# Practical No: 1

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

1. Review
2. Test cases
3. Known Application
   1. [www.google.com](http://www.google.com/)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tes | Test\_case | Test Data | Input | Action | Expected | Actual | Stat | Re |
| t\_c | \_no\_descr |  |  |  | Result | Result | us | mar |
| ase | iption |  |  |  |  |  |  | k |
| \_no |  |  |  |  |  |  |  |  |
| url | url\_01\_va | url | Correct | Write | Successf | Successf | Test |  |
| \_01 | lid | [https://ww](https://www.google.com/) | url | proper | ully open | ully open | pass |
|  |  | [w.google.](https://www.google.com/) |  | url and | the | the | es |
|  |  | [com/](https://www.google.com/) |  | press | [www.go](http://www.google.com/) | [www.go](http://www.google.com/) |  |
|  |  |  |  | enter | [ogle.com](http://www.google.com/) | [ogle.com](http://www.google.com/) |  |
|  |  |  |  | key |  |  |  |
| url | url\_02\_in | url | Incorre | Write | Successf | Error | Test |  |
| \_02 | valid | [https://ww](https://www.google.com/) | ct | proper | ully open |  | fail |
|  |  | [w.google.](https://www.google.com/) | url | url and | the |  | ed |
|  |  | [com/](https://www.google.com/) |  | press | [www.go](http://www.google.com/) |  |  |
|  |  |  |  | enter | [ogle.com](http://www.google.com/) |  |  |
|  |  |  |  | key |  |  |  |
| Gl\_ | Gmail\_log | User | Correct | Write | Successf | Successf | Test |  |
| 01 | in\_valid\_0 | name and | email id | proper | ully | ully open | pass |
|  | 1 | password | and | email id | Open | the home | ed |
|  |  |  | correct | and | home of | page of |  |
|  |  |  | passwor | passwor | user | user |  |
|  |  |  | d | d and | gmail | gmail |  |
|  |  |  |  | press | page |  |  |
|  |  |  |  | the next |  |  |  |
| Gl\_ | Gmail\_log | User | Correct | Write | Error | Error | Test |  |
| 02 | in\_invalid | name and | email id | proper | message | message | pass |
|  | \_02 | password | and | email id |  |  | ed |
|  |  |  | incorrec | and |  |  |  |
|  |  |  | t | passwor |  |  |  |
|  |  |  | passwor | d and |  |  |  |
|  |  |  | d | press |  |  |  |
|  |  |  |  | the next |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| gm  \_01 | Google\_m ap\_valid\_ 01 | Destinatio n Address, GPS has to on | correct destinat ion address and on the GPS | Enter correct destinat ion address and on  the GPS | Successf ully Show the correct route | Successf ully Show the correct route | Test pass ed |  |
| gm  \_02 | Google\_m ap\_invalid  \_02 | Destinatio n Address, GPS has to on | Correct destinat ion address but GPS off | Enter correct destinat ion address and off the  GPS | Error message turn on device location | Error message turn on device location | Test pass es |  |
| gl\_ 01 | Google\_le ns\_valid\_ 01 | scan the image, turn on net | Set camera on image | Scan the proper image | Successf ully show the informati on of  image | Successf ully show the informati on of  image | Test pass ed |  |
| gl\_ 02 | Google\_le ns\_invalid  \_02 | scan the image, turn off net | set camera on image but off  the net | Scannin g proper image but net  off | Error message Somethi ng went wrong | Error message Somethi ng went wrong | Test pass ed |  |
| Ga  \_01 | Google\_A pps\_valid  \_01 | Shows the Google apps | Show the all google apps | Click on google apps icon | Successf ully show the all google  apps | Successf ully Show the all google  apps | Test pass ed |  |
| Ga  \_02 | Google\_A pps\_invali d\_02 | Shows the Google apps | Not showin g 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 pass ed |  |

* 1. [www.youtube.com](http://www.youtube.com/)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tes t\_c ase  \_no | Test\_case  \_no\_descr iption | Test Data | Input | Action | Expected Result | Actual Result | Stat us | Re mar k |
| url  \_01 | url\_01\_va lid | url [https://ww](https://www.youtube.com/) [w.youtube](https://www.youtube.com/)  [.com/](https://www.youtube.com/) | Correct url | Write proper url and press enter key | Successf ully open the [www.yo](http://www.youtube.com/) [utube.co](http://www.youtube.com/) [m](http://www.youtube.com/) | Successf ully open the [www.yo](http://www.youtube.com/) [utube.co](http://www.youtube.com/) [m](http://www.youtube.com/) | Test pass es |  |
| url  \_02 | url\_02\_in valid | url [https://ww](https://www.youtube.com/) [w.youtube](https://www.youtube.com/)  [.com/](https://www.youtube.com/) | Incorre ct  url | Write proper url and press enter  key | Successf ully open the [www.yo](http://www.youtube.com/) [utube.co](http://www.youtube.com/)  [m](http://www.youtube.com/) | Error | Test fail ed |  |
| cr\_ 01 | Create\_vi deo\_valid  \_01 | Upload the video | Upload the video with  .mp4 file | Properl y upload the  video file | It will show the successf ully  upload video | it will show the successf ully  upload video | Test pass ed |  |
| cr\_ 02 | Create\_vi deo\_invali d\_02 | Upload the video | Upload the file instead of .mp4  extensi on | Properl y upload the  video file | Error message invalid file format. | Error Message invalid file format. | Test pass ed |  |
| sb\_ 01 | Search\_bo x\_valid\_0 1 | Search box | Search the video as the name type in search box | Enter correct any thing in search box (Accord ing to  require d) | Successf ully Show the video of accordin g to requirem ent. | Successf ully Show the video of accordin g to requirem ent. | Test pass ed |  |
| sb\_ 02 | Search\_bo x\_invalid\_ 02 | Search box | Search the name | Enter the irreleva | Error it will give the | Error it will give the | Test pass es |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | (what we require  d) | nt name in search  box | irrelevan t video | irrelevan t video |  |  |
| hi\_ 01 | History\_v alid\_01 | Video stored in history | View video | Sign in with YouTu be and see the  video | Successf ully stored viewed video in  history | Successf ully stored viewed video in  history | Test pass ed |  |
| hi\_ 02 | History\_in valid\_02 | Video stored in history or not | View video without login | Without login view the video of YouTu  be | Error message sign in with You Tube | Error sign in with you Tube | Test pass ed |  |
| sb\_ 01 | Subscribe  \_btn\_valid  \_01 | Subscribe channels | Subscri be channel s with Sign in  You Tube | Click on subscri ption button | Successf ully subscript the channel | Successf ully Subscript the channel | Test pass ed |  |
| sb\_ 02 | Subscribe  \_btn\_inval id\_02 | Subscribe the channels | Subscri be channel s without Sign in  You Tube | Click on subscri ption button without sign in | Error Sign in with You Tube | Error sign in with You Tube. | Test pass ed |  |

* 1. [www.facebook.com](http://www.facebook.com/)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tes t\_c ase  \_no | Test\_case  \_no\_descr iption | Test Data | Input | Action | Expected Result | Actual Result | Stat us | Re mar k |
| url  \_01 | url\_01\_va lid | url [https://ww](https://www.facebook.com/) [w.faceboo](https://www.facebook.com/) [k.com/](https://www.facebook.com/) | Correct url | Write proper url and press enter key | Successf ully open the [www.fac](http://www.facebook.com/) [ebook.co](http://www.facebook.com/) [m](http://www.facebook.com/) | Successf ully open the [www.fac](http://www.facebook.com/) [ebook.co](http://www.facebook.com/) [m](http://www.facebook.com/) | Test pass es |  |
| url  \_02 | url\_02\_in valid | url [https://ww](https://www.facebook.com/) [w.faceboo](https://www.facebook.com/) [k.com/](https://www.facebook.com/) | Incorre ct  url | Write proper url and press enter  key | Successf ully open the [www.fca](http://www.fcaebook.com/) [ebook.co](http://www.fcaebook.com/)  [m](http://www.fcaebook.com/) | Error | Test fail ed |  |
| cr\_ 01 | login\_vali d\_01 | User name and password | Correct User id and passwor d | Write proper user id and passwor d and press  the next | Successf ully Open home of user facebook page | Successf ully open the home page of user facebook | Test pass ed |  |
| cr\_ 02 | login\_inva lid\_02 | User name and password | Correct User id and wrong passwor d | Write proper user id and wrong passwor d and  press the next | Error | Error, please enter the correct password | Test pass ed |  |
| sb\_ 01 | Search\_bo x\_valid\_0 1 | Search box | Search the any person as the name type in search box | Enter correct any person name search box | Successf ully Show the list names of person as we search  for. | Successf ully Show list of persons name. | Test pass ed |  |

