**Flight App**

**Tech Stack:** React - Express - MongoDB – Node JS

### ****Description:****

### The **Flight Management App** is a simple web application that allows users to view, add, edit, and delete flight records. The app provides a straightforward interface where users can see a list of available flights, including key details such as airline, source and destination cities, fare, and flight duration. Users can also add new flight information, modify existing entries, or remove flights they no longer need. The app is designed to be easy to use, helping users manage flight data efficiently without the need for complex features.

### ****Actors:****

1. **User**

**User Operations:**

In the Flight Management App, the **User** is the primary actor who interacts with the system to manage flight records. The user can:

**View Flights:** Users can view a list of all available flights, including details like airline, source and destination cities, fare, and duration.

**Add Flights:** Users have the ability to add new flights to the system by providing the necessary flight details.

**Edit Flights:** Users can edit the information of existing flights, such as updating the airline, fare, or flight duration.

**Delete Flights:** Users can remove any flight records they no longer need from the system.

### ****Table & Collections:****

Collection: flights

Each document in the **flights** collection will contain the following fields:

**id**: A unique number representing the flight's ID.

**airline**: The name of the airline operating the flight.

**source**: The city from which the flight departs.

**destination**: The city where the flight arrives.

**fare**: The price of the flight.

**duration**: The length of the flight in hours and minutes.

### ****MongoDB Database:****

**Database Name**: MyDB (as defined in your connection string).

**Collection Name**: flights.

### ****MongoDB Operations****:

Operations for interacting with the **flights** collection:

1. **Get All Flights**:

Endpoint: /getAllFlights

Operation: Retrieves all flight records from the flights collection.

1. **Insert a Flight**:

Endpoint: /insertFlight

Operation: Adds a new flight document to the flights collection.

1. **Update a Flight**:

Endpoint: /updateFlight/:id

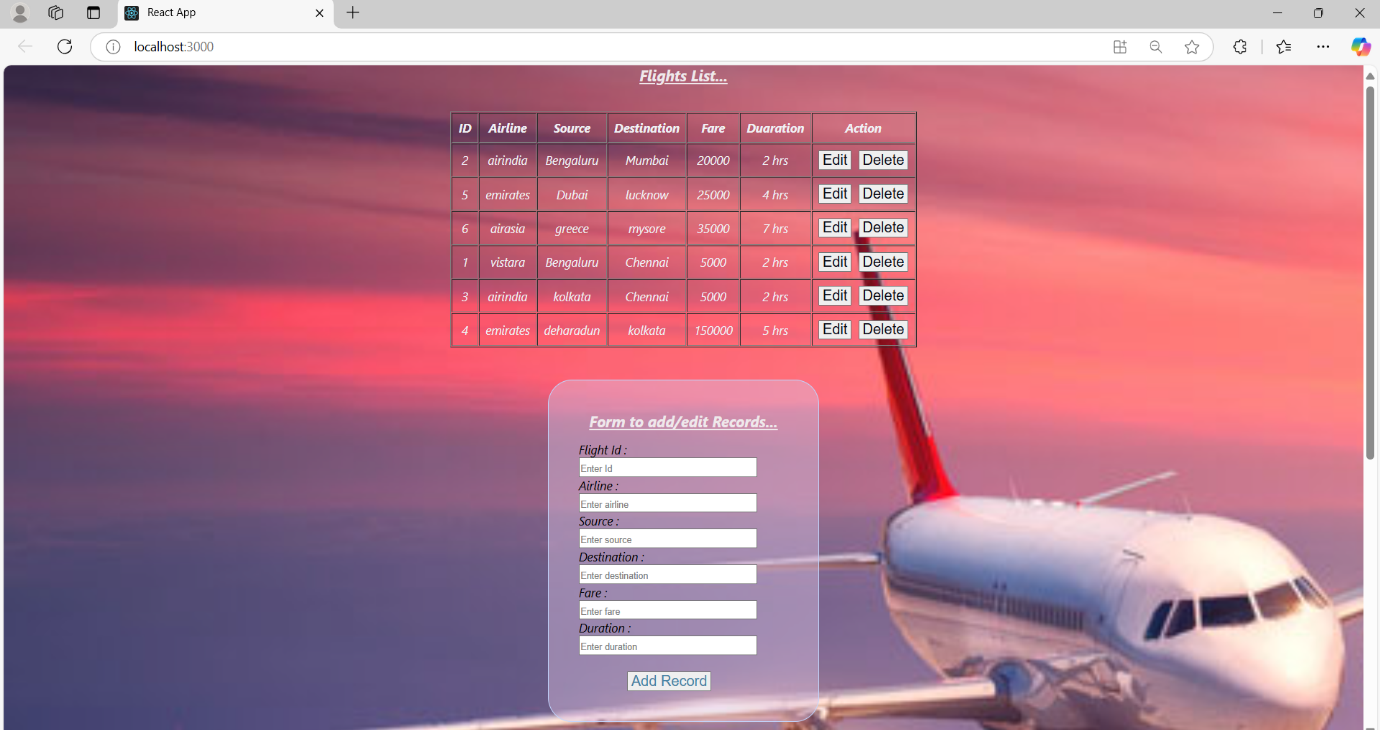
Operation: Updates the flight document identified by the id in flights collection.

