**8. Perform automation testing for launching website and searching for a module & opening it, included in the website using Selenium and Intellij**

**Prerequisites**

1. **Java Development Kit (JDK)** installed.
2. **IntelliJ IDEA** IDE installed.
3. **Selenium WebDriver** library added to the project.

**Steps to Set Up the Project**

**1. Create a New Project in IntelliJ IDEA**

1. Open IntelliJ IDEA and create a new Java project.
2. Add the Selenium WebDriver dependency to your project by adding the following lines to your pom.xml file if you're using Maven:

xml

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

</dependencies>

**2. Write the Test Script**

Create a new Java class in your project and write the following test script to launch the website, search for a module, and open it.

**Example Test Script**

java

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class LaunchSearchOpenModule {

public static void main(String[] args) {

// Set the path of the ChromeDriver executable

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

// Initialize WebDriver

WebDriver driver = new ChromeDriver();

try {

// Launch the website

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

// Maximize the browser window

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

// Locate the search bar and enter the module name

WebElement searchBar = driver.findElement(By.name("q"));

searchBar.sendKeys("desired module");

searchBar.submit();

// Wait for the results to load

Thread.sleep(2000);

// Locate and click on the module link

WebElement moduleLink = driver.findElement(By.linkText("Module Name"));

moduleLink.click();

// Additional actions can be added here

} catch (InterruptedException e) {

e.printStackTrace();

} finally {

// Close the browser

driver.quit();

}

}

}

**Explanation of the Code**

* **Set Up WebDriver**: Configure the path to your chromedriver executable.

java

System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");

* **Initialize WebDriver**: Create an instance of ChromeDriver.

java

WebDriver driver = new ChromeDriver();

* **Launch Website**: Use driver.get("https://www.example.com") to open the desired website.

java

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

* **Maximize Browser Window**: Maximize the browser to ensure all elements are accessible.

java

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

* **Search for Module**: Locate the search bar element by its name attribute, enter the module name, and submit the search form.

java

WebElement searchBar = driver.findElement(By.name("q"));

searchBar.sendKeys("desired module");

searchBar.submit();

* **Wait for Results to Load**: Pause the execution for a few seconds to wait for search results to load.

java

Thread.sleep(2000);

* **Open Module**: Locate the module link from the search results by its link text and click on it.

java

WebElement moduleLink = driver.findElement(By.linkText("Module Name"));

moduleLink.click();

* **Close Browser**: Ensure the browser is closed after the test execution.

java

driver.quit();

**Running the Test**

1. Save the Java file with the test script.
2. Run the main method in IntelliJ IDEA to execute the test.

This script will open the browser, navigate to the specified URL, search for a module, open the module, and then close the browser after a few seconds. You can adjust the URL, search term, and other parameters as needed.

Let me know if you need any further assistance or additional details! 😊