

Practical No: 1

Aim: Take a review and write down test cases for any known application.

Description:

- a) Review
- b) Test cases
- c) Known Application
 - i) www.google.com

Test case no	Test_case_no_description	Test Data	Input	Action	Expected Result	Actual Result	Status	Remark
url_01	url_01_valid	url https://www.google.com/	Correct url	Write proper url and press enter key	Successfully open the www.google.com	Successfully open the www.google.com	Test passes	
url_02	url_02_invalid	url https://www.google.com/	Incorrect url	Write proper url and press enter key	Successfully open the www.google.com	Error	Test failed	
G1_01	Gmail_login_valid_01	User name and password	Correct email id and correct password	Write proper email id and password and press the next	Successfully Open home of user gmail page	Successfully open the home page of user gmail	Test passed	
G1_02	Gmail_login_invalid_02	User name and password	Correct email id and incorrect password	Write proper email id and password and press the next	Error message	Error message	Test passed	

gm_01	Google_map_valid_01	Destination Address, GPS has to on	correct destination address and on the GPS	Enter correct destination address and on the GPS	Successfully Show the correct route	Successfully Show the correct route	Test passed	
gm_02	Google_map_invalid_02	Destination Address, GPS has to on	Correct destination address but GPS off	Enter correct destination address and off the GPS	Error message turn on device location	Error message turn on device location	Test passes	
gl_01	Google_lens_valid_01	scan the image, turn on net	Set camera on image	Scan the proper image	Successfully show the information of image	Successfully show the information of image	Test passed	
gl_02	Google_lens_invalid_02	scan the image, turn off net	set camera on image but off the net	Scanning proper image but net off	Error message Something went wrong	Error message Something went wrong	Test passed	
Ga_01	Google_Apps_valid_01	Shows the Google apps	Show the all google apps	Click on google apps icon	Successfully show the all google apps	Successfully Show the all google apps	Test passed	
Ga_02	Google_Apps_invalid_02	Shows the Google apps	Not showing google apps	Click on google apps icon but net is off	Error It will not shown any google apps	Error it will not show any google apps	Test passed	

ii) www.youtube.com

Test case no	Test case description	Test Data	Input	Action	Expected Result	Actual Result	Status	Remark
url_01	url_01_valid	url https://www.youtube.com/	Correct url	Write proper url and press enter key	Successfully open the www.youtube.com	Successfully open the www.youtube.com	Test passes	
url_02	url_02_invalid	url https://www.youtube.com/	Incorrect url	Write proper url and press enter key	Successfully open the www.youtube.com	Error	Test failed	
cr_01	Create_video_valid_01	Upload the video	Upload the video with .mp4 file	Properly upload the video file	It will show the successfully upload video	it will show the successfully upload video	Test passed	
cr_02	Create_video_invalid_02	Upload the video	Upload the file instead of .mp4 extension	Properly upload the video file	Error message invalid file format.	Error Message invalid file format.	Test passed	
sb_01	Search_box_valid_01	Search box	Search the video as the name type in search box	Enter correct anything in search box (According to requirement)	Successfully Show the video of according to requirement.	Successfully Show the video of according to requirement.	Test passed	
sb_02	Search_box_invalid_02	Search box	Search the name	Enter the irrelevant	Error it will give the	Error it will give the	Test passes	

			(what we require d)	nt name in search box	irrelevant video	irrelevant video		
hi_01	History_valid_01	Video stored in history	View video	Sign in with YouTube and see the video	Successfully stored viewed video in history	Successfully stored viewed video in history	Test passed	
hi_02	History_invalid_02	Video stored in history or not	View video without login	Without login view the video of YouTube	Error message sign in with YouTube	Error sign in with you Tube	Test passed	
sb_01	Subscribe_btn_valid_01	Subscribe channels	Subscribe channels with Sign in YouTube	Click on subscription button	Successfully subscribe the channel	Successfully Subscribe the channel	Test passed	
sb_02	Subscribe_btn_invalid_02	Subscribe the channels	Subscribe channels without Sign in YouTube	Click on subscription button without sign in	Error Sign in with YouTube	Error sign in with YouTube.	Test passed	

iii) www.facebook.com

Test case_no	Test_case_no_description	Test Data	Input	Action	Expected Result	Actual Result	Status	Remark
url_01	url_01_valid	url https://www.facebook.com/	Correct url	Write proper url and press enter key	Successfully open the www.facebook.com	Successfully open the www.facebook.com	Test passes	
url_02	url_02_invalid	url https://www.facebook.com/	Incorrect url	Write proper url and press enter key	Successfully open the www.facebook.com	Error	Test failed	
cr_01	login_valid_01	User name and password	Correct User id and password	Write proper user id and password and press the next	Successfully Open home of user facebook page	Successfully open the home page of user facebook	Test passed	
cr_02	login_invalid_02	User name and password	Correct User id and wrong password	Write proper user id and wrong password and press the next	Error	Error, please enter the correct password	Test passed	
sb_01	Search_box_valid_01	Search box	Search the any person as the name type in search box	Enter correct any person name search box	Successfully Show the list names of person as we search for.	Successfully Show list of persons name.	Test passed	

sb_02	Search_box_invalid_02	Search box	Search the any person as the name type in search box	Enter in-correct any person name search box	It will show the irrelevant persons list.	Can't find actual person	Test passed	
sf_01	Suggested_frds_invalid_01	Suggested friends/ may you know the people	Correct phone number	Enter the correct phone number .	Successfully show the people who are in phone book list	Successfully show the people who are in phone book	Test passed	
sg_02	Suggested_frds_invalid_01	Suggested friends/ may you know the people	Not entering phone no. or incorrect phone number	Enter the incorrect phone number or not enter	Error it will not showing the person who are in phone book	Error not showing the person who are in phone book	Test passed	
sm_01	Send_message_valid_01	message	Able to send the message to friends or not	Install messenger and send message to friends	Successfully send the message to friends.	Successfully Send the message to the friends.	Test passed	
sm_02	Send_message_invalid_02	message	Able to send the message to friends or not	Not Install messenger and send message to friends	Error install messenger	Error install messenger .	Test passed	

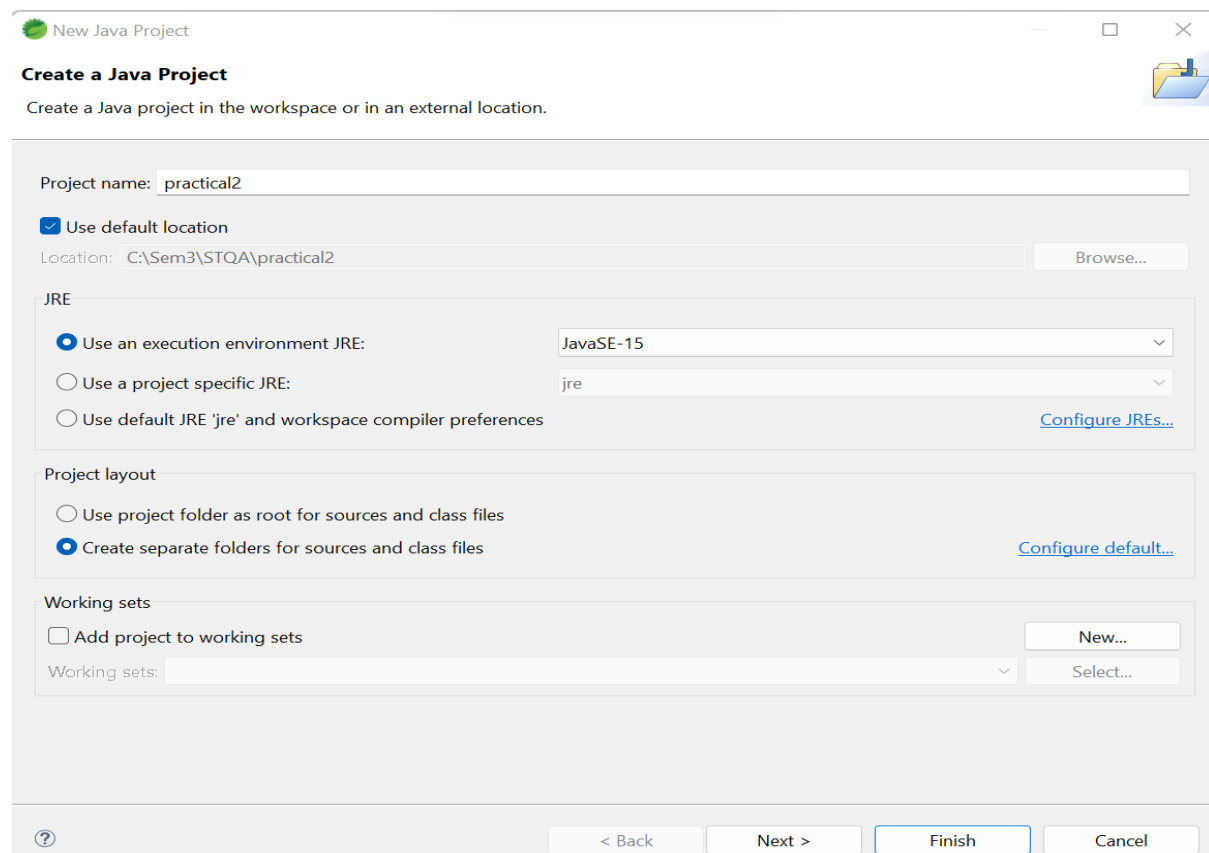
Practical No: 2

Aim: Implement Web Drivers on Chrome & FireFox browsers.

