

# Flight Match

## Airline Management System

### Project Description

Develop a web application that allows users to search for flights, view flight details, and add themselves to a flight manifest. Administrators can manage flights and passenger lists.

### Key Features

- **User Authentication:**
  - Users can sign up, log in, and log out.
- **Flight Search:**
  - Users can search for flights based on origin, destination, date, and other filters.
- **Flight Details:**
  - Display detailed information about selected flights.
- **Manifest Management:**
  - Users can add themselves to a flight's manifest.
  - Users can view and remove themselves from their manifests.
- **Admin Panel:**
  - Admins can create, update, and delete flight entries.
  - Manage passenger lists for each flight.

### Technical Requirements

- **Frontend:**
  - Responsive design using HTML and CSS.
  - Forms for user registration, login, flight search, and manifest management.
  - Client-side validation using JavaScript regex for inputs like email, password strength, flight dates, etc.
- **Backend:**
  - Express.js server with RESTful routes (GET, POST, PUT, DELETE).
  - Data storage using JSON files managed via the `fs` module.
  - Authentication handling (could use simple token-based authentication).
- **Asynchronous Operations:**

- Use `async/await` for all fetch requests and file read/write operations.
- **Additional Features:**
  - Error handling and user-friendly messages.
  - Input sanitization to prevent security issues.

## Project Milestones

### 1. Setup and Authentication:

- Set up the project structure and implement user registration and login.

### 2. Flight Management:

- Develop flight search and display functionality.

### 3. Manifest Functionality:

- Allow users to add/remove themselves from flight manifests.

### 4. Admin Features:

- Implement flight creation and management by admins.

### 5. Final Touches:

- Enhance UI/UX, perform testing, and prepare presentations.