Practical -1

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

Description:

1)Review

2)Test Cases

3)Known Applications:

www.google.com
www.youtube.com
www.facebook.com

4)Code

1.www.google.com

Test_case_ no	Test_case_ desciption	Test_ casedata	Input	Action	Excepted_Result	Actual_ Rsult	Status	Rema rk
url_A	url_A_valid	URL www.google .com	Correc t url	Write Correct url and press Enter key	Successfully opened www.google .com	Success fully opened www.go ogle.co m	Test case passe d	
url_B	url_B_Invalid	URL www.google .com	Incorr ect url	Write Correct url and press Enter key	Successfully opened www.google .com	Error	Test case Failed	
Search_A	Search_A_valid	Search bar	Click on search bar	Click on search bar	Can type on search bar successfully	Can type on search bar successf ully	Test case passe d	
Search_B	Search_B_Invali d	Search bar	search bar is not clickab le	Click on search bar	Can type on search bar successfully	Test case passed	Test case Failed	
Button_A	Button_A_Valid	Search button	Click on search button	Click on search button	Successfully click on search button	Success fully click on search button	Test case passe d	

2.www.youtube.com

Test_case_ no	Test_case_desc iption	Test_casedata	Input	Action	Excepted_ Result	Actual_ Result	Status

Homepage_A	**Homepage_A_valid**	**Homepage of www.youtub.com**	**Enter YouTube url correctly in any of the supported browser**	**Go to YouTube home page**	**YouTube homepage open successfully**	**YouTube homepage open successfully**	**Test case Passed**
Homepage_B	**Homepage_B_Invalid**	**Homepage of www.youtub.com**	**Incorrect url**	**Write Correct url and press Enter key**	**YouTube homepage open successfully**	**Error**	**Test case Failed**
Change_setting_A	**Change_setting_A_ Valid**	**Change setting**	**Click able setting button**	**Click on setting after signing in**	**Successfully able to change setting**	**Successfully change setting**	**Test passed**
Change_setting_B	**Change_setting_B_ Valid**	**Change setting**	**Click able setting button**	**Click on setting without signing in**	**Successfully able to change setting**	**Error**	**Test Failed**
History	**Check the History_valid**	**Check History**	**Click able History button**	**Click on History without signing in**	**Successfully able to check history**	**Error**	**Test Failed**

3.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.fcaebook.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 we search**	**Successfully Show list of persons name.**	**Test passed**	

Practical 2

Aim:-Implement Web Drivers on Chrome & Firefox Browser

To perform STQA Practicals we required Driver

1.Eclipse

2.Chrome Drivers

3.JAKO(Firefox Driver)

4.Download Selenium Standalone jar file

Practcial 2.1

Step1:- Open Eclipse and create new Java Project and give the project(our jdk should be from 1.7)

Step2:- Select src , right click on src and create package and in that package create a class.

Step3:-import the jar file 'selenium server standalone-3.141.59'

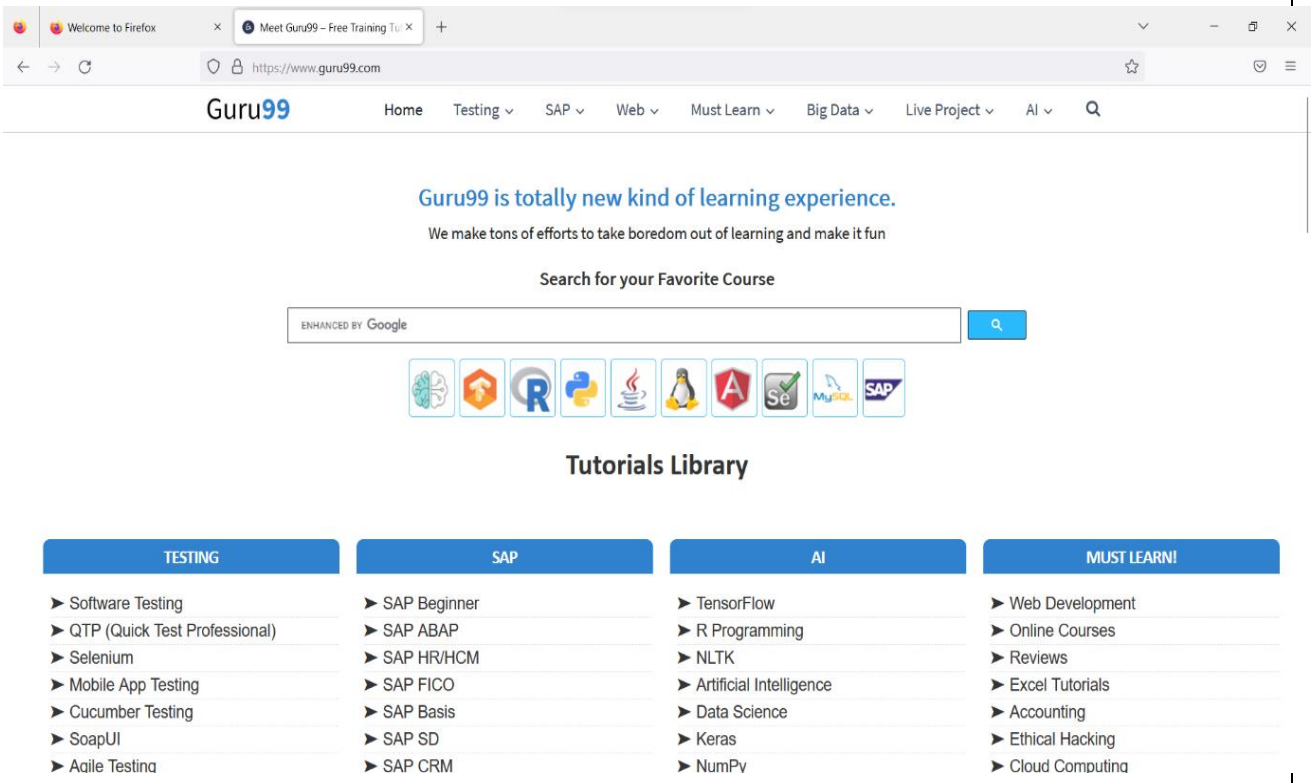
Step4:- Write a code

```
package FireFoxDriver;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class firefoxdriver {

	public static void main(String[] args) {
		System.setProperty("webdriver.gecko.driver","D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new  FirefoxDriver();
		driver.get("https://www.guru99.com/");
		driver.manage().window().maximize();

	}

}
```

Output:-
Run the java file

```
1666238975436   geckodriver     INFO    Listening on 127.0.0.1:4405
1666238976148   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla F
1666238976542   Marionette      INFO    Marionette enabled
1666238976549   Marionette      INFO    Listening on port 7178
1666238976739   RemoteAgent     WARN    TLS certificate errors will be ignored for this sessio
console.warn: SearchSettings: "get: No settings file exists, new profile?" (new NotFoundError(
Oct 20, 2022 9:39:38 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
console.error: Region.jsm: "Error fetching region" (new Error("TIMEOUT", "resource://gre/modul
console.error: Region.jsm: "Failed to fetch region" (new Error("TIMEOUT", "resource://gre/modu
```

It will directly open the firefox and redirect to the url:- "https://www.guru99.com/"



Practcial 2.1

Step1:- Open Eclipse and create new Java Project and give the project(our jdk should be from 1.7)

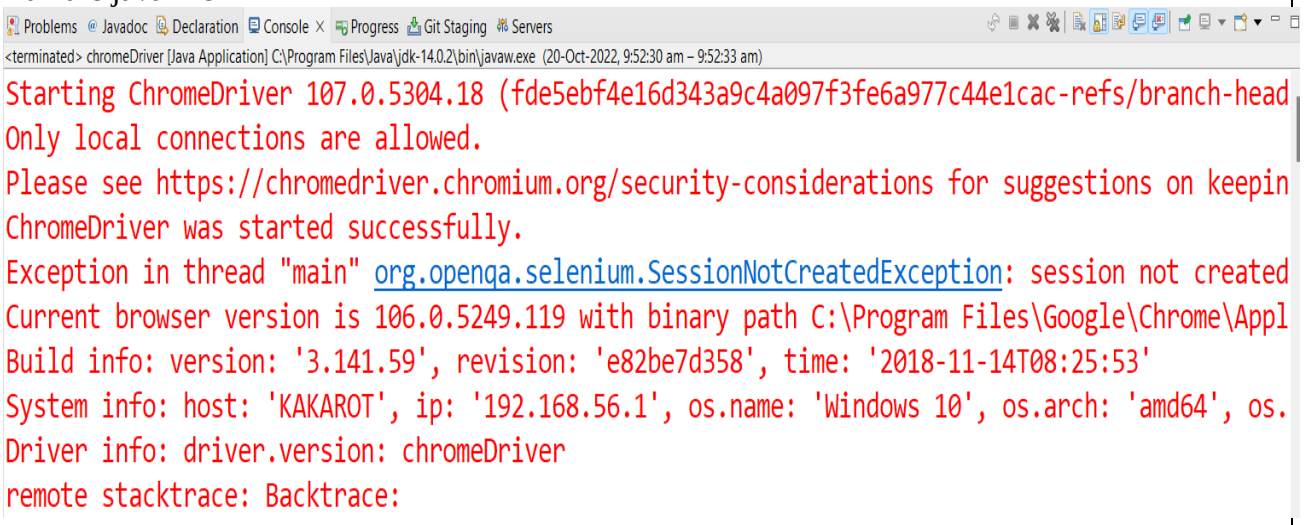
Step2:- Select src , right click on src and create package and in that package create a class.