Description:

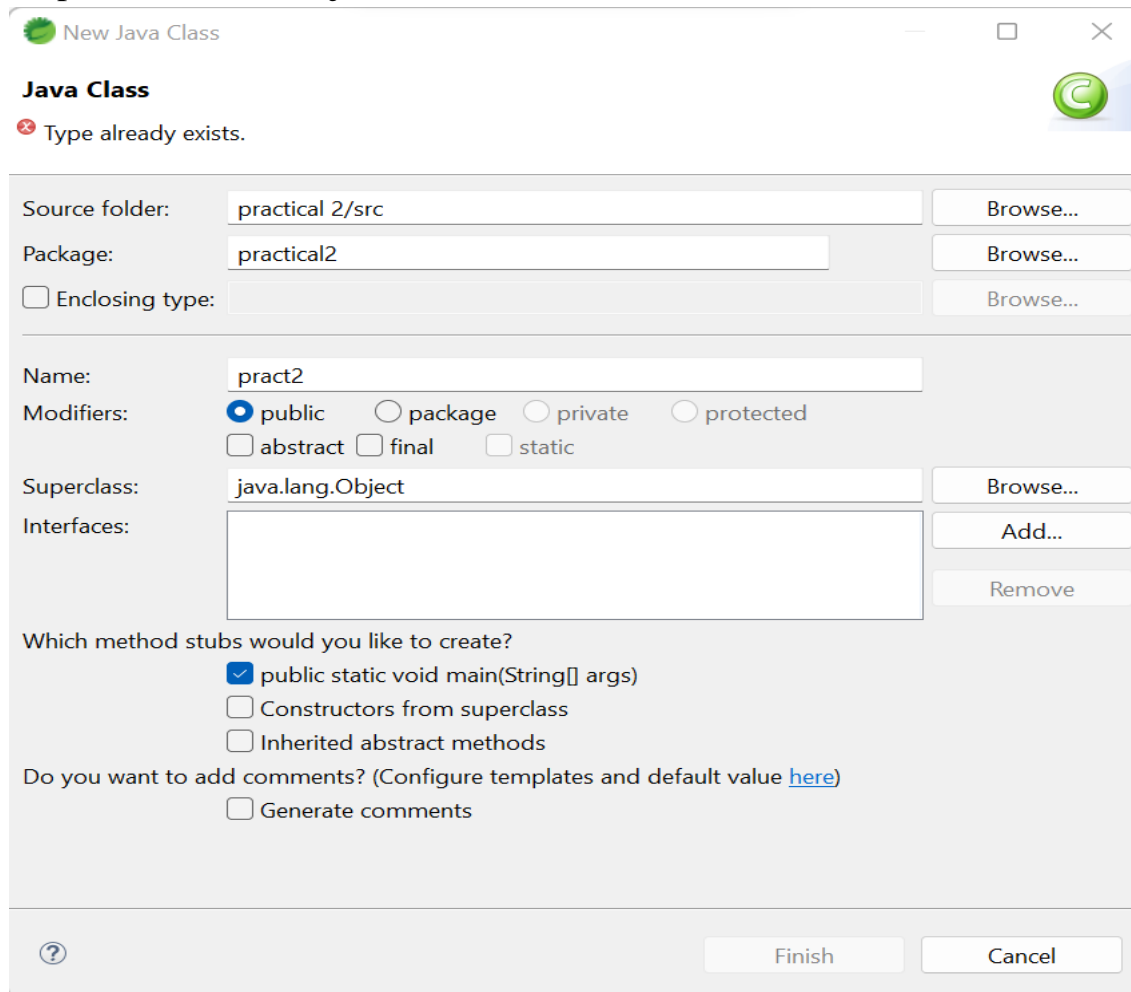
- 1) drivers(required to perform the cross browsing)
 - a) chrome driver
 - b) geko driver
- 2) we required selenium standalone server used for synchronaton of various tools, dependency
- 3) Ecalips OR STS (required to write testcases which will run in java)
- 4) jdk 1.7 onwards

Create a new java project



The screenshot shows the 'New Java Project' dialog box in the Eclipse IDE. The title bar reads 'New Java Project'. Below the title bar, there is a section 'Create a Java Project' with a subtext 'Create a Java project in the workspace or in an external location.' and a folder icon. The main area contains several sections: 'Project name:' with the text 'practical2'; 'Use default location' checked, with 'Location:' showing 'C:\Sem3\STQA\practical2' and a 'Browse...' button; 'JRE' section with three radio buttons: 'Use an execution environment JRE:' (selected) with a dropdown showing 'JavaSE-15', 'Use a project specific JRE:' with a dropdown showing 'jre', and 'Use default JRE 'jre' and workspace compiler preferences' with a 'Configure JREs...' link; 'Project layout' section with two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected) with a 'Configure default...' link; and 'Working sets' section with 'Add project to working sets' unchecked, a 'New...' button, and a 'Working sets:' dropdown with a 'Select...' button. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

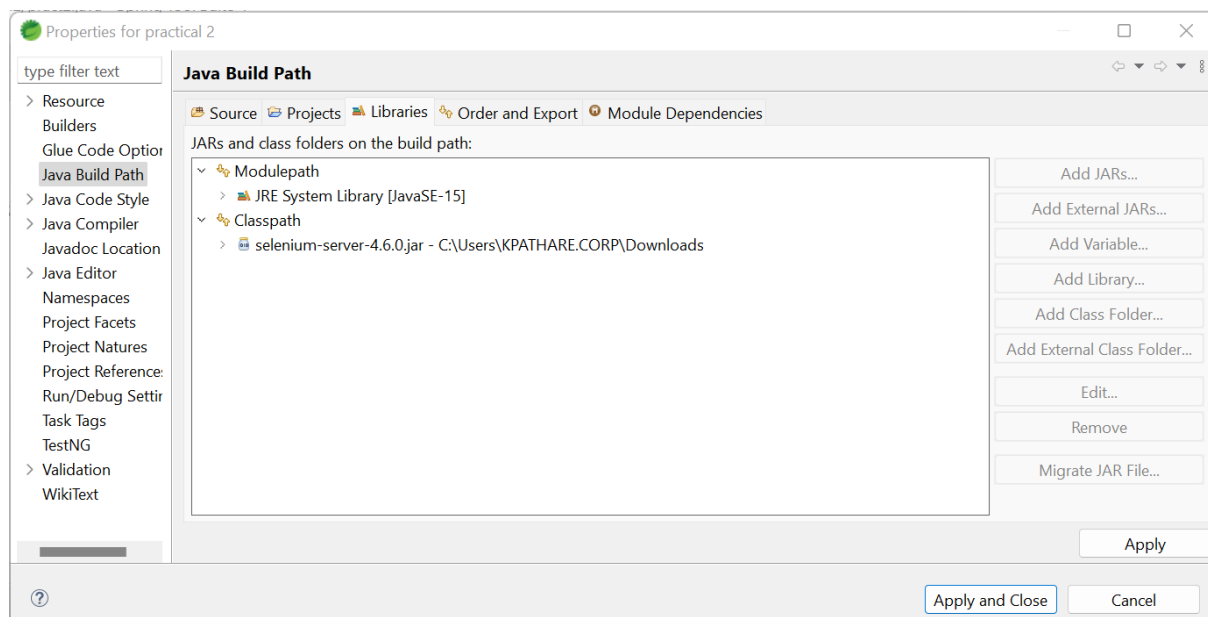
Step2: - create a new java class



The screenshot shows the 'New Java Class' dialog box. At the top, it says 'Java Class' and has a red error message: 'Type already exists.' The 'Source folder' is set to 'practical 2/src', 'Package' is 'practical2', and 'Enclosing type' is empty. The 'Name' is 'pract2'. Under 'Modifiers', 'public' is selected. The 'Superclass' is 'java.lang.Object'. Under 'Which method stubs would you like to create?', 'public static void main(String[] args)' is checked. At the bottom, there are 'Finish' and 'Cancel' buttons.

Modifier should be public it going on web Brower

Step3:- Add external jar.



The screenshot shows the 'Properties for practical 2' dialog box, specifically the 'Java Build Path' tab. The 'Libraries' section shows 'JRE System Library [JavaSE-15]' and 'selenium-server-4.6.0.jar - C:\Users\KPATHARE.CORP\Downloads'. The 'Add External JARs...' button is highlighted. At the bottom, there are 'Apply and Close' and 'Cancel' buttons.

Step4:Code


```

package practical2;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class pract2 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://www.selenium.dev/documentation/overview/");
        driver.manage().window().maximize();
    }
}

```

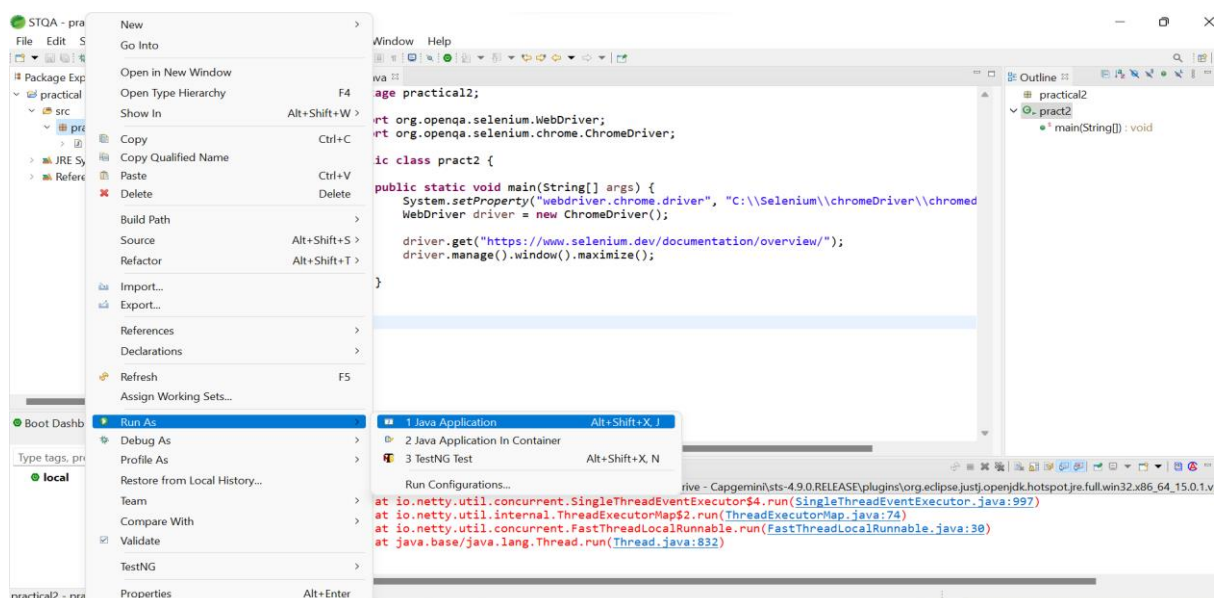
&

```

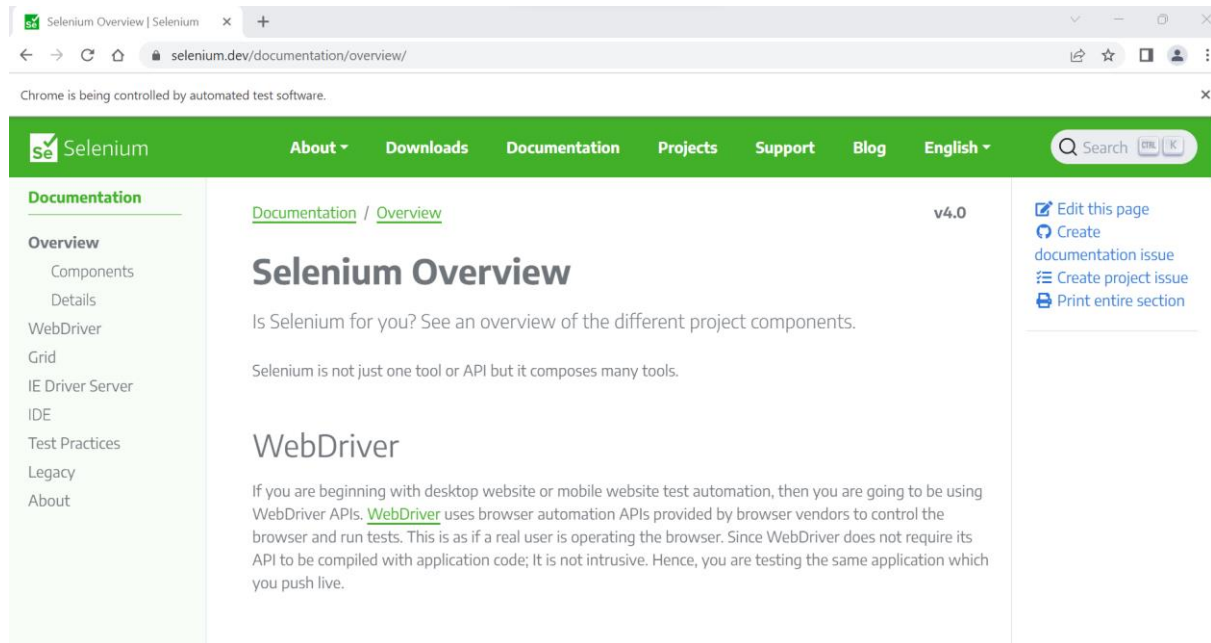
package practical2;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class practical_2 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.setProperty("webdriver.gecko.driver", "C:\\geckodriver-v0.30.0-
win64\\geckodriver.exe");
        WebDriver driver=new FirefoxDriver();
        driver.get("https://www.selenium.dev/documentation/overview/");
        driver.manage().window().maximize();
    }
}

