

Software Testing Assignment

Module – 5(Selenium IDE)

* What is Automation Testing ?

- Automation testing means we can test the app or a web using a tools and scripts rather than manual testing. We create a script in different languages like java, python and by that script we can automate the Test Cases.

* Which Are The Browsers Supported By Selenium Ide?

- Browsers are: Chrome, Mozilla Firefox, Safari, Microsoft edge, Opera etc.

* What are the benefits of Automation Testing?

- Automation testing executes tests faster than manual testing, accelerating the development cycle.
- Test scripts can be reused across different versions of the application, saving time and effort.
- Easily handles a large number of test cases and supports parallel execution on multiple machines.
- Integrates well with CI/CD pipelines to catch issues early in the development process.
- Ensures tests are executed in a consistent and standardized manner every time they are run and eliminates human errors.

* What are the advantages of Selenium?

- **Open Source:** Free to use with no licensing fees and supported by a large community.
- **Cross-Browser Testing:** Supports major browsers like Chrome, Firefox, Safari, and Edge.
- **Multi-Language Support:** Compatible with languages such as Java, Python, C#, and JavaScript.
- **Integration Capabilities:** Easily integrates with CI/CD tools and testing frameworks.
- **Scalable and Flexible:** Handles complex scenarios and large test suites, and supports parallel execution with Selenium Grid.

* Why testers should opt for Selenium and not QTP?

-> Because of following reasons:

- Selenium is open-source and free, while QTP/UFT is a commercial product with licensing fees.
- Selenium supports multiple programming languages (Java, Python, C#, etc.), whereas QTP/UFT primarily uses VBScript.

Software Testing Assignment

- Selenium offers extensive cross-browser compatibility (Chrome, Firefox, Safari, Edge), whereas QTP/UFT has more limited browser support.
- Selenium has a large and active open-source community, providing extensive resources, plugins, and integrations, unlike the more closed ecosystem of QTP/UFT.
- Selenium integrates well with modern CI/CD tools and frameworks, such as Jenkins and Maven, which aligns better with contemporary development practices compared to QTP/UFT.

* To validate the Swaglab website Login and logout process page
<https://www.saucedemo.com>.

```
package ui;

import org.openqa.selenium.By;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import io.github.bonigarcia.wdm.WebDriverManager;

public class SauceDemo {

    public static void main(String[] args) throws InterruptedException {

        WebDriverManager.chromedriver().setup();
        ChromeDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.saucedemo.com/");

        driver.findElement(By.xpath("//input[@name='user-
name']")).sendKeys("standard_user");

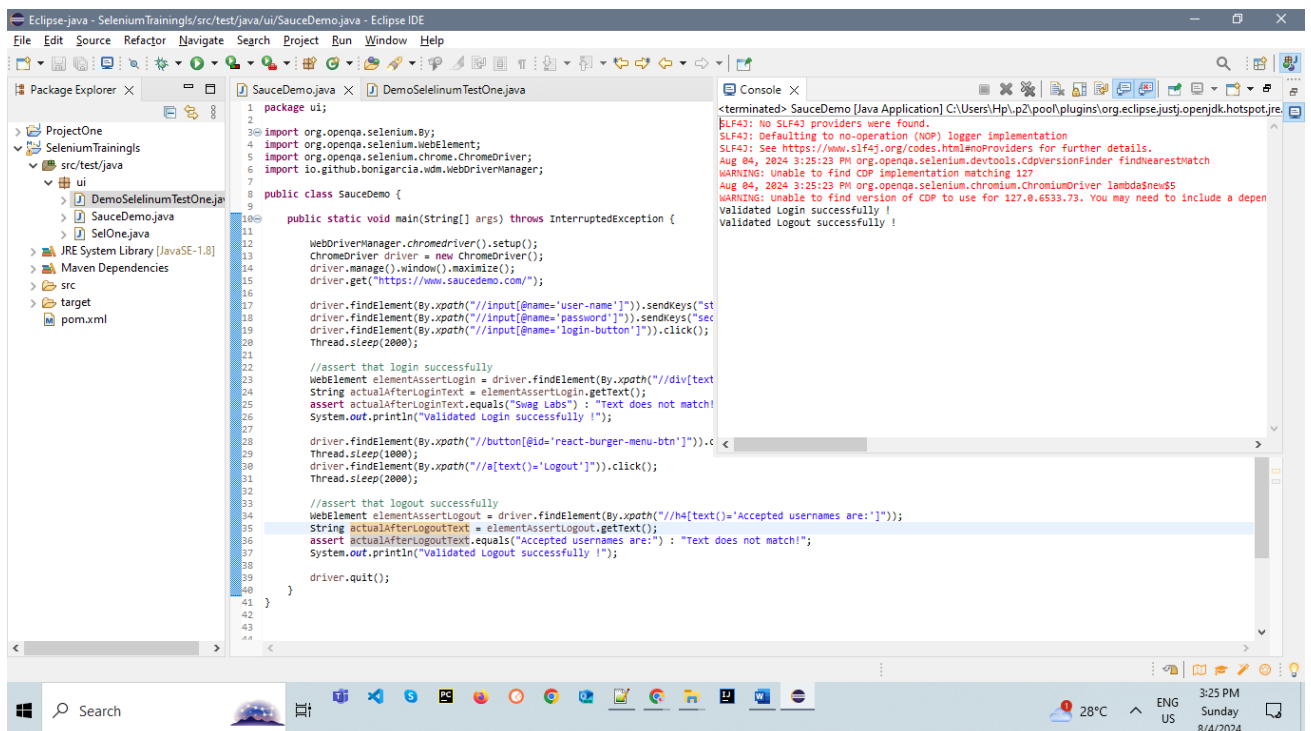
        driver.findElement(By.xpath("//input[@name='password']")).sendKeys("secret_sauce");
        driver.findElement(By.xpath("//input[@name='login-
button']")).click();
        Thread.sleep(2000);

        //assert that login successfully
        WebElement elementAssertLogin =
driver.findElement(By.xpath("//div[text()='Swag Labs']"));
        String actualAfterLoginText = elementAssertLogin.getText();
        assert actualAfterLoginText.equals("Swag Labs") : "Text does not
match!";
        System.out.println("Validated Login successfully !");
    }
}
```

Software Testing Assignment

```
driver.findElement(By.xpath("//button[@id='react-burger-menu-  
btn']")).click();  
Thread.sleep(1000);  
driver.findElement(By.xpath("//a[text()='Logout']")).click();  
Thread.sleep(2000);  
  
//assert that logout successfully  
WebElement elementAssertLogout =  
driver.findElement(By.xpath("//h4[text()='Accepted usernames are:]"));  
String actualAfterLogoutText = elementAssertLogout.getText();  
assert actualAfterLogoutText.equals("Accepted usernames are:") :  
"Text does not match!";  
System.out.println("Validated Logout successfully !");  
  
driver.quit();
```

O/P:



The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure: ProjectOne, SeleniumTrainings, and its sub-packages. The main editor shows the source code of DemoSeleniumTestOne.java, which includes imports for Selenium WebDriver, ChromeDriver, and WebDriverManager. The code defines a SauceDemo class with a main method that performs a login and logout test. The console on the right shows the output of the test, including SLF4J warnings and the final status: "Validated Login successfully !" and "Validated Logout successfully !".