Step3:-import the jar file ‘selenium server standalone-3.141.59’

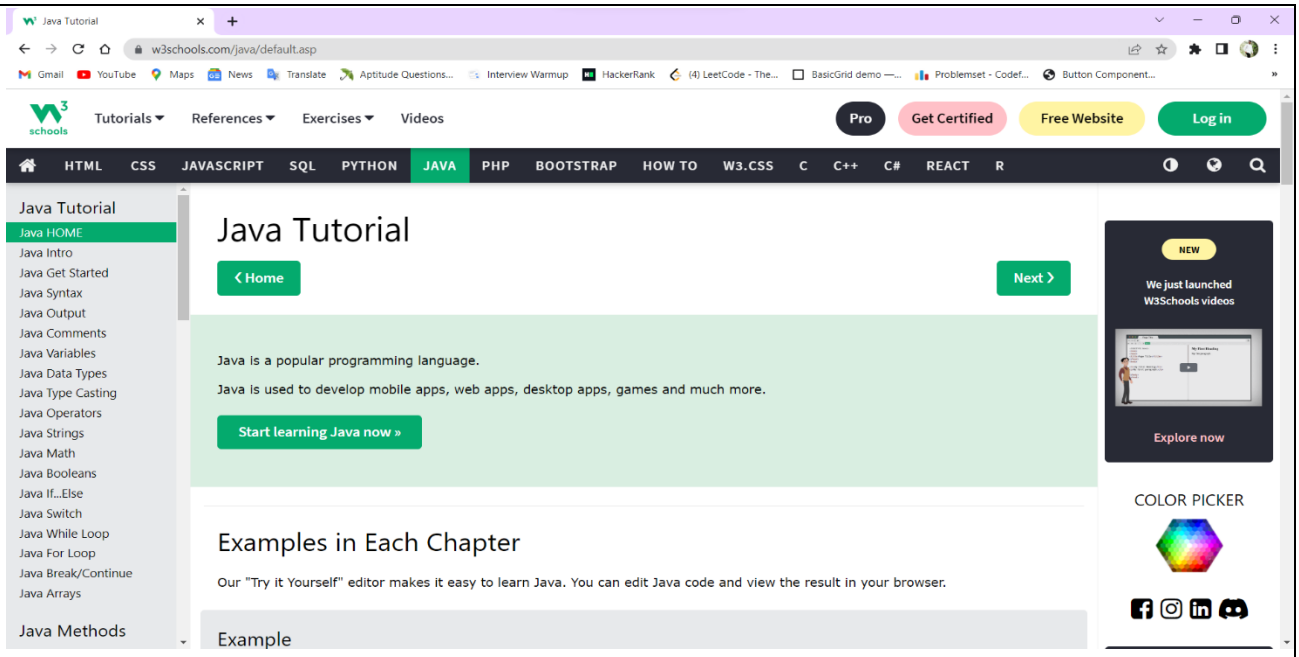
Step4:-Write code:-

```
package ChromeDriver;

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


public class chromeDriver {

	public static void main(String[] args) {

		System.setProperty("webdriver.chrome.driver","D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\chromedriver_win32(1)\\chromedriver.exe");
		WebDriver driver = new  ChromeDriver();
		driver.get("https://www.w3schools.com/java/default.asp");
		driver.manage().window().maximize();
	}

}
```

Output:-
Run the java file



It will directly open the chrome and redirect to the url:-
“https://www.w3schools.com/java/default.asp”


Java Tutorial
w3schools.com/java/default.asp
Tutorials
References
Exercises
Videos
Pro
Get Certified
Free Website
Log in
HTML CSS JAVASCRIPT SQL PYTHON JAVA PHP BOOTSTRAP HOW TO W3.CSS C C++ C# REACT R
Java Tutorial
Java HOME
Java Intro
Java Get Started
Java Syntax
Java Output
Java Comments
Java Variables
Java Data Types
Java Type Casting
Java Operators
Java Strings
Java Math
Java Booleans
Java If...Else
Java Switch
Java While Loop
Java For Loop
Java Break/Continue
Java Arrays
Java Methods
Java Tutorial
Home
Next
Java is a popular programming language.
Java is used to develop mobile apps, web apps, desktop apps, games and much more.
Start learning Java now »
Examples in Each Chapter
Our "Try it Yourself" editor makes it easy to learn Java. You can edit Java code and view the result in your browser.
Example
NEW
We just launched W3Schools videos
Explore now
COLOR PICKER

Practical 3

Aim:-Demonstrate Handling multiple frames in selenium.

iFrame in Selenium Webdriver is a web page of an inline frame which is embedded in another web page or an HTML document embedded inside another HTML document.

1)How to identify the iFrame

Method 1:- Right click on a frame if you get an option of “This frame ” in panel box then that is iFrame.

Method 2:-Right click on the page and click “View page source ” and search the “iFrame”.

Practical 3.1

Step 1:-Create java project

Step 2:- Create a package in src folder

Step 3:- Create a class name of SwitchToFrameByID.java

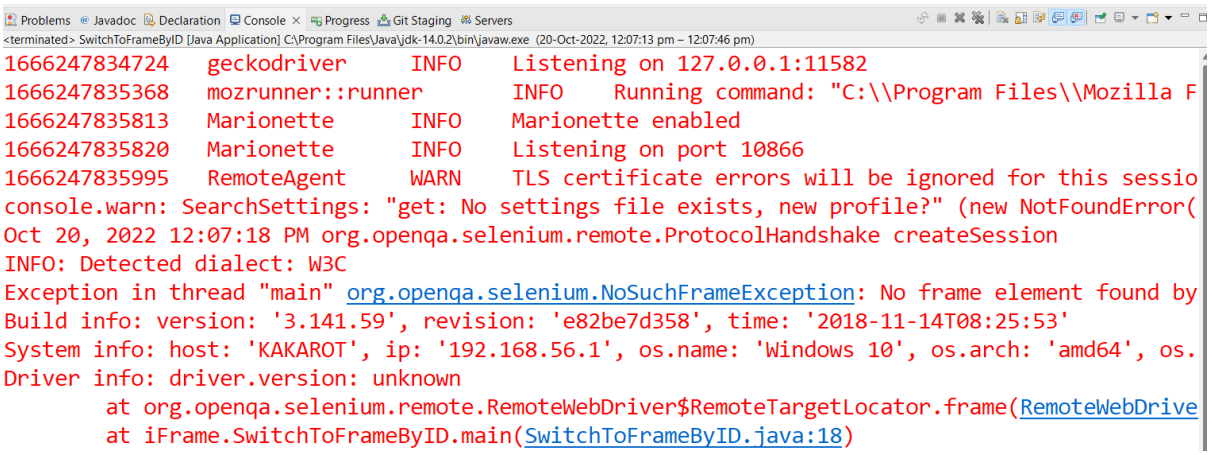
Step4:-write a code

```
package iFrame;

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

public class SwitchToFrameByID {

	public static void main(String[] args)
	{
		//step 1
		System.setProperty("webdriver.gecko.driver","D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser
		driver.get("https://www.w3schools.com/css/default.asp");//navigates to the page consisting an iframe
		driver.manage().window().maximize();

		//step 2
		driver.switchTo().frame("GTM-KTCFC3S"); //switching the frame by ID
		System.out.println("********We are switch to iframe********");
		driver.findElement(By.xpath("html/body/a/img")).click(); //Clicks the iframe

		System.out.println("********We are done********");
```

