

Automation Testing using Behave and Selenium

Introduction

This project aims to automate the testing of a web application using Behave and Selenium. The goal is to create a set of tests that can be run automatically to verify the functionality of the application.

Approach

To complete this project, I used Behave, a BDD (Behavior-Driven Development) testing framework, and Selenium, a browser automation tool. I wrote a set of feature files that define the test scenarios, and step definitions that implement the test steps. The tests were run using the `run_tests.sh` script, which executes the Behave tests and generates the test results.

Tools and Technologies Used

- Behave: A BDD testing framework for Python
- Selenium: A browser automation tool for Python
- Python: The programming language used to write the test scripts
- Git: A version control system used to manage the code repository

Challenges and Issues

During the project, I encountered some challenges and issues, including:

- Setting up the Selenium environment and configuring the browser drivers
- Writing effective test scripts that cover all the required scenarios
- Debugging and troubleshooting test failures

How to Run the Program

To run the program, follow these steps:

1. Clone the Git repository containing the code and test scripts.
2. Install the required dependencies, including Behave and Selenium.

3. Run the **run_tests.sh** script to execute the tests.
4. Review the test results to verify that all tests passed successfully.

Interpreting the Test Results

The test results are displayed in a hierarchical format, with each scenario and step shown in a tree-like structure. The results indicate whether each test passed or failed, and provide information on the number of tests executed, passed, and failed.

Bugs and Defects

During testing, I identified some bugs and defects in the application, including:

- The login functionality does not handle invalid usernames and passwords correctly.
- The order functionality does not validate the user's input correctly.

Conclusion

In this project, I successfully automated the testing of a web application using Behave and Selenium. The tests cover various scenarios, including login functionality and order functionality. The project demonstrates the effectiveness of automation testing in identifying bugs and defects in the application.