```

Step5: run as java application



OUTPUT:-



The screenshot shows a web browser displaying the Selenium Overview page. The browser's address bar shows the URL `selenium.dev/documentation/overview/`. The page has a green header with the Selenium logo and navigation links: About, Downloads, Documentation, Projects, Support, Blog, and English. A search bar is located on the right side of the header. The main content area is divided into a left sidebar and a main body. The sidebar contains a 'Documentation' section with a list of links: Overview, Components, Details, WebDriver, Grid, IE Driver Server, IDE, Test Practices, Legacy, and About. The main body features the title 'Selenium Overview' and a subtitle 'v4.0'. Below the title, there is a paragraph stating 'Is Selenium for you? See an overview of the different project components.' and another paragraph stating 'Selenium is not just one tool or API but it composes many tools.' The 'WebDriver' section is also visible, starting with the text 'If you are beginning with desktop website or mobile website test automation, then you are going to be using WebDriver APIs. WebDriver uses browser automation APIs provided by browser vendors to control the browser and run tests. This is as if a real user is operating the browser. Since WebDriver does not require its API to be compiled with application code; It is not intrusive. Hence, you are testing the same application which you push live.'

Documentation / Overview v4.0

Selenium Overview

Is Selenium for you? See an overview of the different project components.

Selenium is not just one tool or API but it composes many tools.

WebDriver

If you are beginning with desktop website or mobile website test automation, then you are going to be using WebDriver APIs. [WebDriver](#) uses browser automation APIs provided by browser vendors to control the browser and run tests. This is as if a real user is operating the browser. Since WebDriver does not require its API to be compiled with application code; It is not intrusive. Hence, you are testing the same application which you push live.

[Edit this page](#)
[Create documentation issue](#)
[Create project issue](#)
[Print entire section](#)

Practical No: 3

Aim: Demonstrate handling multiple frames in selenium

Description: Iframe in selenium Webdriver is a webpage of an inline frame which is embedded in another webpage or an HTML document embedded inside the HTML document.

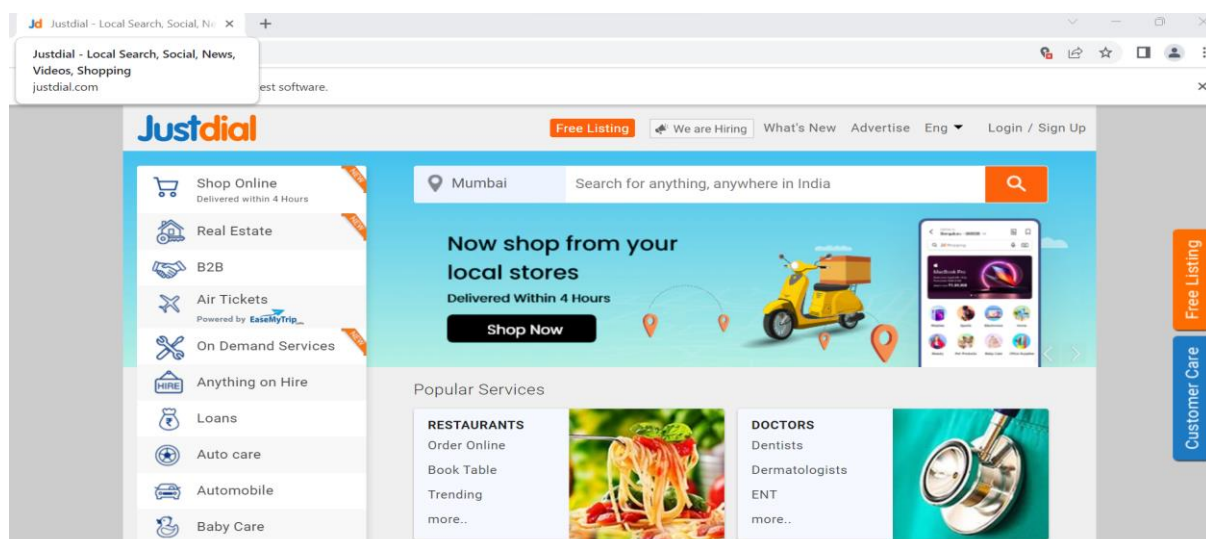
Code:

```
package Practical3;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Pract3 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.justdial.com/");
        driver.manage().window().maximize();
        driver.switchTo().frame("GTM-5CQCS9");
        System.out.println("***We are switch to the iframe***");
        driver.findElement(By.xpath("html/body/a/img")).click();
        System.out.println("***We are done***");
    }
}
```

Output:



Practical No: 4

Aim: Implement browser command and navigation commands.

A] Implement Browser Command

- 1) Absolute Xpath
- 2) Relative Xpath

Syntax for Xpath

Xpath=//target [@ attribute = 'value']

Absolute path will be come as

/html/body/div[2]/div/h4[1]/b/html[1]/body[1]/div[2]/div[1]/div[1]/

- 1) Absolute xpath

Goto the <https://demo.guru99.com/test/selenium-xpath.html> this url and select any then right click -> selectorHub -> copy abs xpath

Abs path is /html[1]/body[1]/div[4]/div[1]/div[1]/ul[1]/li[1]/a[1]

```
package practical4A;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Pract4A {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.javapoint.com/");
        String Title = driver.getTitle();
        System.out.println("WebPage Title is "+Title);
        String CurrentURL=driver.getCurrentUrl();
        System.out.println("WebPage CurrentURL is "+CurrentURL);
        String getPageSource =driver.getPageSource();
        System.out.println("WebPage PageSource is " +getPageSource);
    }
}
```

Welcome to javapoint.com

javapoint.com

Chrome is being controlled by automated test software.

Contributors 10698

Rent \$100.00/mo Buy Contribute & Earn Inquire

Play Sponsor Earn

Hey you! Join and help us build the next challenge platform

STQA - practical4A/src/practical4A/Pract4A.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

Problems Javadoc Declaration Console

```
<terminated> Pract4A [Java Application] C:\Users\KPATHARE\OneDrive - Capgemini\sts-4.9.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v2021027-0507\jre\bin\javaw.exe (29-Nov-2022, 2:04:20)
Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aeee66d25075c159aa-refs/branch-heads/53048{#942}) on port 26120
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
WebPage Title is =Welcome to javapoint.com
WebPage CurrentURL is =https://www.javapoint.com/
WebPage PageSource is =<html lang="en"><head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Welcome to javapoint.com</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta http-equiv="content-language" content="en-US">
<meta name="title" content="javapoint.com - Welcome to javapoint.com">
<meta name="description" content="Join a vibrant community of developers, influencers, and entrepreneurs on javapoint.com, all using the versatile CONTRIB token to powe
<meta name="keywords" content="java, point">
<meta name="author" content="javapoint.com">
<link rel="canonical" href="https://javapoint.com/">
<meta name="robots" content="INDEX, FOLLOW">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94wRhftJDbrCEXSL
<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@100;200;300;400;500;600;700;800;900&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.6.1/font/bootstrap-icons.css">
<!-- <script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*'new Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','//www.google-analytics.com/analytics.js','ga');

ga('create', '', '');
ga('send', 'pageview');
</script> -->

<!--<link rel="stylesheet" href="/css/custom.css">-->
<style>
```

B] Implement Navigation Commands

Code:

```
package Practical4B;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Pract4B {

    public static void main(String[] args) {
        System.getProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.google.com/");
        driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
        driver.navigate().to("https://artoftesting.com");
        driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
        driver.navigate().back();
        String str1 = driver.getCurrentUrl();
        System.out.println(str1);
        driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
        driver.navigate().forward();
        String str2 = driver.getCurrentUrl();
        System.out.println(str2);
        driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
        driver.navigate().refresh();
        String str3 = driver.getCurrentUrl();
        System.out.println(str3);
        driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
        driver.quit();
    }
}
```

Output:

```
Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 28092
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
https://www.google.com/
https://artoftesting.com/
https://artoftesting.com/
```

Learn Testing Online with Free Tutorials

artoftesting.com

Chrome is being controlled by automated test software.

ArtOfTesting

Home Manual Automation Performance Java Interview Blog

"Be a part of over 2 million testing enthusiasts that have been benefited from our free tutorials."

ArtOfTesting

Want to learn software testing but don't know how to begin? Let us walk you through everything you need to know to start your testing journey.

ArtOfTesting presents free tutorials on different testing topics ranging from manual, automation, and performance testing along with interview preparation for the same.

Here you will find step by step guide to some of the most popular test automation and performance testing tools like – Selenium, Katalon Studio,

Practical No: 5

Aim: Implement the find element command

Description:

Interaction with a web page requires a user to locate the web element. Find Element command is used to uniquely identify a (one) web element within the web page. Whereas, Find Elements command is used to uniquely identify the list of web elements within the web page. There are multiple ways to uniquely identify a web element within the web page such as ID, Name, Class Name, Link Text, Partial Link Text, Tag Name

Code:

```
package Practical5;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Pract5 {
    public static void main(String[] args) throws Exception {
        System.getProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.calculator.net/mass-calculator.html");
        //by id
        driver.findElement(By.id("cdensity")).clear();
        Thread.sleep(5000);
        driver.findElement(By.id("cdensity")).sendKeys("500");
        Thread.sleep(5000);

        //by name
        driver.findElement(By.name("cvolume")).clear();
        Thread.sleep(5000);
        driver.findElement(By.name("cvolume")).sendKeys("5");
        Thread.sleep(5000);

        //by classname
        driver.findElement(By.className("inhalf")).clear();
        Thread.sleep(5000);
        driver.findElement(By.className("inhalf")).sendKeys("200");
```



```
Thread.sleep(5000);

//by linktext
driver.findElement(By.linkText("Age")).click();
driver.navigate().back();
Thread.sleep(5000);

//by partial link text
driver.findElement(By.partialLinkText("Time")).click();
driver.navigate().back();
Thread.sleep(5000);

//by CSS selector
driver.findElement(By.cssSelector("table.paneltbody:nth-child(1) tr:nth-child(3) td:nth-child(1) <input:nth-child(2)")).click();
driver.navigate().back();
Thread.sleep(5000);

//by xpath
driver.findElement(By.xpath("//tbody/tr[3]/td[1]/img[1]")).click();
Thread.sleep(5000);
driver.quit();
}
}
```

Output:

Mass Calculator

home / other / mass calculator

Mass Calculator

This is a basic mass calculator based on density and volume. This calculator takes and generates results of many common units.

