AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
ADDING VALUE TO ENGINEERING

Department of Computer Engineering

A SOFTWARE TESTING AND QUALITY ASSURANCE

PROJECT REPORT ON

E-Commerce Website Automation Testing Using Selenium

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERINGAISSMS IOIT

BE Computer Engineering

SUBMITTED BY

STUDENT NAME	**ERP No:**
Onasvee Banarse	**09**
Kaustubh Kabra	**37**
Akash Mete	**50**
Harsh Shah	**65**

सत्याला मरण नाही

2022 -2023

AISSMS
INSTITUTE OF INFORMATION TECHNOLOGY
ADDING VALUE TO ENGINEERING

Department of Computer Engineering

CERTIFICATE

This is to certify that the project report
“E-Commerce Website Automation Testing Using Selenium”

Submitted by

STUDENT NAME	ERP No:
Onasvee Banarse	09
Kaustubh Kabra	37
Akash Mete	50
Harsh Shah	65

is a bonafide students at this institute and the work has been carried out by them under the supervision of **Prof. Prajwal Gaikwad** and it is approved for the partial fulfillment of the Department of Computer Engineering AISSMS IOIT.

(**Prof. Prajwal Gaikwad**) (**Dr. S.N.Zaware**)

Mini-Project Guide Head of Computer Department

Place: Pune Date:

Abstract

Tests with the purpose of validating the product works are named clean tests, or positive tests. The drawbacks are that it can only validate that the software works for the specified test cases. A finite number of tests cannot validate that the software works for all situations. On the contrary, only one failed test is sufficient enough to show that the software does not work. Dirty tests, or negative tests, refers to the tests aiming at breaking the software, or showing that it does not work. A piece of software must have sufficient exception handling capabilities to survive a significant level of dirty tests.

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language. In order to create a small web-based application by selecting relevant system environment we need selenium web driver and IDE. In order to gain better understanding and familiarity about selenium , we will target three things: Java, Selenium Webdriver, Testing to learn testing with Selenium. we will Identify the bugs using Selenium Web-Driver and IDE and we will generate test reports by testing.The tests can then run against most modern websites. Here we will test it on the login pages like instagram ,twitter,
facebook etc. There are many types of testing that can be done with Selenium. You can do smoke testing, sanity, testing, UI testing, regression testing, and more.For most testing scenarios, Selenium Automation Testing is considered an ideal option. But, when it comes to unit testing, developers need a modern automated unit testing framework that can create automated unit test cases and quickly integrates with Selenium.

Contents

1. Introduction

Software testing is a set of activities conducted for finding errors in software. It is a process used to measure the quality of the software. Manual testing and automation testing are the two ways of testing. Manual testing is also called as static testing. It is carried out by the tester. Automation testing is also called as dynamic testing. But the problem is it is very time consuming process and requires more effort. So, automation testing is used to solve these problems. Automated testing is divided into four types such as reliability testing, security testing, correctness testing, and performance testing. It automates the steps of manual testing using automation tools . Automated tests are fast to execute and they are repeatable in nature. There are various tools available in the market which are used to test the process and targeted specific test environment. The environment may be functional, performance or exceptional testing etc. Testing tool should be selected on the basis of its compatibility with checklist for that purpose pilot round of the corresponding tool should be done. Cost is also an important factor for selection of tool.

2. Problem Statement

Create a small web-based application by selecting relevant system environment platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDE and generate test reports encompassing exploratory testing.

In this Project we will focus on following questions:

- to provide a reliable and easy to use software tool that allows test creation and test taking capabilities.
- to provide a database that stores questions, which can be used to automatically generate tests based on specified criteria
- to provide a secure test taking application and powerful automated grading tool and non-automated grading capabilities

3. Software and Hardware Requirement Specification

Software Used:

- Python (version 3 or above)-

  Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. The sentiment analysis is performed using python language and packages.

- VScode or any IDE–

  Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Hardware Used:

The detailed hardware used for the project are:

Item	Description
System	Asus TUF A15
Processor	AMD Ryzen 5 4600H
RAM	8 GB
System Type	64-bit operating system, x64-based processor
SSD	512 GB Solid State Drive
HDD	1 TB Hard Disk Drive
Graphics	NVIDIA 4 GB Graphic Card
Operating System	Windows 10 Operating System

5. Theory

What is Selenium?

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. Selenium is a suite of software tools to automate Web Browsers. It is an Open-source suite of tools mainly used for Functional and Regression Test Automation. Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred to as Selenium Testing

● Selenium supports various Operating environments.
Microsoft Windows
Linux
Macintosh

● Selenium supports various Browsers.
Mozilla Firefox
IE
Google Chrome
Safari
Opera etc…

● Selenium supports various programming environments to write programs (Test scripts)
Java
C#
Python
Perl
Ruby
PHP

History of the Selenium Project

Selenium first came to life in 2004.

● In 2006, Selenium WebDriver was launched at Google.

