CISC 327 GROUP-51

In the test_integration.py file, there are 3 individual integration tests. Respectively they are:

1) wrong user log in -> log in -> pay - **Hassan**
2) log in -> cancel payment -> log back in -> pay- **Sungmoon**
3) create user -> log in -> pay - **Steven**

Test 1: wrong user log in -> log in -> pay

- This function tests the use case where the user initially enters incorrect login information, then retries and enters the correct information and finishes off by completing the payment
- It begins by submitting incorrect login information
- It stays on the login page while displaying a message that informs the user that their login information is wrong
- It submits the correct login information to be taken to the payment page
- It processes the payment information included in the function and confirms that it went through successfully

Test 2: log in -> cancel payment -> log back in -> pay

- This function tests the user experience case of where they log in to pay for the ticket, but decide to cancel payment back to the login page, log back in and make a purchase
- It begins by creating a test user login information
- It sends the test user log in information to confirm that user is logged in
- It imitates user cancelling the purchase by sending /cancel_payment route and confirming that status code matches the redirect and user is in /login route
- It rechecks the log in user information to log the user back in
- It processes the payment provided by the test function and confirms that the right response is returned

Test 3: create user -> log in -> pay

- This function tests the complete workflow of a user creating an account, logging in, and successfully making a payment. It ensures that the /register, /login, and /process_payment endpoints operate as expected, with no conflicts or errors.
- It begins by deleting any existing user with the same email to avoid conflicts.
- It sends information through the /register, /login and /process_payment endpoints respectively with the information situated in the setup. Then, asserts that it has successfully run.
- It is expected that registration, login, and payment will all return HTTP 200 codes to ensure that it is a success.

Steps to running the integration test:

- Move working directory to Assignment-5 folder
- Run the code **python -m unittest discover -s tests -p "test_integration.py"** (quotation marks included) to run the test
- The terminal should indicate that 3 tests have been ran and tested successfully as shown in screenshot below

Task Distribution:

Hassan Hamid:

- Created integration test for incorrect login information, correct login information, and successful payment
- Recorded the results of this integration test
- Helped debug code to allow the three integration tests to work together

Sungmoon Choi:

- Created integration test for logging in, cancelling payment, logging back in, and payment
- Recorded the result of the integration test
- Wrote the instructions to run the integration test

Steven Guan:

- Created integration test for successful registration, logging in and payment
- Added explanation to this test case (test 3) in the 327 Integration Test report document
- Contributed to the instructions for running the test_integration.py