Modify the values and click the Calculate button to use

Density	<input type="text" value="500"/>	<input type="text" value="kilogram/cubic meter [kg/m³]"/>
Volume	<input type="text" value="5"/>	<input type="text" value="cubic meter [m³]"/>

What is mass?

Practical No: 6

Aim: Demonstrate the Locator(id, css selector, path)

Description:

ID is the most common way of locating elements since ID's are supposed to be unique for each element.

SS Selectors are one of the locator strategies offered by Selenium to identify the web elements. The CSS Selectors mainly use the character sequence pattern, which identifies the web elements based on their HTML structure.

XPath is a Selenium technique to navigate through a page's HTML structure. It enables testers to navigate through any document's XML structure, which can be used on both HTML and XML documents

Code:

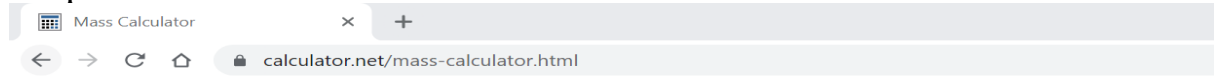
```
package Practical6;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class pract6 {
    public static void main(String[] args) throws Exception {
        System.getProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.calculator.net/mass-calculator.html");
        //by id
        driver.findElement(By.id("cdensity")).clear();
        Thread.sleep(5000);
        driver.findElement(By.id("cdensity")).sendKeys("500");
        Thread.sleep(5000);

        //by CSS selector
        driver.findElement(By.cssSelector("table.paneltbody:nth-child(1) tr:nth-
child(3) td:nth-child(1) <input:nth-child(2)")).click();
        driver.navigate().back();
        Thread.sleep(5000);

        //by xpath
        driver.findElement(By.xpath("//tbody/tr[3]/td[1]/img[1]")).click();
        Thread.sleep(5000);
    }
}
```

```
}  
    driver.quit();  
}
```


Output:




[home](#) / [other](#) / [mass calculator](#)

Mass Calculator

This is a basic mass calculator based on density and volume. This calculator takes and generates results of many common units.

 Modify the values and click the Calculate button to use

Density	<input type="text" value="500"/>	<input type="text" value="kilogram/cubic meter [kg/m³]"/>
Volume	<input type="text" value="1"/>	<input type="text" value="cubic meter [m³]"/>
<div><div>Calculate </div><div>Clear</div></div>		

What is mass?

Practical No: 7

Aim: Demonstrate synchronization in selenium

Description:

Synchronization has a very vital role to play in automation. Code execution and application need to be in sync to perform the operation. If the application slows down for any reasons like network, heavy load, etc then the code keeps on checking for the particular web element.

Code:

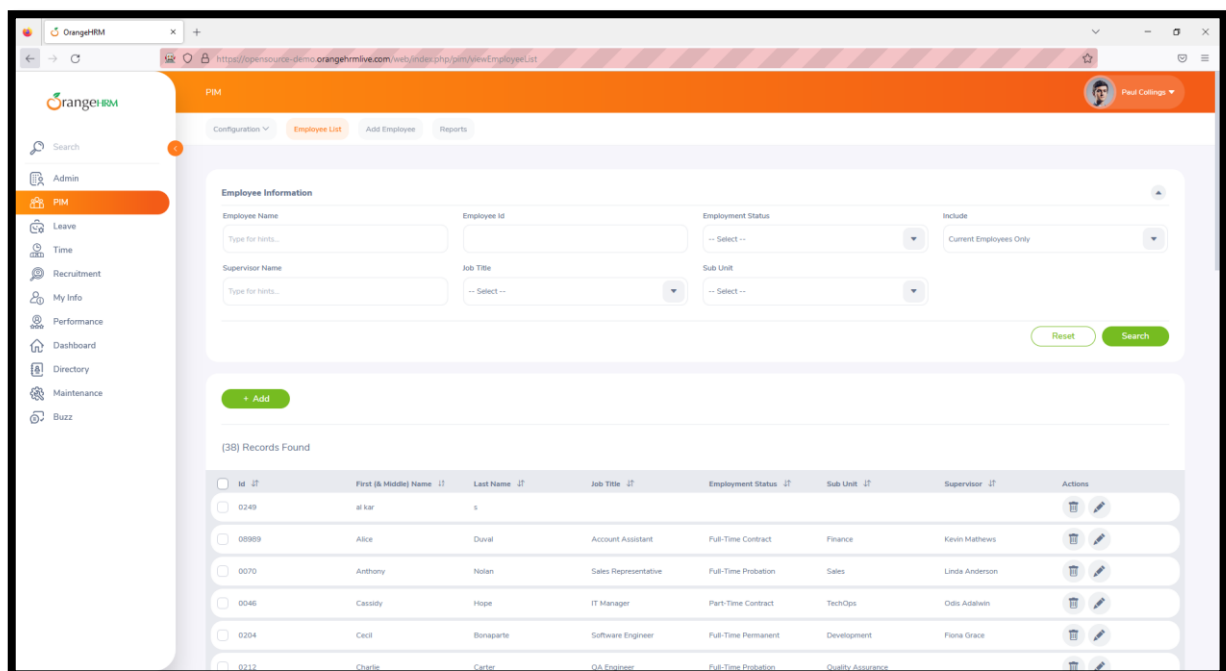
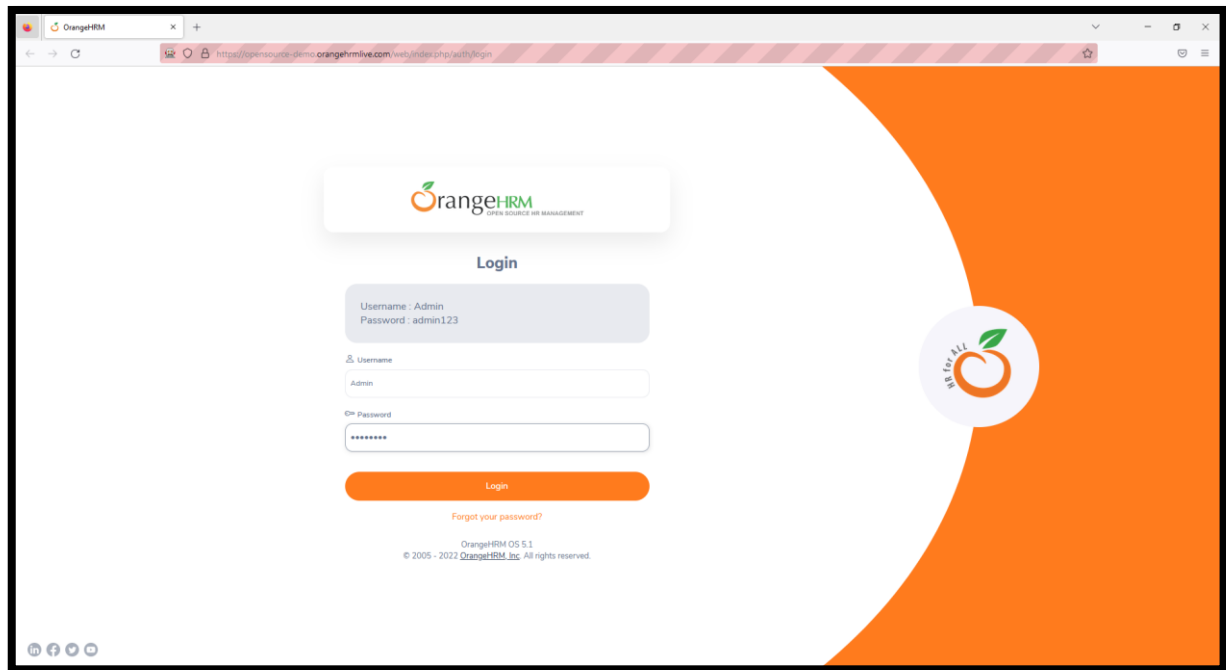
```
package Practical7A;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Pract7A {
    public static void main(String[] args) {

        System.setProperty("webdriver.chrome.driver", "C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://opensource-demo.orangehrmlive.com");

        driver.findElement(By.id("txtusername")).sendKeys("admin");
        driver.findElement(By.id("txtusername")).sendKeys("admin123");
        driver.findElement(By.id("button")).click();

        driver.findElement(By.partialLinkText("Welcome")).click();
        driver.findElement(By.xpath("//*[@id=\"Welcome\"]")).click();
        driver.findElement(By.linkText("Logout")).click();
    }
}
```

Output:



B]package Practical7B;

Code:

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
```

```
public class Pract7B{  
    public static void main(String[] args) {  
        System.setProperty("webdriver.chrome.driver","C:\\Selenium\\chromeDriver\\chromedriver107.exe"); WebDriver driver = new ChromeDriver();  
        WebDriverWait wt=new WebDriverWait(driver,10);  
        driver.get("https://opensource-demo.orangehrmlive.com");  
        driver.findElement(By.id("txtusername")).sendKeys("admin");  
        driver.findElement(By.id("txtusername")).sendKeys("admin123");  
        driver.findElement(By.id("button")).click();  
        driver.findElement(By.partialLinkText("Welcome")).click();  
        driver.findElement(By.xpath("//*[@id=\"Welcome\"]")).click();  
        wt.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Logout")));  
        driver.findElement(By.linkText("Logout")).click(); } }
```

Practical No: 8

Aim: Demonstrate different types of alerts

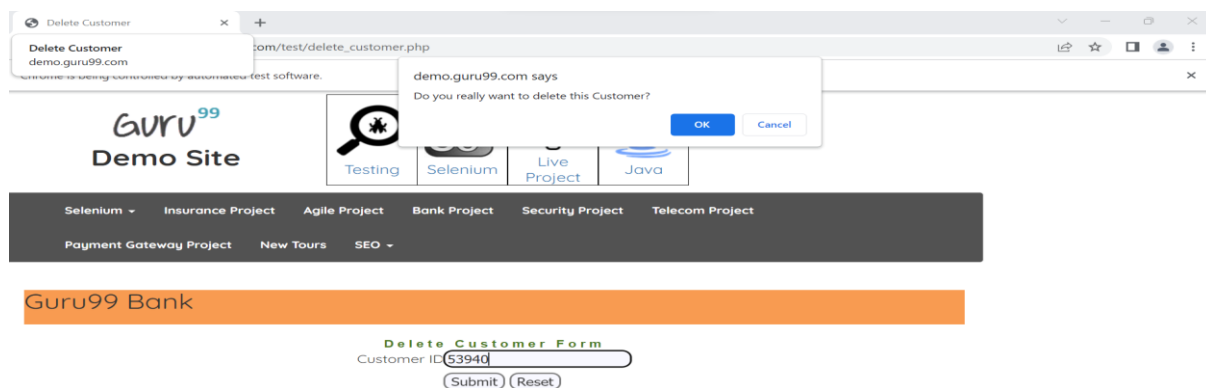
Description: An Alert in Selenium is a small message box which appears on screen to give the user some information or notification. It notifies the user with some specific information or error, asks for permission to perform certain tasks and it also provides warning messages as well.

Types of alerts are simple alert, prompt alert, Confirmation alert.

Code:

