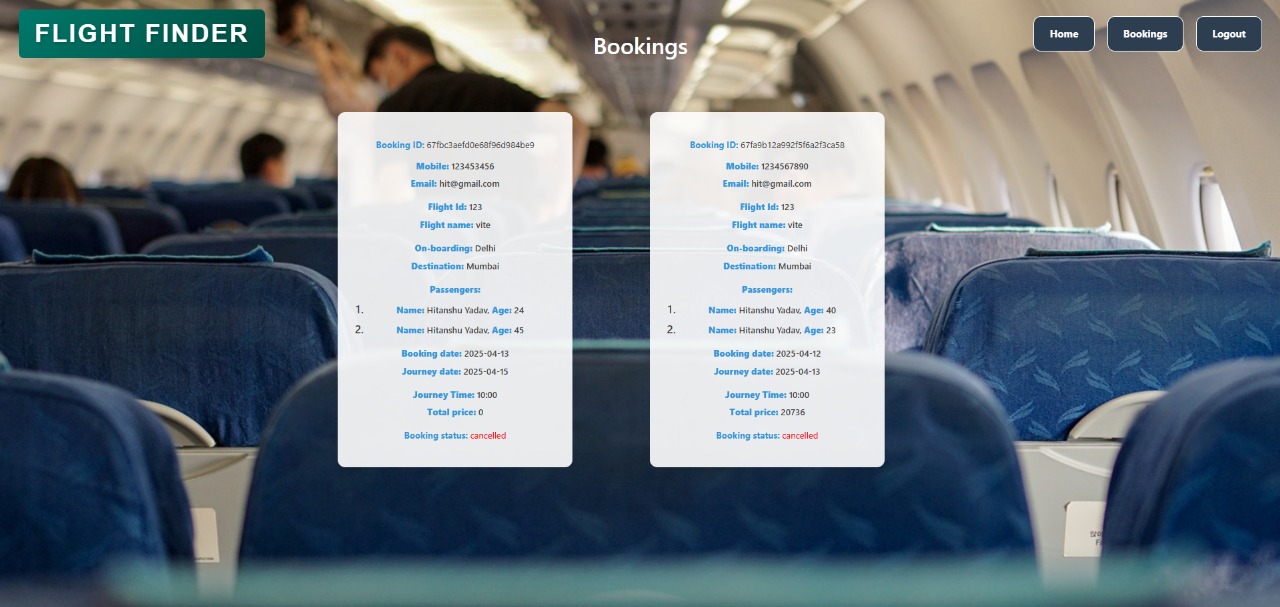
* Flight search form
* Results table with filters
* Booking confirmation page
* User login/signup and dashboard

