**Task 15 – Selenium**

**1.Explain the difference between Selenium IDE, Selenium Web Driver, and Selenium Grid:**

* **Selenium IDE (Integrated Development Environment):** Selenium IDE is a simple and user-friendly tool primarily used for recording and playing back interactions with a web application. It is essentially a browser extension that helps users create automated test scripts without requiring in-depth programming knowledge. Selenium IDE records user actions, such as clicking buttons and entering text, and converts them into test scripts. However, Selenium IDE has limitations when it comes to complex testing scenarios and cross-browser testing, making it more suitable for quick tests, demonstrations, and simple use cases.
* **Selenium WebDriver**:

Selenium WebDriver is the core component of the Selenium framework. It provides a programming interface that allows developers to create more powerful and flexible automated test scripts. It allows you to interact with web elements, simulate user actions, navigate through web pages, and validate application behavior. Unlike Selenium IDE, WebDriver doesn’t rely on recording and playback; instead, you write code in programming languages like Java, Python, C#, etc., to create robust and customizable test scripts. WebDriver supports a wide range of browsers and provides more control over testing scenarios and browser configurations.

* **Selenium Grid:**

Selenium Grid is designed to facilitate parallel and distributed testing across multiple machines and browsers simultaneously. It allows you to run your tests on different operating systems, browsers, and versions in a distributed environment, reducing the time required for testing. With Selenium Grid, you can set up a hub and multiple nodes, where the hub coordinates the test distribution and the nodes execute the tests on different configurations. This is particularly useful for cross-browser and cross-platform testing, ensuring that your web application works consistently across various environments.

**2. Write a Selenium script in Java to open Google and search for "Selenium Browser Driver."**

* Given in github

**3. What is Selenium? How it is useful in Automation Testing?**

Selenium is an open-source software framework primarily used for automating web browser interactions. It provides a suite of tools and libraries that allow testers and developers to automate the testing of web applications, ensuring that they function correctly across different browsers, platforms, and scenarios. Selenium is widely used in the field of automation testing to streamline and improve the testing process.

Selenium is used in automation testing:

1. Automating User Interactions: Selenium allows you to automate user interactions with web applications like clicking buttons, entering text, selecting options, navigating through pages, and more.
2. Cross-Browser Testing: Web applications are accessed using different web browsers (like Chrome, Firefox, Safari, etc.).Selenium helps you test your application’s compatibility across various browsers, ensuring consistent functionality.
3. Functional Testing: Selenium is often used to perform functional testing, where the goal is to verify that the application’s features and functionalities work as expected. Test cases are written to simulate different scenarios and verify that the application responds correctly.
4. Regression Testing: When new code is added to an application, there’s a risk of introducing bugs that might affect existing functionalities. Selenium can automate the process of running regression tests to ensure that new changes haven’t broken previously working parts of the application.
5. Performance Testing: While not the primary tool for performance testing, Selenium can be used to simulate user interactions under load, helping to identify performance bottlenecks and issues.
6. Data-Driven Testing: Selenium allows you to perform data-driven testing, where you can run the same test with different sets of data. This is useful for testing various scenarios and inputs.
7. Parallel and Distributed Testing: With tools like Selenium Grid, you can run tests on multiple machines and browsers simultaneously. This speeds up the testing process and allows you to cover more test cases efficiently.
8. Custom Test Frameworks: Selenium provides the flexibility to design custom test frameworks tailored to your project’s requirements, allowing you to create reusable components and enhance test maintenance.

**4.What are all Browser driver used in Selenium?**

1. ChromeDriver: The driver for Google Chrome browser. It’s widely used due to Chrome’s popularity and developer tools.
2. GeckoDriver: The driver for Mozilla Firefox browser. It’s used to automate interactions with Firefox.
3. MicrosoftEdge WebDriver (EdgeDriver): The driver for Microsoft Edge browser. It allows automation of Edge browser actions.
4. SafariDriver: The driver for Safari browser. It’s used for automating interactions with Safari on macOS.
5. OperaDriver: The driver for the Opera browser. It enables automation of Opera browser actions.
6. InternetExplorerDriver: The driver for Microsoft Internet Explorer browser. It’s used to automate interactions with IE, although its usage is decreasing due to IE’s reduced popularity.

**5. What are all the steps to create a simple web driver script? Explain with a code**

Step 1: Create a New maven Project

Open your Eclipse and create a new Maven project.

Step 2: Add Selenium WebDriver Library in porn.xml

Add the Selenium WebDriver library (JAR files) to your project. You can usually download the WebDriver JAR files from a build tool like Maven to manage dependencies.Copy and paste those dependencies in porn.xml

Step 3: Write the WebDriver Script

Create a Java class and write the WebDriver script. Below is an example code that opens Google, searches for a query, and prints the page title.

**package** Task13;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.Keys;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** OpenBrowser {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//setup driver and open browser

ChromeDriver driver = **new** ChromeDriver();

//Pass url to load in browser

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

//To maximize window of browser

driver.manage().window().maximize();

//To locate the search box using locator and enter text to be searched, and to select enter we used keys tag

driver.findElement(By.*name*("q")).sendKeys("Selenium browser driver",Keys.***ENTER***);

//to close the browser

driver.close();

}

}

Step 4: Run the Script

Run the script from your Eclipse. The Chrome browser should open, navigate to Google, perform the search, and then close.

