Full Stack Mern Development

Introduction

Project Title:
Flight Finder: Navigation of Your Air Travel Options

Team Members:

- Divyanjali Nookarapu
- Mouniya Sheik
- Amrutha Tummala
- Veenam Venkata Sravani

Project Overview

Purpose:
The purpose of the Flight Finder project is to create a user-friendly platform that allows travelers to easily search, compare, and book flights based on their preferences, such as price, duration, airlines, and layovers. With the growing demand for convenient air travel planning, this project aims to simplify the flight booking process by providing real-time flight data, intuitive search options, and customized recommendations.

Goals:

- Real-Time Flight Information
- Cost-Effective Travel Planning
  To help users find the most affordable and optimal flight options by comparing
- To provide seamless redirection to airline or third-party booking platforms for ticket purchase, enhancing the overall travel planning experience.

Key Features:

- Users can opt-in for price drop alerts, flight status updates.
- Directs users to the respective airline.
- Real-Time Flight Data.

Architecture :

Front End Architecture :

Technology Stack

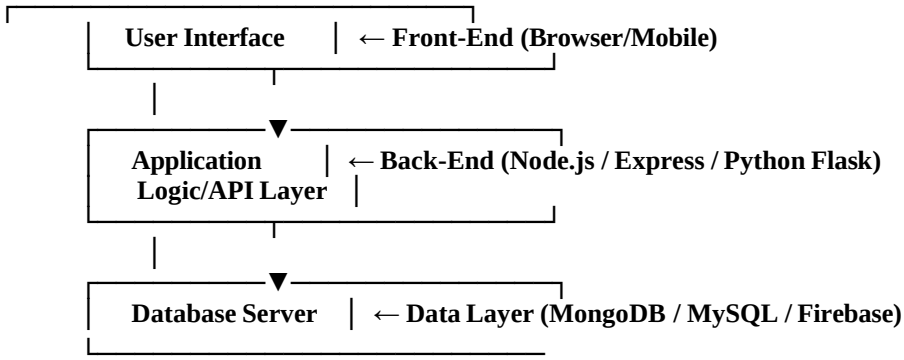
HTML5, CSS3 – Structure and styling

JavaScript (ES6+)

React.js or Angular – For building UI components

Axios/Fetch – For calling APIs

Tailwind CSS / Bootstrap – For responsive design



Backend:
Runtime Environment: Node.js

Web Framework: Express.js (or Python Flask/Django as alternatives)

Database: MongoDB / Firebase (NoSQL) or MySQL (SQL)

Flight API Integration: Skyscanner, Amadeus, or any open flight data API

Authentication: JWT (JSON Web Token) or OAuth 2.0

Hosting: Heroku, AWS, or Render

API Testing: Postman or Swagger

/backend

```
│
├── server.js            ← Entry point of the server
├── config/              ← API keys, DB config
│   └── db.js
│
├── routes/              ← REST API endpoints
│   └── flightRoutes.js
│
├── controllers/         ← Logic for handling flight searches
│   └── flightController.js
│
├── services/            ← External API communication logic
│   └── flightAPIService.js
│
```

├── models/ ← Database schema (MongoDB models or SQL tables)
│ └── User.js
│
├── middleware/ ← Auth middleware, error handlers
│ └── auth.js
│
└── utils/ ← Helper functions

Database:

You can also use Firebase, MongoDB or MySQL depending on your team's stack preference.

Prerequisites:

- HTML5, CSS3, JavaScript fundamentals
- Familiarity with React.js (or Angular/Vue.js)
- Node.js and Express.js
- Knowledge of MongoDB, Firebase, or MySQL

Installation Steps:

STEP 1: Install Required Software

Make sure the following tools are installed:
Node.js (v14 or above)
MongoDB Community Server (or use MongoDB Atlas)
Git
VS Code or any code editor

STEP 2: Clone the Project

```
git clone https://github.com/your-username/flight-finder.git
cd flight-finder
```

STEP 3: Set Up the Front-End

```
cd client          # Navigate to the front-end directory
npm install        # Install dependencies
npm start          # Start the React development server
```

STEP 4: Set Up the Back-End

```
cd server          # Navigate to the back-end directory
npm install        # Install server dependencies
```

STEP 5: Configure Environment Variables

```
Create a .env file inside the /server folder with the following (sample)
content:
PORT=5000
MONGODB_URI=mongodb://localhost:27017/flightfinder
API_KEY=your_flight_api_key_here
```

STEP 6: Start the Back-End Server

```
npm run start    # or node server.js
```

> This will run the back-end on http://localhost:5000

STEP 7: Test the Application

1. Open http://localhost:3000 in your browser.
2. Enter flight search details and view results.
3. Ensure API responses are working and data is rendering correctly.

