Project Title:  
 FLIGHT FINDER: NAVIGATING YOUR AIR TRAVEL OPTIONS

# **Team ID:**

LTVIP2025TMID48507

# **Team Name:**

Team Flight Finder

# **Team Members:**

1.Mungara Abhinaya

2.Mangali Poojitha

3.Mala Dasari Kasturi

4.Malluri Naga Kiran

# 

# 

# 

# 

# **Table of Contents**

|  |  |
| --- | --- |
| **S. No.** | **Section Title** |
| **1.** | Introduction |
| **2.** | Features |
| **3.** | Technology Stack |
| **4.** | Folder Structure |
| **5.** | Backend Implementation |
| **6.** | Frontend Implementation |
| **7.** | Challenges Faced |
| **8.** | Screenshots & Results |
| **9.** | Advantages & Disadvantages |
| **10.** | Conclusion |
| **11.** | Future Scope |
| **12.** | Appendix |

# **1. INTRODUCTION**

## **1.1 Project Overview**

**Flight Finder** is a full-stack web application developed to simplify online flight ticket bookings. The platform allows users to register, search flights, book seats, and manage bookings. Admins can add and manage flight details and view user bookings. This application demonstrates seamless UI and functional backend integration, built using the MERN stack.

## **1.2 Purpose**

The purpose of Flight Finder is to:

* Provide a user-friendly online flight booking system.
* Enable secure user login and role-based authentication.
* Allow users to view and book flights.
* Enable admin-level flight and booking management.
* Showcase real-world full-stack web application development

# **2. FEATURES**

## **User Features**

* User Registration & Login
* Flight Search
* Book Flights
* View My Bookings
* Responsive User Interface

- **Admin Features**

* Admin Login
* Add/Edit/Delete Flights
* View All Flights
* View All Booking

# **3. TECHNOLOGY STACK**

|  |  |
| --- | --- |
| **Layer** | **Tools/Technologies** |
| **Frontend** | React, Vite, Bootstrap, Axios |
| **Backend** | Node.js, Express.js, MongoDB, Mongoose |
| **Others** | JWT Authentication, Thunder Client, GitHub |

# **4. FOLDER STRUCTURE**

flightfinder/

│

├── backend/

│ ├── models/

│ ├── routes/

│ ├── controllers/

│ └── server.js

│

└── frontend/

├── public/

├── src/

├── api/

├── components/

├── pages/

├── HomePage.jsx

├── LoginPage.jsx

├── RegisterPage.jsx

├── SearchPage.jsx

├── BookingPage.jsx

├── MyBookingsPage.jsx

├── AdminDashboard.jsx

├── App.jsx

├── index.css

└── main.jsx

# **5. BACKEND IMPLEMENTATION**

* Configured Express.js server in server.js.
* Connected MongoDB using Mongoose.
* Created models:
* User
* Flight
* Booking
* Defined routes for:
* User registration/login
* Flight search & booking
* Admin flight & booking management
* Implemented role-based access control using JWT tokens.
* Tested endpoints using Thunder Client/Postman

# **6. FRONTEND IMPLEMENTATION**

* Developed React app using Vite for fast performance.
* Used React Router for multi-page navigation.
* Integrated Bootstrap for responsive UI.
* Created pages:
* HomePage with navigation
* Login/Register forms
* Flight Search page with filters
* Booking page with passenger info
* Admin dashboard for flight & booking management
* Axios used for connecting to backend APIs.
* Used conditional rendering based on login role (user/admin).

-**7. CHALLENGES FACED**

* Handling CORS between frontend & backend.
* Dynamic rendering of flight results.
* Integrating JWT authentication properly.
* Managing bookings logic and validation.
* Styling with Bootstrap for responsive layouts.

# **8. SCREENSHOTS & RESULTS**

(Add your screenshots here when preparing the final report. Examples include Home Page, Products listing page, Registration/Login screens, and Cart page view.)

# **9. ADVANTAGES & DISADVANTAGES**

### **✅ Advantages**

* Real-world booking system simulation.
* Clean modular code with clear structure.
* Modern UI and responsive design.
* Role-based secure access system.

### **❌ Disadvantages**

* No real-time flight API integration.
* Lacks payment gateway for booking confirmation.
* Admin role is predefined without separate signup.
* No cloud deployment included yet.

# **10. CONCLUSION**

FlightFinder achieves its goal of offering a complete flight booking experience with key features like user login, search, booking, and admin control. It demonstrates proficiency in MERN stack development, and sets the foundation for further enhancements.

# **11. FUTURE SCOPE**

* Integrate payment gateways (Razorpay/Stripe).
* Upload flight images or logos.
* Admin dashboard UI enhancements.
* Search flights based on time/date/city.
* Cloud deployment on platforms like Vercel, Netlify, or AWS.
* Add notifications and booking confirmation via email.
* Implement seat selection and ticket PDF download.

# **12. APPENDIX**

Source Code Repository:  
 https://github.com/mangalipoojitha/Flightfinder-Navigating-your-air-travel-options.git

# **Author**

Mangali Poojitha  
 Email id:sessannam086@gmail.com  
 Hall Ticket No.: 22AT1A0575  
 G. Pullaiah College of Engineering and Technology