**Flight Finder: Navigating Your Flight Travel Options**

**Team Members:**  
Team Leader: Neeharika Rayachoti  
Team Member: S. Reddy Poojitha  
Team Member: R. Vijay Kumar Reddy  
Team Member: P. Nagi Reddy

## Introduction

In today’s fast-paced world, travel has become an essential part of our lives. Thanks to flight booking apps, booking flights is now accessible and convenient. A flight booking app allows users to search, compare, and book flights easily from their mobile devices.

## Description

This Flight Booking App revolutionizes flight ticket booking, offering users a seamless experience through a user-friendly interface. Whether you’re a frequent traveler or flying occasionally, this app helps you discover and reserve flights based on your preferences.

Key features include: - Intuitive interface - Efficient search and booking - Personalization options - Robust security - Continuous improvement

## Use Case Scenario

John, a business professional, needs to book a flight from New York to Paris for a conference: 1. Opens the app and enters travel details. 2. The app retrieves flight options. 3. Filters for direct flights and preferred airline. 4. Selects a seat and pays. 5. Receives e-ticket and itinerary.

This scenario showcases the convenience and real-time assistance the app provides.

## Technical Architecture

* **Frontend**: User Interface including Authentication, Flight Search, and Booking.
* **Backend**: Node.js APIs for Users, Flights, Admin, Bookings with Authentication and Dashboard.
* **Database**: MongoDB for storing Users, Flights, and Bookings.

## ER Diagram

### Entities:

* **USER**: Books flights; can make multiple bookings/payments.
* **BOOKING**: Includes flight and passenger info.
* **FLIGHT**: Contains flight details.
* **ADMIN**: Manages backend tasks.

## Prerequisites

* **Node.js & npm**: [Download](https://nodejs.org/en/download/)
* **MongoDB**: [Download](https://www.mongodb.com/try/download/community)
* **Express.js**: npm install express
* **React.js**: [Setup Guide](https://reactjs.org/docs/create-a-new-react-app.html)
* **HTML/CSS/JS Basics**
* **Mongoose** for DB connectivity
* **Git**: [Download](https://gitscm.com/downloads)
* **IDE**: VS Code, Sublime, or WebStorm

## Setup & Installation

1. **Clone Repo**:  
   git clone https://github.com/harsha-vardhan-reddy-07/Flight-Booking-App-MERN
2. **Install Dependencies**:  
   Navigate to repo → npm install
3. **Run App**:  
   npm run dev
4. **Access**:  
   Visit: http://localhost:3000

## Project Structure

* **Client**: React JS frontend
* **Server**: Node.js, Express backend

## Application Flow

* **User**:
  + Registers/Login
  + Searches & books flights
  + Makes payment & cancels bookings
* **Admin**:
  + Manages bookings, flights, and users

## Project Flow

### Milestone 1: Setup

* Create frontend (client) and backend (server) directories
* Install React (Bootstrap, Axios) & Node packages (bcrypt, body-parser, cors, express, mongoose)

### Milestone 2: Backend Development

* Setup MongoDB
* Build Express.js server
* Create APIs for users, flights, bookings
* Implement Mongoose models and authentication
* Add admin functionalities

### Milestone 3: Database

* Create MongoDB Schemas from ER diagram
* Connect DB to backend

### Milestone 4: Frontend

* Login/Register forms
* Flight search & booking modals
* Bookings display & cancel option
* Admin dashboard for managing flights and bookings

### Milestone 5: Implementation

* Run full app
* Test and debug

## Demo Video

[Flight Booking App Demo](https://drive.google.com/file/d/12cVg1Q8Hovdo86cTHOKZGyNa0G0uo8BE/view?usp=drivesdk)

**Happy Coding!**