# Full Stack Development with MERN

## Project Documentation – Flight Finder

## 1. Introduction

• Project Title: Flight Finder – Flight Booking Web Application

• Team ID: LTVIP2025TMID58636

• Team Size: 5

• Team Leader: Malliboyina Pooja Sri

• Team Member: Mudamala Prathyusha

• Team Member: Malreddy Budasri

• Team Member: M Pavithra

• Team Member: Pobbarlapalli Sai Priya

• Repository: https://github.com/malliboyinapoojasri/Flightfinder.git

• Contact Email: malliboyinapoojasri@gmail.com

## 2. Project Overview

• Purpose: The purpose of this project is to build a full-stack flight booking application using the MERN stack as part of the SmartInternz internship.

• Features:  
 - User registration and login  
 - Search and book flights  
 - Admin dashboard  
 - Responsive UI using React  
 - REST APIs with MongoDB backend

## 3. Architecture

• Frontend: Built using React.js with components for user interaction.

• Backend: Built using Node.js and Express.js for REST API functionality.

• Database: MongoDB for storing users, flights, and bookings.

## 4. Setup Instructions

• Prerequisites:  
 - Node.js  
 - MongoDB  
 - Git

• Installation:  
 1. Clone the repo: git clone https://github.com/malliboyinapoojasri/Flightfinder.git  
 2. cd Flightfinder  
 3. cd client && npm install  
 4. cd ../server && npm install  
 5. Create .env file with MONGO\_URI and JWT\_SECRET

## 5. Folder Structure

Flight-Finder/  
├── client/ → React frontend  
│ ├── src/  
│ └── public/  
├── server/ → Node.js backend  
│ ├── controllers/  
│ ├── routes/  
│ ├── models/  
│ └── config/  
├── .gitignore  
├── README.md  
└── package.json

## 6. Running the Application

• Frontend:  
 npm start (inside client folder)

• Backend:  
 node index.js (inside server folder)

## 7. API Documentation

• User APIs: /register, /login, /my-bookings

• Flight APIs: /flights, /book

• Admin APIs: /admin/add-flight, /admin/delete-flight/:id

## 8. Authentication

• JWT-based authentication with protected routes  
• Token stored in local storage and verified via middleware

## 9. User Interface

• React with Bootstrap styling  
• Components: Login/Register, Booking Form, Admin Panel

## 10. Testing

• Manual testing using Postman  
• Console-based checks in frontend for validation

## 11. Screenshots or Demo

• Add screenshots from app (home, login, booking) here.  
• Alternatively, provide demo link if deployed.

## 12. Known Issues

• No password reset feature  
• No email confirmation on booking  
• Admin route not login-restricted

## 13. Future Enhancements

• Add payment gateway  
• Deploy app online  
• Add filters and confirmation emails