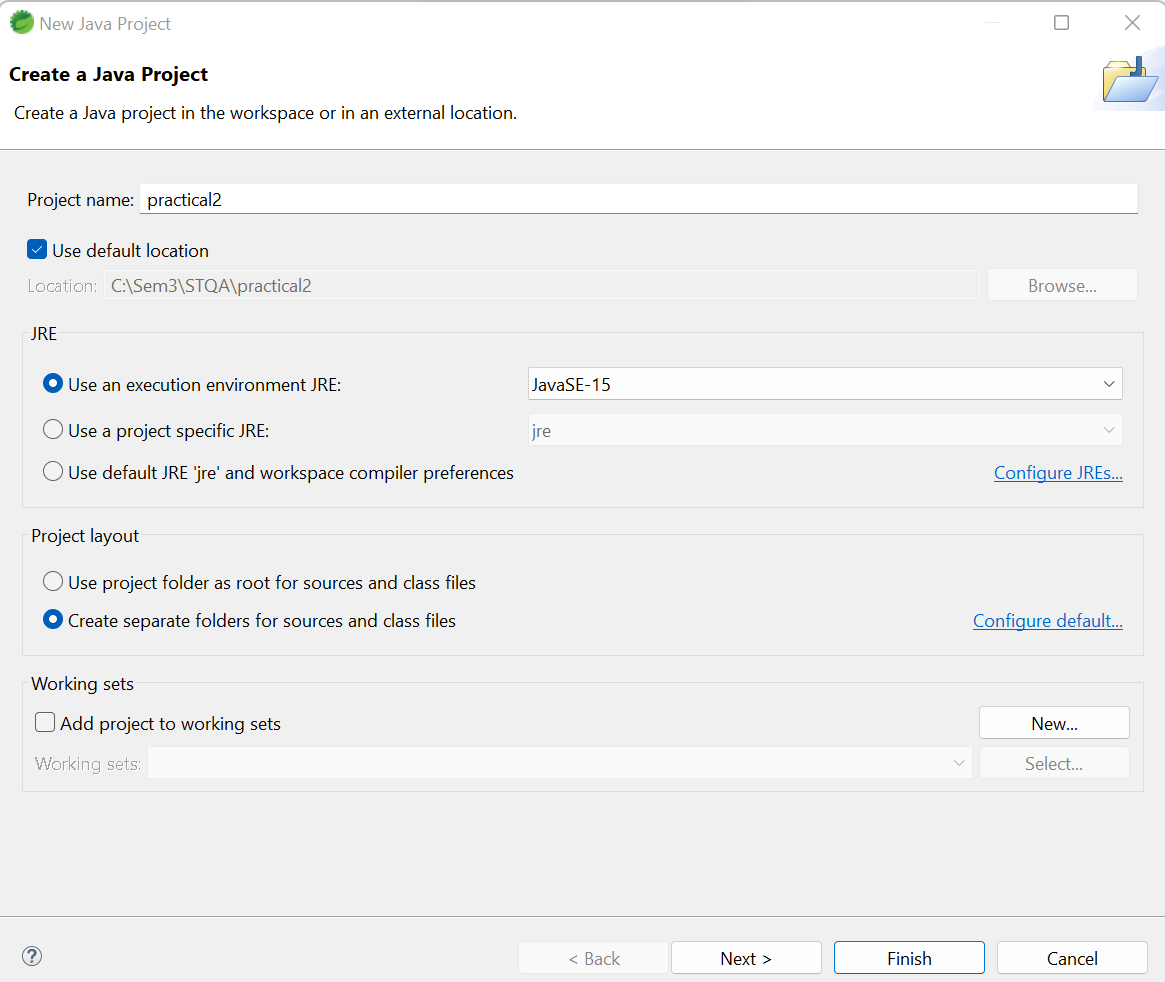
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| sb\_ 02 | Search\_bo x\_invalid\_ 02 | Search box | Search the any person as the name type in search box | Enter in- correct any person name search box | It will show the irrelevan t persons list. | Can’t find actual person | Test pass ed |  |
| sf\_ 01 | Suggested  \_frds\_inva lid\_01 | Suggested friends/ may you know the people | Correct phone number | Enter the correct phone number  . | Successf ully show the people who are  in phone book list | Successf ully show the people who are  in phone book | Test pass ed |  |
| sg\_ 02 | Suggested  \_frds\_inva lid\_01 | Suggested friends/ may you know the people | Not entering phone no. or incorrec t phone number | Enter the incorrec t 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 pass ed |  |
| sm  \_01 | Send\_mes sage\_valid  \_01 | message | Able to send the messag e to  friends or not | Install messen ger and send messag  e to friends | Successf ully send the message to friends. | Successf ully Send the message to the friends. | Test pass ed |  |
| sm  \_02 | Send\_mes sage\_inval id\_02 | message | Able to send the messag e to friends or not | Not Install messen ger and send messag e to  friends | Error install messeng er | Error install messeng er  . | Test pass ed |  |

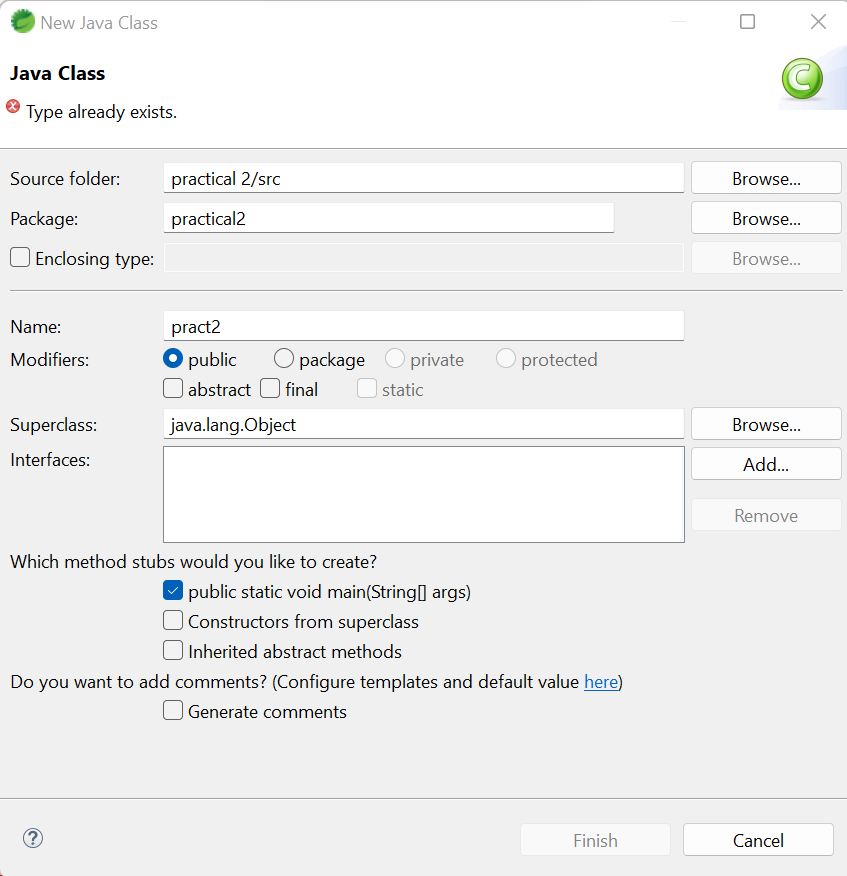
# Practical No: 2

**Aim: Implement Web Drivers on Chrome & FireFox browsers. Description:**

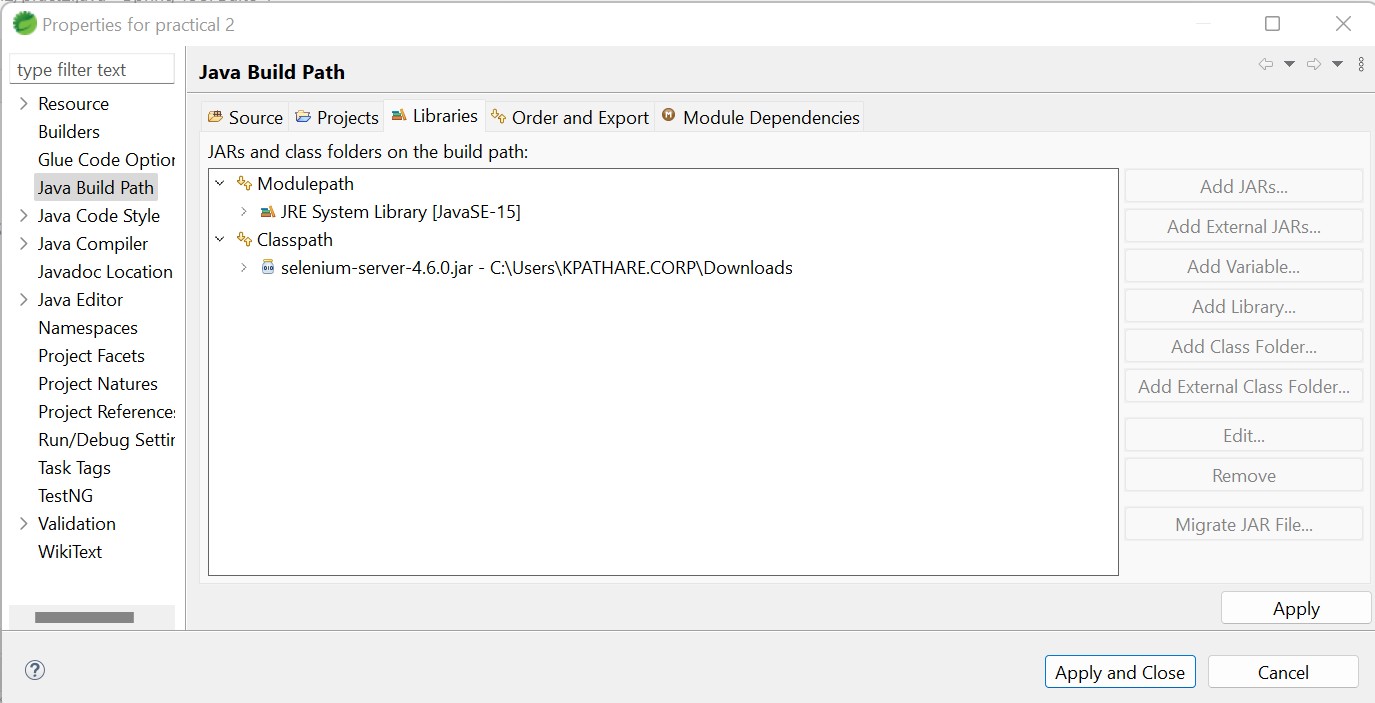
1. drivers(requied to perform the cross browsing)
   1. chrome driver
   2. geko driver
2. we required selenium standalone server used for synchronation of various tools, dependency
3. Ecalips OR STS (required to write testcases which will run in java)
4. jdk 1.7 onwards