How to Use:

1. Open the application
2. Launch the application by visiting:
   http://localhost:3000 (if running locally)
3. Enter Flight Search Details
4. Source Location (e.g., Hyderabad) / Destination Location (e.g., Delhi)
5. Number of Passengers / Click the "Search Flights" button
6. Browse and Filter Flight Results
7. View a list of available flights with:

   Airline Name
   Departure & Arrival Times
   Duration
   Price

8. View Flight Details
9. Layovers
   Terminal info
   Baggage details

Proceed to Booking:

- Click “Book Now” which will redirect to:
  Airline website

Authentication Implementation

- Authentication in the Flight Finder project ensures that only authorized users can access certain features such as saving flight searches

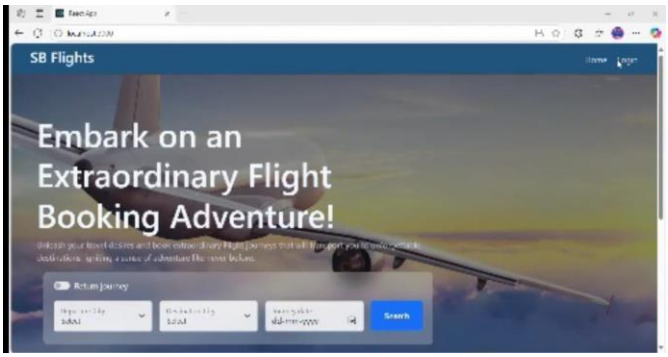
User Interface

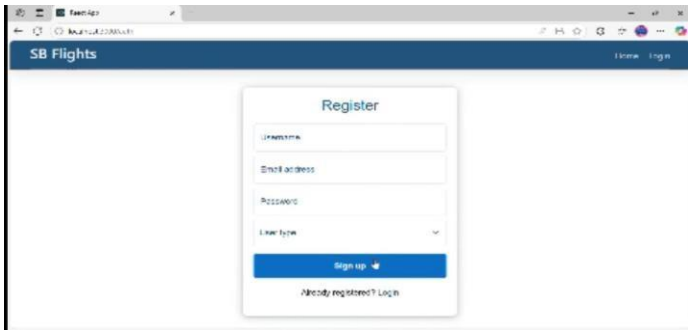
- The User Interface of the Flight Finder project is designed to be clean, intuitive, and user-friendly, enabling users to easily search, compare, and explore flight options. It is built using modern front-end technologies

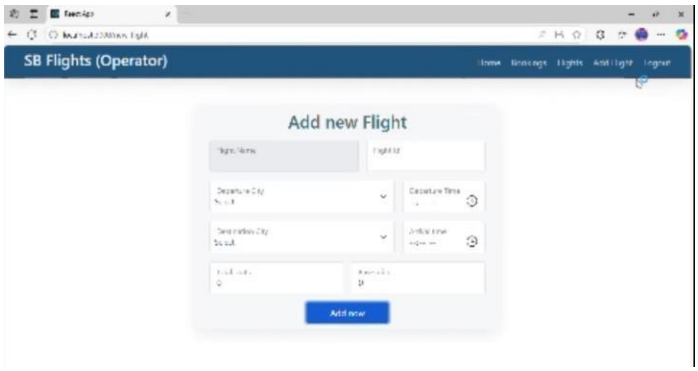
Testing

- Testing ensures that the Flight Finder application functions correctly, is free of bugs, and offers a smooth user experience.

Screenshots or Demo







Known Issues

- API Rate Limiting
- No Real Booking System
- Mobile UI Glitches.
- Incomplete Authentication (If Basic Only)

Future Enhancements

- Add an admin panel to manage users, monitor API usage, and view logs..
- Offline Mode.
- Advanced Authentication & Role Management.
- Multi-Language Support.
- Benefit: Enhances usability and inclusivity