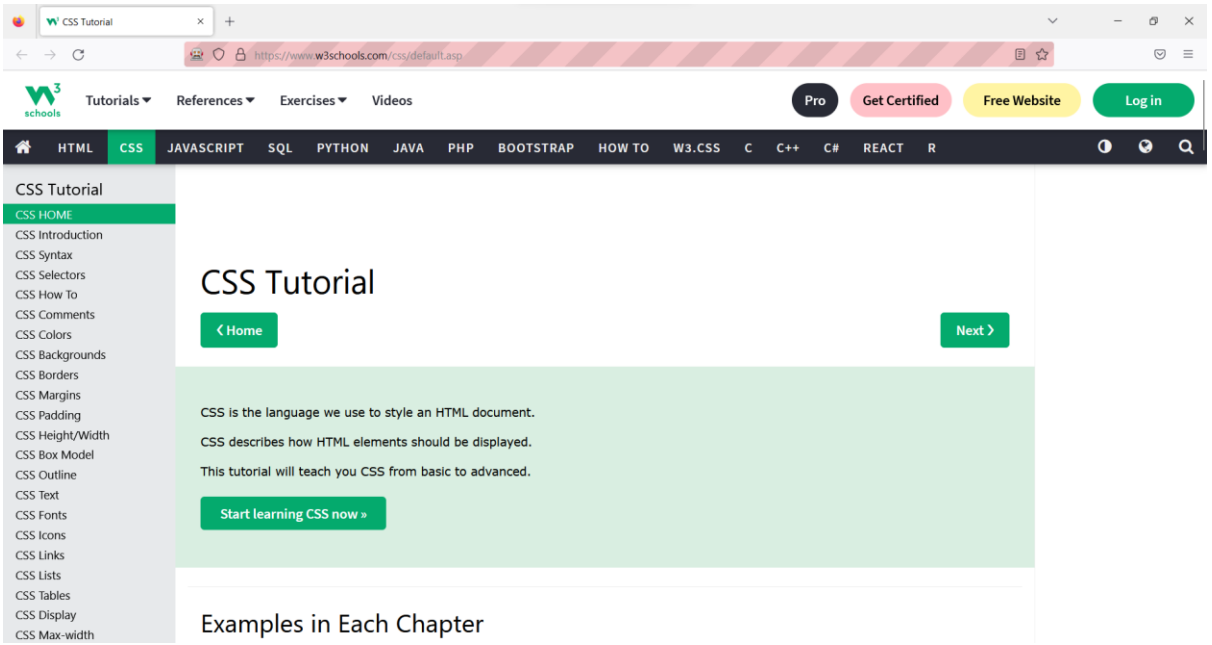
```
        }

}
```

Output:-

Run the java file



After Running the java file and open the firefox and it will redirectly open the url you have given.



Practical 3.2

Step 1:-Create java project

Step 2:- Create a package in src folder

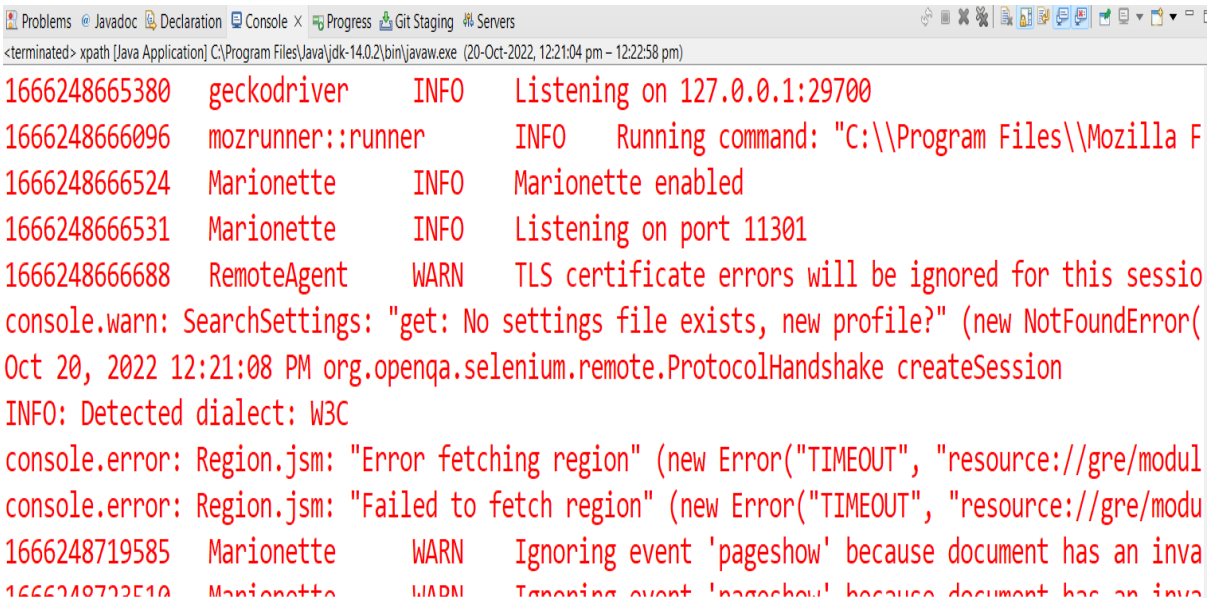
Step 3:- Create a class name of xpath.java

Step 4:- Write the code of java xpath:-

```
package iFrame;

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

public class xpath {

	public static void main(String[] args) {
		//step 1
		System.setProperty("webdriver.gecko.driver","D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser
		driver.get("https://www.w3schools.com/css/default.asp");//navigates to the page consisting an iframe
		driver.manage().window().maximize();

		driver.get("https://www.calculator.net/rent-calculator.html");
		driver.findElement(By.xpath("//input@name='monthlydebt'"));
	}

}
```

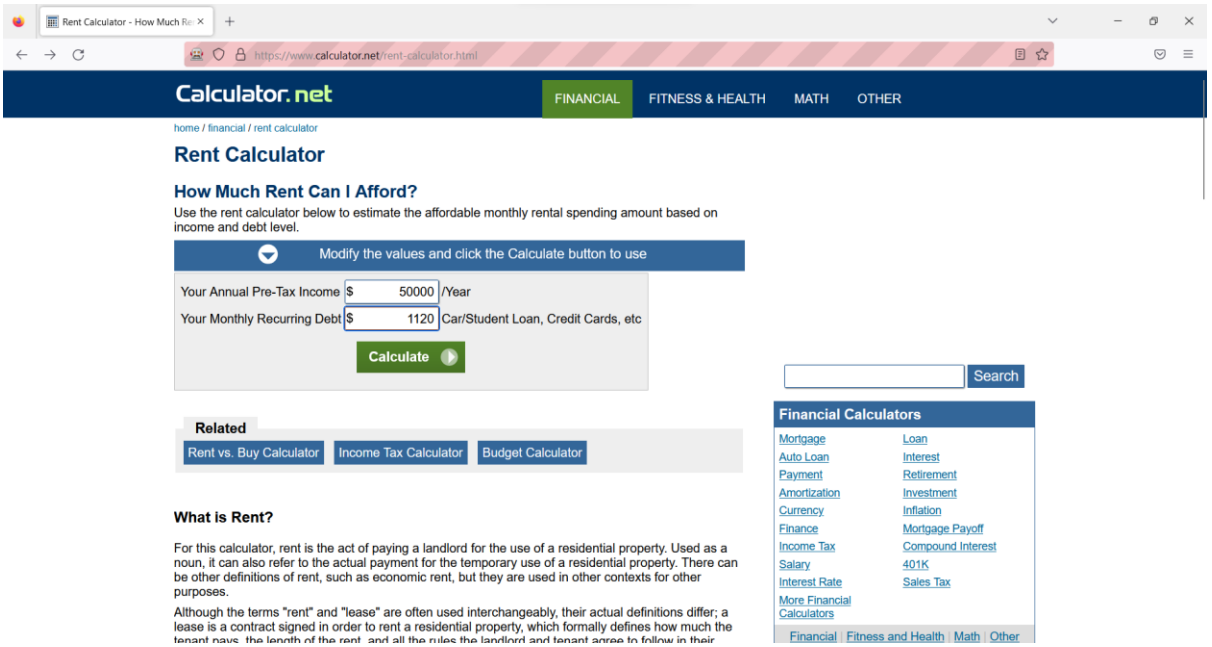
Output:-

Run the xpath.java file

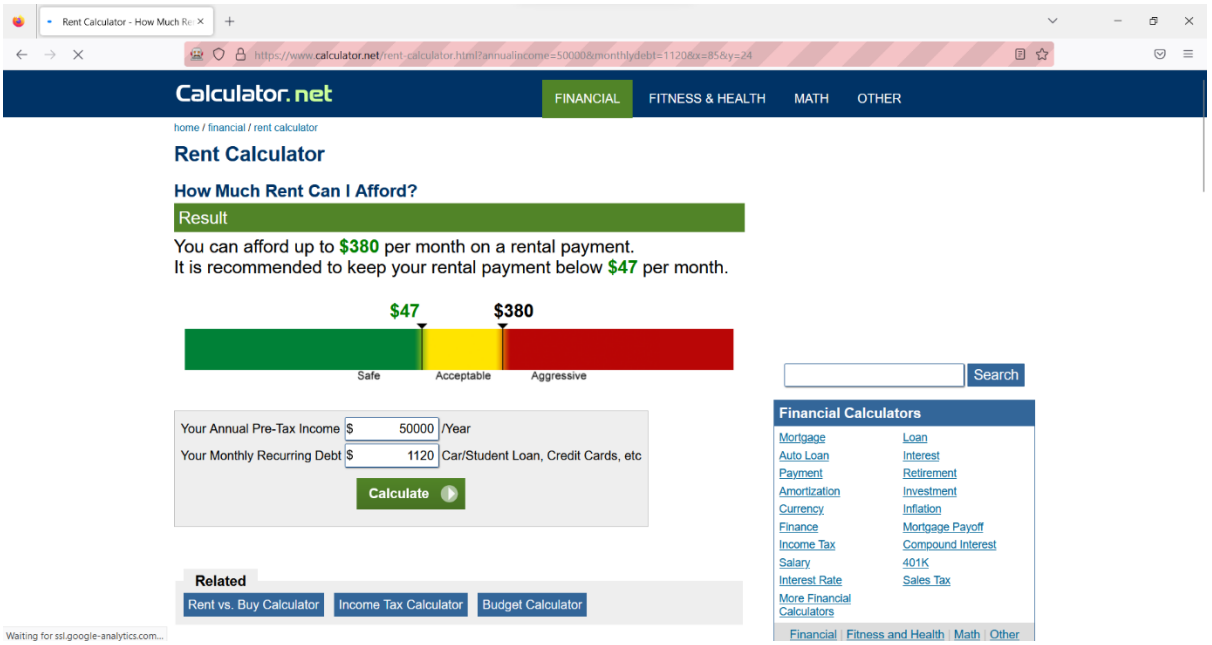


It will open the firefox and redirect to the link you have given

And check the website is working or not by calculating the rent



If the website is calculating properly then it will give the answer to the next page.



