**Automating the Amazon application using Selenium Webdriver**

**AIM:**

To Automating the Amazon application using Selenium Webdriver

**Steps:**

automating various actions on the Amazon homepage using Selenium WebDriver and the Page Object Model (POM) design pattern. It includes the step-by-step process

Step-by-Step Process:

1.Create a GitHub Repository:

Create a new GitHub repository to track your automation project.

2.Set Up Your Development Environment:

Ensure you have Java and Selenium WebDriver set up in your development environment.

Install a WebDriver executable (e.g., ChromeDriver) and configure it.

3.Create a Maven Project:

Set up a Maven project in your IDE. This will help manage dependencies and automate project build.

4.Create Page Object Model (POM) Classes:

Create POM classes to represent the Amazon homepage and its elements. Each page should have its own class with methods to interact with the elements.

5.Navigate to Amazon Homepage:

In your test script, navigate to the Amazon homepage (https://amazon.in/).

Verify that the page has loaded successfully and that the user has signed in (you can use elements like "Sign In" or "Account & Lists" to check).

6.Scroll the Page:

Use WebDriver actions or JavaScript to scroll the page. Verify that scrolling works as expected.

7.Search for a Product:

Use your POM classes to enter "iPhone 13" in the search bar and click the search button.

Check that filters are applied correctly for the "Mobile" category.

8.Add Products to Wishlist:

Select products from the search results and add them to the wishlist using your POM classes.

Verify that the products have been added successfully.

9.Check for Confirmation Box:

After adding products to the wishlist, check if a confirmation box appears.

Use WebDriver to interact with the confirmation box if it does.

10.Implement Test Framework:

Set up a test framework using TestNG or JUnit to organize your test cases.

11.Write Test Cases:

Create test cases for each step of your automation scenario using your POM classes.

12.Add Logging and Reporting:

Implement logging (e.g., log4j) and reporting (e.g., ExtentReports or Allure) to capture test execution details and results.

13.Create a .gitignore File:

Create a .gitignore file in your project directory to specify which files or directories should be ignored by Git. Common entries include build artifacts, IDE-specific files, and logs.

14.Initialize Git and Commit Your Code:

Initialize Git in your project directory and commit your code to the local repository.

Use clear and meaningful commit messages for each step or feature you implement.

15.Push Your Code to GitHub:

Link your local repository to the GitHub repository you created earlier and push your code to GitHub.

Make sure you don't push files specified in the .gitignore file.

16.Documentation:

In your GitHub repository, create a README.md file to document the purpose of your project, setup instructions, and how to run the tests.

Update the README with any additional information about your automation project.

**Source Code:**

**TestSignInpage**

**package in.amazon.project.lesson1;**

**import java.time.Duration;**

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

**import org.openqa.selenium.WebDriver;**

**import org.openqa.selenium.WebElement;**

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

**import org.openqa.selenium.interactions.Actions;**

**public class TestSignInPage {**

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

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

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

**driver.manage().deleteAllCookies();**

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

**// wait for 2 seconds the page to Load**

**driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(5));**

**driver.get("https://www.amazon.in/");**

**// locating the dropdown that has sign in button -> signin - accounts and lists**

**WebElement e1 = driver.findElement(By.xpath("//\*[@class='nav-line-2 ']"));**

**Actions a = new Actions(driver);**

**a.moveToElement(e1).perform();**

**Thread.sleep(3000);**

**driver.findElement(By.linkText("Sign in")).click();**

**// Inspect the Email and continue button on sign page**

**Thread.sleep(3000);**

**driver.findElement(By.id("ap\_email")).sendKeys("kalalabhilash495@gmail.com");**

**driver.findElement(By.xpath("//input[@id='continue']")).click();**

**// inspect password input box and click on sign button**

**driver.findElement(By.id("ap\_password")).sendKeys("password@123");**

**driver.findElement(By.id("signInSubmit")).click();**

**// fetch the error text**

**String text = driver.findElement(By.xpath("//div[@class='a-box-inner a-alert-container']/descendant::span")).getText();**

