Full Stack Development with MERN

Project Documentation format

1. Introduction

- **Project Title: AirVoyager - Flight Booking System**
- **Team Members:** Harshitha – Authentication
  Keerthi-Database Management
  Tirupatayya-Frontend
  Varshini-backend

2. Project Overview

- **Purpose:** Briefly describe the purpose and goals of the project.

- AirVoyager is a web-based flight booking system where Users can search and book flights, Flight Operators can manage flights and view bookings, and Admins can manage users and approve flight operators.

- **Features:** Highlight key features and functionalities.
  - ✅ User Registration & Login
  - ✅ Flight Search with Filters
  - ✅ Flight Booking & My Bookings View
  - ✅ Booking Cancellation for Users
  - ✅ Operator: Add, Edit Flights & View Bookings and Users
  - ✅ Admin: View All Users, Approve Operators
  - ✅ About Page

3. Architecture

- **Frontend:** Describe the frontend architecture using React.

  Built with React.js

  Component-based structure (e.g., Login, Register, FlightList, UserBookings, Admin Panel)

  Axios for API calls

- **Backend:** Outline the backend architecture using Node.js and Express.js.

  Node.js with Express.js

  RESTful API Design

  Middleware for JWT Authentication and Role-Based Authorization

- **Database:** Detail the database schema and interactions with MongoDB.

- MongoDB Atlas
- Mongoose Models for Users, Flights, Bookings
- Relations handled via MongoDB ObjectIDs

4. Setup Instructions

- **Prerequisites:** List software dependencies (e.g., Node.js, MongoDB).

  ✅ Node.js
  ✅ MongoDB Atlas Account
  ✅ npm

- **Installation:** Step-by-step guide to clone, install dependencies, and set up the environment variables.

```
# Clone the project
git clone https://github.com/Harshitha-2210/FlightFinder

# Setup Backend
 cd server
 npm install
Create .env file:
MONGODB_URI=your_mongo_uri
JWT_SECRET=your_secret_key
npm start

# Setup Frontend
cd ../client
npm install
npm start
```

5. Folder Structure

- **Client:** Describe the structure of the React frontend.

```
client/
├── src/
│   ├── components/
│   ├── App.js
│   └── index.js
└── package.json
```

- **Server:** Explain the organization of the Node.js backend.

```
server/
├── controllers/
├── models/
```

```
├── routes/
├── middleware/
├── index.js
└── package.json
```

6. Running the Application

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

7. API Documentation

- Document all endpoints exposed by the backend.
- Include request methods, parameters, and example responses.

8. Authentication

- Explain how authentication and authorization are handled in the project.
- Include details about tokens, sessions, or any other methods used.

- JWT based Authentication

- Roles:

✅ User
✅ Flight Operator
✅ Admin

- Tokens are stored in LocalStorage on client side.

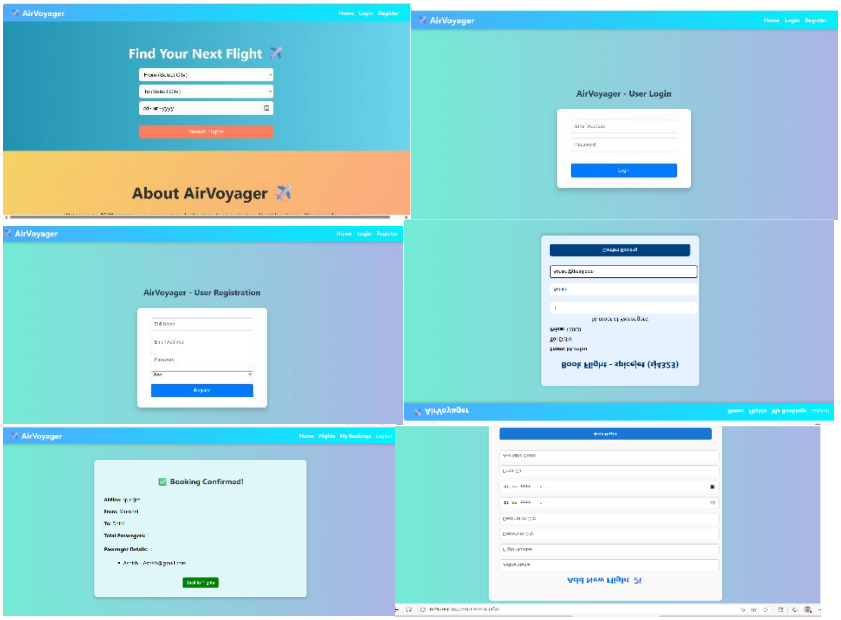
- Backend protects routes using middlewares like `verifyUser`, `verifyAdmin`, `verifyOperator`.

9. User Interface

✅ **Simple, responsive UI built with React**
✅ **Navigation based on user role**
✅ **Forms with validation**
✅ **UI Pages for:**

- **Login**
- **Register**
- **Flight Search**
- **My Bookings**
- **Operator Dashboards**
- **Admin Dashboards**
- **About Page**

- Provide screenshots or GIFs showcasing different UI features.



10. Testing

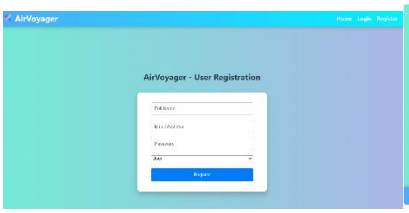
- Describe the testing strategy and tools used.
- Manual Testing with:
  - ✅ Hoppscotch / Postman for APIs
  - ✅ Browser Testing for UI

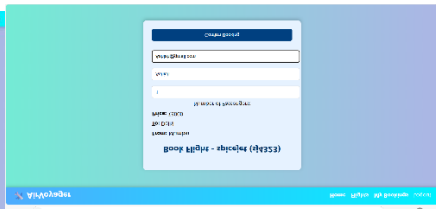
Tested:

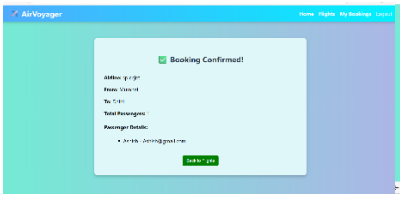
- User flows
- Booking process
- Role-based access
- Error handling

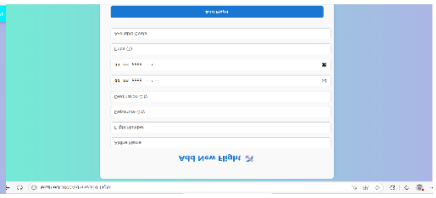
11. Screenshots or Demo

- Provide screenshots or a link to a demo to showcase the application.









12. Known Issues

- Document any known bugs or issues that users or developers should be aware of.

No real payment gateway

Email confirmation still pending

No pagination on flight list

13. Future Enhancements

- Outline potential future features or improvements that could be made to the project.

✅ Integrate Payment Gateway
✅ Add Email Notifications
✅ Add Pagination and Sorting
✅ Deploy with CI/CD pipeline
✅ Add Mobile Responsiveness