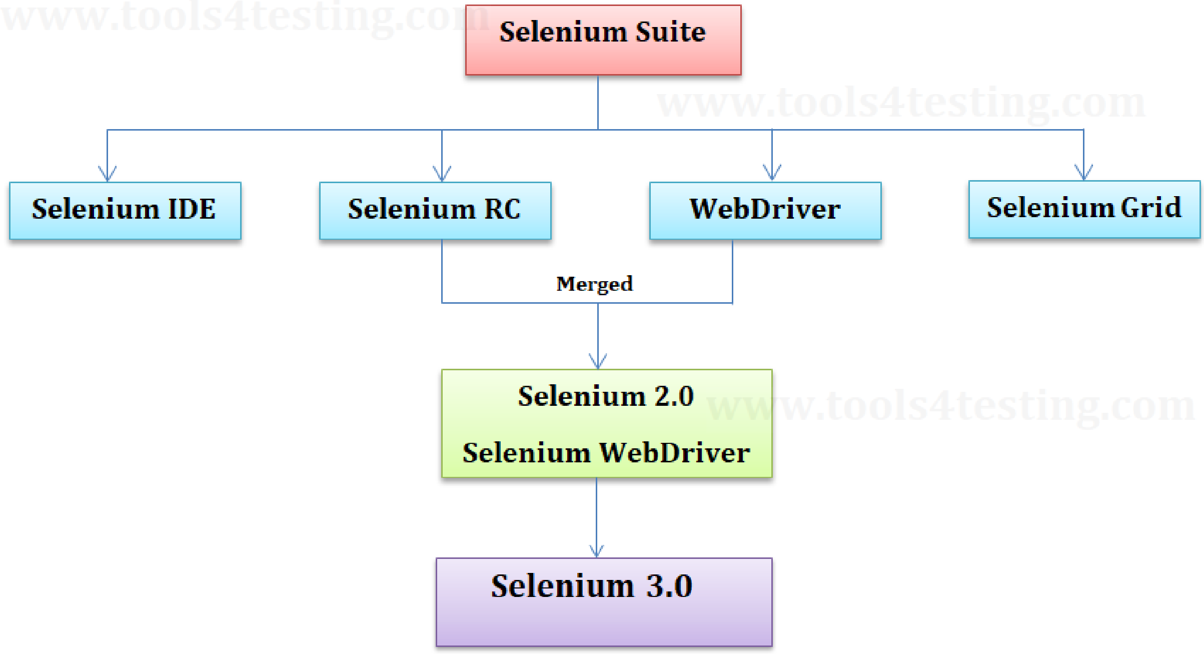
● In 2008, the whole Selenium team decided to merge Selenium WebDriver with Selenium RC to form a more powerful tool called Selenium 2.0

Selenium 1

■ (Selenium IDE + Selenium RC + Selenium Grid)

○ Selenium 2

■ (Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)



Selenium' Tools Suite

Selenium is not just a single tool but a suite of software's, each catering to different testing needs of an organization. It has four components.

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid

Selenium IDE Features :-

Create Test Cases, Test suites (We can Record test cases or type Test steps using element locators and Selenese commands)

- Edit Test Cases
- Execute Test cases, Test suites
- Debug Test Cases.
- Enhance Test Cases

Drawbacks of Selenium IDE :-

- It supports Mozilla Firefox browser only.
- It doesn’t support Programming logic/features to enhance Test cases.
- It doesn’t support Data-Driven Testing.
- It is not suitable for complex test case design.
- No centralized maintenance of Objects/Elements

Selenium Remote Control (Selenium RC) :-

Selenium RC was the flagship testing framework of the whole Selenium project for a long time. This is the first automated web testing tool that allowed users to use a programming language they prefer. As of version 2.25.0, RC can support the following programming languages:

- **J**ava
- C#
- PHP
- Python
- Perl
- Ruby

Selenium WebDriver :-

- It is a Programming interface to create and execute Test cases.
- Selenium IDE has IDE but doesn’t have Programming interface.
- Selenium WebDriver has Programming interface but doesn’t have IDE.
- It communicates directly to the browser.
- No need of Separate Server such as RC ServerUFT/QTP has both IDE as well as a Programming interface.
- Faster Execution than IDE & RC
- Selenium WebDriver supports various programming environments to write programs.
    - Java
    - C#
    - Perl
    - Python
    - Ruby
    - PHP

- Using Element/Object locators/properties and Webdriver Methods we can create and execute Test cases.
- Selenium Webdriver supports various browsers to create and execute a test case/test script/test

Note: Browser driver varies from one browser to another.

Drawbacks of Selenium WebDriver

- It doesn’t generate detailed Test Reports.
- No centralized maintenance of Object/elements
- It requires Programming Knowledge
- Cannot support the readily new browser
- Installation is More Complicated than Selenium IDE

No built-in mechanism for logging runtimemessage

Selenium Grid :-

- Selenium Grid is used to execute tests across multiple browsers, operating environments and machines in parallel.
- Selenium Grid 2 supports Selenium RC Tests as well as Selenium WebDriver Tests.
- Selenium WebDriver to create Test cases using Element locators and Webdriver methods.
- Java Programming to enhance test cases.
- TestNG Framework to group test cases, execute test batches and generate detailed test reports.

