IRCTC-like Example using Higher-Order Functions

Coding Assignment

Objective:

In this coding assignment, you will work on implementing a train ticket booking system similar to IRCTC using

JavaScript Higher-Order Functions. The goal is to filter, transform, and summarize booking data using functions

like `filter`, `map`, and `reduce`.

You will be provided with a list of train bookings, and you will apply filters to get bookings based on specific criteria.

Then, you will use `map` to extract the relevant data and `reduce` to compute a summary (e.g., total available seats).

Functions to Implement:

---

1. filterBookings(bookings, filterCriteria):

   - Filters bookings based on criteria provided in a callback function.

2. getTrainNames(bookings):

   - Uses `map` to get the names of trains from a filtered list of bookings.

3. getTotalAvailableSeats(bookings):

- Uses `reduce` to calculate the total number of available seats from a filtered list of bookings.

Step 1: Sample Data

You are provided with the following sample data representing train bookings:

```
const bookings = [
  { trainName: "Rajdhani Express", destination: "Delhi", classType: "AC", availableSeats: 5 },
  { trainName: "Shatabdi Express", destination: "Mumbai", classType: "Sleeper", availableSeats: 0 },
  { trainName: "Duronto Express", destination: "Kolkata", classType: "AC", availableSeats: 10 },
  { trainName: "Garib Rath", destination: "Delhi", classType: "Sleeper", availableSeats: 15 },
];
```

Step 2: Filtering Bookings

You will implement the `filterBookings` function that takes an array of bookings and a callback function as arguments.

The callback function will define the filtering criteria. For example, filtering bookings to only "Delhi" can be

implemented using a callback function like:

```
const filterByDestination = (booking) => booking.destination === "Delhi";
```

Use this function to filter out bookings that match specific criteria such as:

- Destination (e.g., Delhi)
- Available seats (e.g., availableSeats > 0)

Step 3: Extracting Train Names using Map

After filtering the bookings, you will implement `getTrainNames` to extract the train names from the filtered list.

Use the `map` method to achieve this.

For example:

```
const trainNames = bookings.map((booking) => booking.trainName);
```

Step 4: Summing Available Seats using Reduce

Next, you will implement the `getTotalAvailableSeats` function to calculate the total number of available seats

from the filtered bookings. Use the `reduce` method to sum up the seat numbers.

For example:

```
const totalSeats = bookings.reduce((total, booking) => total + booking.availableSeats, 0);
```