```
package Practical8;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Pract8 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.guru99.com/test/delete_customer.php");
        driver.findElement(By.name("cusid")).sendKeys("53940");
        driver.findElement(By.name("submit")).submit();
        driver.switchTo().alert();
        driver.switchTo().alert().getText();
    }
}
```

Output:



Practical No: 9

Aim: Demonstrate Handling Drop Down & List Boxes

Description:

The 'Select' class in Selenium WebDriver is used for selecting and deselecting option in a dropdown. The objects of Select type can be initialized by passing the dropdown webElement as parameter to its constructor.

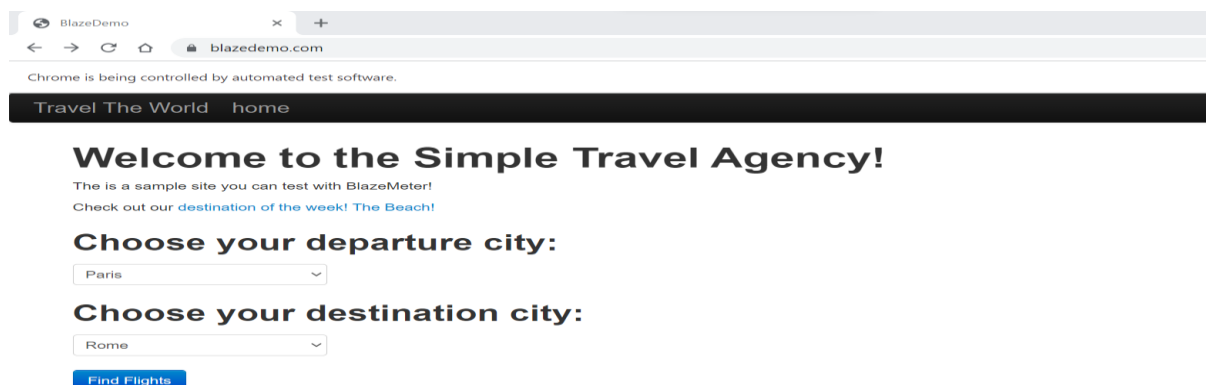
Code:

```
package Practical9;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
public class Pract9 {

    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://blazedemo.com/");
        Select f = new Select(driver.findElement(By.name("fromPort")));
        Select t = new Select(driver.findElement(By.name("toPort")));
        f.selectByVisibleText("Paris");
        t.selectByVisibleText("Rome");
    }
}
```

Output:



Practical No:10

Aim: Demonstrate Command Button, Radio button & text boxes, Waits command in selenium

Description:

Radio Buttons too can be toggled on by using the click() method.

Code:

```
package Practical10;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.*;
public class Pract10 {
    public static void main(String[] args) {
        // declaration and instantiation of objects/variables
        System.setProperty("webdriver.chrome.driver", "C:\\Selenium\\chromeDriver\\chromedriver107.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://demo.guru99.com/test/radio.html");
        WebElement radio1 = driver.findElement(By.id("vfb-7-1"));
        WebElement radio2 = driver.findElement(By.id("vfb-7-2"));
        //Radio Button1 is selected
        radio1.click();
        System.out.println("Radio Button Option 1 Selected");
        //Radio Button1 is de-selected and Radio Button2 is selected
        radio2.click();
        System.out.println("Radio Button Option 2 Selected");
    }
}
```

Output:

Radio

☐Option1

☒Option2

☐Option3

Practical No: 11

Aim: Demonstrate action classes in selenium

Description:

It's important to create the object of action class for used the method of action class unless and until not Create the object of action class we can't perform the events.

Actions class object is created with any name but it's the good or professional practice to write builder.

```
package practical11;
```

```
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.Action;  
import org.openqa.selenium.interactions.Actions;
```

```
public class Pract11 {
```

```
    public static void main(String[] args) {  
        System.setProperty("webdriver.chrome.driver",  
"C:\\Selenium\\chromeDriver\\chromedriver107.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.get("http://demo.guru99.com/test/newtours/");  
        driver.manage().window().maximize();  
        WebElement link_home=driver.findElement(By.linkText("Home"));  
        WebElement  
td_home=driver.findElement(By.xpath("//html/body/div"+"table/tbody/tr/td"+"table/tbody  
y/tr/td"+"table/tbody/tr/td"+"table/tbody/tr"));  
  
        Actions builder=new Actions(driver);  
        Action mouseOverHome=builder.moveToElement(link_home).build();  
        String bgcolor=td_home.getCssValue("background-color");  
        System.out.println("Before hover:" +bgcolor);  
        ((Action) mouseOverHome).perform();  
        bgcolor=td_home.getCssValue("background-color");
```