Practical 4

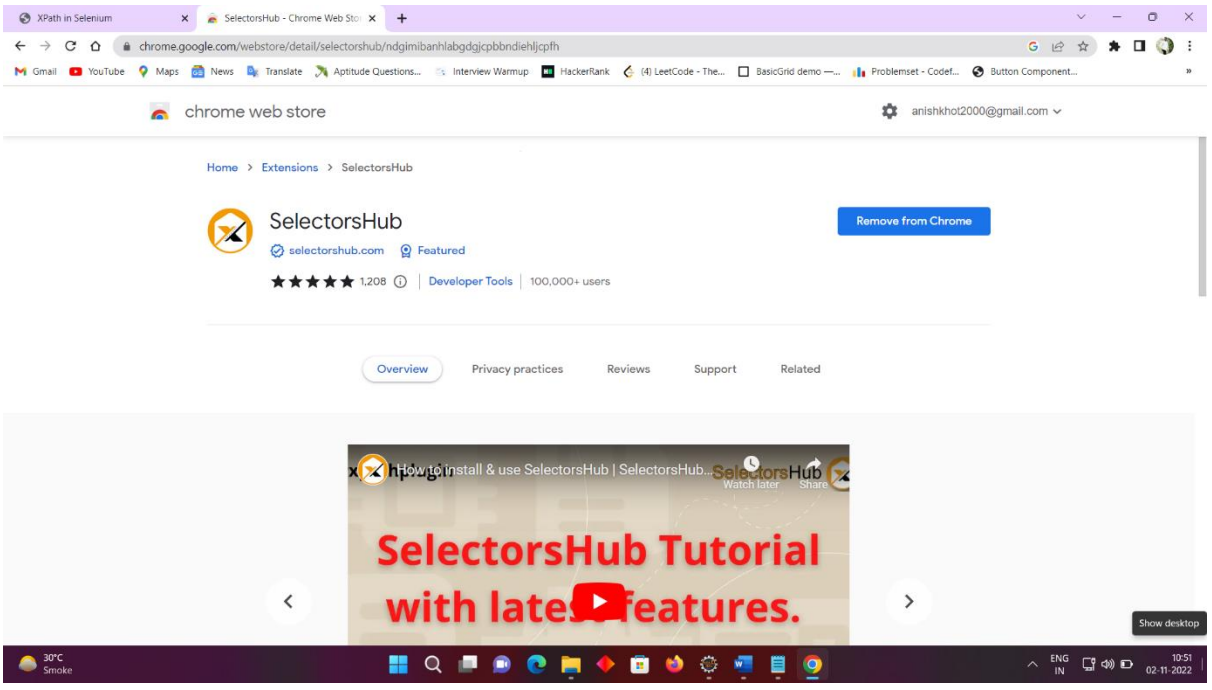
Aim:- Implement Browser command and navigation Commands.

Syntax of Xpath:-

Xpath=//tagname[@Attribute=”value”]

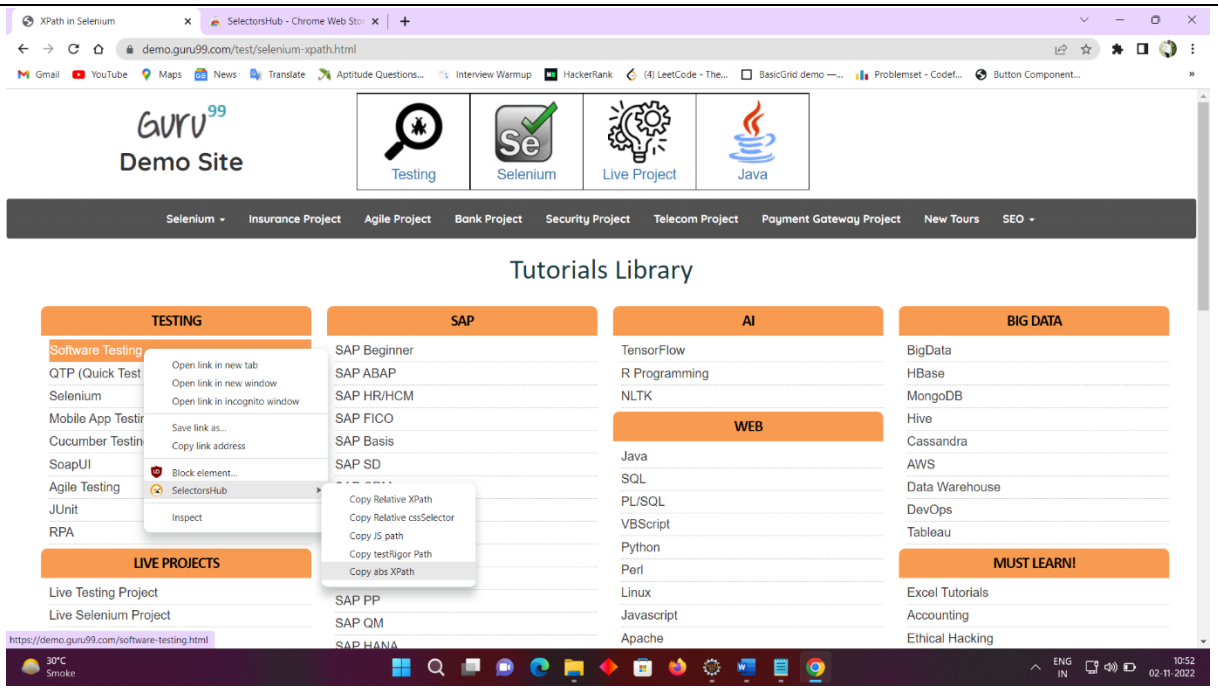
In this practical we have to add extension in chrome of

SelectorHub



Now got the website

“https://demo.guru99.com/test/selenium-xpath.html”



Now right click on any element select SelectorHub and select Copy as Xpath we will get below format path

/html[1]/body[1]/div[4]/div[1]/div[1]/ul[1]/li[4]/a[1]

a)

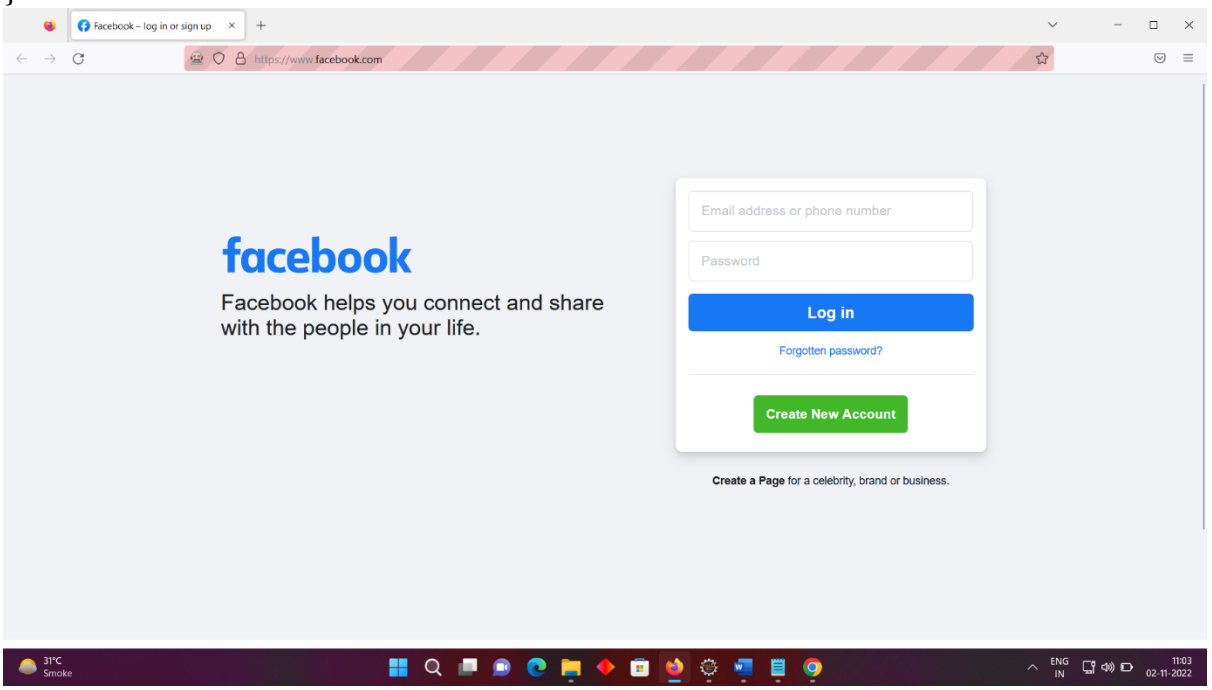
Step1:-Create java project and add jar file of selenium and create package and inside it java class.

