**9. Perform automation testing for launching website and getting logged into the web application with valid credentials 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 and log into the web application using valid credentials.

**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 LoginAutomationTest {

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 username field and enter the username

WebElement usernameField = driver.findElement(By.id("username"));

usernameField.sendKeys("validUsername");

// Locate the password field and enter the password

WebElement passwordField = driver.findElement(By.id("password"));

passwordField.sendKeys("validPassword");

// Locate the login button and click it

WebElement loginButton = driver.findElement(By.id("loginButton"));

loginButton.click();

// Wait for the login process to complete

Thread.sleep(3000);

// Verify the login is successful by checking for a specific element on the home page

WebElement homePageElement = driver.findElement(By.id("homePageElementId"));

if (homePageElement.isDisplayed()) {

System.out.println("Login successful.");

} else {

System.out.println("Login failed.");

}

} 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();

* **Enter Username**: Locate the username field by its ID attribute and enter the username.

java

WebElement usernameField = driver.findElement(By.id("username"));

usernameField.sendKeys("validUsername");

* **Enter Password**: Locate the password field by its ID attribute and enter the password.

java

WebElement passwordField = driver.findElement(By.id("password"));

passwordField.sendKeys("validPassword");

* **Click Login Button**: Locate the login button by its ID attribute and click it.

java

WebElement loginButton = driver.findElement(By.id("loginButton"));

loginButton.click();

* **Wait for Login to Complete**: Pause the execution for a few seconds to wait for the login process to complete.

java

Thread.sleep(3000);

* **Verify Login Success**: Check for the presence of a specific element on the home page to confirm a successful login.

java

WebElement homePageElement = driver.findElement(By.id("homePageElementId"));

if (homePageElement.isDisplayed()) {

System.out.println("Login successful.");

} else {

System.out.println("Login failed.");

}

* **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, log in using the provided credentials, verify the login success, and then close the browser. Adjust the URL, credentials, and element identifiers as needed.

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