Travel Planner

Array methods

Required methods for the following exercises: `map`, `filter`, `reduce`, `spread`, `destructuring`.

1. **Retrieve Destinations**
    - **Input:** An array of travel destinations.
    - **Task:** Return a new array that contains all the travel destinations.
    - **Example:**
        - Input: `["New York", "London", "Paris"]`
        - Expected Output: `["New York", "London", "Paris"]`
2. **Filter Flights by Destination**
    - **Input:** An array of flight objects, a destination string.
    - **Task:** Return a new array containing only flights with the specified destination.
    - **Example:**
        - Input:
          `[{ destination: "New York" }, { destination: "London" }, { destination: "Paris" }], destination = "London"`
        - Expected Output: `[{ destination: "London" }]`
3. **Calculate Total Flight Duration**
    - **Input:** An array of flight objects.
    - **Task:** Calculate and return the total duration of all flights.
    - **Example:**
        - Input:
          `[{ duration: "3h 30m" }, { duration: "2h 45m" }, { duration: "1h 15m" }]`
        - Expected Output: `7560` (total duration in seconds)

> Helper function for Exercise 3: Get Flight Duration in Seconds

```
const getFlightDurationInSeconds = duration => {
  const [hours, minutes] = duration.split("h ");
  return parseInt(hours) * 3600 + parseInt(minutes) * 60;
};
```

4. **Combine Flight Numbers and Destinations**

   - **Input:** An array of flight objects.
   - **Task:** Return a new array with strings combining the flight number and the destination. Each string should be formatted as "{flightNumber} - {destination}".
   - **Example:**
     - Input: `[{ flightNumber: "FL001", destination: "New York" }, { flightNumber: "FL002", destination: "London" }]`
     - Expected Output: `["FL001 - New York", "FL002 - London"]`

5. **Update Flight Duration**

   - **Input:** An array of flight objects, a new duration value, and a flight number.
   - **Task:** Update the duration of the specified flight in the array and return the updated array.
   - **Example:**
     - Input: `[{ flightNumber: "FL001", duration: "3h 30m" }, { flightNumber: "FL002", duration: "2h 45m" }], newDuration = "4h 15m", flightNumber = "FL001"`
     - Expected Output: `[{ flightNumber: "FL001", duration: "4h 15m" }, { flightNumber: "FL002", duration: "2h 45m" }]`

6. **Filter Flights by Duration**

   - **Input:** An array of flight objects, a minimum duration value.
   - **Task:** Return a new array containing only flights with a duration equal to or greater than the provided value.
   - **Example:**
     - Input: `[{ duration: "3h 30m" }, { duration: "2h 45m" }, { duration: "1h 15m" }], minDuration = 180`

- Expected Output: `[{ duration: "3h 30m" }, { duration: "2h 45m" }]`

7. **Retrieve Flight Numbers**

   - **Input:** An array of flight objects.
   - **Task:** Return a new array that contains all the flight numbers.
   - **Example:**
     - Input:
       ```
       [{ flightNumber: "FL001" }, { flightNumber: "FL002" }, { flightNumber: "FL003" }]
       ```
     - Expected Output: `["FL001", "FL002", "FL003"]`

8. **Sort Flights by Duration**

   - **Input:** An array of flight objects.
   - **Task:** Return a new array sorted by flight duration in ascending order.
   - **Example:**
     - Input:
       ```
       [{ duration: "3h 30m" }, { duration: "2h 45m" }, { duration: "1h 15m" }]
       ```
     - Expected Output:
       ```
       [{ duration: "1h 15m" }, { duration: "2h 45m" }, { duration: "3h 30m" }]
       ```

9. **Retrieve Flight Destinations**

   - **Input:** An array of flight objects.
   - **Task:** Return a new array that contains all the destinations of the flights.
   - **Example:**
     - Input:
       ```
       [{ destination: "New York" }, { destination: "London" }, { destination: "Paris" }]
       ```
     - Expected Output: `["New York", "London", "Paris"]`