**System.out.println("Error test is >> " + text);**

**driver.close();**

**}**

**}**

**Mobile search test**

**package in.amazon.project.lesson1;**

**import java.time.Duration;**

**import java.util.ArrayList;**

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

**import org.openqa.selenium.WebDriver;**

**import org.openqa.selenium.WebElement;**

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

**import org.openqa.selenium.interactions.Actions;**

**public class TestMobilePage {**

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

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

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

**driver.manage().deleteAllCookies();**

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

**// wait for 2 seconds the page to Load**

**//driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(5));**

**driver.get("https://www.amazon.in/");**

**// click on the link Mobiles:**

**Thread.sleep(1500);**

**driver.findElement(By.linkText("Mobiles")).click();**

**// Move the mouse to the element**

**WebElement mobile = driver.findElement(By.xpath("//\*[@id='nav-subnav']/descendant::span[2]"));**

**Actions a = new Actions(driver);**

**a.moveToElement(mobile).build().perform();**

**Thread.sleep(3000);**

**// click on the mobile brand Apple**

**driver.findElement(By.linkText("Apple")).click();**

**// Click on apple phone link**

**driver.findElement(By.xpath("(//div[@class='sg-col-inner'])[6]/descendant::span[9]")).click();**

**// switch to the new tab window**

**ArrayList<String> tabs = new ArrayList<String>(driver.getWindowHandles());**

**driver.switchTo().window(tabs.get(1));**

**Thread.sleep(1500);**

**driver.findElement(By.id("add-to-cart-button")).click();**

**}**

**}**

**TestSearchMobilePage**

**package in.amazon.project.lesson1;**

**import java.util.ArrayList;**

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

**import org.openqa.selenium.WebDriver;**

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

