

# Integration Test Explanations

## **test\_integration\_page\_navigation.py:**

This test simulates a user's traversal through all of the pages of the site, ensuring that all of the links work and the website can function as a whole. To do this, we first navigate to the home page and check the status and contents of the page. Then for each subsequent redirect, we first check that the link to the next page exists in the current page data and then we have a GET request to that page. This continues until we have traversed through all of the pages on the website in a way that is possible by the user.

## **test\_integration\_viewFlightsDB:**

This test ensures that the flight viewing page and functionality accurately represents what is in the database. First, we issue a GET request to the website itself, getting the content of the page via the application logic. Then we manually create a database connection and query the data directly from the database. Once we have both sources of information, we compare to see that all of the direct database results are present in the page that is rendered by navigating to the view flights page for a given UID.

## **test\_integration\_flightSearch:**

First we test the integration of `get_codes`. We query the flight table to retrieve departure and arrival code values, then call the function and compare the data output. Then we test `read_table` by making a request to the `/read-table` route, and verify that the response status code is 200 (page loaded successfully). Then we query the database again for the departure and arrival code values and check that the rendered HTML response includes all those values. Last, we test the `/search` route by defining some test cases that consist of a departure code, and arrival code, and the expected flight ID for that flight. We simulate a request to the search route with the test data, ensure the response code is 200, and check for the expected results (either the expected flight ID or expected message).

```
3025073100122025 - main - main
[(base) MacBookPro:Group-37-CH nasreen$ pytest -v ]
===== test session starts =====
platform darwin -- Python 3.12.4, pytest-7.4.4, pluggy-1.0.0 -- /opt/anaconda3/bin/python
cachedir: .pytest_cache
rootdir: /Users/nasreen/Group37-CH/Group-37-CH
plugins: anyio-4.2.0
collected 14 items

app/test_db.py::test_create_db PASSED [ 7%]
app/test_flightSearchUnitTests.py::test_read_table_route PASSED [ 14%]
app/test_flightSearchUnitTests.py::test_valid_flight_found PASSED [ 21%]
app/test_flightSearchUnitTests.py::test_no_flight_found PASSED [ 28%]
app/test_flightSearchUnitTests.py::test_same_departure_and_arrival PASSED [ 35%]
app/test_flightSearchUnitTests.py::test_invalid_departure_and_arrival PASSED [ 42%]
app/test_integration_flightSearch.py::test_get_codes_integration PASSED [ 50%]
app/test_integration_flightSearch.py::test_read_table_integration PASSED [ 57%]
app/test_integration_flightSearch.py::test_search_integration PASSED [ 64%]
app/test_integration_page_navigation.py::test_navigation_between_pages PASSED [ 71%]
app/test_integration_viewFlightsDB.py::test_view_flights_integration PASSED [ 78%]
app/test_viewFlightsUnitTests.py::test_view_route PASSED [ 85%]
app/test_viewFlightsUnitTests.py::test_view_id_doesnt_exist PASSED [ 92%]
app/test_viewFlightsUnitTests.py::test_view_0001 PASSED [100%]

===== 14 passed in 0.31s =====
```