Step2 :- Write the code inside the java file

Testingfornavigation.java

```
package navigate;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Testingfornavigation {

        public static void main(String[] args) {
```

```
                System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem
3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-
win64\\geckodriver.exe");
                WebDriver driver = new FirefoxDriver(); //navigates to the browser
                driver.manage().window().maximize();
                driver.get("https://www.facebook.com/");// to go to particular
page.javatpoint.com
                String Title =driver.getTitle();
                System.out.println("WebPage Title is ="+Title);
                String CurrentURL=driver.getCurrentUrl(); //2
                System.out.println("WebPage CurrentURL is ="+CurrentURL);
                String getPageSource =driver.getPageSource(); //3
                System.out.println("WebPage PageSource is =" +getPageSource);

        }

}
```



b)

Step1:-Create java project and add jar file of selenium and create package and inside it java class.

Step2 :- Write the code inside the java file

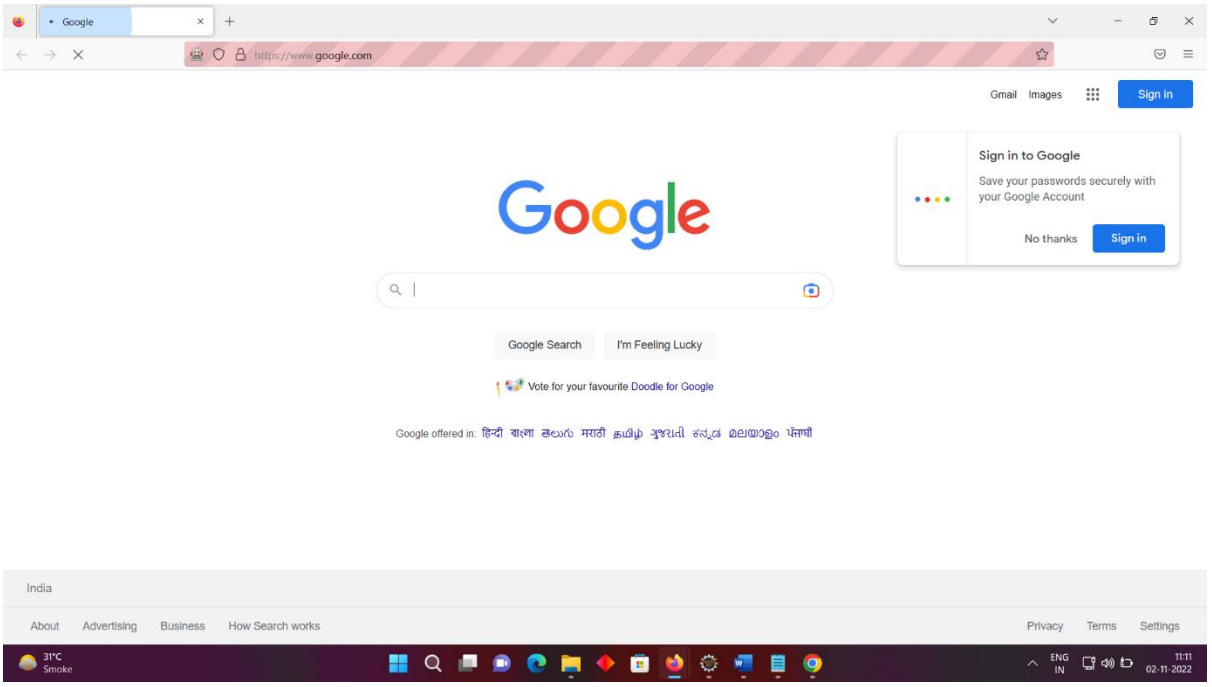
Testingfornavigation2.java

```
package navigate;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class TestingforNavigation2 {

	public static void main(String[] args)
	{
		System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser
		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();//2
		String str1=driver.getCurrentUrl();
		System.out.println(str1);
		driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
		driver.navigate().forward();//3
		String str2 = driver.getCurrentUrl();
		System.out.println(str2);
		driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
		driver.navigate().refresh();//4
		String str3 = driver.getCurrentUrl();
		System.out.println(str3);
		driver.manage().timeouts().implicitlyWait(5000, TimeUnit.SECONDS);
		driver.quit();

	}
```