Features:

- Enables simultaneous running of tests in multiple browsers and environments.
- Saves time enormously.
- Utilizes the hub-and-node concept. The hub acts as a central source of Selenium
- commands to each node connected to it.

Memory usage is another important aspect of sorting algorithms, which in some cases can be more important than the execution time. Ideally, one wants to keep the value for this parameter small, but extra memory can often improve execution times, which leads to a time-space trade-off. One cannot have both, and therefore needs to find a reasonable compromise for what is best for the intended usage. For this reason, in-place sorting algorithms are often slower than its counterpart.

6. TEST PLAN

Features to be tested:-

- Open the given e-commerce website
- Select a product with quantity more than 1 and add to cart
- Select a product and update the delivery date and add to cart
- View cart and checkout as a guest
- Fill in all the delivery and checkout details
- Confirm order and check for the order confirmation message

Test Environment:-

• Environment

Windows 10
Chromedriver
Selenium
Python
Pycharm
VSCode

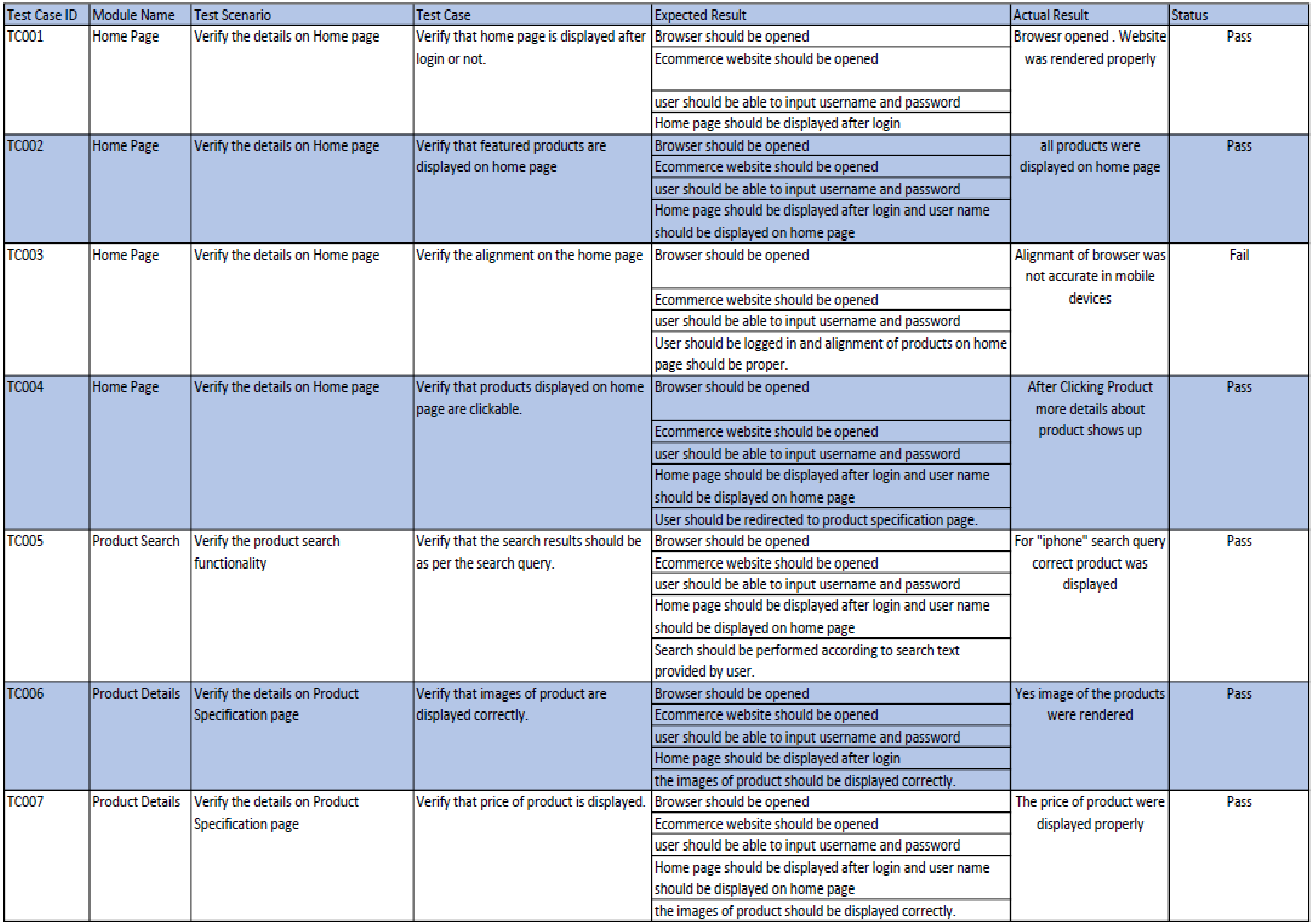
• Programming Languages

HTML
CSS
Python

• Testing tool used

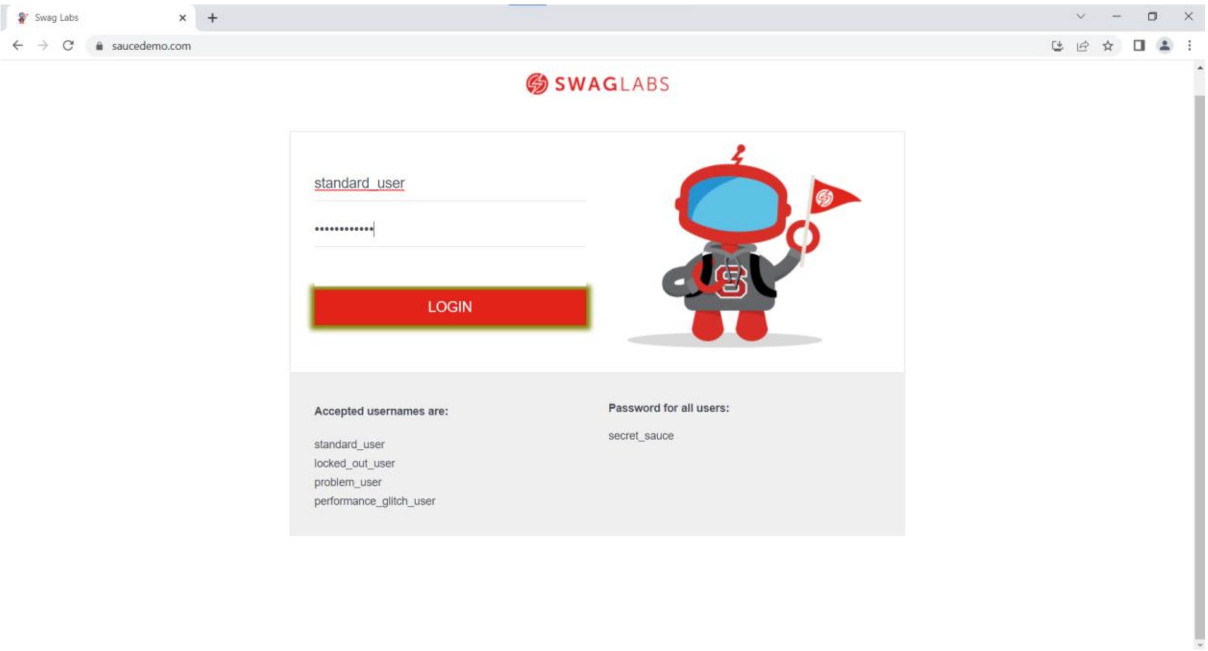
Selenium

Test Case Scenarios:-



Test Case ID	Module Name	Test Scenario	Test Case	Expected Result	Actual Result	Status
TC001	Home Page	Verify the details on Home page	Verify that home page is displayed after login or not.	Browser should be opened Ecommerce website should be opened user should be able to input username and password Home page should be displayed after login	Browesr opened . Website was rendered properly	Pass
TC002	Home Page	Verify the details on Home page	Verify that featured products are displayed on home page	Browser should be opened Ecommerce website should be opened user should be able to input username and password Home page should be displayed after login and user name should be displayed on home page	all products were displayed on home page	Pass