Create a new java project



Step2: - create a new java class

Modifier should be public it going on web Brower Step3:- Add external jar.



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/](http://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-

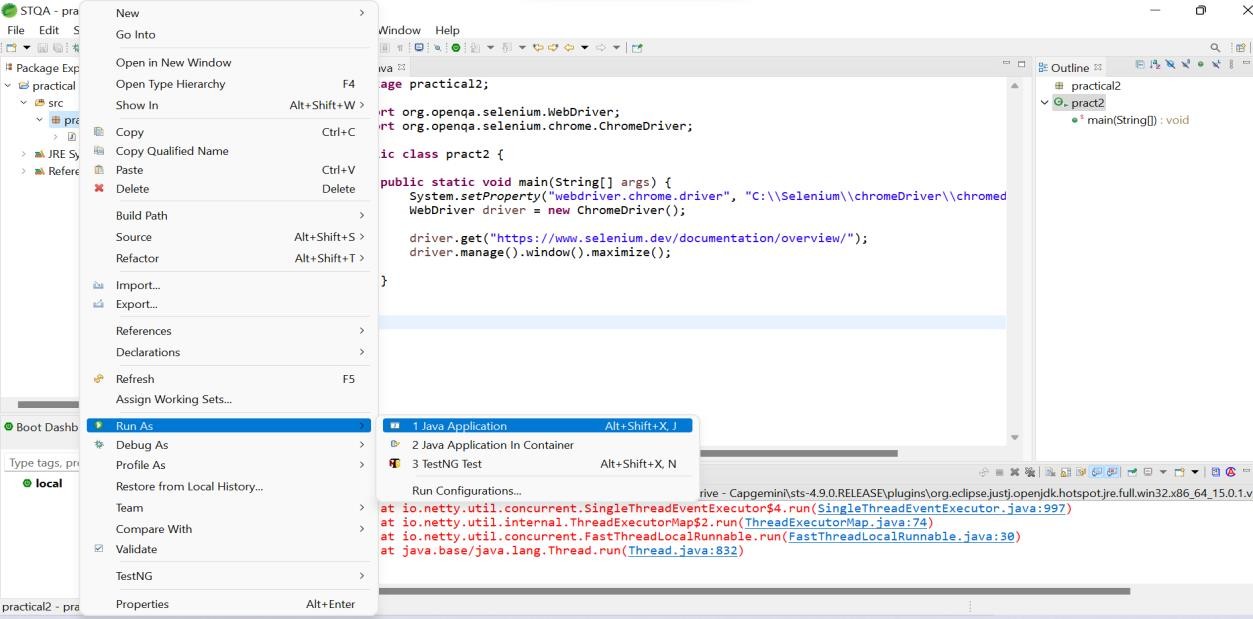
win64\\geckodriver.exe");

WebDriver driver=**new** FirefoxDriver(); driver.get("https://[www.selenium.dev/documentation/overview/](http://www.selenium.dev/documentation/overview/)"); driver.manage().window().maximize();

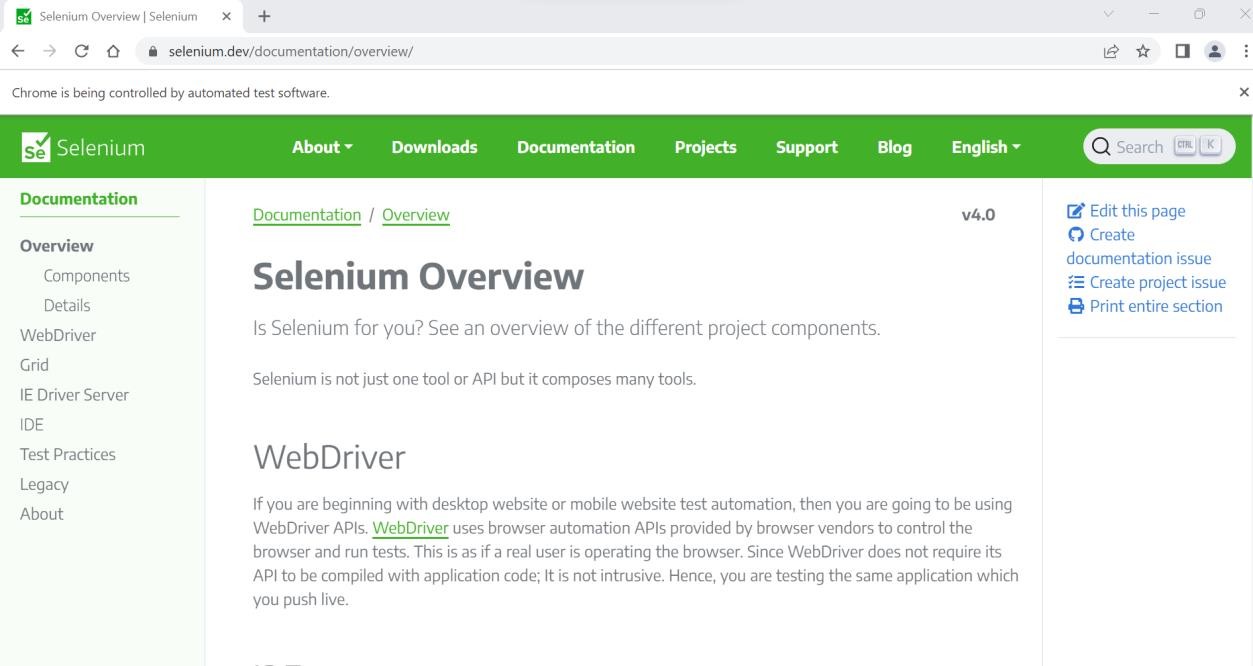
}

}

Step5: run as java application



OUTPUT:-



.

# 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",

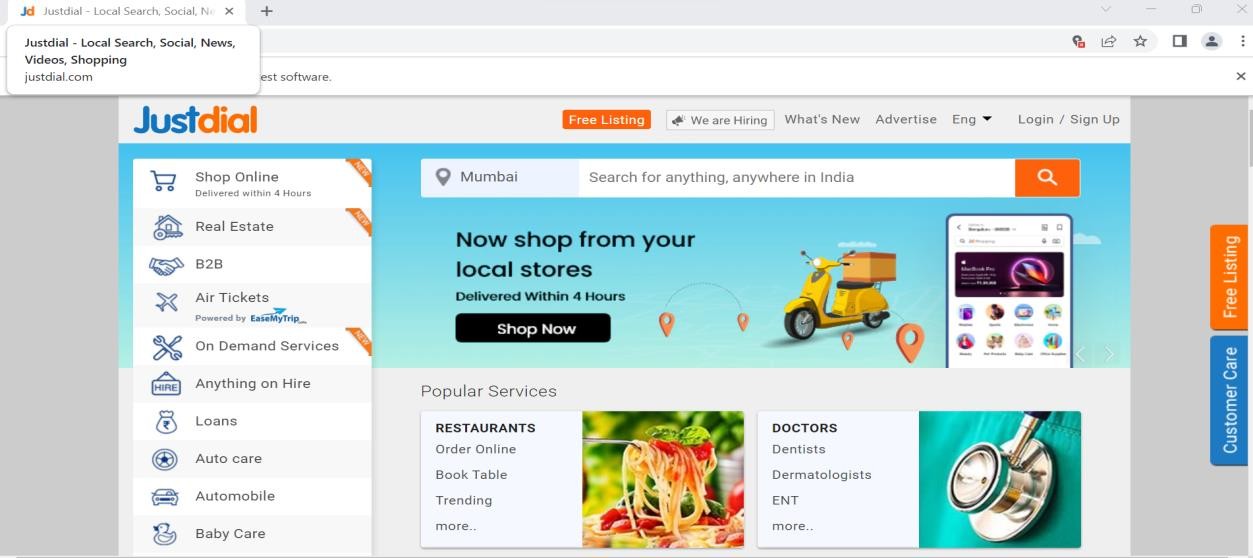
"C:\\Selenium\\chromeDriver\\chromedriver107.exe"); WebDriver driver = **new** ChromeDriver(); driver.get("https://[www.justdial.com/](http://www.justdial.com/)"); driver.manage().window().maximize(); driver.switchTo().frame("GTM-5CQCS9");