1. **Delete a Flight**:

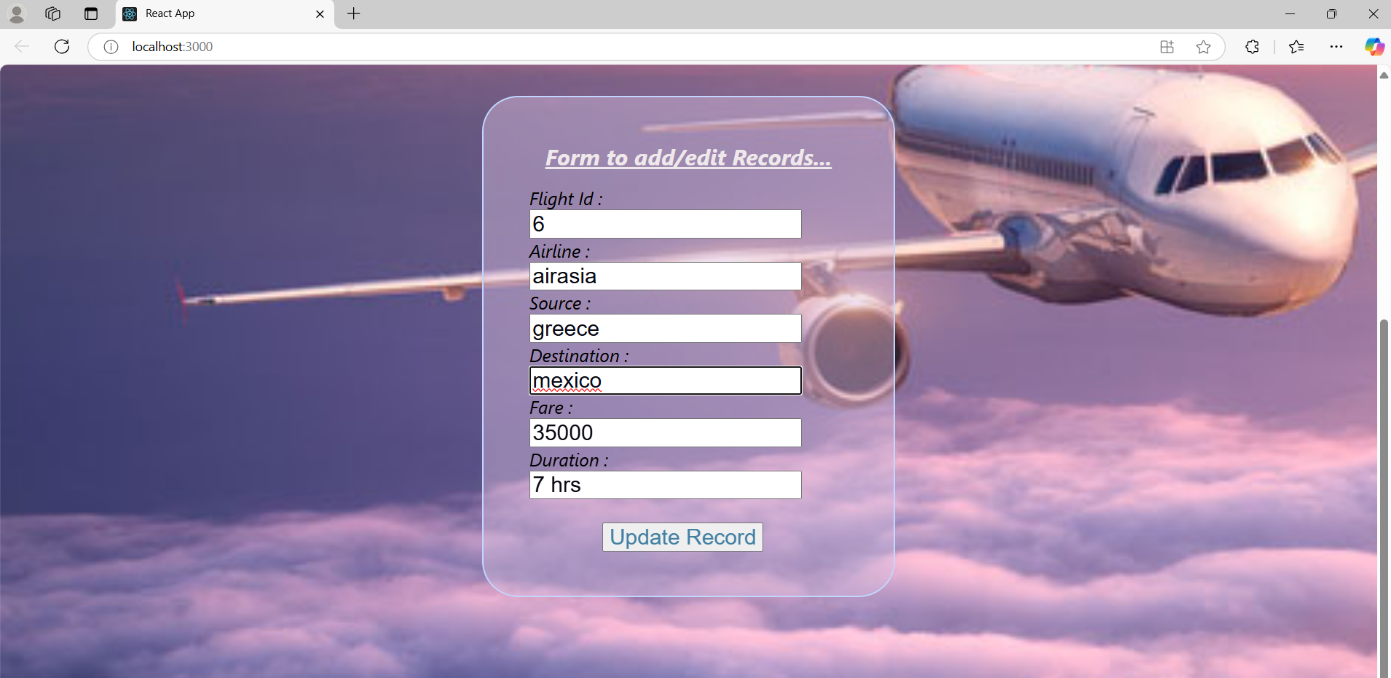
Endpoint: /deleteFlight/:id