TC003	Home Page	Verify the details on Home page	Verify the alignment on the home page	Browser should be opened Ecommerce website should be opened user should be able to input username and password User should be logged in and alignment of products on home page should be proper.	Alignmant of browser was not accurate in mobile devices	Fail
TC004	Home Page	Verify the details on Home page	Verify that products displayed on home page are clickable.	Browser should be opened Ecommerce website should be opened user should be able to input username and password Home page should be displayed after login and user name should be displayed on home page User should be redirected to product specification page.	After Clicking Product more details about product shows up	Pass
TC005	Product Search	Verify the product search functionality	Verify that the search results should be as per the search query.	Browser should be opened Ecommerce website should be opened user should be able to input username and password Home page should be displayed after login and user name should be displayed on home page Search should be performed according to search text provided by user.	For "iphone" search query correct product was displayed	Pass
TC006	Product Details	Verify the details on Product Specification page	Verify that images of product are displayed correctly.	Browser should be opened Ecommerce website should be opened user should be able to input username and password Home page should be displayed after login the images of product should be displayed correctly.	Yes image of the products were rendered	Pass
TC007	Product Details	Verify the details on Product Specification page	Verify that price of product is displayed.	Browser should be opened Ecommerce website should be opened user should be able to input username and password Home page should be displayed after login and user name should be displayed on home page the images of product should be displayed correctly.	The price of product were displayed properly	Pass

