**How to run tests:**

* Make sure you are in the root directory of the folder after creating the virtual environment.
* First you need to run the flask application by typing python app.py in the terminal.
* Then create a new terminal navigate back to the root directory of the folder and activate the virtual environment again: (Activate the virtual environment by running the following command: **source venv/bin/activate** or **venv\Scripts\activate**
* After activating the virtual environment use py name\_of\_test\_script to run the test script

**Register and login testing:**

**login\_register\_endtesting.py**

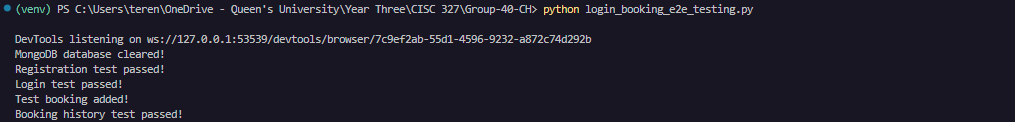
A computer screen with text

Description automatically generated

* Automates the testing of both registration and simultaneous login
* Database is cleared so that if you run the test more than once the user will not already be registered
* Script starts on the /register page locates all required field and submits dummy data
* After submitting it waits for redirection to the /login page and enters the users data there as well
* If we are successful we will land on the / page which is the homepage

**Login and Booking History Testing:**

**login\_booking\_e2e\_testing.py**

****

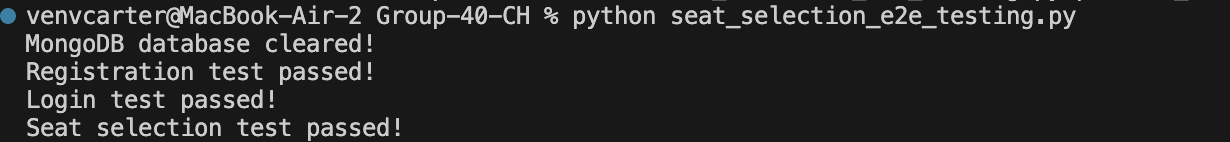
**Features Considered for Testing:**

1. User Registration: Automates the registration process to ensure a new user can register and test with the log in functionality.
2. User Login:  Automates the login process to verify a registered user can log in.
3. Booking Addition: Adds a booking for the test user directly to the database.
4. Booking History: Checking the booking history page displays correct information for the logged-in user.

The scripts start on the registration page to add a new user to the database for testing with the login page. After redirecting to the /login page, the user information is filled out and submitted, leading to the flight search page. A new booking is then added, and a redirection to /booking-history occurs to check that the booking history is properly displayed.

**Register, Login, Seat Selection testing:**

**seat\_selection\_e2e\_testing.py**

****

This script automates the process of:

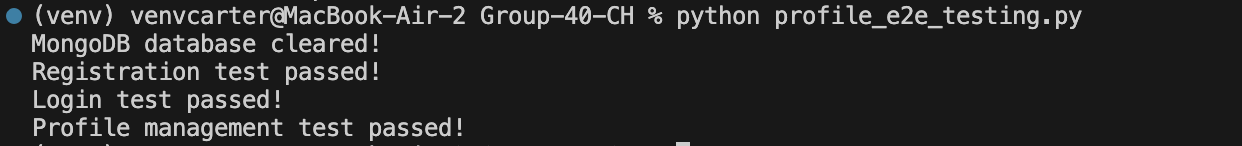
* Clearing the database, navigates to register page
* Registering a new user, navigates to login page
* Logging in the new user, navigates to seat selection page
* There is a time.sleep utilized to help JavaScript have more time to load the seat grid
* Selects available seats for both departure and return flights (if round trip)⁠
* Clicks the proceed button after seat selection
* Verifies successful redirection or alert message after seat selection

Important Testing Notes:

* The script includes proper error handling and cleanup in the finally block
* Uses WebDriverWait to ensure elements are loaded before interacting with them
* Handles both one-way and round-trip scenarios
* Verifies the entire flow from registration through seat selection

**Register, Login, Profile Testing:**

**profile\_e2e\_testing.py**

****

This script automates the process of:

* Clearing the database and navigates to register page
* Registering a new user with test data and navigates to login page
* Logging in the new user and navigates to profile page
* Updates profile information with new test data:
  + New first and last name
  + New email address
  + New password
  + Verifies the profile changes were saved successfully

Important Testing Notes:

* The script includes proper error handling and cleanup in the finally block
* Uses WebDriverWait to ensure form elements are loaded before interacting with them
* Performs field validation by clearing existing data before entering new information
* Includes assertions to verify profile updates were successful
* Tests the complete user flow from registration through profile management