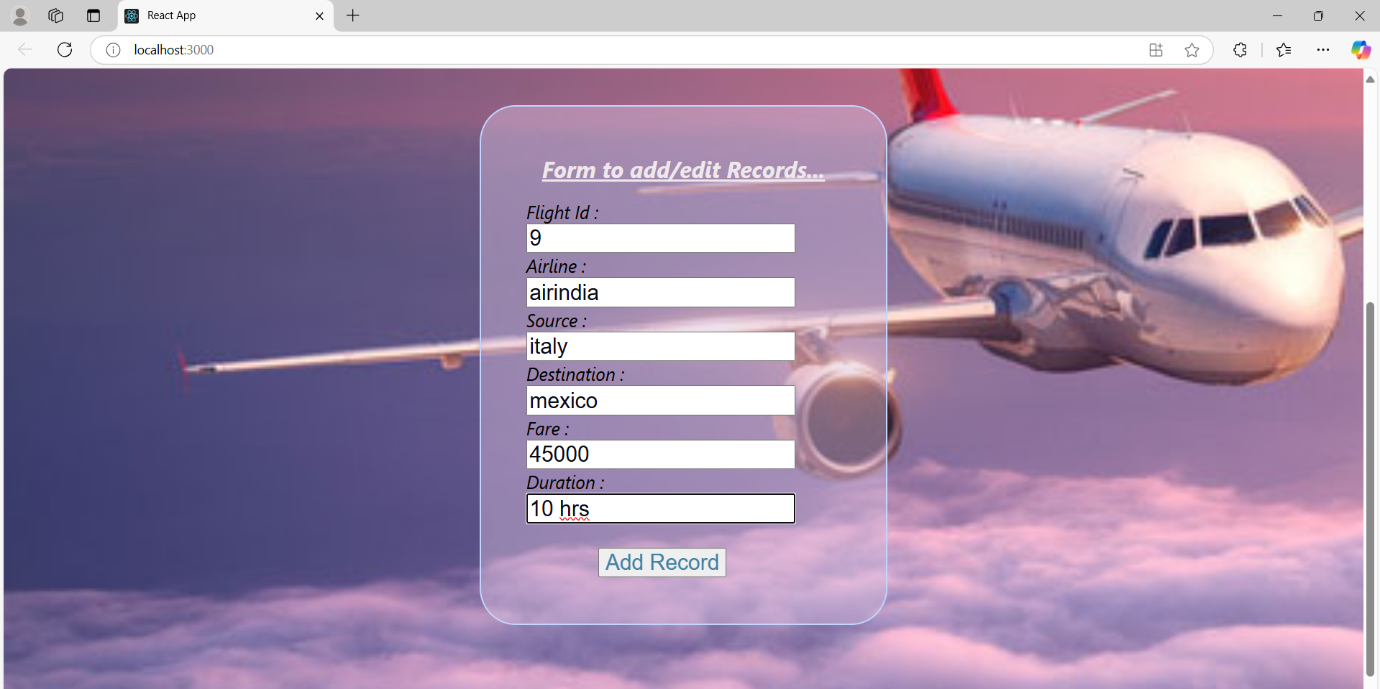
Operation: Deletes the flight document identified by the id in the flights collection.

