Haroon Rashid

Registration No 16549

Semester 6th

Paper: Software Verification and Validation

Submitted To: Sir Zain Shaukat

Selenium Testing Tool

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional test without the need to learn a test scripting language (Selenium IDE). The test can then run against most modern web browsers. Selenium runs on Windows, Linux, and mac OS.

Part(a)

pros/cons

Pros:

- Open source, active contributions 24/7
- Wide range of supported languages
- Lots of online resources
- Good multi browser support / Parallel testing
- Excellent for what it does (when you learn to implement it properly to prevent flaky/brittle tests)
- Easy integration with the likes of Maven/TestNG/Jenkins etc.
- Multi OS support
- Lightweight in terms of resourcing compared to some alternative tools

Cons:

- Can be a tad harder to use than some other things, Selenium IDE isn't great.
- Web based stuff only, e.g. no windows applications.
- Cannot interact with everything in the browser, e.g. some pop out dialogs require additional handling elsewhere.
- No official support channels.
- Not ideal for image testing stuff.

Part(b)

Functionality

- It also supports parallel test execution which reduces time and increases the efficiency of tests.
- Selenium can be integrated with frameworks like Ant and Maven for source code compilation.
- Selenium can also be integrated with testing frameworks like TestNG for application testing and generating reports.
- Selenium requires fewer resources as compared to other automation test tools.

- WebDriver API has been indulged in selenium which is one of the most important modifications done to selenium.
- Selenium web driver does not require server installation, test scripts interact directly with the browser.
- Selenium commands are categorized in terms of different classes which make it easier to understand and implement.
- Selenium Remote Control (RC) in conjunction with WebDriver API is known as Selenium 2.0. This version was built to support the vibrant web pages and Ajax.

Part(c)

Supporting Languages

- Java
- Python
- Ruby
- C#
- JavaScript
- Perl
- PHP

Part(d)

Supporting Tests

- ### Functional Testing

  FUNCTIONAL TESTING is a type of software testing whereby the system is tested against the functional requirements/specifications. Functions (or features) are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properly satisfied by the application.

- ### Regression Testing

  Regression testing is re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change. If not, that would be called a regression.

- ### Sanity Testing

  A sanity test or sanity check is a basic test to quickly evaluate whether a claim or the result of a calculation can possibly be true. It is a simple check to see if the produced material is rational

- **Smoke Testing**

  In computer programming and software testing, smoke testing is preliminary testing to reveal simple failures severe enough to, for example, reject a prospective software release

- **Responsive Testing**

  Responsive web design (RWD) is an approach to web design that makes web pages render well on a variety of devices and window or screen sizes. Responsive layouts automatically adjust and adapt to any device screen size, whether it is a desktop, a laptop, a tablet, or a mobile phone.

- **Cross Browser Testing**

  Cross Browser testing is a type of non-functional testing that lets you check whether your website works as intended when accessed through.

- **UI Testing (black box)**

  BLACK BOX TESTING is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications

- **Integration Testing**

  INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

Part(e)

Write a short(faulty) code, Test using this tool, and show the bugs in the code.

Answer

1. **Google test search (automated search through code)**
2. **Add implicit wait of 10 second**
3. **maximiser window**
4. **I used selenium python in pychrm**