7. Output



Swag Labs

saucedemo.com

SWAGLABS

standard_user

••••••••••••

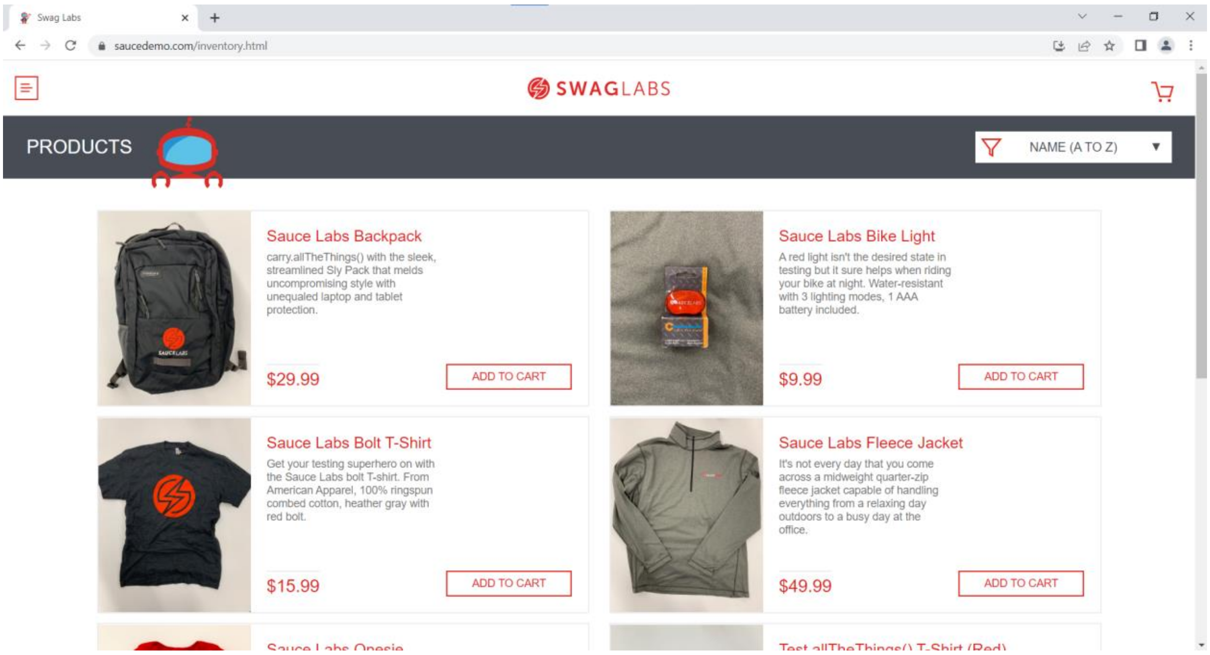
LOGIN

Accepted usernames are:

standard_user
locked_out_user
problem_user
performance_glitch_user

Password for all users:

secret_sauce



Swag Labs

saucedemo.com/inventory.html

SWAGLABS

PRODUCTS

NAME (A TO Z) ▼

Sauce Labs Backpack

carry.allTheThings() with the sleek, streamlined Sly Pack that melds uncompromising style with unequaled laptop and tablet protection.

$29.99 ADD TO CART

Sauce Labs Bike Light

A red light isn't the desired state in testing but it sure helps when riding your bike at night. Water-resistant with 3 lighting modes, 1 AAA battery included.

$9.99 ADD TO CART

Sauce Labs Bolt T-Shirt

Get your testing superhero on with the Sauce Labs bolt T-shirt. From American Apparel, 100% ringspun combed cotton, heather gray with red bolt.