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",

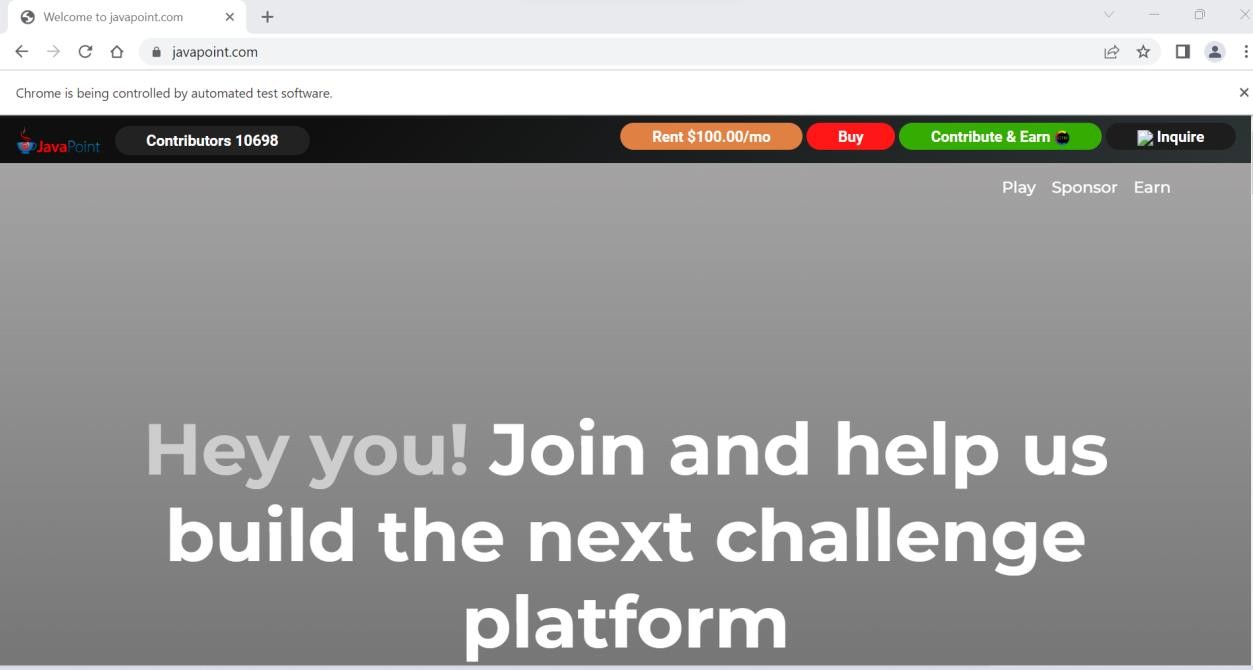
"C:\\Selenium\\chromeDriver\\chromedriver107.exe"); WebDriver driver = **new** ChromeDriver();

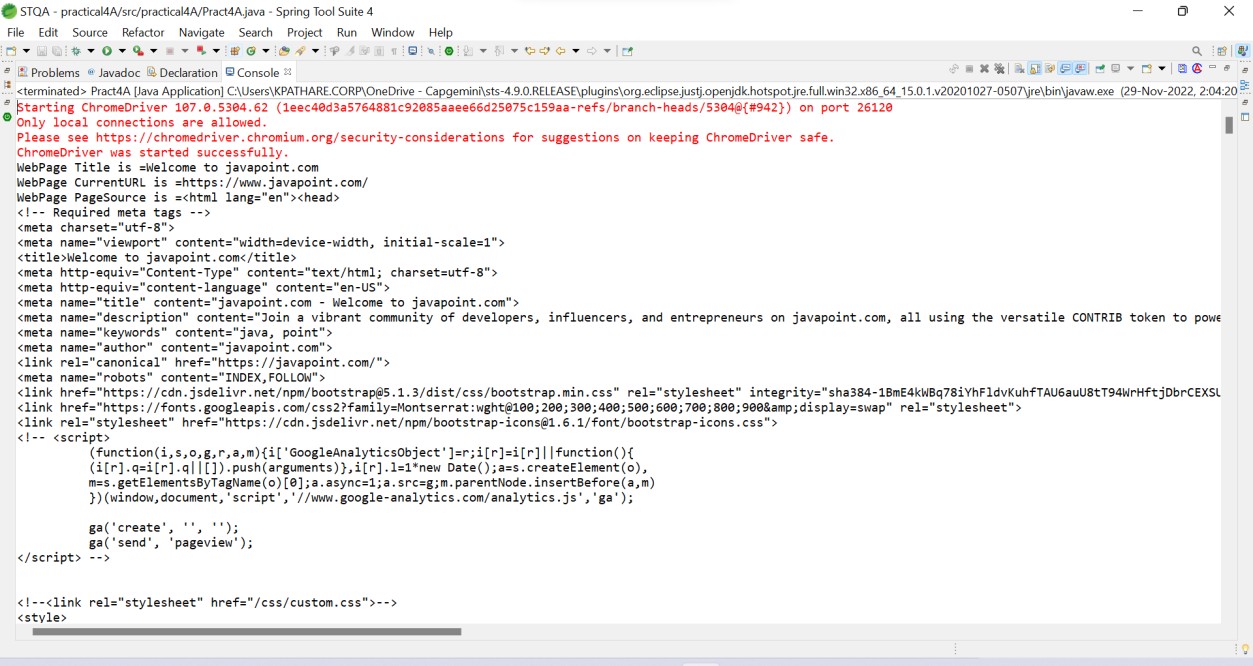
driver.manage().window().maximize(); driver.get("https://[www.javapoint.com/](http://www.javapoint.com/)"); String Title = driver.getTitle(); System.***out***.println("WebPage Title is ="+Title); String CurrentURL=driver.getCurrentUrl();

System.***out***.println("WebPage CurrentURL is ="+CurrentURL); String getPageSource =driver.getPageSource(); System.***out***.println("WebPage PageSource is =" +getPageSource);

}

}





**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/](http://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();

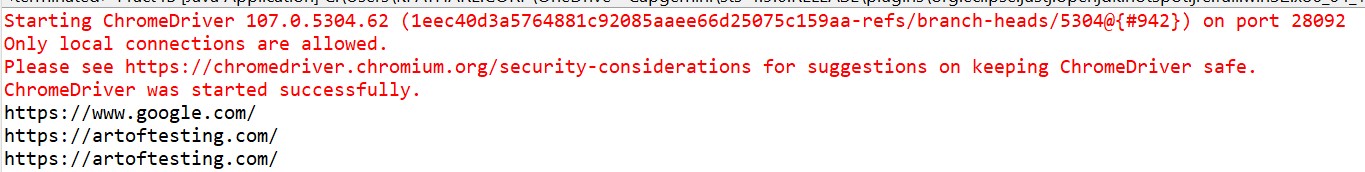
String str3 = driver.getCurrentUrl(); System.***out***.println(str3);

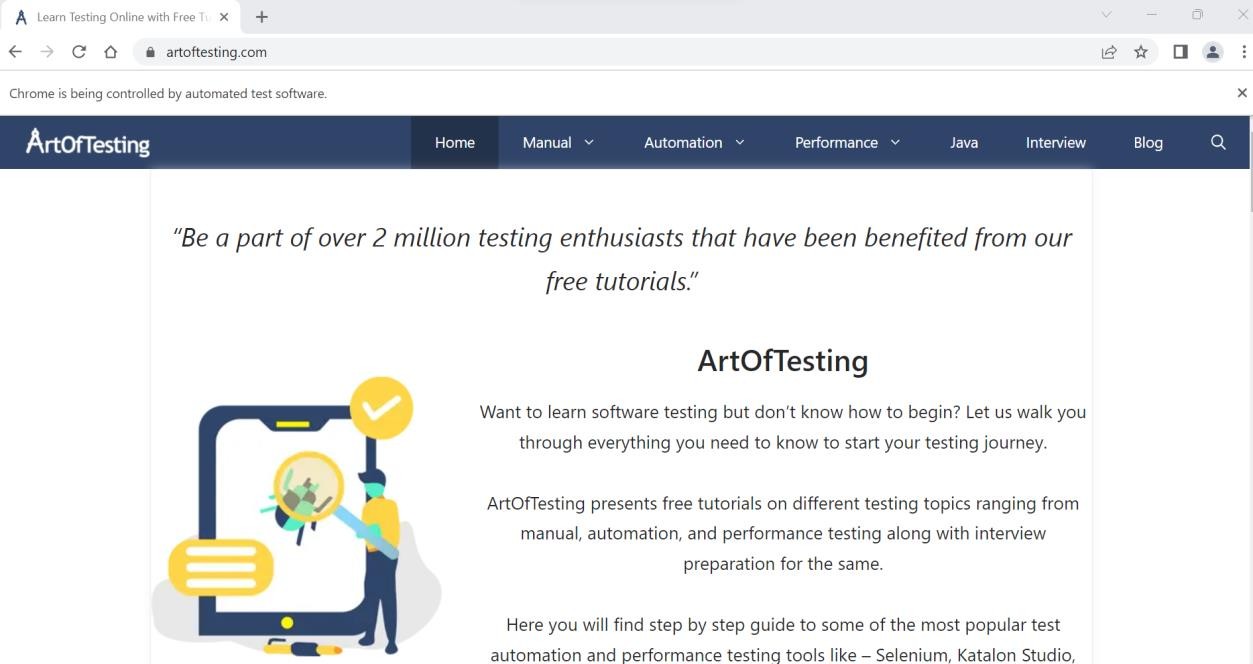
driver.manage().timeouts().~~implicitlyWait~~(5000, TimeUnit.***SECONDS***); driver.quit();