```
        System.out.println("After hover:" +bgcolor);  
        driver.close();  
    }  
}
```

Output:

Before hover:rgba(255, 165, 0, 1)

After hover:rgba(0, 0, 0, 0)

Practical No :12

Aim: Installation of TestNG, running testing and TestNG annotation

Description:

Next generation nothing but we are using annotation for our testing , because of test ng we can control execution of our test cases throw annotation (proration of task)

Test-cases are equally important and execute at the same time.

Test ng is frame work or different type of plugging provided by selenium

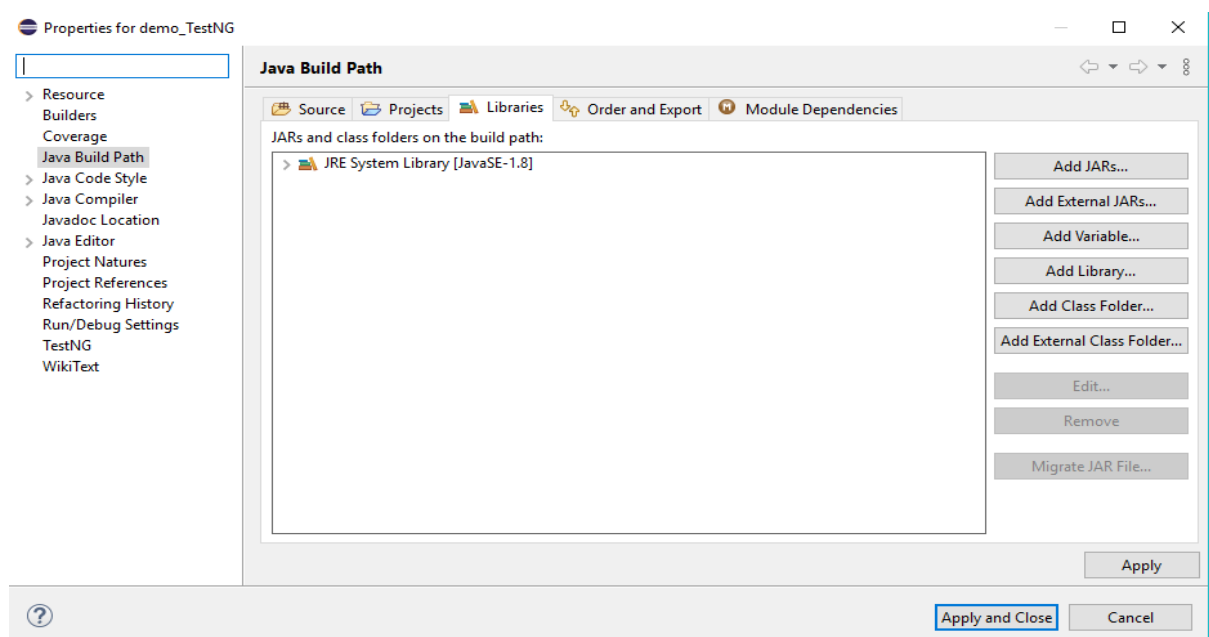
While doing practical select add library and select **JUnit**

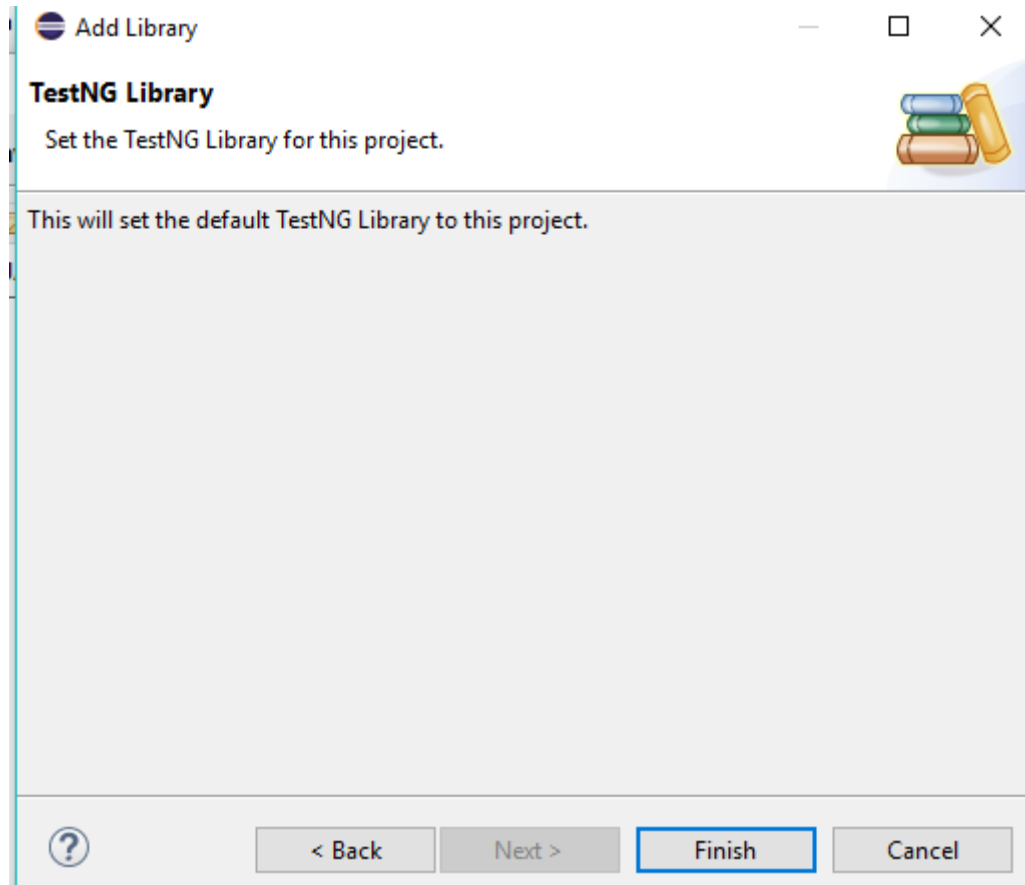
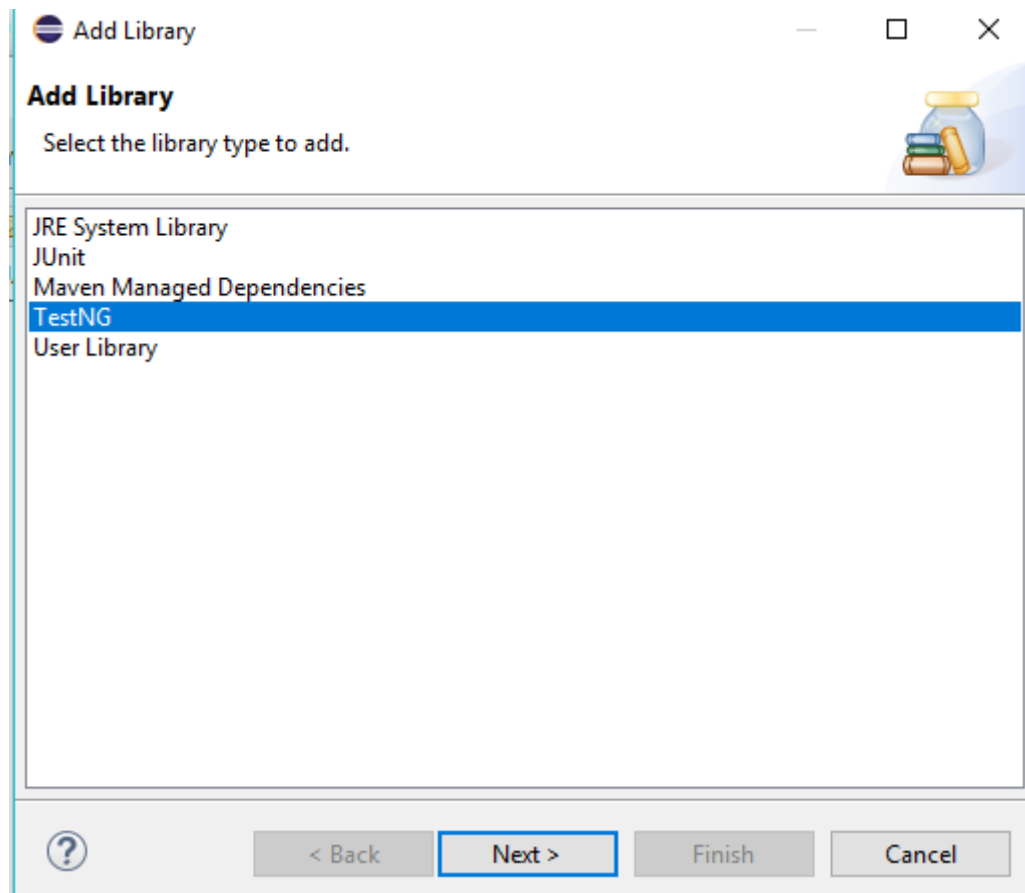
Selenium code -> TestNG -> XLS Files -> Logs (maintain records up to test the cases)-> Reports generated in html

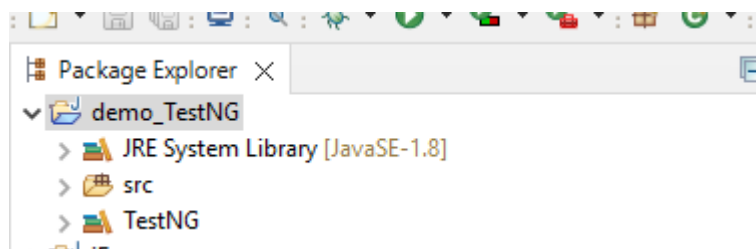
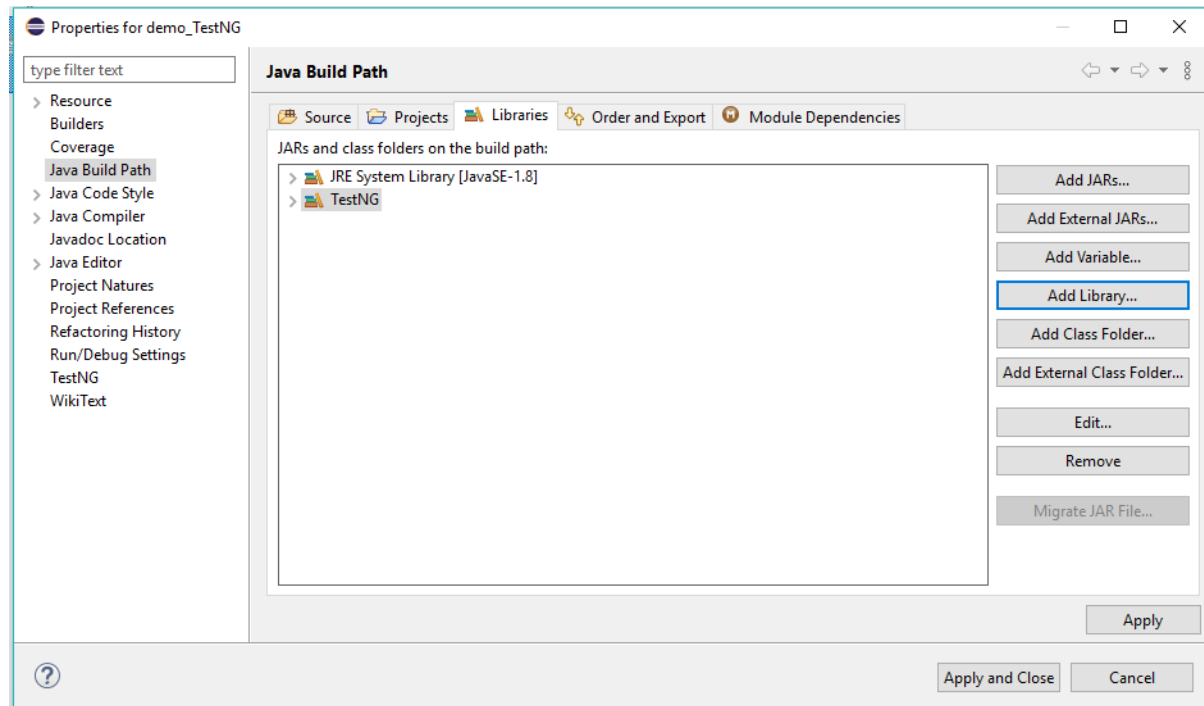
Setps for install TestNg

- 1) Open eclipse
- 2) Go to help and select eclipse marketplace
- 3) Search TestNG and install it
- 4)

Add libabry







Practical No : 13

AIM: Demonstrate data driven Framework.

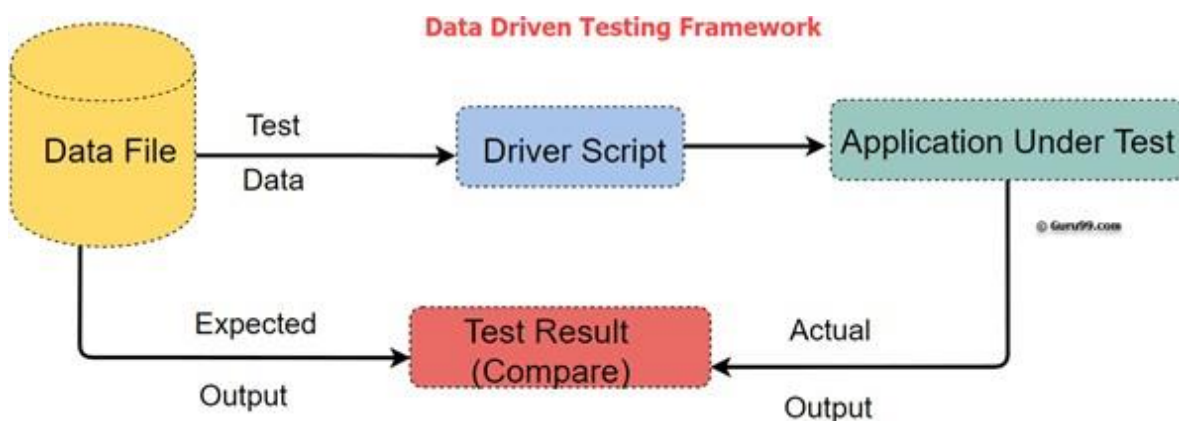
Theory:

Data Driven Testing

Data Driven Testing is a software testing method in which test data is stored in table or spreadsheet format. Data driven testing allows testers to input a single test script that can execute tests for all test data from a table and expect the test output in the same table. It is also called table-driven testing or parameterized testing.

Data Driven Framework

Data Driven Framework is an automation testing framework in which input values are read from data files and stored into variables in test scripts. It enables testers to build both positive and negative test cases into a single test. Input data in data driven framework can be stored in single or multiple data sources like .xls, .xml, .csv and databases.



Code:

```
package pracs;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver; import
org.testng.Assert;
import org.testng.annotations.AfterMethod; import
org.testng.annotations.DataProvider; import
org.testng.annotations.Test;

public class Thirteenth { WebDriver driver;
    @Test(dataProvider="testdata")
    public void demoClass(String username, String password) throws
InterruptedException {
        System.setProperty("webdriver.gecko.driver","geckodriver.exe"); driver = new
        FirefoxDriver();

        driver.get("https://www.phptravels.net/login");
        driver.findElement(By.name("email")).sendKeys(username);
        driver.findElement(By.name("password")).sendKeys(password);

        driver.findElement(By.xpath("/html/body/div[1]/div/div[2]/div[2]/div/form/ div[3]/button")).click();
        Thread.sleep(5000); Assert.assertTrue(driver.getTitle().matches("Dashboard -
        PHPTRAVELS"), "Invalid credentials");
        System.out.println("Login successful");
    }