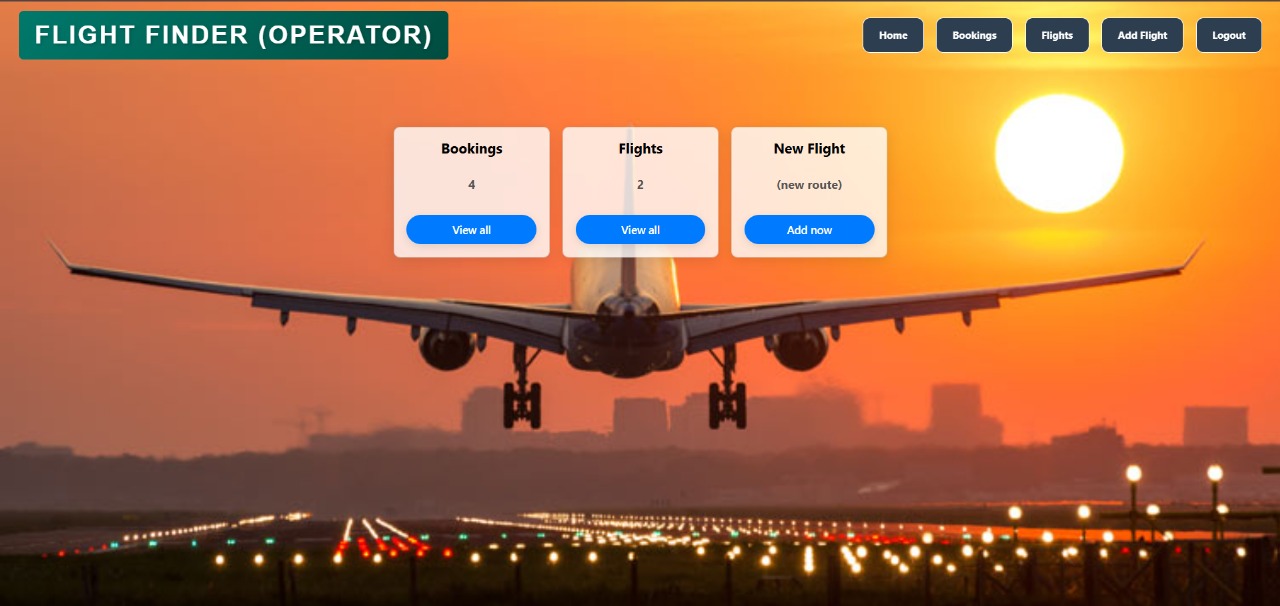
**Sample UI Screenshots:**  
  


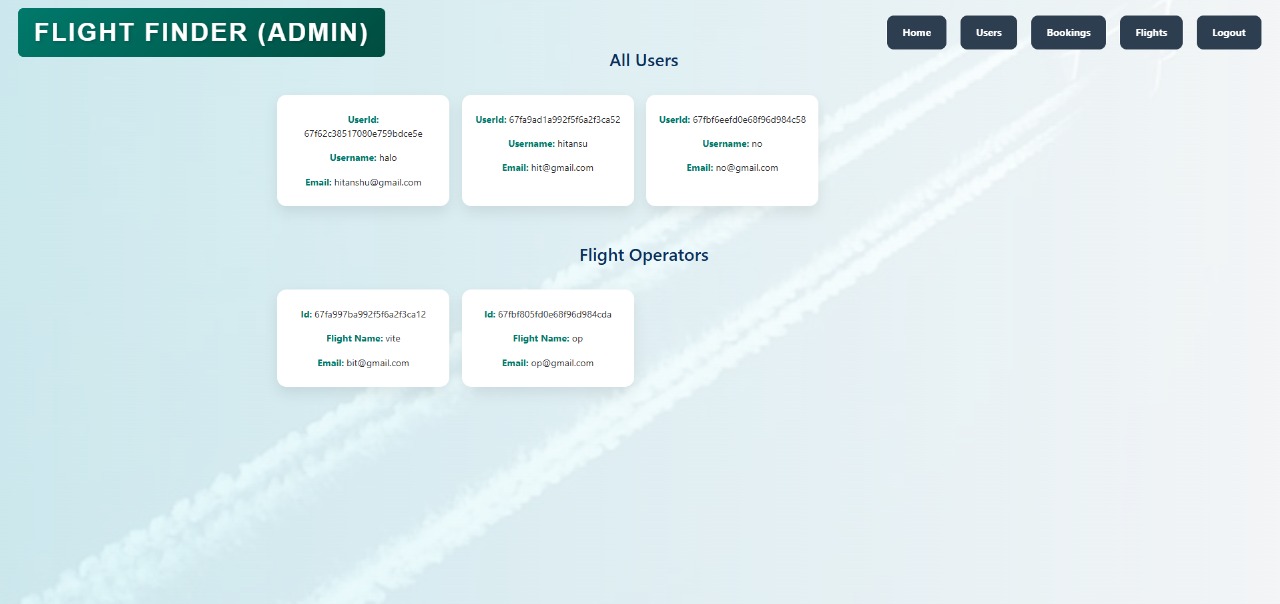
*Figure 9.1*



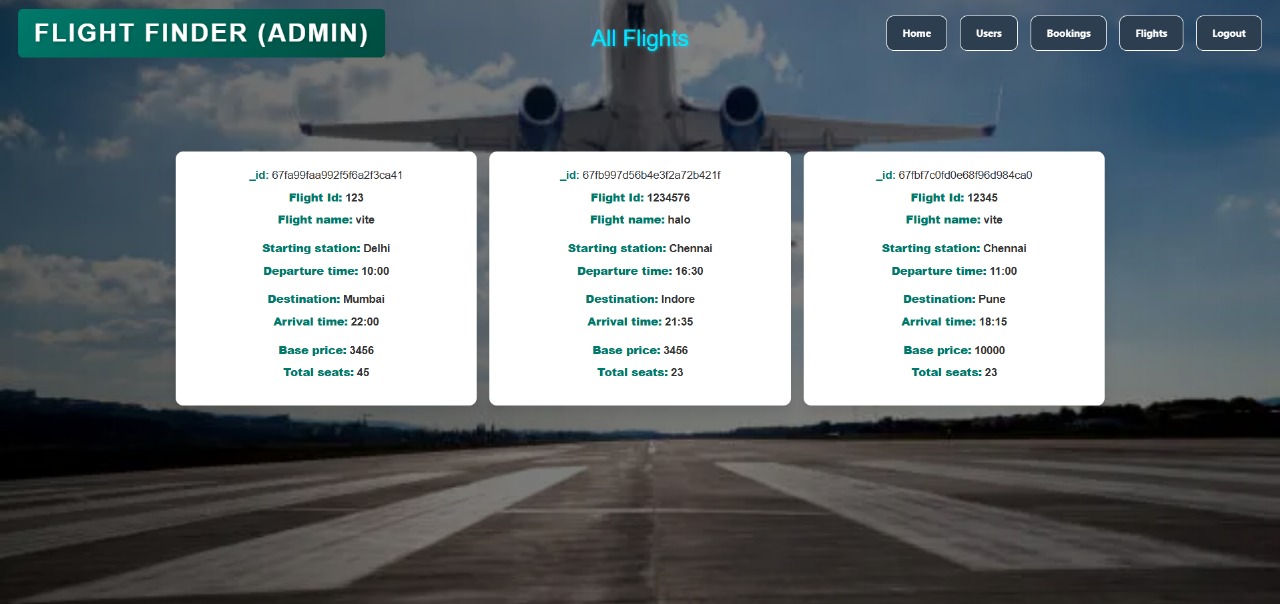
*Figure 9.2*



*Figure 9.3*  
  
  
  
  
  
  
  
*Figure 9.4*



*Figure 9.5*



*Figure 9.6*

# 10. Testing

API endpoints were tested using Postman. Frontend components were manually tested across devices. Performance was analyzed using Chrome DevTools. Example test scenarios include booking workflows and login edge cases.

# 11. Demo Video

Live demo available at:   
[GitHub Link](https://github.com/Haloween-arch/Flight_Finder.git)  
[Google Drive](https://drive.google.com/drive/folders/1sDyig-13VsQO0ZHKS_uzPwb-JAcwv86V?usp=sharing)  
  


# 12. Known Issues

* No native mobile application yet
* Offline access not supported
* Only English supported
* Limited flight provider APIs integrated

# 13. Future Enhancements

* Integrate hotel and car rental booking services
* Develop native iOS and Android apps
* Support multi-language and currency localization
* AI-based price prediction and booking suggestion
* Real-time flight status updates and loyalty program integration