}

}

**Output:**





# 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](http://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);

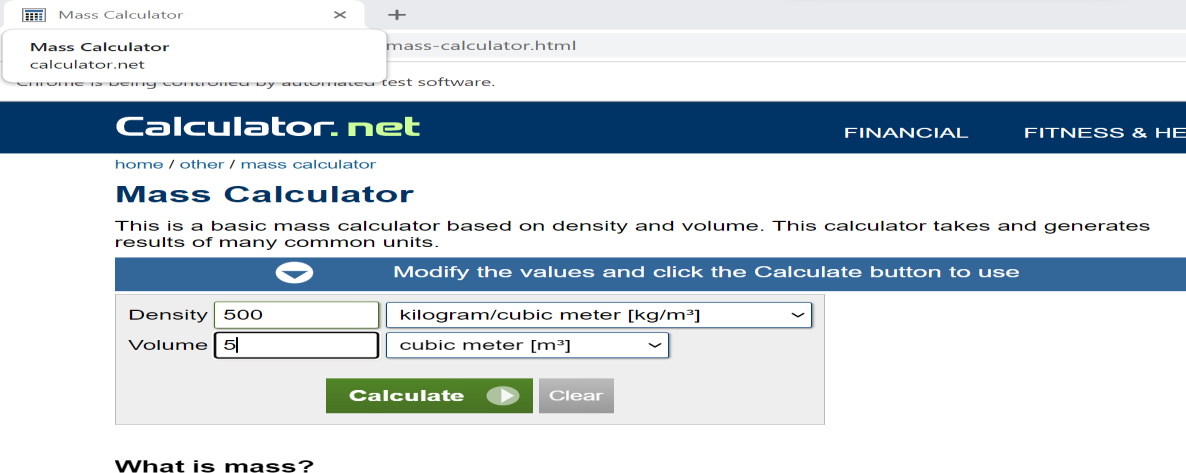
//by xpath driver.findElement(By.*xpath*("//tbody/tr[3]/td[1]/img[1]")).click(); Thread.*sleep*(5000);

driver.quit();

}

}

Output:



# 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](http://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);

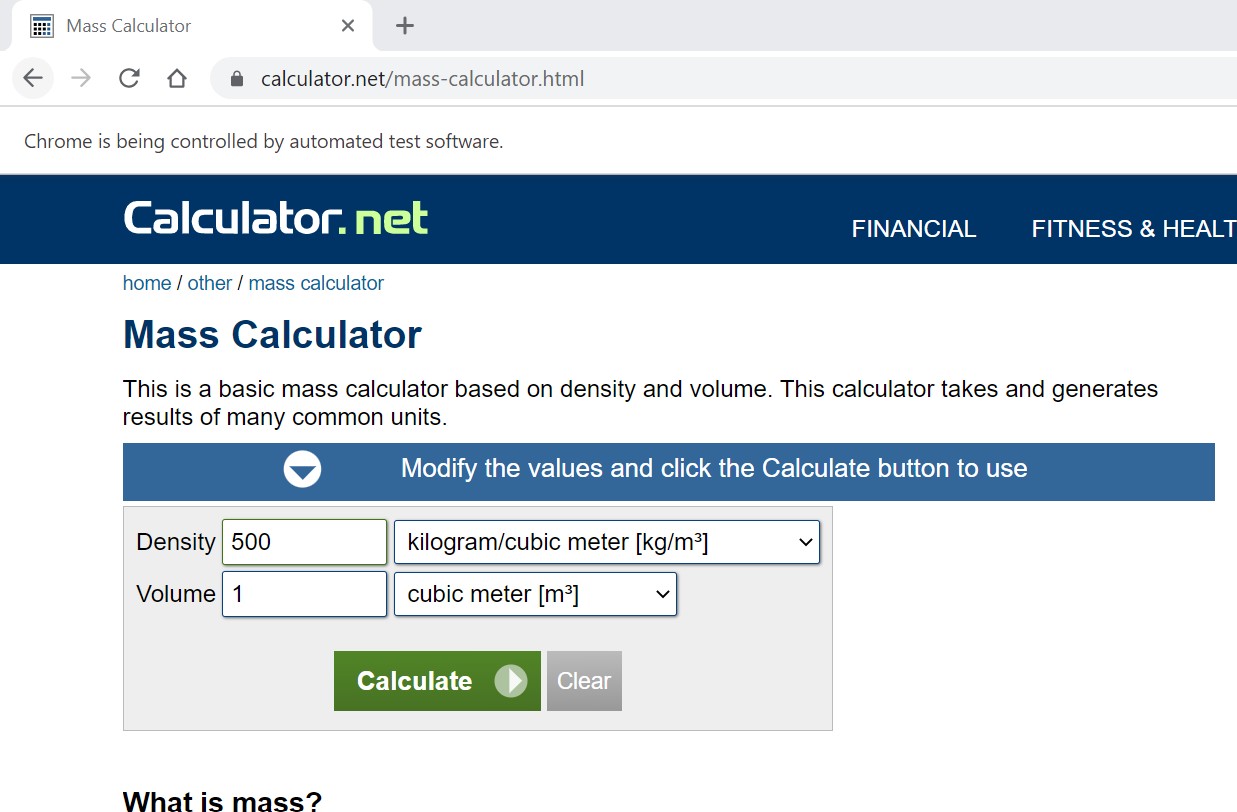
//by xpath driver.findElement(By.*xpath*("//tbody/tr[3]/td[1]/img[1]")).click(); Thread.*sleep*(5000);

driver.quit();

}

}

Output:



# 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\\chro medriver107.exe");

WebDriver driver = **new** ChromeDriver(); driver.get("https://opensource-demo.orangehrmlive.com");

