

Name:- Kunal Bhat

Roll No:- CE023

Identity :- 22CEUOG155

1. What are the advantages of automated testing?

- Saving Costs
- Faster Feedback Loop
- Better Allocation of Resources
- Guarantees Higher Accuracy
- Increased Test Coverage
- Detects bugs earlier
- Test at Scale
- Maximizes ROI

2. What is Selenium?

Selenium is an open-source, automated testing tool used to test web applications across various browsers. Selenium can only test web applications, unfortunately, sodesktop and mobile apps can't be tested.

Selenium is an open-source automation testing framework that is used for automationtesting on web applications across varying browsers and platforms. It can utilize a number of programming languages including C#, Java, and Python, among others.

Because developers often do not have the time to run manual tests on their web applications, tools like Selenium offer automated testing, avoiding time-consuming or repetitive tasks. In this sense, it is a significant time saver for teams that have a limitedwindow for testing. They can prioritize more complex testing for developers to executemanually and save time-consuming tests for Selenium to run without the need for developers' manual input.

3. What are the benefits of using Selenium as an automation tool?

- Open Source Availability
- Support for Major Languages
- Flexible Test Management
- Multi-Browser Use
- Easy Cross-Device Testing
- Reusable Automation Test Suites
- Simple Framework for Users

4. What are the limitations of Selenium as a tool?

- Selenium does not support automation testing for desktop applications.
- Selenium requires high skill sets in order to automate tests more effectively.

- Since Selenium is open source software, you have to rely on community forums to get your technical issues resolved.
- We can't perform automation tests on web services like SOAP or REST using Selenium.
- We should know at least one of the supported programming languages to create test scripts in Selenium WebDriver.
- It does not have built-in Object Repository like UFT/QTP to maintain objects/elements in a centralized location. However, we can overcome this limitation using Page Object Model.
- Selenium does not have any built-in reporting capability; you have to rely on plug-ins like **JUnit** and **TestNG** for test reports.
- It is not possible to perform testing on images. We need to integrate Selenium with **Sikuli** for image-based testing.
- Creating test environment in Selenium takes more time as compared to vendor tools like UFT, RFT, Silk test, etc.

- No one is responsible for new features usage; they may or may not work properly.
- Selenium does not provide any test tool integration for Test Management.

5. How to install selenium? Describe the steps.

How to install Selenium: To install Selenium: Choose your preferred programming language (e.g., Python, Java). ○ Install the Selenium WebDriver bindings for the selected language using package managers like pip (for Python) or Maven (for Java).

Set up the development environment and configure dependencies as needed.

6. Which testing types are supported by Selenium?

- Functional testing
- Regression testing
- Compatibility testing
- End-to-end testing

7.What is Selenium WebDriver?

Selenium WebDriver was first introduced as a part of Selenium v2.0. The initial version of Selenium i.e Selenium v1 consisted of only IDE, RC and Grid. However,with the release of Selenium v3, RC has been deprecated and moved to legacy package.

In WebDriver, test scripts can be developed using any of the supported programminglanguages and can be run directly in most modern web browsers. Languages supported by WebDriver include C#, Java, Perl, PHP, Python and Ruby.

8.Which browsers provide support for Selenium WebDriver?

- a) Chrome
- b) Firefox
- c) Safari
- d) Edge

9.Which are the ways to locate an element of a webpage using Selenium? :

- ID
- Name
- XPath
- CSS selector
- Class name
- Link text

10. Demonstrate the use of selenium to test web page using 1. Selenium Chrome Extension 2. Python Code.

1. Using Selenium Chrome Extension:

Open Selenium IDE:

Click on the Selenium IDE icon in your Chrome toolbar to open it. Record Test:

Click on the record button in Selenium IDE. Perform actions on the web page that you want to test.

Stop Recording:

Click on the stop button in Selenium IDE when you're done recording. Save Test:

Save the test case with a relevant name. Playback Test:

Click on the play button in Selenium IDE to playback the recorded test and verify if it works as expected.

2. Using Python Code:

```
from selenium import webdriver chromedriver_path = '/path/to/  
chromedriver'
```

```
driver =  
webdriver.Chrome(executable_path=chromedriver_path)  
driver.get('https://example.com')
```

```
assert 'Example Domain' in driver.title driver.quit()
```