$15.99 ADD TO CART

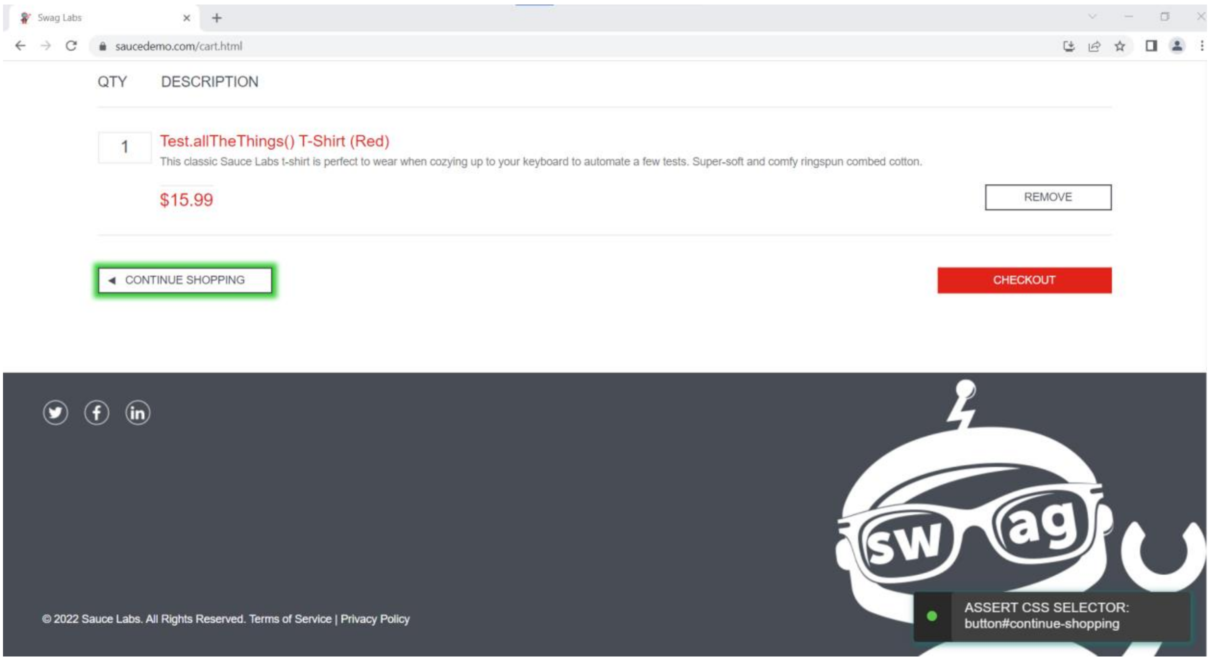
Sauce Labs Fleece Jacket

It's not every day that you come across a midweight quarter-zip fleece jacket capable of handling everything from a relaxing day outdoors to a busy day at the office.

$49.99 ADD TO CART

Sauce Labs Onesie

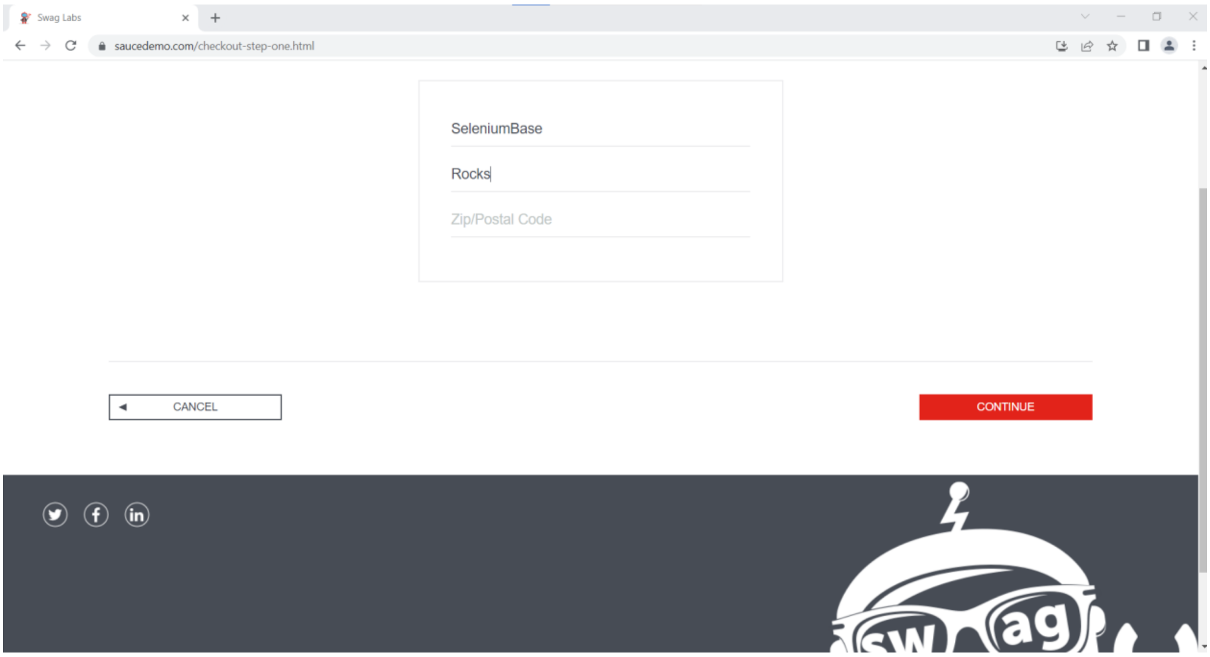
Test.allTheThings() T-Shirt (Red)



QTY	DESCRIPTION
1	Test.allTheThings() T-Shirt (Red) This classic Sauce Labs t-shirt is perfect to wear when cozying up to your keyboard to automate a few tests. Super-soft and comfy ringspun combed cotton. $15.99