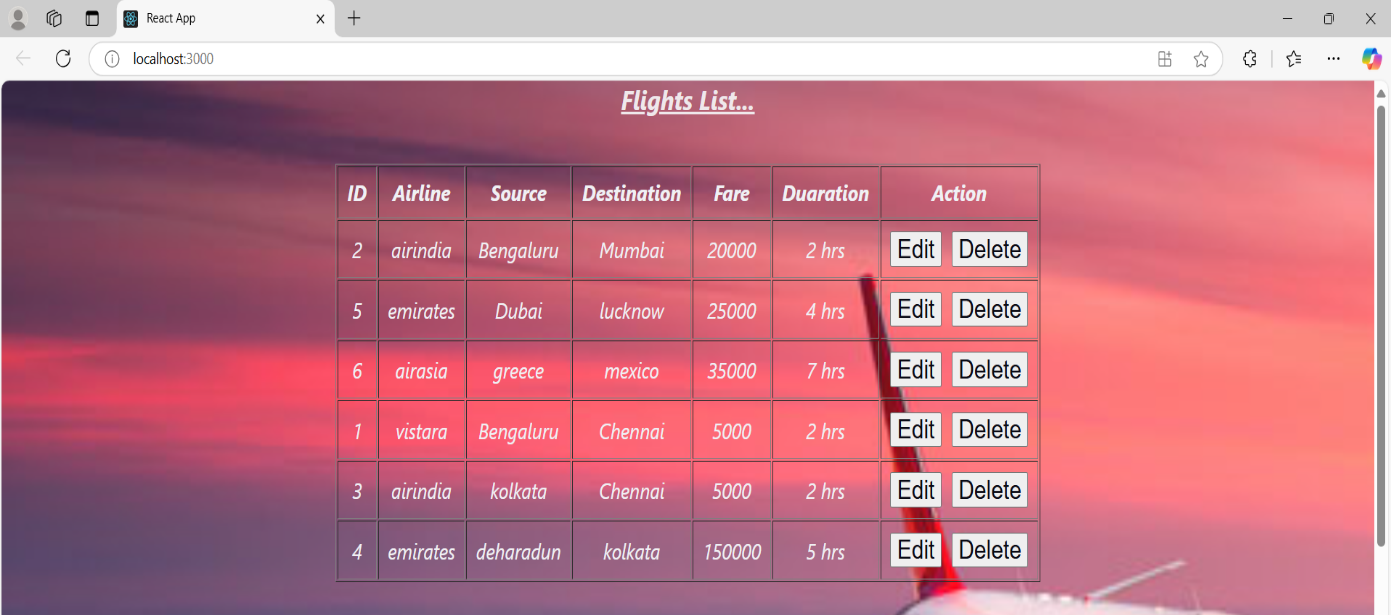
**Screenshots:**

User Interface 

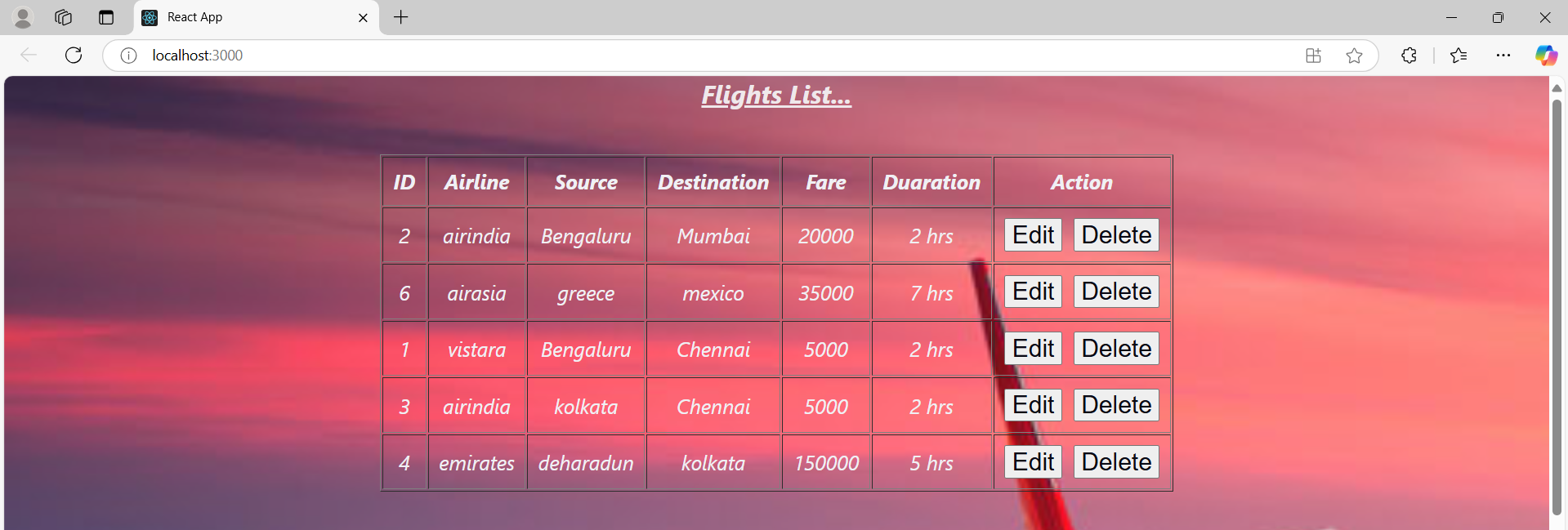
Updating record



Inserting new record

Deleting a record with id=5

View the flights list



**Conclusion:**

**Flight Management App** is a simple yet effective tool designed to manage flight records through basic **CRUD (Create, Read, Update, and Delete)** operations. Built using **MongoDB** for data storage and **Mongoose** for handling database interactions, the app allows users to easily view, add, edit, and delete flight information such as airline, source, destination, fare, and duration. The app is designed with simplicity in mind, making it user-friendly and accessible to anyone who needs to manage flight data. While the current version focuses on core functionalities, it lays a solid foundation for future enhancements, such as adding flight search features or integrating user authentication. This app serves as a practical solution for managing flight records in a small-scale, real-world scenario.