}
Output

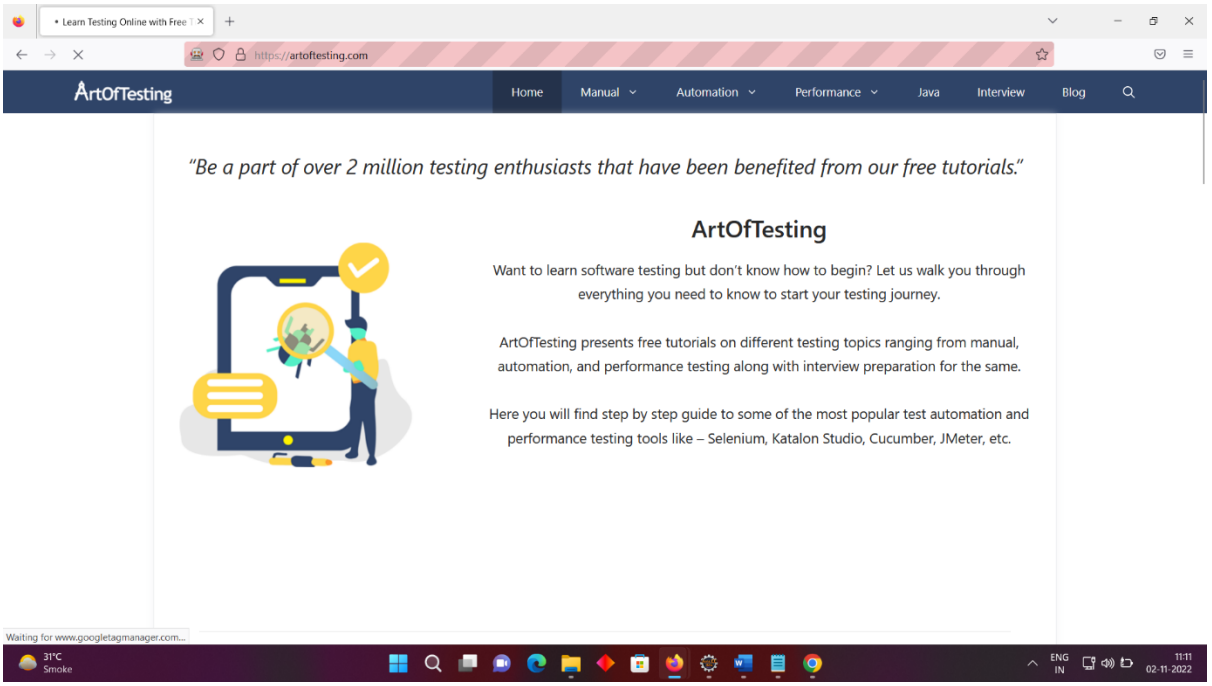
It will first redirect to

https://www.google.com/



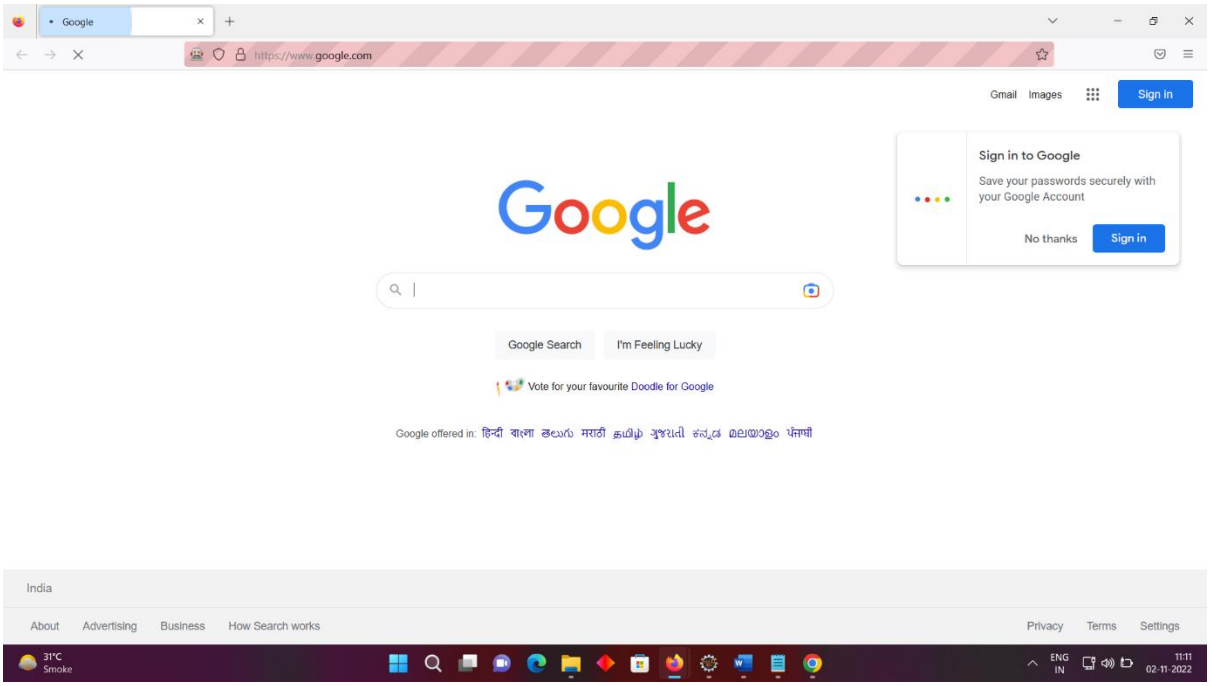
After 5 seconds it will redirect to

https://artoftesting.com/



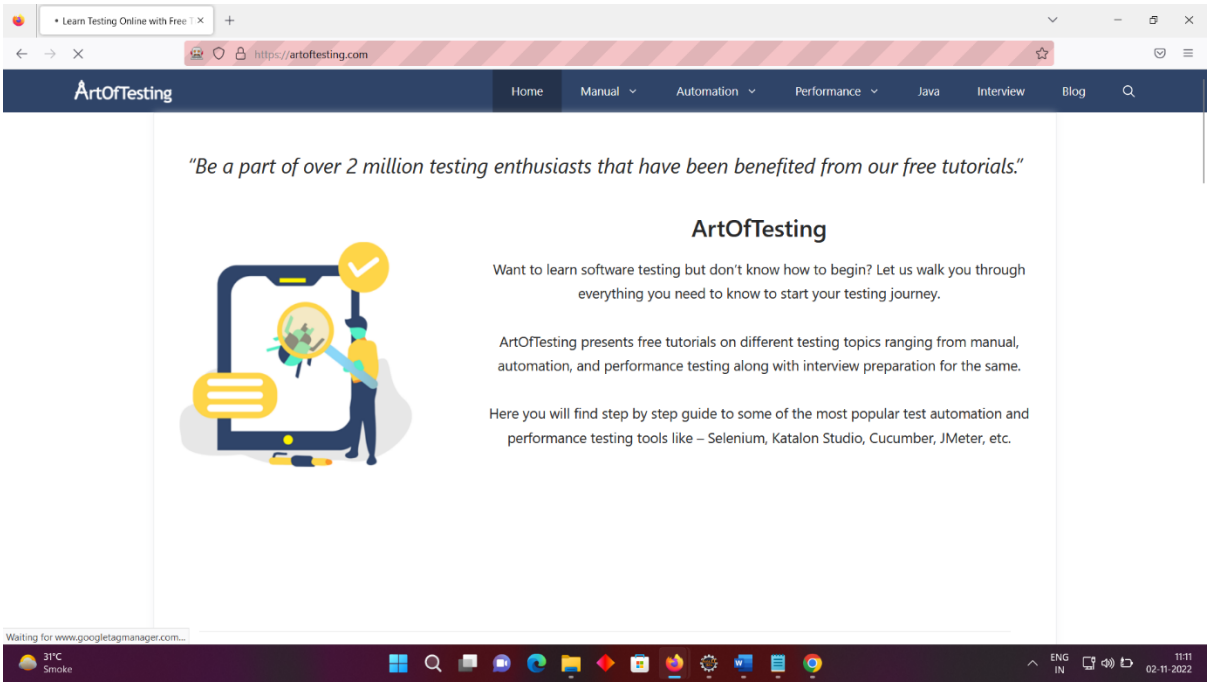
After 5 seconds it will navigate back to

https://www.google.com/



After 5 seconds it will navigate back to

https://artoftesting.com/



After this page it will close the window

Practical 5 and 6

Aim:- Implement the find element command

Step1:-Create java project and add jar file of selenium and create package and inside it java class.

Step2 :- Write the code inside the java file

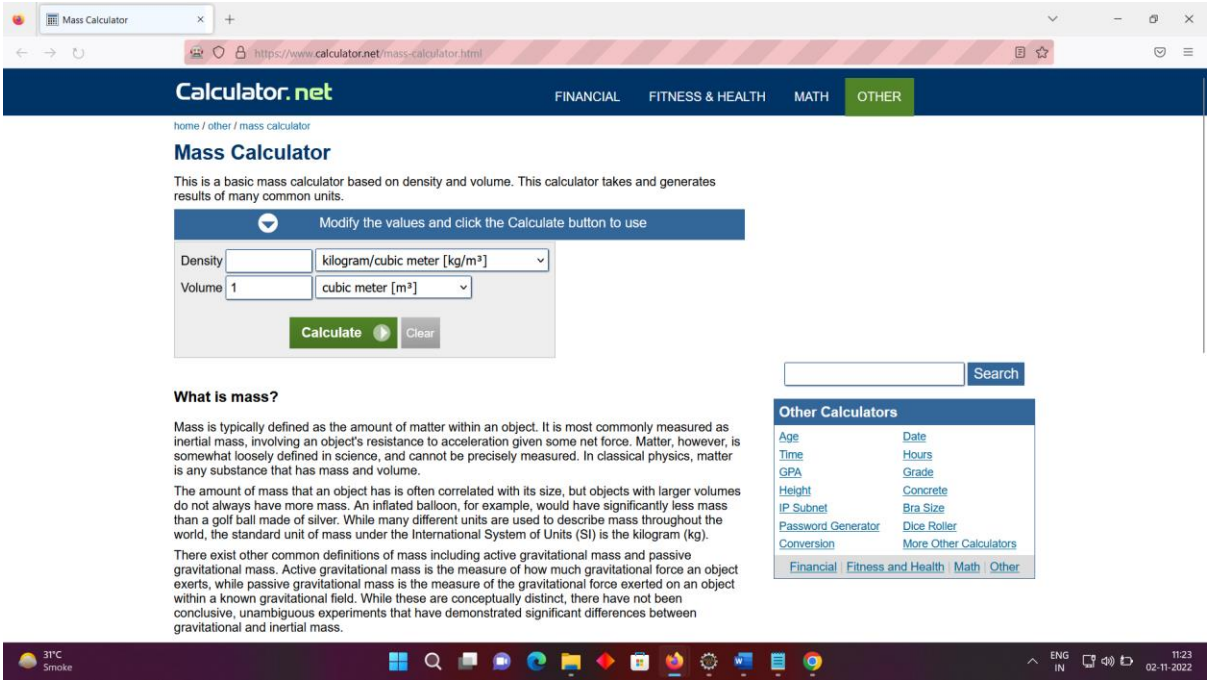
FindElement.java

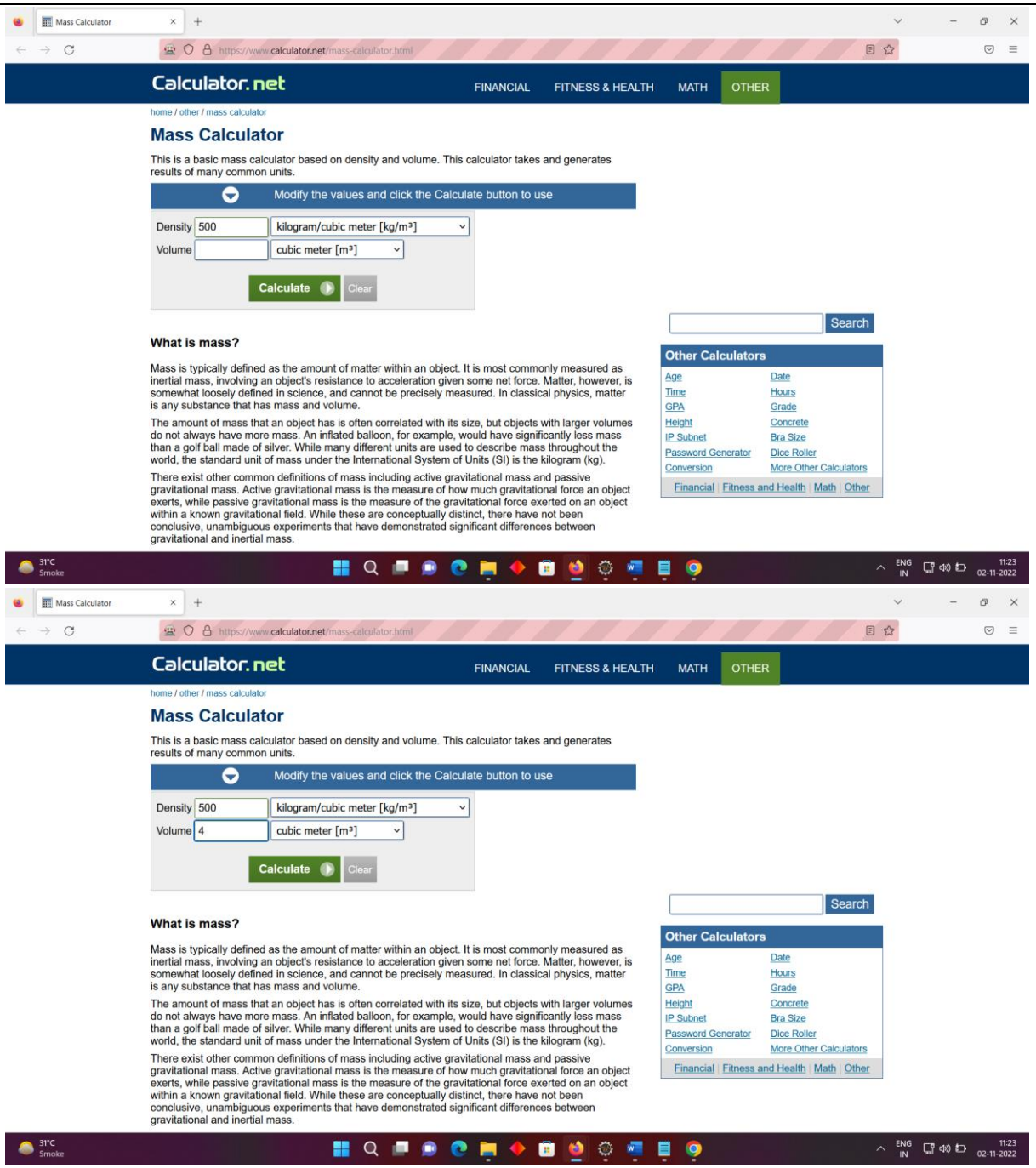
```
package LocatorsAndFindElement;

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

public class FindElement {

	public static void main(String[] args) throws Exception
	{
		System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser
		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("4");
			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 Test
                    driver.findElement(By.partialLinkText("Time")).click();
                    driver.navigate().back();
                    Thread.sleep(5000);

                    // By CSS selector (Limitation on Browser)
                    driver.findElement(By.cssSelector("table.panel tbody:nth-child(1)
tr:nth-child(3) td:nth-child(1) &gt; 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();

        }

}
```