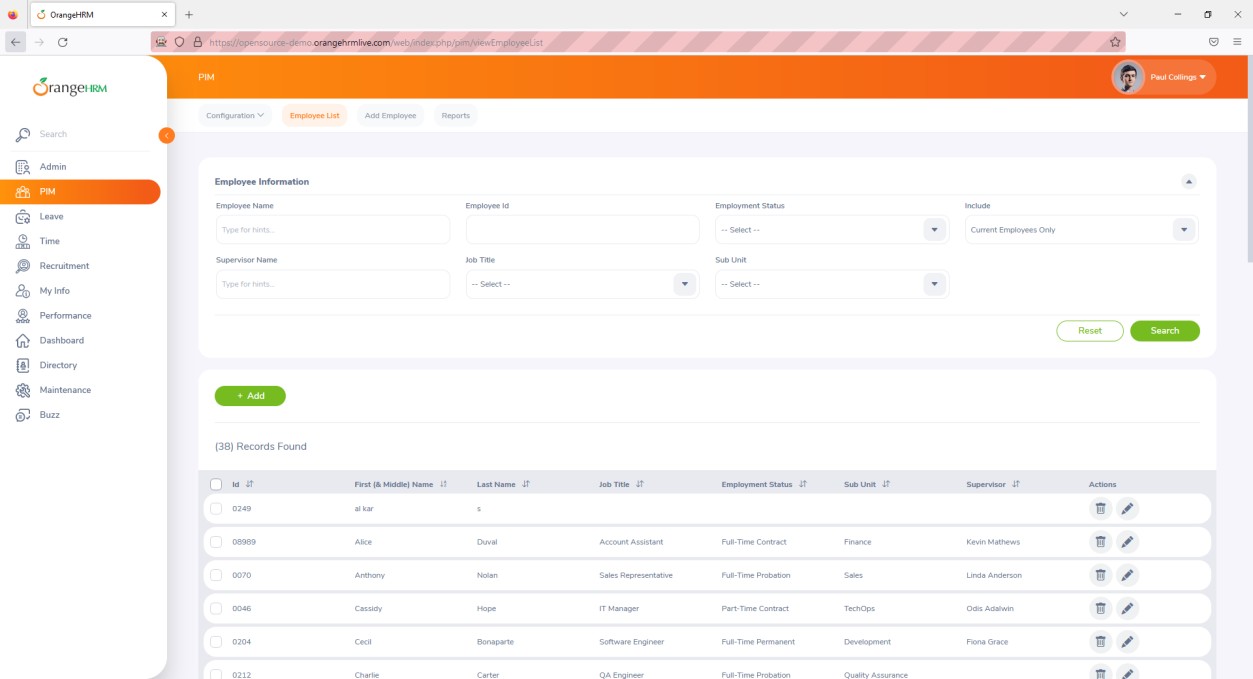
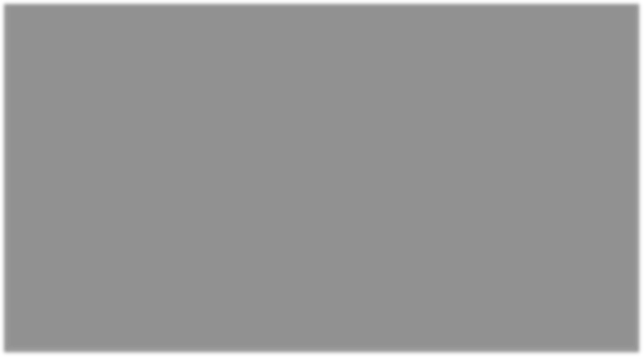
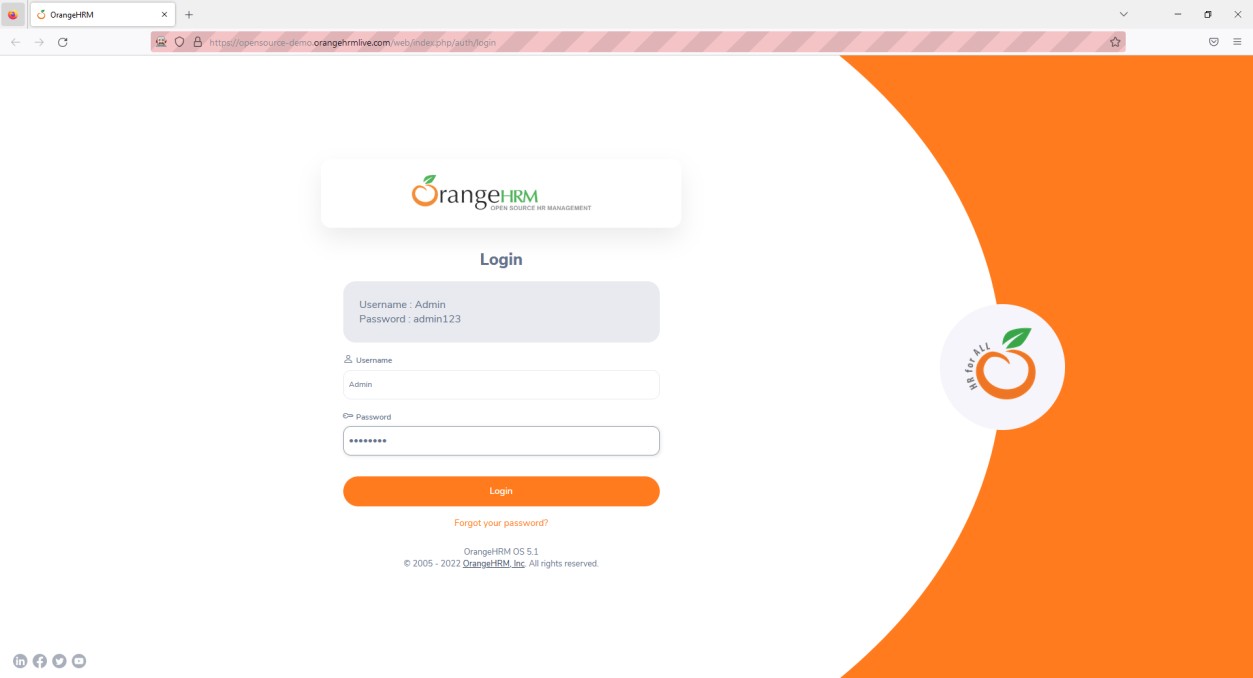
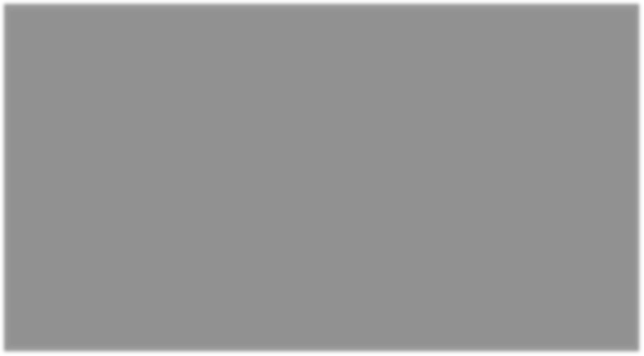
driver.findElement(By.*id*("txtusername")).sendKeys("admin"); driver.findElement(By.*id*("txtusername")).sendKeys("admin123"); driver.findElement(By.*id*("button")).click();

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

}

}

Output:



1. 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\\chromedriv er107.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("Log out"))); 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();

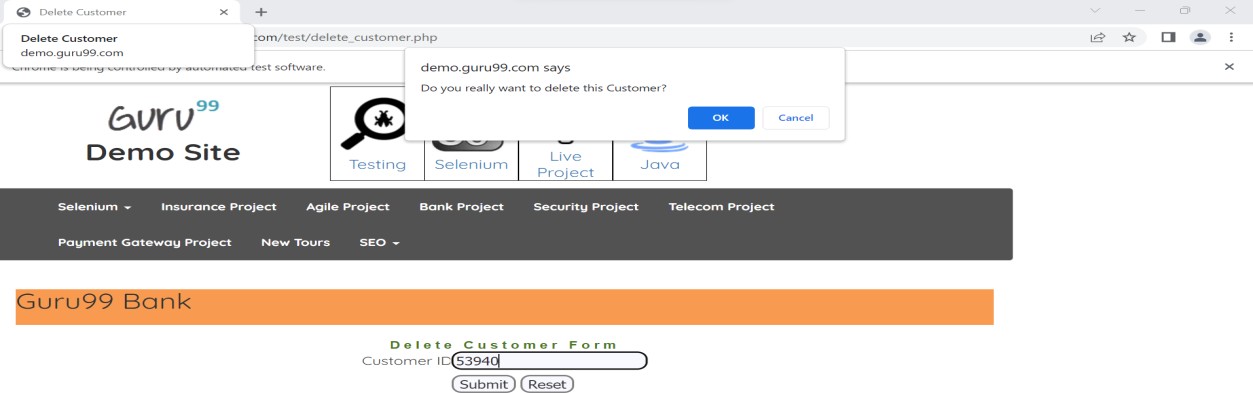
driver.get("<http://demo.guru99.com/test/delete_customer.php>"); driver.findElement(By.*name*("cusid")).sendKeys("53940"); driver.findElement(By.*name*("submit")).submit(); driver.switchTo().alert();

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/>");

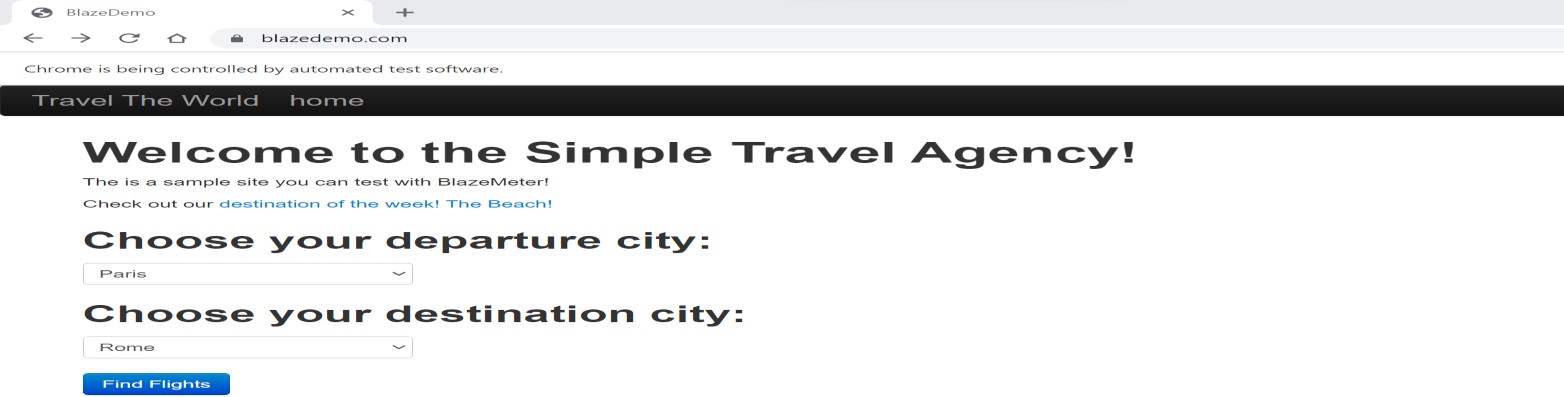
Select f = **new** Select(driver.findElement(By.*name*("fromPort"))); Select t = **new** Select(driver.findElement(By.*name*("toPort"))); f.selectByVisibleText("Paris");

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\\chromedriv er107.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");

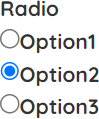
//Radio Button1 is de-selected and Radio Button2 is selected radio2.click();

System.***out***.println("Radio Button Option 2 Selected");

}

}

Output:



# 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 Crete 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/tbod 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 Descripion:**

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.