**public class TestSearchMobilePage {**

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

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

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

**driver.manage().deleteAllCookies();**

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

**// wait for 2 seconds the page to Load**

**//driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(5));**

**driver.get("https://www.amazon.in/");**

**// Inspect the search box and enter the input as iPhone 13**

**driver.findElement(By.id("twotabsearchtextbox")).sendKeys("iphone 13");**

**driver.findElement(By.id("twotabsearchtextbox")).submit();**

**Thread.sleep(1500);**

**driver.findElement(By.xpath("(//div[@class='a-section'])[2]/descendant::span[11]")).click();**

**ArrayList<String> tabs = new ArrayList<String>(driver.getWindowHandles());**

**driver.switchTo().window(tabs.get(1));**

**Thread.sleep(1500);**

**driver.findElement(By.id("wishlistButtonStack")).click();**

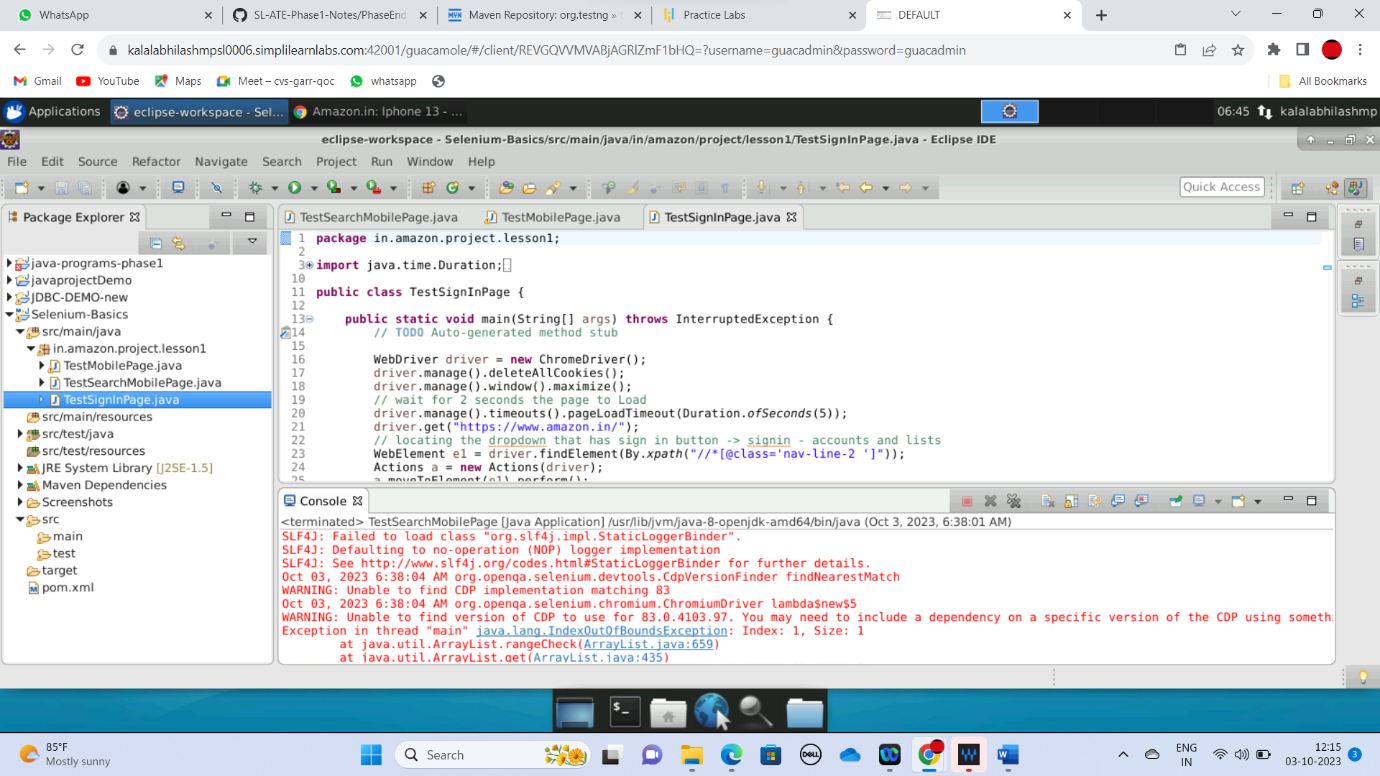
**driver.quit();**

**}**

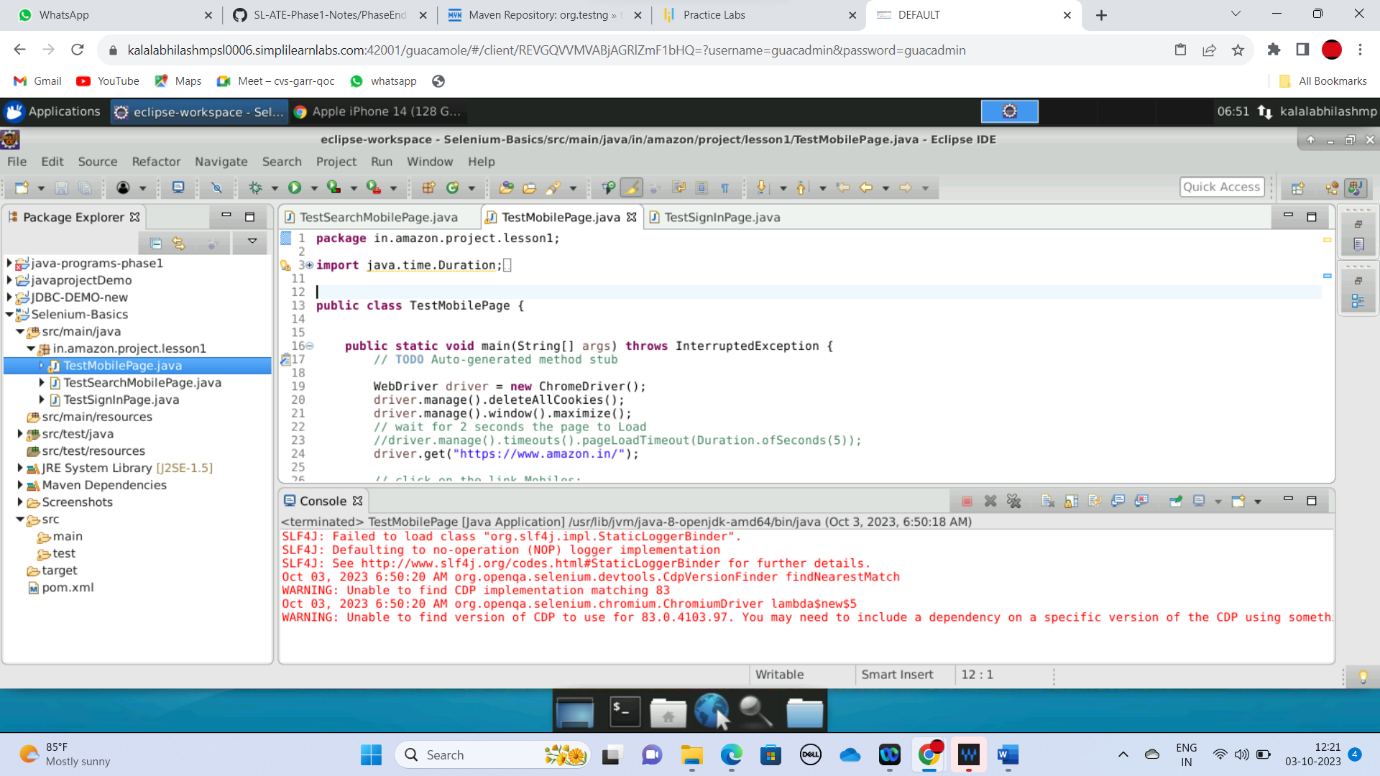
**}**

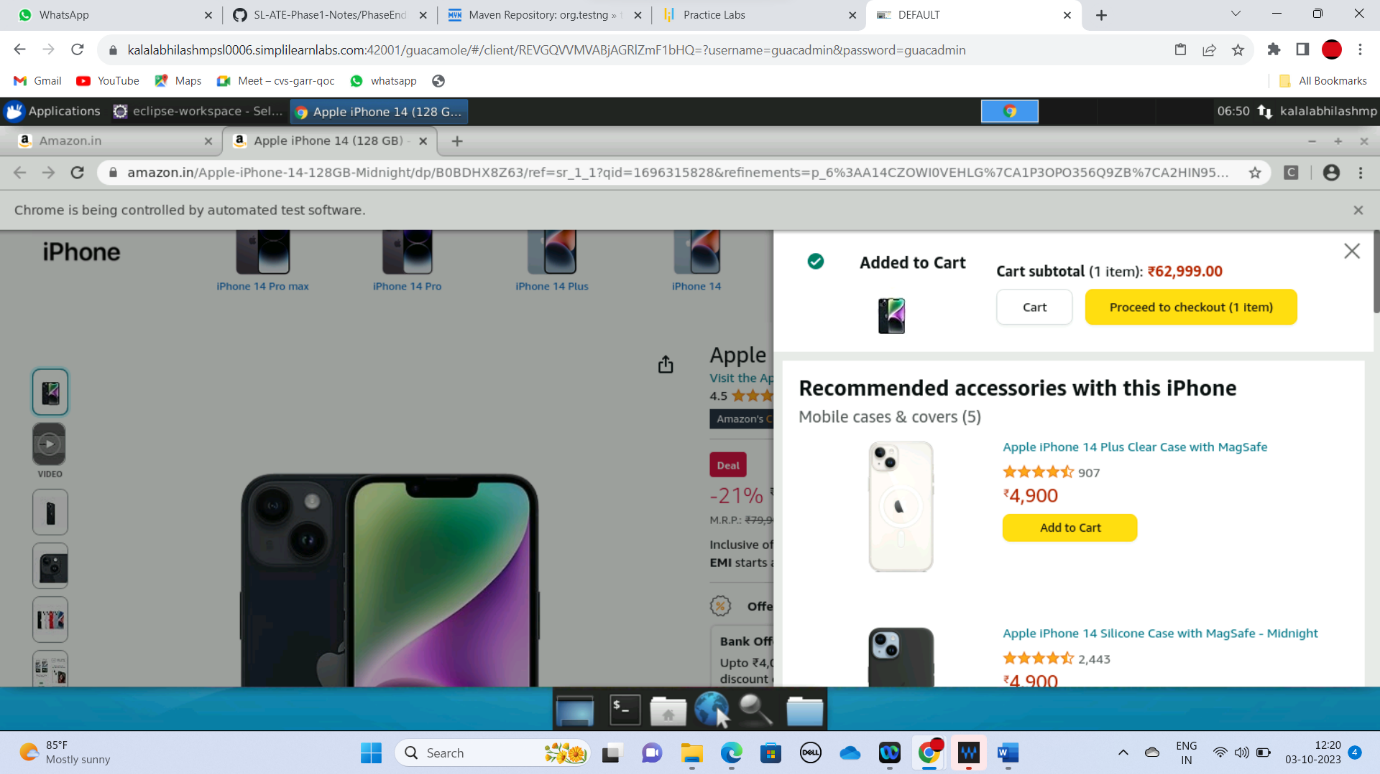
**SCREENSHOTS**

**TestSignInPage**

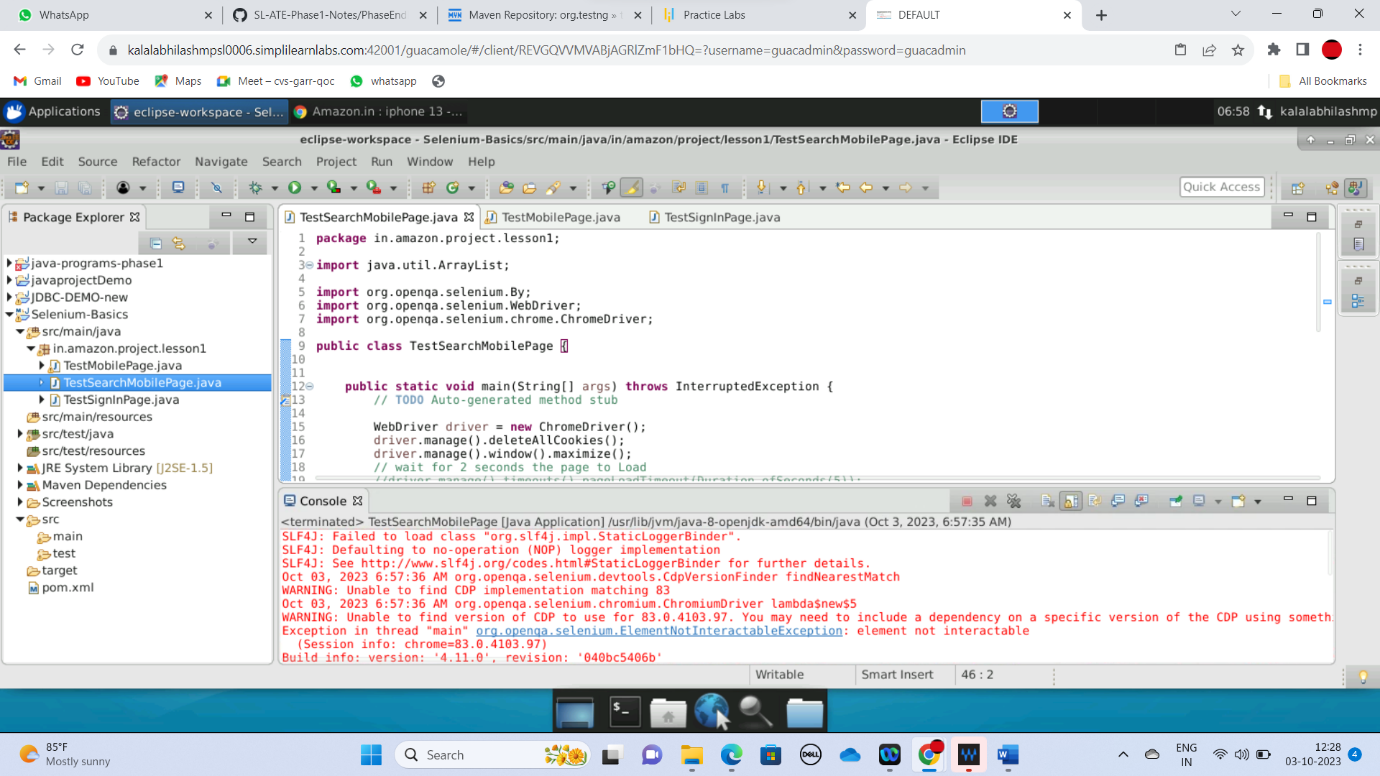
****

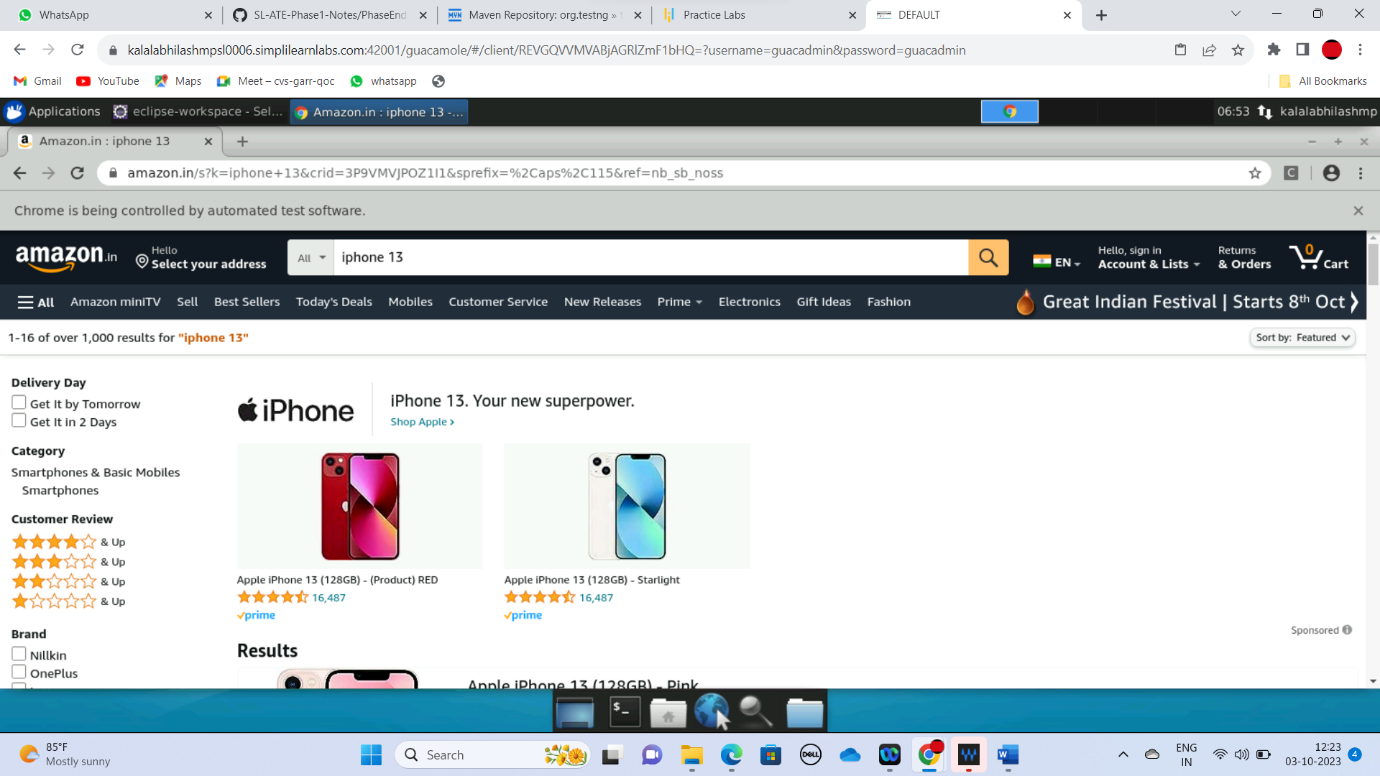
**TestMobilePage**

****

****

**TestSearchMobilePage**

****

****