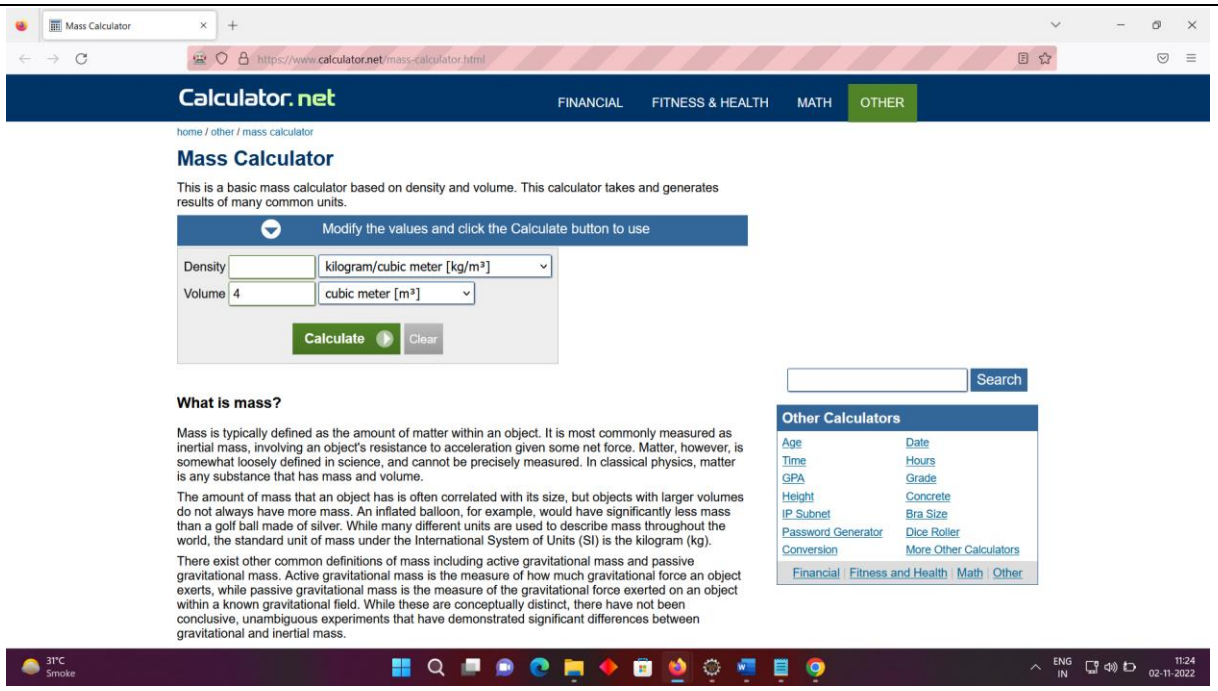
Output:-

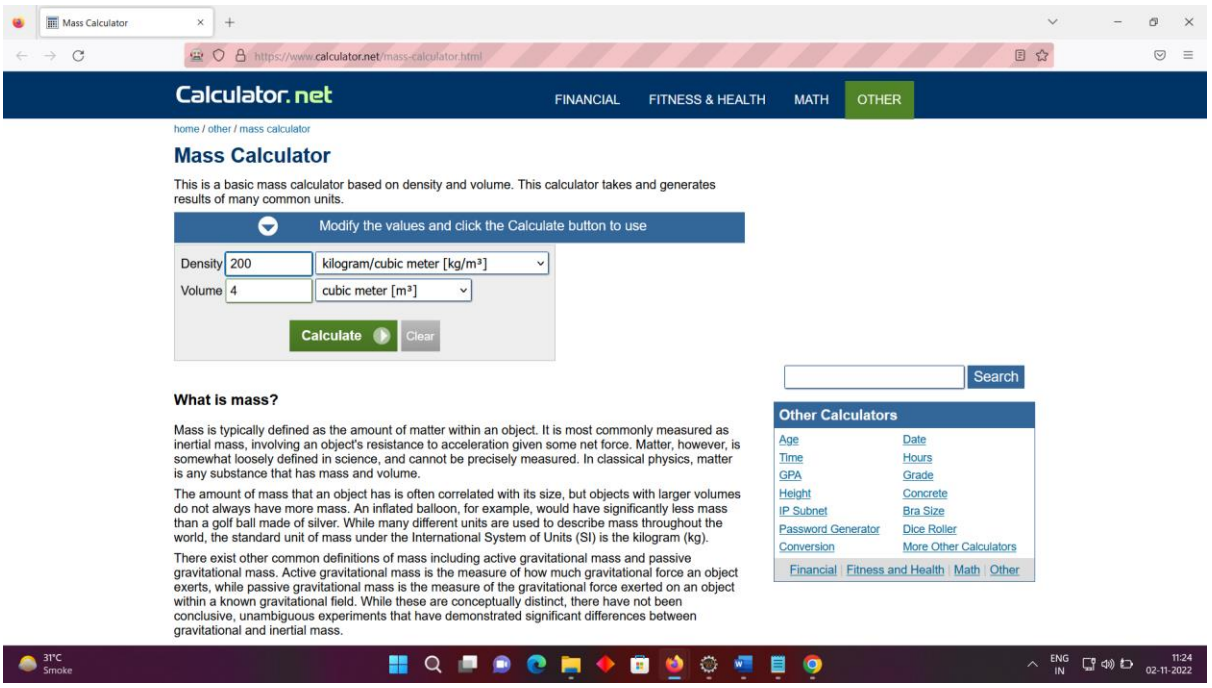
It will automatically add the values and calculate the value as you can see the screenshots