© 2022 Sauce Labs. All Rights Reserved. Terms of Service | Privacy Policy

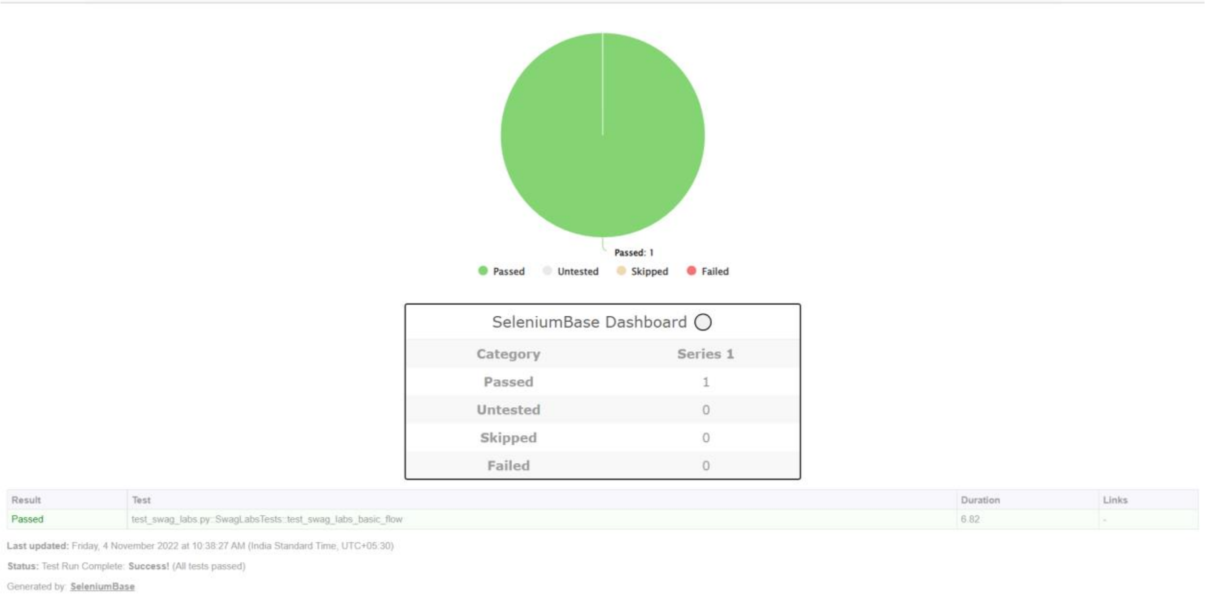
ASSERT CSS SELECTOR:
button#continue-shopping



SeleniumBase

Rocks

Zip/Postal Code



SeleniumBase Dashboard ○

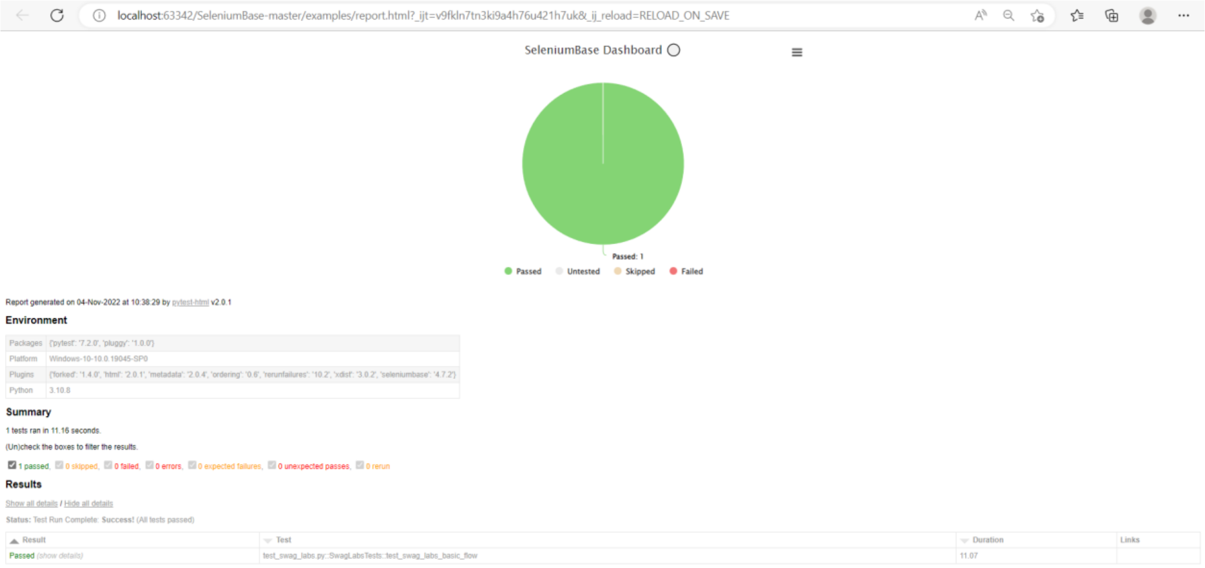
Category	Series 1
Passed	1
Untested	0
Skipped	0
Failed	0

Result	Test	Duration	Links
Passed	test_swag_labs.py::SwagLabsTests::test_swag_labs_basic_flow	6.82	-

Last updated: Friday, 4 November 2022 at 10:38:27 AM (India Standard Time, UTC+05:30)

Status: Test Run Complete: **Success!** (All tests passed)

Generated by: **SeleniumBase**


localhost:63342/SeleniumBase-master/examples/report.html?_ijt=v9fkln7tn3ki9a4h76u421h7uk&_ij_reload=RELOAD_ON_SAVE
SeleniumBase Dashboard ○
Passed: 1
● Passed ● Untested ● Skipped ● Failed

Report generated on 04-Nov-2022 at 10:38:29 by pytest-html v2.0.1

Environment

Packages	{'pytest': '7.2.0', 'pluggy': '1.0.0'}
Platform	Windows-10-10.0.19045-SP0
Plugins	{'forked': '1.4.0', 'html': '2.0.1', 'metadata': '2.0.4', 'ordering': '0.6', 'rerunfailures': '10.2', 'xdist': '3.0.2', 'seleniumbase': '4.7.2'}
Python	3.10.8

Summary

1 tests ran in 11.16 seconds.