    @AfterMethod
    void ProgramTermination() { driver.quit();
```



```
}
```

```
@DataProvider(name="testdata") public  
Object[][] testDataExample(){
```

```
    ReadExcelFile configuration = new ReadExcelFile("F:\\STQA  
WORKSPACE\\stqa\\src\\pracs\\XYZ.xlsx");  
    int rows = configuration.getRowCount(0); Object[][]signin_credentials = new  
    Object[rows][2];  
  
    for(int i=0;i<rows;i++)  
    {  
  
        signin_credentials[i][0] = configuration.getData(0, i, 0); signin_credentials[i][1] =  
        configuration.getData(0, i, 1);  
    }  
  
    return signin_credentials;  
}
```

Output:

Data in Excel sheet:


atharva@gmail.com	xyz
tony@gmail.com	abc
user@phptravels.com	demouser

Wrong credentials

[Home](#) [Hotels](#) [Flights](#) [Tours](#) [Cars](#) [Visa](#) [Blog](#) [Company](#) ▾


Login
Please enter your account credentials below

Email



atharva@gmail.com

Password



...

☐ Remember Me


[Reset Password](#)

Signup

[Home](#) [Hotels](#) [Flights](#) [Tours](#) [Cars](#) [Visa](#) [Blog](#) [Com](#)


Login
Please enter your account credentials below

Email



tony@gmail.com

Password



...


☐ Remember Me

[Reset Password](#)

Signup

Login

Please enter your account credentials below

 Wrong credentials. try again!

Email

 Email

Password


 Password☐ Remember Me[Reset Password](#)**Login**[Signup](#)**Correct credentials**[Home](#) [Hotels](#) [Flights](#) [Tours](#) [Cars](#) [Visa](#) [B](#)**Login**

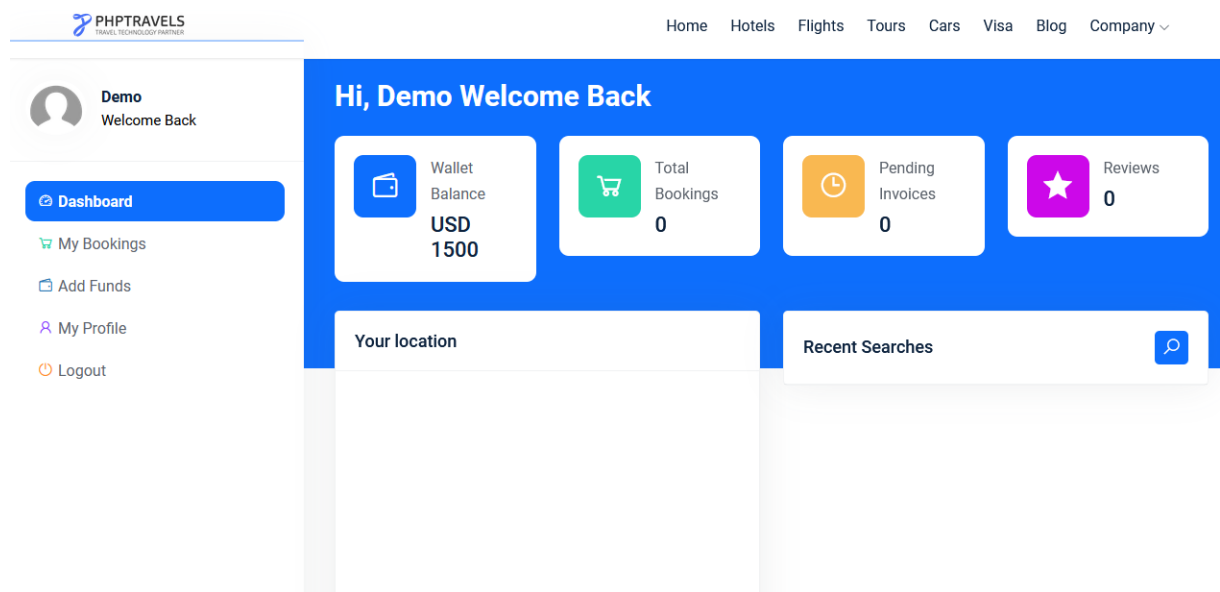
Please enter your account credentials below

Email

 user@phptravels.com

Password

☐ Remember Me[Reset Password](#)
[Signup](#)



Console OP

```
=====
Default test
Tests run: 1, Failures: 2, Skips: 0
=====

=====
Default suite
Total tests run: 3, Passes: 1, Failures: 2, Skips: 0
=====

PASSED: demoClass("user@phptravels.com", "demouser")
FAILED: demoClass("atharva@gmail.com", "xyz")

FAILED: demoClass("tony@gmail.com", "abc")
```

Practical No : 14

AIM : Asserts and Verify methods are commonly used in [Selenium](#) for verifying or validating applications.

Assertions (also known as Asserts)

The word **Assert** means to state a fact or belief confidently or forcefully. In Selenium, Asserts are validations or checkpoints for an application. Assertions state confidently that application behavior is working as expected. One can say that Asserts in Selenium are used to validate the test cases. They help testers understand if tests have passed or failed.

Types of Assertions

- Hard Assertions
- Soft Assertions (Verify Method)

Hard vs Soft Asserts in Selenium

Hard Assertions	Soft Assertions
Test Execution will be aborted if assert condition is not met	Test execution will continue till the end of the test case even if assert condition is not met
Does not have to invoke any methods to capture the assertions	To view assertions result at the end of the test, tester has to invoke <code>assertAll()</code>

Difference between Assert and Verify in selenium

- In the case of assertions, if the assert condition is not met, test case execution will be aborted. The remaining tests are skipped, and the test case is marked as failed. These assertions are used as checkpoints for testing or validating business-critical transactions.
- In case of verify, tests will continue to run until the last test is executed even if assert conditions are not met. Verify or Soft Asserts will report the errors at the end of the test. Simply put, tests will not be aborted if any condition is not met. Testers need to invoke the `assertAll()` method to view the results.

Both Hard and Soft Assertions are very important for designing and running [Selenium webdriver](#) tests. They are instrumental in verifying application behavior at critical stages. By using assertions, testing teams can determine if an application is working as it is expected to. They can also save teams the trouble of running tests that don't need to be run if a condition is not met.

Hard Assertions

Hard Assertions are ones in which test execution is aborted if the test does not meet the assertion condition. The test case is marked as failed. In case of an assertion error, it will throw the “*java.lang.AssertionError*” exception.

- **assertEquals()** is a method that takes a minimum of 2 arguments and compares actual results with expected results. If both match, then the assertion is passed and the test case is marked as passed. assertEquals() can compare Strings, Integers, Doubles and many more variables, as shown in the image below.

Below is an example of assertEquals().

Code Snippet for assertEquals() in Selenium

```
package com.tests;
import org.junit.Assert;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class BrowserStackTutorials {
    @Test
    public void testAssertFunctions() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to("https://www.browserstack.com/");
        String ActualTitle = driver.getTitle();
```

```
String ExpectedTitle = "Most Reliable App & Cross Browser Testing  
Platform | BrowserStack";  
Assert.assertEquals(ExpectedTitle, ActualTitle);  
}  
}
```

- **assertNotEquals()** is a method that does the opposite of the `assertEquals()` method. In this case, the method compares the actual and expected result. But if the assertion condition is met if the two are not identical. If actual and expected results are not the same, the test case is marked as passed.

Code For `assertNotEquals()` in Selenium

```
package com.tests;  
import org.junit.Assert;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.testng.annotations.Test;  
public class BrowserStackTutorials {  
    @Test  
    public void testAssertFunctions() {  
        System.setProperty("webdriver.chrome.driver",  
            "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
        driver.navigate().to("https://www.browserstack.com/");  
        String ActualTitle = driver.getTitle();  
        String ExpectedTitle = "Most Reliable App & Cross Browser Testing  
Platform | browserstack";  
        Assert.assertNotEquals(ActualTitle, ExpectedTitle);  
    }  
}
```

- **assertTrue():** This Assertion verifies the Boolean value returned by the condition. If the Boolean value is true, then the assertion passes the test case.

Code For assertTrue() in Selenium

```
package com.tests;
import static org.testng.Assert.assertTrue;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class BrowserStackTutorials {
    @Test
    public void testAssertFunctions() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to("https://www.browserstack.com/");
        Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable
        App & Cross Browser Testing Platform | BrowserStack");
        assertTrue(verifyTitle);
    }
}
```

Code For assertFalse() in Selenium

```
package com.tests;
import static org.testng.Assert.assertFalse;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class BrowserStackTutorials {
    @Test
    public void testAssertFunctions() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
```



```
driver.navigate().to("https://www.browserstack.com/");
Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable
App & Cross Browser Testing Platform");
assertFalse(verifyTitle);
}
}
```

-
- **assertNull():** This method verifies if the expected output is null. If not, the value returned is false.

Code Snippet For assertNull() in Selenium

```
package com.tests;
import static org.testng.Assert.assertNull;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
public class BrowserStackTutorials {
    @Test
    public void testAssertFunctions() {
        System.setProperty("webdriver.chrome.driver",
            "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.navigate().to("https://www.browserstack.com/");
        String verifyAssertNull = null;
        assertNull(verifyAssertNull);
    }
}
```

-
- **assertNotNull():** This method works opposite to that of the assertNull() method. The assertion condition is met when the method validates the expected output to be not null.

Code For assertNotNull() in Selenium

```
package com.tests;
import static org.testng.Assert.assertNotNull;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class BrowserStackTutorials {
// Author: Chaitanya Pujari
@Test
public void testAssertFunctions() {
System.setProperty("webdriver.chrome.driver",
"C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.navigate().to("https://www.browserstack.com/");
Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable
App & Cross Browser Testing Platform");
assertNotNull(verifyTitle);
}
}
```

Example of Hard Assert in Selenium

```
package com.tests;
import static org.testng.Assert.assertEquals;
import static org.testng.Assert.assertFalse;
import static org.testng.Assert.assertNotEquals;
import static org.testng.Assert.assertNotNull;
import static org.testng.Assert.assertNull;
import static org.testng.Assert.assertTrue;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class BrowserStackTutorials {
@Test
```

```
public void testAssertFunctions() {
    System.setProperty("webdriver.chrome.driver",
        "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.navigate().to("https://www.browserstack.com/");
    String ActualTitle = driver.getTitle();
    String verifyAssertNull=null;
    String ExpectedTitle = "Most Reliable App & Cross Browser Testing
    Platform | BrowserStack";
    Boolean verifyTitleIsPresent=driver.getTitle().equalsIgnoreCase("Most
    Reliable App & Cross Browser Testing Platform | BrowserStack");
    Boolean
    verifyTitleIsChanged=driver.getTitle().equalsIgnoreCase("Testing
    Platform | BrowserStack");
    assertEquals(ExpectedTitle, ActualTitle);
    assertNotEquals(ExpectedTitle, "browserstack");
    assertTrue(verifyTitleIsPresent);
    assertFalse(verifyTitleIsChanged);
    assertNotNull(verifyTitleIsPresent);
    assertNull(verifyAssertNull);
}
}
```

Verify in Selenium (also known as Soft Assertion)

In a hard assertion, when the assertion fails, it terminates or aborts the test. If the tester does not want to terminate the script they cannot use hard assertions. To overcome this, one can use soft assertions.

Let's explore the different types of soft assertions with examples (verify).

Example of Soft Assert in Selenium (or Verify in Selenium)

