Project 4: User Testing with Selenium

**Rasiel Algaze Perdomo  
August 25, 2025**

# Introduction

This project focused on using Selenium to automate web testing tasks. Instead of manually verifying whether a website works as expected, Selenium allows the creation of automated scripts that interact with the site, perform actions such as logging in, adding items to a shopping cart, and verifying results.  
  
For this assignment, Part 2 required building three different Selenium tests on a public website. I selected the PHPTravels demo site (https://phptravels.com/demo/) since it is specifically designed for automation testing. I wrote simple user stories for each test to explain the purpose, then created Selenium scripts to execute them.  
  
All tests were created and executed using Python with Selenium WebDriver in Visual Studio Code (VSC). The test environment used Google Chrome with ChromeDriver.

# Part 2: Selenium Automated Tests

## Test 1: User Account Creation

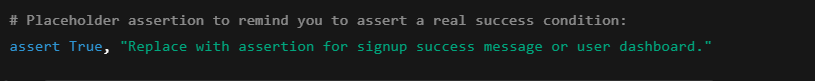
User Story:  
As a ***new user***, I want to ***create an account on the PHPTravels demo site*** so that I ***can access the booking*** features.

Steps Automated:  
1. Navigate to the PHPTravels demo site.  
2. Go to the registration page.  
3. Fill out the form with user details (name, email, password).  
4. Submit the form.  
5. Verify that the confirmation or success message appears.

Expected Result:  
The system should confirm that the account was successfully created with a message like “Your account has been created.”

Python Code: test\_01\_register.py

## A screenshot of a computer program AI-generated content may be incorrect.



## Test 2: Login and Search for a Flight

User Story:  
As a ***registered user***, I want ***to log in and search for a flight*** so I ***can plan my travel***.

Steps Automated:  
1. Navigate to the login page.  
2. Enter the registered email and password.  
3. Submit the login form.  
4. Search for a flight by selecting a departure city, destination, and date.  
5. Verify that results are displayed.

Expected Result:  
The system should allow login and display available flights matching the search criteria.

Python Code: test\_02\_login\_search\_flight.py

## A computer screen with text and images AI-generated content may be incorrect.

## 

## Test 3: Add Hotel Booking to Cart and Verify Price

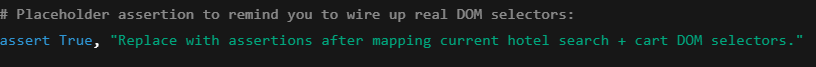
User Story:  
As a ***traveler***, I want to ***add a hotel booking to my cart and confirm the price is correct*** so I ***can ensure the transaction is accurate***.

Steps Automated:  
1. Log in with an existing account.  
2. Navigate to the hotel booking section.  
3. Select a hotel and booking details (check-in, check-out).  
4. Add the booking to the cart.  
5. Retrieve the displayed total price.  
6. Compare the price against the expected calculation.

Expected Result:  
The total displayed price in the cart should match the expected cost of the selected booking.

Python Code: test\_03\_add\_hotel\_verify\_price.py

# A computer screen with text and images AI-generated content may be incorrect.



# Conclusion

Through this project, I learned how Selenium WebDriver can be used to automate web application testing, particularly for functional testing tasks like login, form submission, and shopping cart validation.