(Un)check the boxes to filter the results.

☑ 1 passed, ☑ 0 skipped, ☑ 0 failed, ☑ 0 errors, ☑ 0 expected failures, ☑ 0 unexpected passes, ☑ 0 rerun

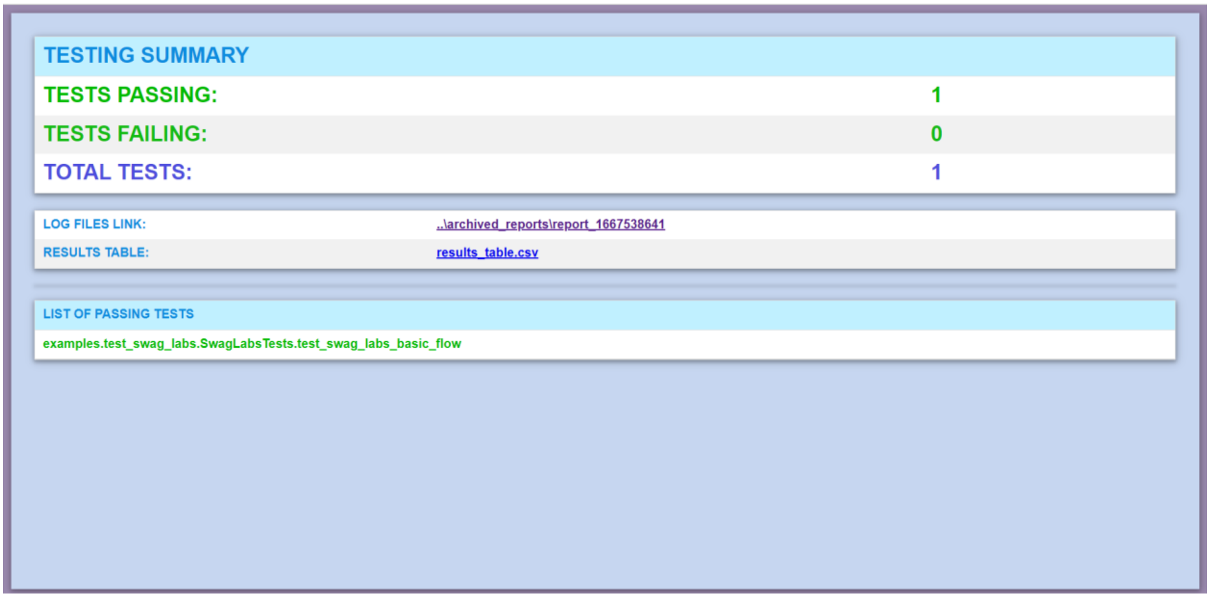
Results

Show all details / Hide all details

Status: Test Run Complete: **Success!** (All tests passed)

Result	Test	Duration	Links
Passed *(show details)*	test_swag_labs.py::SwagLabsTests::test_swag_labs_basic_flow	11.07	

NoseTests Report:



TESTING SUMMARY	
TESTS PASSING:	1
TESTS FAILING:	0
TOTAL TESTS:	1

LOG FILES LINK:	..\archived_reports\report_1667538641
RESULTS TABLE:	results_table.csv

LIST OF PASSING TESTS
examples.test_swag_labs.SwagLabsTests.test_swag_labs_basic_flow

8. Conclusion

Test cases are important document for future prospective. Quality assurance team is the review of software products and related documentation for completeness, correctness, reliability and maintainability. It also includes assurance, that the system meets the specification and the requirements for its intended use and performance. In this research paper, I present an overview of our experience through a case study and also provide some new techniques. Today, most of the programmer/test engineers have face many problems regarding test case documentation.

In our project we studied concept of STLC and implemented on testing website.

9. References

[1] Ramya, Paruchuri; Sindhura, Vemuri; Sagar, P. Vidya (2017). [IEEE 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) - Coimbatore (2017.2.22-2017.2.24)] 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) - Testing using selenium web driver. , (), 1–7. doi:10.1109/ICECCT.2017.8117878

[2] Barab, S. & Squire, K. 2004. Design-based Research: Putting a Stake in the Ground. The Journal of the Learning Sciences, 13(1), 1 –14.

[3] Dustin, E., Garrett, T. & Gauf, B. 2009. Books on Google Play Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality. New York City, New York, United States: Pearson Education Inc.

[4] Itkonen, J., Mäntylä, M.V. & Lassenius, C. 2009. How do testers do it? An exploratory study on manual testing practices, in Proceedings of 3rd International Symposium on Empirical Software Engineering and Measurement.

[5] Peffers, K., Tuunanen, T.A., Rothenberger, M. & Chatterjee, S. 2008. A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems. Abingdon: M.E. Sharpe, Inc.

[6] Westby, E.J.H. 2015. Git for Teams: A User-Centered Approach to Creating Efficient Workflows in Git. Sebastopol: O’Reilly Media, Inc.

[7] Selenium:
https://www.selenium.dev/

[8] Selenium with Python Tutorial:
https://www.javatpoint.com/selenium-python