```
package com.tests;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;
public class BrowserStackTutorials {
```

```
@Test
public void softAssert() {
    System.setProperty("webdriver.chrome.driver",
        "C:\\I2EWebsiteTest\\Driver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    SoftAssert softAssert = new SoftAssert();
    driver.navigate().to("https://www.browserstack.com/");
    String getActualTitle = driver.getTitle();
    Boolean verifyTitle = driver.getTitle().equalsIgnoreCase("Most Reliable
    App & Cross Browser Testing Platform | BrowserStack");
    softAssert.assertEquals(getActualTitle, "Most Reliable App & Cross
    Browser Testing Platform | BrowserStack");
    softAssert.assertNotEquals(getActualTitle, "Most Reliable App & Cross
    Browser Testing Platform | BrowserStack");
    softAssert.assertNull(verifyTitle);
    softAssert.assertNotNull(verifyTitle);
    softAssert.assertTrue("BrowserStack".equals("Browserstack"), "First soft
    assert failed");
    softAssert.assertFalse("BrowserStack".equals("BrowserStack"), "Second
    soft assert failed");
    softAssert.assertAll();
}
```

Practical No : 15

AIM : Regression testing in selenium

[Regression Testing](#) is a kind of testing that is done to check the behavior of an application after a new functionality has been introduced or bug fix has been implemented. It checks whether the new functionality is not affecting the existing application behavior.

Regression Testing with Selenium

[Selenium](#) is a web-based automation testing framework. It helps in automating functional and regression test cases that reduce the manual testing effort. Usually, regression suites include a huge number of test cases and it takes time and effort to execute them manually every time when a code change has been introduced. Hence almost every organization looks after automating regression test cases to reduce the time and effort. Choosing the right automation framework/ tool completely depends upon the application, technology used, testing requirements, and skill sets required for performing automation testing.

There are four components in Selenium – [Selenium Webdriver](#), [Selenium IDE](#), [Selenium RC](#), and [Selenium Grid](#). Each of these is used for different testing purposes. Selenium Webdriver provides an interface that helps us develop automation scripts that interact with the browser and perform. Various browsers like Chrome, Edge, Firefox, IE, and Opera are supported by Selenium. Selenium also supports multiple programming languages like [Java](#), [Python](#), [Javascript](#), Ruby, etc.

Let's see some best practices that should be considered for regression testing.

- **Defining Test Strategy:** The test strategy defined may include the test cases to be considered for regression, estimates for test execution enhancements required to the existing test cases, and the new test cases if required.
- **Maintaining/updating Regression suites:** Testing teams have to regularly maintain the regression suites to check for any new failures, test script enhancements required, etc.

- **Test Automation:** Automating regression tests is a best practice to save the time and efforts required to execute regression tests manually every time during a release. There are multiple approaches for automating test cases like the one mentioned above using Selenium. Selenium can be used along with the Page object model (POM) design pattern, Data-driven, keyword-driven frameworks, etc.

How to Perform Regression Testing Using Selenium?

Automation completely depends on the framework that you choose to develop, and there is no such tool dedicated to performing only regression testing. The automation framework you select should be designed such that it supports regression testing effectively.

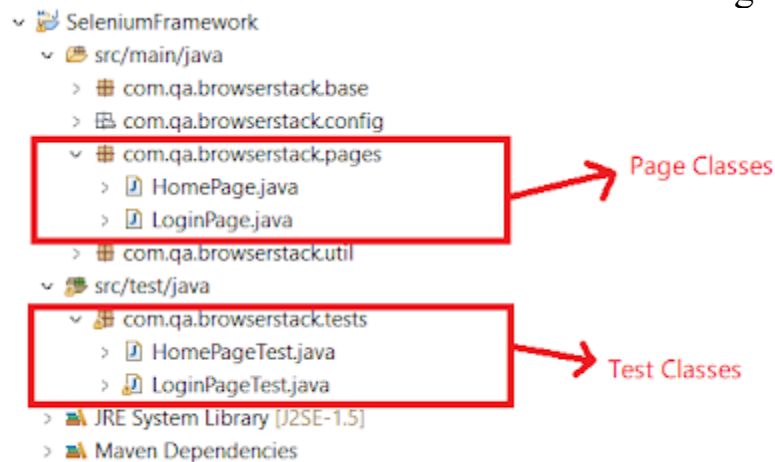
You can develop the regression suite for automation and keep adding new test scripts/test cases as and when required. Selenium Framework contains many reusable modules/functions that make it easy to maintain the existing code or add any new code.

You can integrate Selenium with TDD frameworks like TestNG, Junit Maven, etc. [TestNG annotations](#) help in writing automation scripts effectively. You can also use the [Page Object Model](#) design pattern while building an automation framework.

The page object model is a design pattern that makes it easy to maintain code, reduces complexity, and increases code reusability. In POM there is a separate class for each application web page. In these page classes, there are page objects and corresponding methods that implement these page objects while interacting with the browser.

Also, there are separate Test classes in which you can write your test cases using TestNG or Junit. You can also add assertions and verifications in your Test classes. The fact that verifications are separated from our page operations in page classes makes POM easy to understand and simplified.

Let's see the below framework structure using POM:



In the above structure, there are two Page classes – HomePage and LoginPage. Similarly, there are two corresponding test classes – HomePageTest and LoginPageTest.

LoginPage class

```
package com.qa.browserstack.pages;

import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;

import com.qa.browserstack.base.BasePage;

public class LoginPage extends BasePage {

    WebDriver driver;

    By emailID = By.id("user_email_login");

    By password = By.id("user_password");

    By SignIn = By.cssSelector("li.sign-in-link>a");
```

```
By Login = By.id("user_submit");

By checkBox = By.id("tnc_checkbox");

public LoginPage(WebDriver driver)
{
    this.driver = driver;
}

public String getLoginPageTitle()
{
    return driver.getTitle();
}

public void doLogin(String username,String pwd) {
    driver.findElement(SignIn).click();
    driver.findElement(emailID).sendKeys(username);
    driver.findElement(password).sendKeys(pwd);
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));
    driver.findElement(Login).click();
}

public Boolean signInLinkIsDisplayed()
{
    boolean signIn;
```



```
signIn = driver.findElement(SignIn).isDisplayed();

return signIn;

}

}
```

In this Page class, page objects like emailID, password, signIn are designed first and then there are corresponding methods like getLoginPageTitle, doLogin, signInLinkIsDisplayed that implement these page objects to interact with the browser.

LoginPageTest class

```
package com.qa.browserstack.tests;

import com.qa.browserstack.base.BasePage;

import com.qa.browserstack.pages.LoginPage;

import static org.testng.Assert.assertEquals;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.Properties;

import org.openqa.selenium.By;

import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.remote.DesiredCapabilities;
```

```
import org.openqa.selenium.remote.RemoteWebDriver;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import com.qa.browserstack.util.Constants;


public class LoginPageTest {

    BasePage basePage;

    Properties prop;

    WebDriver driver;

    LoginPage loginPg;

    @BeforeTest

    public void setUp() throws Exception

    {

        basePage = new BasePage();

        prop = basePage.init_properties();

        driver = basePage.init_driver(prop);
```

```
loginPg = new LoginPage(driver);

}

@Test(priority = 3)

public void LoginTest() throws Exception

{

loginPg.doLogin(prop.getProperty("username"),
prop.getProperty("password"));

}

@Test(priority = 2)

public void LoginPageTitleTest()

{

String title = loginPg.getLoginPageTitle();

System.out.println(title);

Assert.assertEquals(title, Constants.LOGIN_PAGE_TITLE);

}

@Test(priority = 1)

public void SignupLinkTest()

{

Assert.assertTrue(loginPg.signInLinkIsDisplayed());

}
```

```
@AfterTest

public void tearDown()

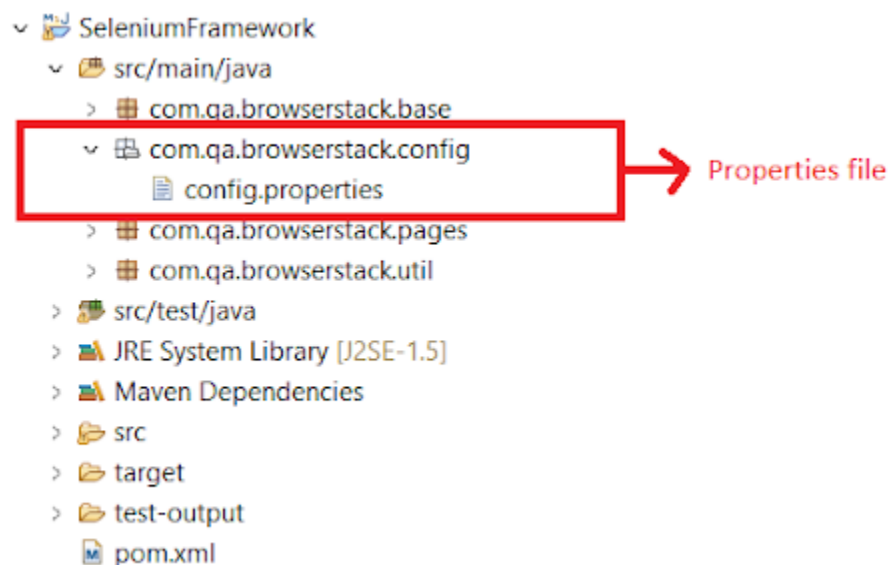
{

driver.quit();

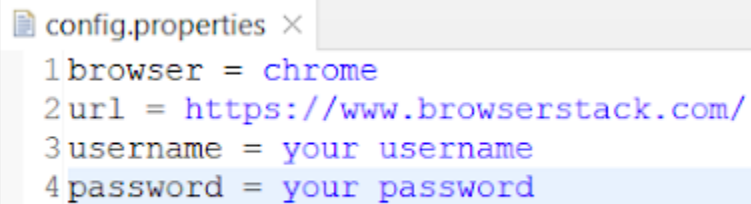
}

}
```

The above test cases are written using TestNG. Through these test cases, you can call the page class methods like doLogin, getLoginPageTitle, etc. You can also maintain the data in the properties file as shown below.



config.properties

A screenshot of a code editor window titled 'config.properties'. The editor contains four lines of code: '1 browser = chrome', '2 url = https://www.browserstack.com/', '3 username = your username', and '4 password = your password'. The fourth line is highlighted with a blue background.

```
1 browser = chrome
2 url = https://www.browserstack.com/
3 username = your username
4 password = your password
```

Properties file plays a crucial role within the automation framework and helps to implement regression testing effectively. The properties file consists of key and value pairs which we require while executing our main automation test scripts. This way, you just have to update the value of any key if required, and no major code change is required.