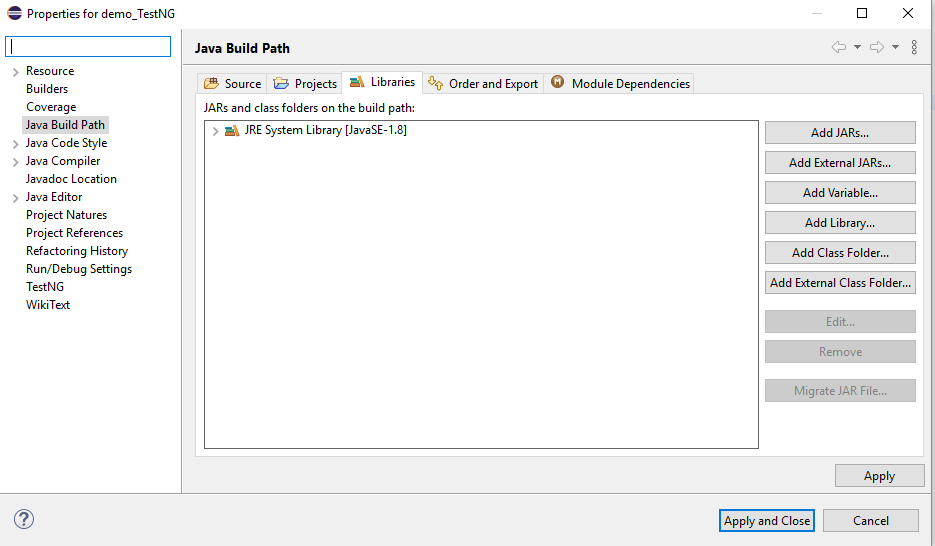
Test ng is frame work or different type of plugging provided by selenium While doing practical select add library and select **JUnit**

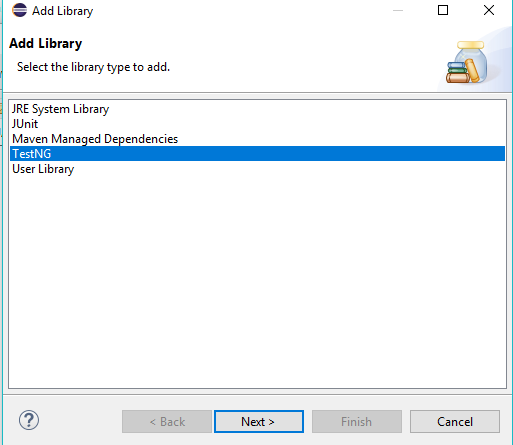
**Selenium code -> TestNG -> XLS Files -> Logs (maintain records up to test the cases)-> Reports generated in html**

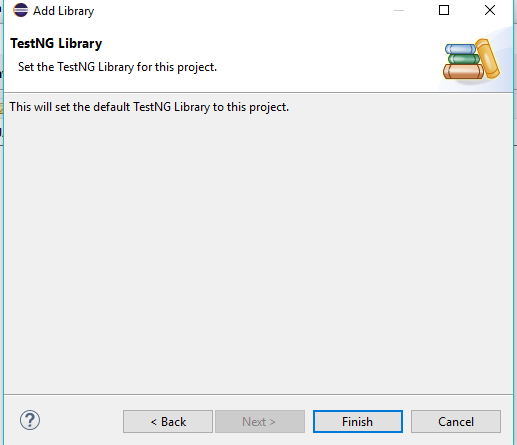
**Setps for install TestNg**

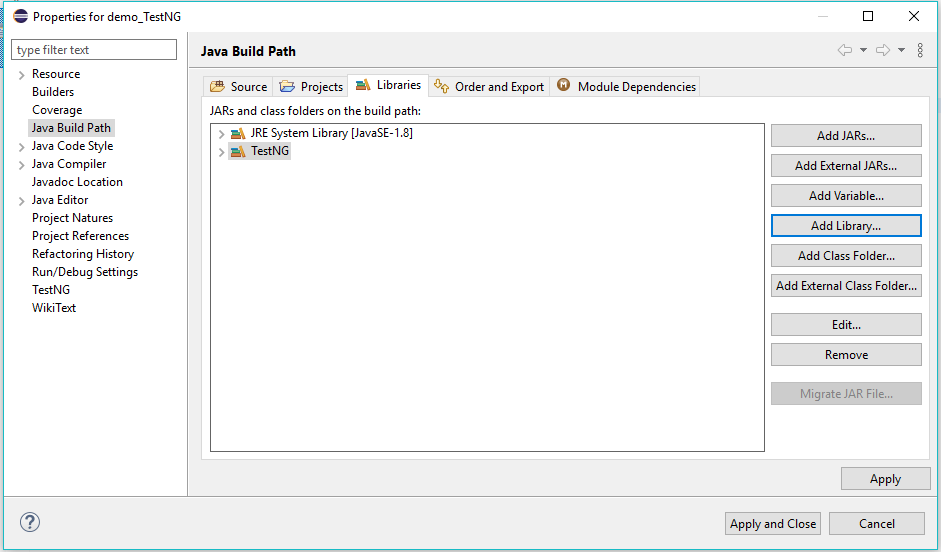
* 1. **Open eclipse**
  2. **Go to help and select eclipse marketplace**
  3. **Search TestNG and install it 4)**

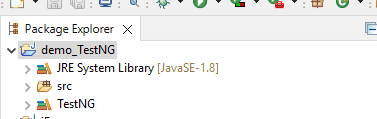
**Add libabry**











## Practical No : 13

**AIM:** Demonstrate data driven Framework.

### Theory:

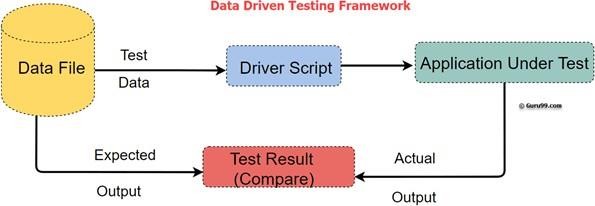
*Data Driven Testing*

**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](http://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++)

{

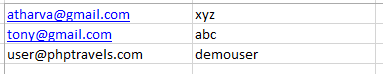
signin\_credentials[i][0] = configuration.getData(0, i, 0); signin\_credentials[i][1] = configuration.getData(0, i, 1);

