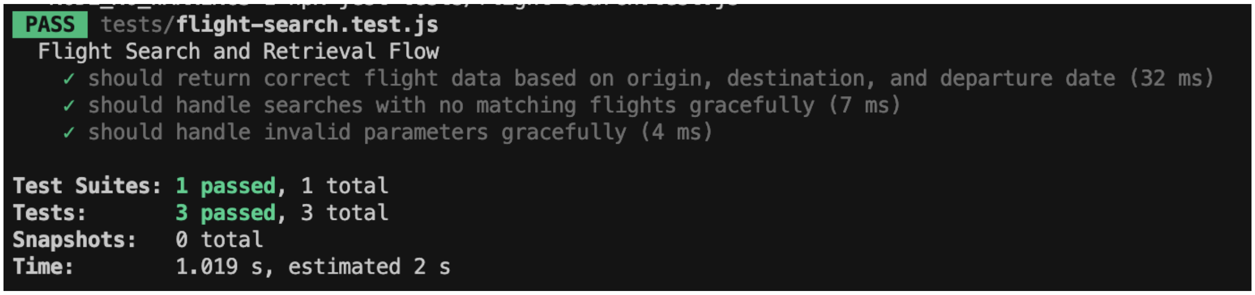
Corey McCann - 21ccm12@queensu.ca
Aatif Mohammad - 22am37@queensu.ca
Benjamin Leray - 21bl45@queensu.ca

All instructions for running the following integration tests are in the readme-first located in the assignment-5 directory.

Flight Search and Retrieval Flow:
This integration test aims to verify the core functionality of searching for flights within the system. It ensures that the application is capable of retrieving available flights based on specific search criteria, such as origin, destination, and departure dates. The test checks the accuracy of the response, ensuring that details like airline and departure time are returned correctly. Furthermore, the test validates the system's behavior when the search input is invalid or does not match any available flights. The main goal is to confirm that the search functionality operates correctly without extending into seat selection or booking steps.

flight-search.test.js

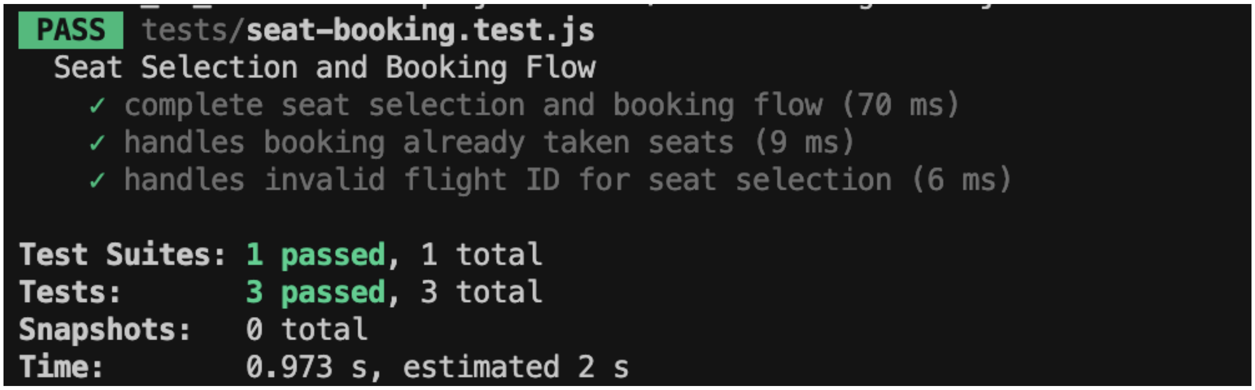


```
 PASS  tests/flight-search.test.js
  Flight Search and Retrieval Flow
    ✓ should return correct flight data based on origin, destination, and departure date (32 ms)
    ✓ should handle searches with no matching flights gracefully (7 ms)
    ✓ should handle invalid parameters gracefully (4 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        1.019 s, estimated 2 s
```

Seat Selection and Booking Flow:
The second integration test focuses on the complete seat selection and booking process. This test picks up where a successful flight search ends. It involves selecting seats for a specific flight and making a booking request via a POST request to the /booking-summary/:id route. The test then verifies whether the selected seats are appropriately marked as unavailable in the database, ensuring data consistency. The primary objective is to validate that the end-to-end process of seat selection and booking works seamlessly and that booking actions are correctly reflected in the database, maintaining accurate records of availability.

seat-booking.test.js

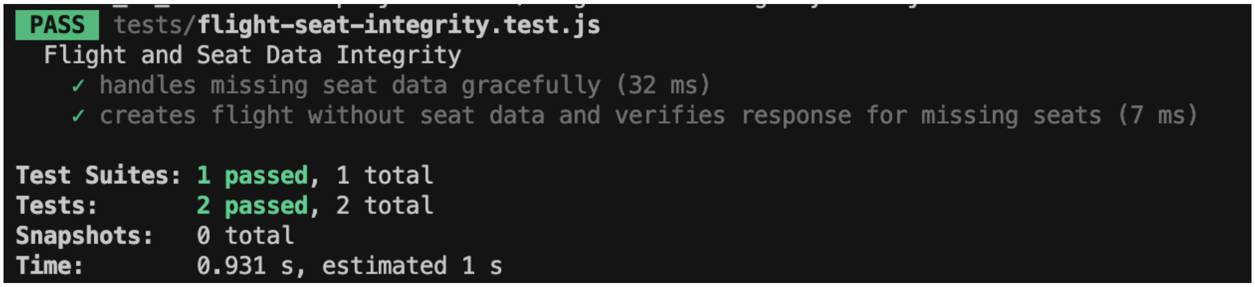


```
 PASS  tests/seat-booking.test.js
  Seat Selection and Booking Flow
    ✓ complete seat selection and booking flow (70 ms)
    ✓ handles booking already taken seats (9 ms)
    ✓ handles invalid flight ID for seat selection (6 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        0.973 s, estimated 2 s
```

Flight and Seat Data Integrity:
The final integration test is designed to verify the system's resilience when there are inconsistencies between flight and seat data. Specifically, this test creates a scenario in which a flight exists without any associated seat data, and it attempts to retrieve seat information for that flight. The focus here is on ensuring that the system handles the missing seat data gracefully, avoiding crashes or incorrect behavior. This test also verifies that the system provides a suitable error message or fallback response for the user, ensuring the integrity between flight details and seat data. The primary goal is to maintain user experience and system reliability in cases of incomplete or missing data.

flight-seat-integrity.test.js



```
 PASS  tests/flight-seat-integrity.test.js
  Flight and Seat Data Integrity
    ✓ handles missing seat data gracefully (32 ms)
    ✓ creates flight without seat data and verifies response for missing seats (7 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.931 s, estimated 1 s
```