

Assignments Module – 5 (Selenium IDE)

(1) What is Automation Testing?

- Automated testing is the application of software tools to automate a human-driven manual process of reviewing and validating a software product.
- The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources.
- Automation Testing is more reliable, as it is performed by tools and scripts.

(2) Which Are The Browsers Supported By Selenium Ide?

- Selenium Ide The Browsers Supported By Currently, Firefox, Chrome, Edge, IE, and Safari are supported and the Selenium IDE.

(3) What are the benefits of Automation Testing?

- (1) Saving Costs
- (2) Faster Feedback Loop
- (3) Better Allocation of Resources
- (4) Guarantees Higher Accuracy
- (5) Increased Test Coverage
- (6) Detects bugs earlier
- (7) Test at Scale
- (8) Maximizes ROI

(4) What are the advantages of Selenium?

- Open Source
- Supports all browsers like IE, Firefox, Mozilla, Safari

- Supports all Operating Systems.
- Supports all programming languages Java, Ruby, C# and Python.
- Run multiple tests at a time.

(5) Why testers should opt for Selenium and not QTP?

- Selenium

- It is an open source. It does not require any license to be used. Its completely free.
- It is just a browser automation tool. Its mainly used to automating test cases for web applications.
- You can write code for Selenium using wide range of IDEs like Visual Studio, Eclipse, Net beans.
- Selenium is written in JAVA, .NET, Ruby, Perl, PHP, and many other programming languages
- Selenium does not have build in object repository, but we can implement POM(Page Object Model)
- Selenium can be integrated with any tool provided that that tool has its API exposed to be consumed.
- It supposed dialog boxes which can be handled by JavaScript. It will not support print dialog or open file dialog.
- We can develop or use addition of plug-ins to include features that are not provided by core Selenium

- QTP

- It is commercial tool and hence it requires a license is expensive
- It is used mainly to test client-server applications. It can test web- based as well as desktop applications
- QTP tests can only be developed using QTP IDE
- It only supports VB script
- It comes with built in object repository. So, object repository development and maintenance is quite easy.
- It integrates with test management tool like HP Quality Center
- All types of dialog box is supported by it.

- It supports different environments like SAP, Oracle, .NET but user need to purchase add-on license for them
- Automation testing is expensive due to licensing cost
- It offers very good technical support
- It supports only VBScript . Hence Object Oriented Approach to Test Automation becomes challenging
- It is more user friendly and scripts are developed quickly.
- Backward compatibility is powerful. Latest version of HP ALM supports code that was developed 5 years back

(6) To validate the Swaglab website Login and logout process page

```
- // Generated by Selenium IDE
import org.junit.Test;
import org.junit.Before;
import org.junit.After;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.is;
import static org.hamcrest.core.IsNot.not;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Alert;
import org.openqa.selenium.Keys;
import java.util.*;
import java.net.MalformedURLException;
import java.net.URL;
public class SaucedemoTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
```

```
        driver.quit();
    }
    @Test
    public void saucedemo() {
        driver.get("https://www.saucedemo.com/");
        driver.manage().window().setSize(new Dimension(1536, 414));
        driver.findElement(By.cssSelector("[data-test=username\\"])).sendKeys("standard_user");
        driver.findElement(By.cssSelector("[data-test=password\\"])).sendKeys("secret_sauce");
        driver.findElement(By.cssSelector("[data-test=login-button\\"])).click();
        driver.findElement(By.id("react-burger-menu-btn")).click();
        driver.findElement(By.cssSelector("[data-test=logout-sidebar-link\\"])).click();
        driver.findElement(By.cssSelector("[data-test=username\\"])).sendKeys("standard_user");
        driver.findElement(By.cssSelector("[data-test=password\\"])).sendKeys("secret_sauce");
    }
}
```