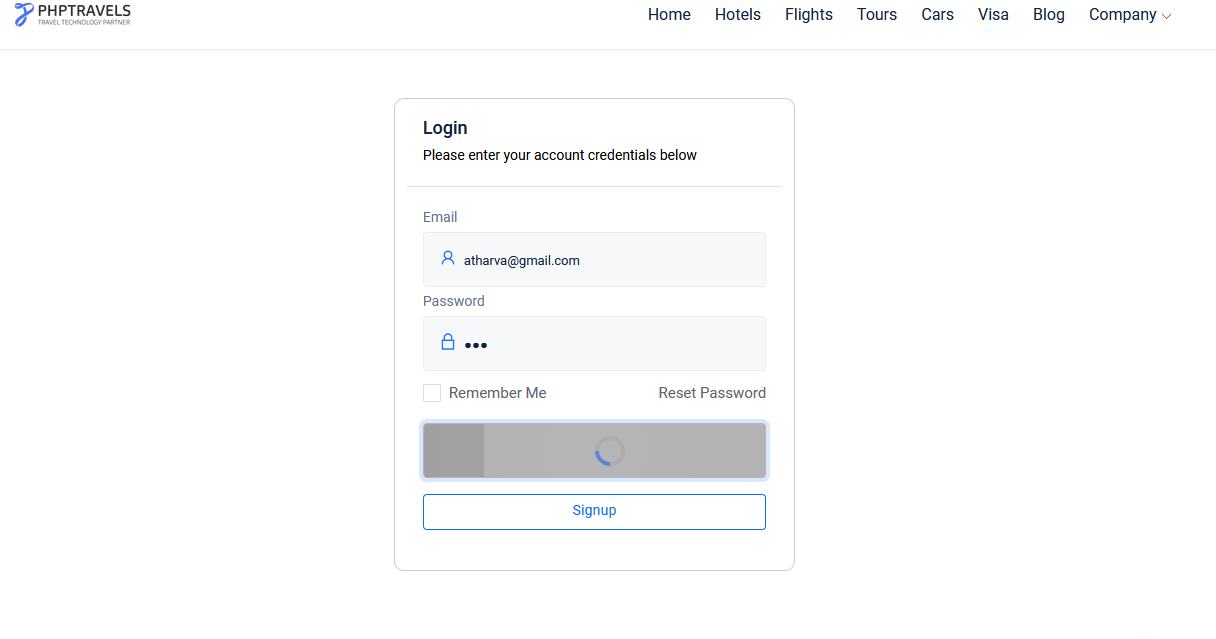
}

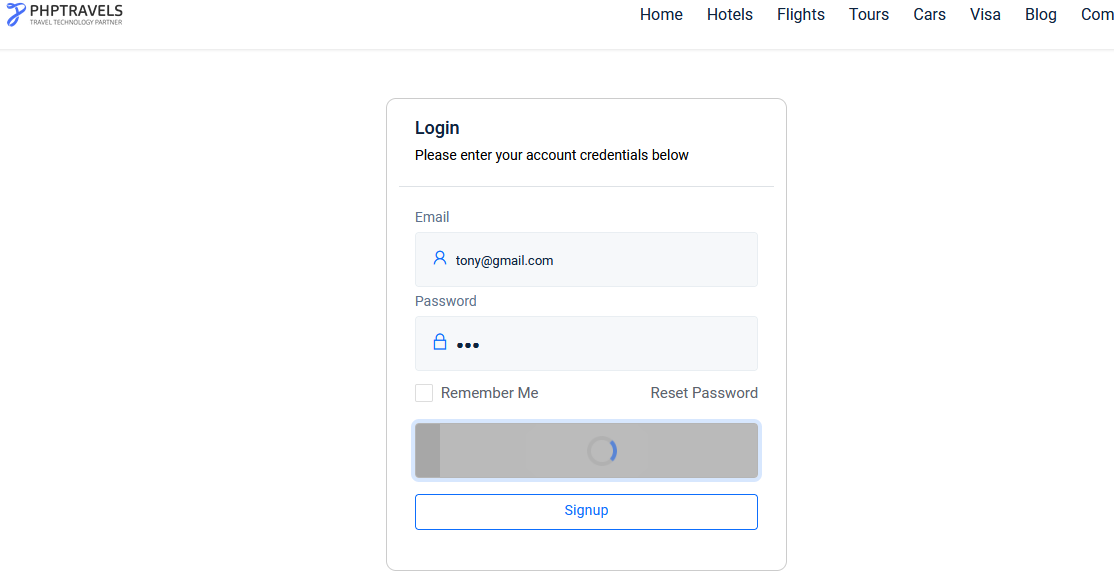
return signin\_credentials;

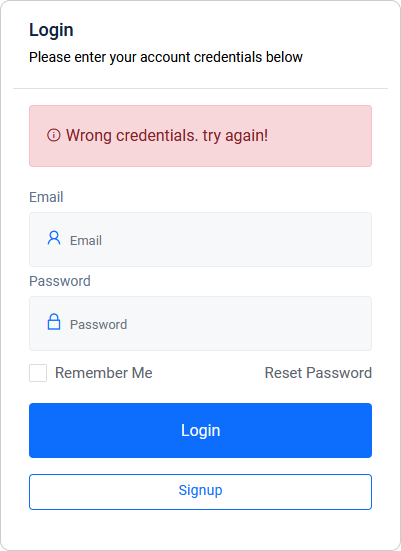
}

*Output:*

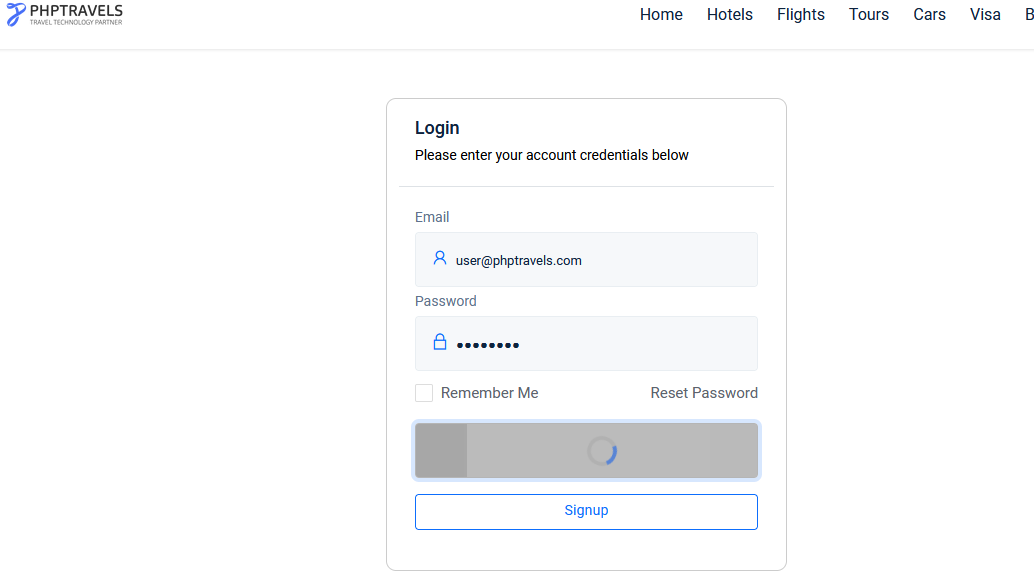
Data in Excel sheet:

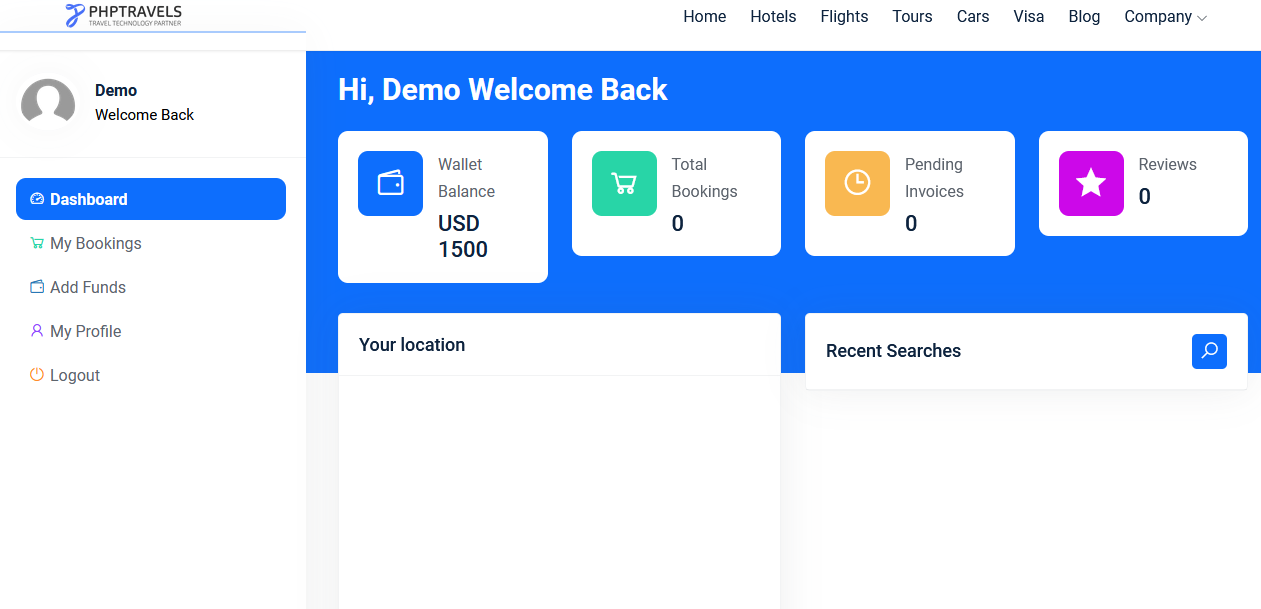
Wrong credentials



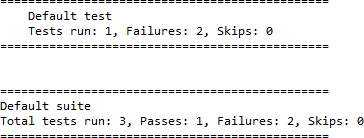


Correct credentials





Console OP





## Practical No : 14

**AIM** : Asserts and Verify methods are commonly used in [Selenium](https://www.browserstack.com/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 assertAll() |

**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](https://www.browserstack.com/guide/selenium-webdriver-tutorial) 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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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.br](http://www.browserstack.com/)o[wserstack.com/")](http://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](https://www.browserstack.com/guide/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](https://www.browserstack.com/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](https://www.browserstack.com/guide/selenium-webdriver-tutorial), [Selenium](https://www.browserstack.com/guide/what-is-selenium-ide) [IDE](https://www.browserstack.com/guide/what-is-selenium-ide), [Selenium RC](https://www.browserstack.com/guide/selenium-rc-tutorial), and [Selenium Grid](https://www.browserstack.com/guide/selenium-grid-tutorial). 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](https://www.browserstack.com/guide/selenium-with-java-for-automated-test), [Python](https://www.browserstack.com/guide/python-selenium-to-run-web-automation-test), [Javascript](https://www.browserstack.com/guide/automation-using-selenium-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.

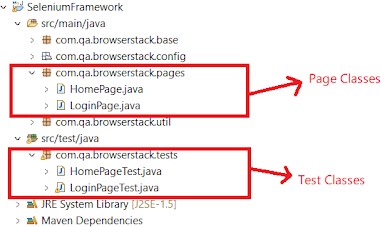
You can develop the regression suite for automation and keep adding new test scripts/test cases as and when required. Selenium Framework contains many reusable modules/functions that make it easy to maintain the existing code or add any new code.

You can integrate Selenium with TDD frameworks like TestNG, Junit Maven, etc. [TestNG annotations](https://www.browserstack.com/guide/testng-annotations-in-selenium) help in writing automation scripts effectively. You can also use the [Page Object Model](https://www.browserstack.com/guide/page-object-model-in-selenium-python) 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()

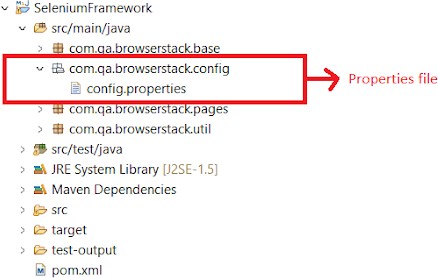
{

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



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.