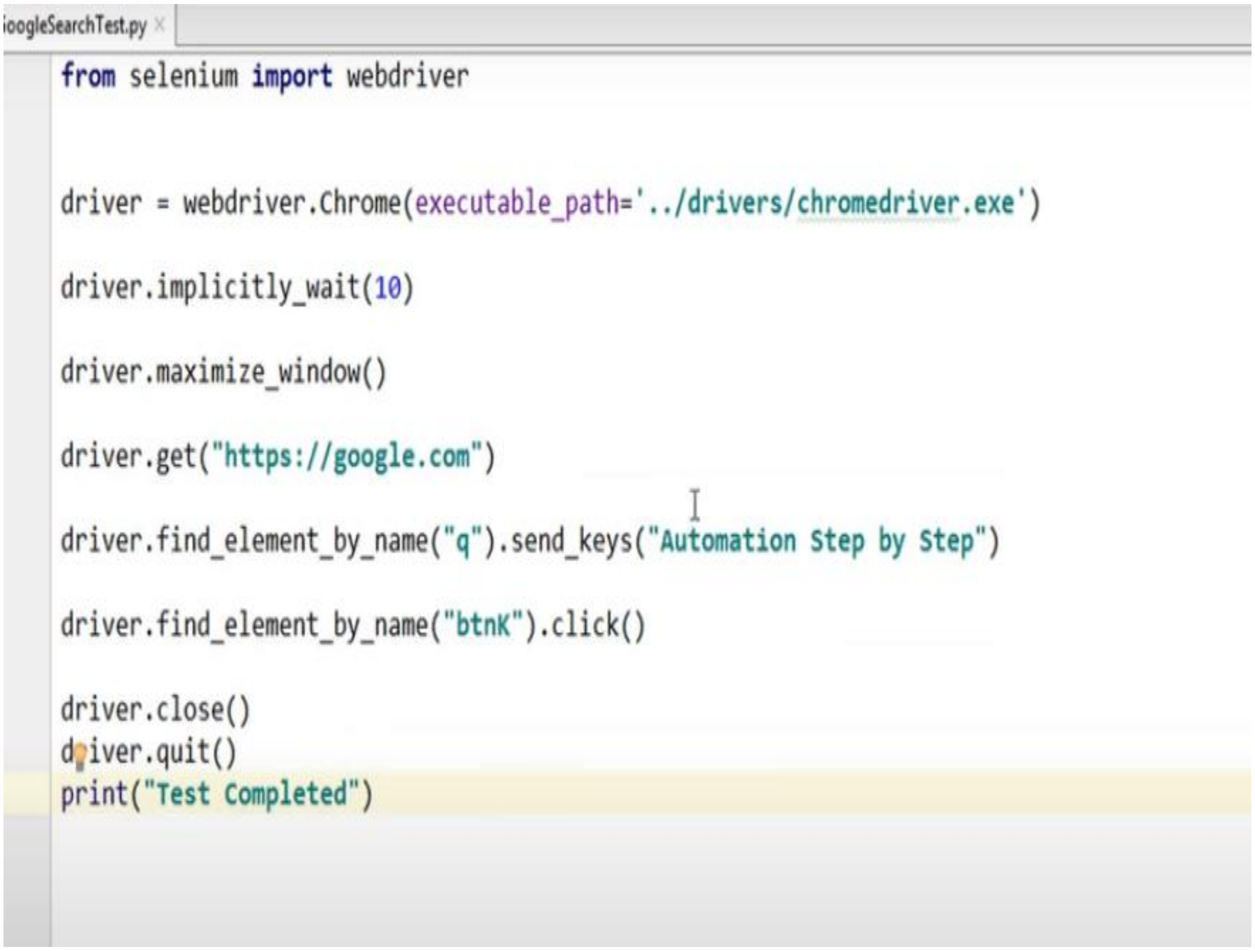
```
from selenium import webdriver

driver = webdriver.Chrome(executable_path='../drivers/chromedriver.exe')

driver.maximize_window()

driver.get(“https://google.com”)

driver.find_element_by_name(“q”).send_keys(“Automation step by step”)

driver.find_element_by_name(“btnk”) .click()

driver.close()
driver.quit()
print(“Test completed”)
```

“code will automatic seach for the word “Automation step by step” through google chrome”

Bntk is used for name function in google chrome.

So selenium is used for testing webdriver

The following o=are the scrrenshot of the programming in pychrm IDE



```python
from selenium import webdriver


driver = webdriver.Chrome(executable_path='../drivers/chromedriver.exe')

driver.implicitly_wait(10)

driver.maximize_window()

driver.get("https://google.com")

driver.find_element_by_name("q").send_keys("Automation Step by Step")

driver.find_element_by_name("btnK").click()

driver.close()
driver.quit()
print("Test Completed")
```