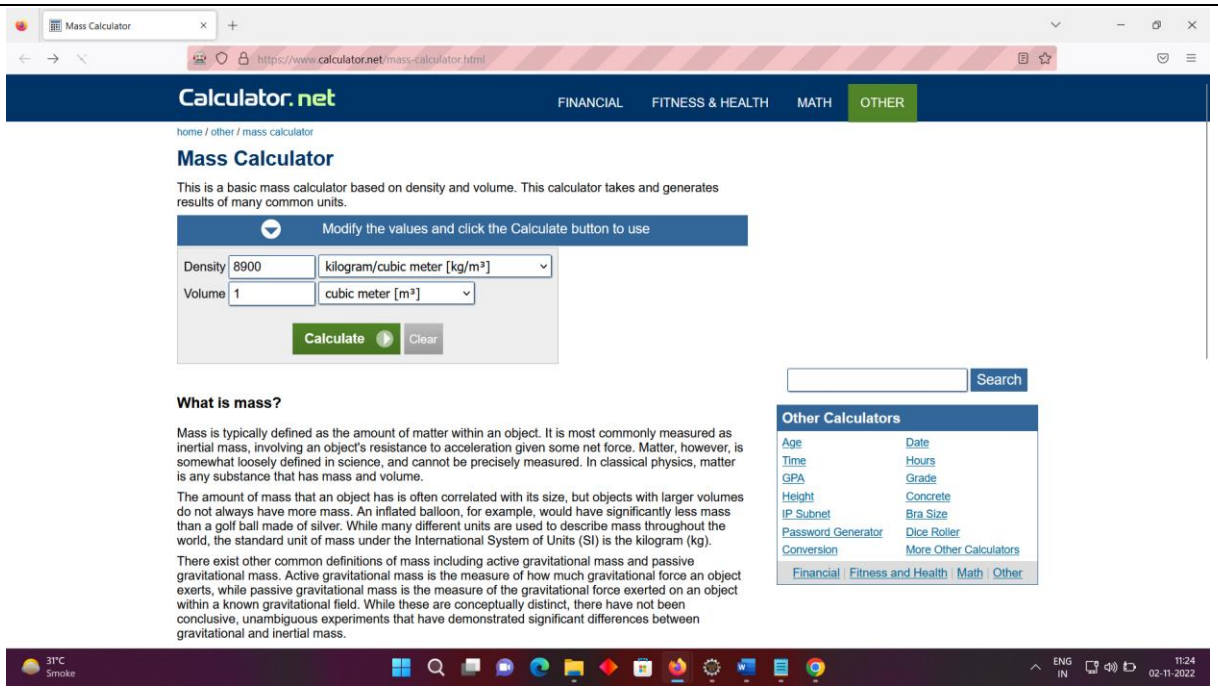



Mass Calculator
https://www.calculator.net/mass-calculator.html
Calculator.net
FINANCIAL
FITNESS & HEALTH
MATH
OTHER
home / other / mass calculator
Mass Calculator
This is a basic mass calculator based on density and volume. This calculator takes and generates results of many common units.
Modify the values and click the Calculate button to use
Density 500 kilogram/cubic meter [kg/m³]
Volume cubic meter [m³]
Calculate Clear
What is mass?
Mass is typically defined as the amount of matter within an object. It is most commonly measured as inertial mass, involving an object's resistance to acceleration given some net force. Matter, however, is somewhat loosely defined in science, and cannot be precisely measured. In classical physics, matter is any substance that has mass and volume.
The amount of mass that an object has is often correlated with its size, but objects with larger volumes do not always have more mass. An inflated balloon, for example, would have significantly less mass than a golf ball made of silver. While many different units are used to describe mass throughout the world, the standard unit of mass under the International System of Units (SI) is the kilogram (kg).
There exist other common definitions of mass including active gravitational mass and passive gravitational mass. Active gravitational mass is the measure of how much gravitational force an object exerts, while passive gravitational mass is the measure of the gravitational force exerted on an object within a known gravitational field. While these are conceptually distinct, there have not been conclusive, unambiguous experiments that have demonstrated significant differences between gravitational and inertial mass.
Search
Other Calculators
Age Date
Time Hours
GPA Grade
Height Concrete
IP Subnet Bra Size
Password Generator Dice Roller
Conversion More Other Calculators
Financial | Fitness and Health | Math | Other
31°C Smoke
ENG IN
11:23 02-11-2022

Mass Calculator
https://www.calculator.net/mass-calculator.html
Calculator.net
FINANCIAL
FITNESS & HEALTH
MATH
OTHER
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 500 kilogram/cubic meter [kg/m³]
Volume 4 cubic meter [m³]
Calculate Clear
What is mass?
Mass is typically defined as the amount of matter within an object. It is most commonly measured as inertial mass, involving an object's resistance to acceleration given some net force. Matter, however, is somewhat loosely defined in science, and cannot be precisely measured. In classical physics, matter is any substance that has mass and volume.
The amount of mass that an object has is often correlated with its size, but objects with larger volumes do not always have more mass. An inflated balloon, for example, would have significantly less mass than a golf ball made of silver. While many different units are used to describe mass throughout the world, the standard unit of mass under the International System of Units (SI) is the kilogram (kg).
There exist other common definitions of mass including active gravitational mass and passive gravitational mass. Active gravitational mass is the measure of how much gravitational force an object exerts, while passive gravitational mass is the measure of the gravitational force exerted on an object within a known gravitational field. While these are conceptually distinct, there have not been conclusive, unambiguous experiments that have demonstrated significant differences between gravitational and inertial mass.
Search
Other Calculators
Age Date
Time Hours
GPA Grade
Height Concrete
IP Subnet Bra Size
Password Generator Dice Roller
Conversion More Other Calculators
Financial | Fitness and Health | Math | Other
31°C Smoke
ENG IN
11:23 02-11-2022


Mass Calculator
https://www.calculator.net/mass-calculator.html
Calculator.net
FINANCIAL
FITNESS & HEALTH
MATH
OTHER
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
kilogram/cubic meter [kg/m³]
Volume
4
cubic meter [m³]
Calculate
Clear
What is mass?
Mass is typically defined as the amount of matter within an object. It is most commonly measured as inertial mass, involving an object's resistance to acceleration given some net force. Matter, however, is somewhat loosely defined in science, and cannot be precisely measured. In classical physics, matter is any substance that has mass and volume.
The amount of mass that an object has is often correlated with its size, but objects with larger volumes do not always have more mass. An inflated balloon, for example, would have significantly less mass than a golf ball made of silver. While many different units are used to describe mass throughout the world, the standard unit of mass under the International System of Units (SI) is the kilogram (kg).
There exist other common definitions of mass including active gravitational mass and passive gravitational mass. Active gravitational mass is the measure of how much gravitational force an object exerts, while passive gravitational mass is the measure of the gravitational force exerted on an object within a known gravitational field. While these are conceptually distinct, there have not been conclusive, unambiguous experiments that have demonstrated significant differences between gravitational and inertial mass.
Search
Other Calculators
Age
Date
Time
Hours
GPA
Grade
Height
Concrete
IP Subnet
Bra Size
Password Generator
Dice Roller
Conversion
More Other Calculators
Financial | Fitness and Health | Math | Other
31°C Smoke
ENG IN
11:24 02-11-2022


Mass Calculator
https://www.calculator.net/mass-calculator.html
Calculator.net
FINANCIAL
FITNESS & HEALTH
MATH
OTHER
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
200
kilogram/cubic meter [kg/m³]
Volume
4
cubic meter [m³]
Calculate
Clear
What is mass?
Mass is typically defined as the amount of matter within an object. It is most commonly measured as inertial mass, involving an object's resistance to acceleration given some net force. Matter, however, is somewhat loosely defined in science, and cannot be precisely measured. In classical physics, matter is any substance that has mass and volume.
The amount of mass that an object has is often correlated with its size, but objects with larger volumes do not always have more mass. An inflated balloon, for example, would have significantly less mass than a golf ball made of silver. While many different units are used to describe mass throughout the world, the standard unit of mass under the International System of Units (SI) is the kilogram (kg).
There exist other common definitions of mass including active gravitational mass and passive gravitational mass. Active gravitational mass is the measure of how much gravitational force an object exerts, while passive gravitational mass is the measure of the gravitational force exerted on an object within a known gravitational field. While these are conceptually distinct, there have not been conclusive, unambiguous experiments that have demonstrated significant differences between gravitational and inertial mass.
Search
Other Calculators
Age
Date
Time
Hours
GPA
Grade
Height
Concrete
IP Subnet
Bra Size
Password Generator
Dice Roller
Conversion
More Other Calculators
Financial | Fitness and Health | Math | Other
31°C Smoke
ENG IN
11:24 02-11-2022


Mass Calculator
https://www.calculator.net/mass-calculator.html
Calculator.net
FINANCIAL
FITNESS & HEALTH
MATH
OTHER
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 8900 kilogram/cubic meter [kg/m³]
Volume 1 cubic meter [m³]
Calculate
Clear
What is mass?
Mass is typically defined as the amount of matter within an object. It is most commonly measured as inertial mass, involving an object's resistance to acceleration given some net force. Matter, however, is somewhat loosely defined in science, and cannot be precisely measured. In classical physics, matter is any substance that has mass and volume.
The amount of mass that an object has is often correlated with its size, but objects with larger volumes do not always have more mass. An inflated balloon, for example, would have significantly less mass than a golf ball made of silver. While many different units are used to describe mass throughout the world, the standard unit of mass under the International System of Units (SI) is the kilogram (kg).
There exist other common definitions of mass including active gravitational mass and passive gravitational mass. Active gravitational mass is the measure of how much gravitational force an object exerts, while passive gravitational mass is the measure of the gravitational force exerted on an object within a known gravitational field. While these are conceptually distinct, there have not been conclusive, unambiguous experiments that have demonstrated significant differences between gravitational and inertial mass.
Search
Other Calculators
Age
Date
Time
Hours
GPA
Grade
Height
Concrete
IP Subnet
Bra Size
Password Generator
Dice Roller
Conversion
More Other Calculators
Financial | Fitness and Health | Math | Other
31°C
Smoke
ENG
IN
11:24
02-11-2022

Practcial 7

Aim:- Understanding Synchronization in Selenium.

Synchronization in selenium is combining different components in selenium

Two Types of Synchronization

a)Conditional:-Two Types, Implicit and Explicit.

cmd:- driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(30));

The Implicit wait takes one argument: TimeSpan(timeToWait)

b)Unconditional:- cmd - System.Threading.Thread.Sleep();

a)

Step1:-Create java project and add jar file of selenium and create package and inside it java class.

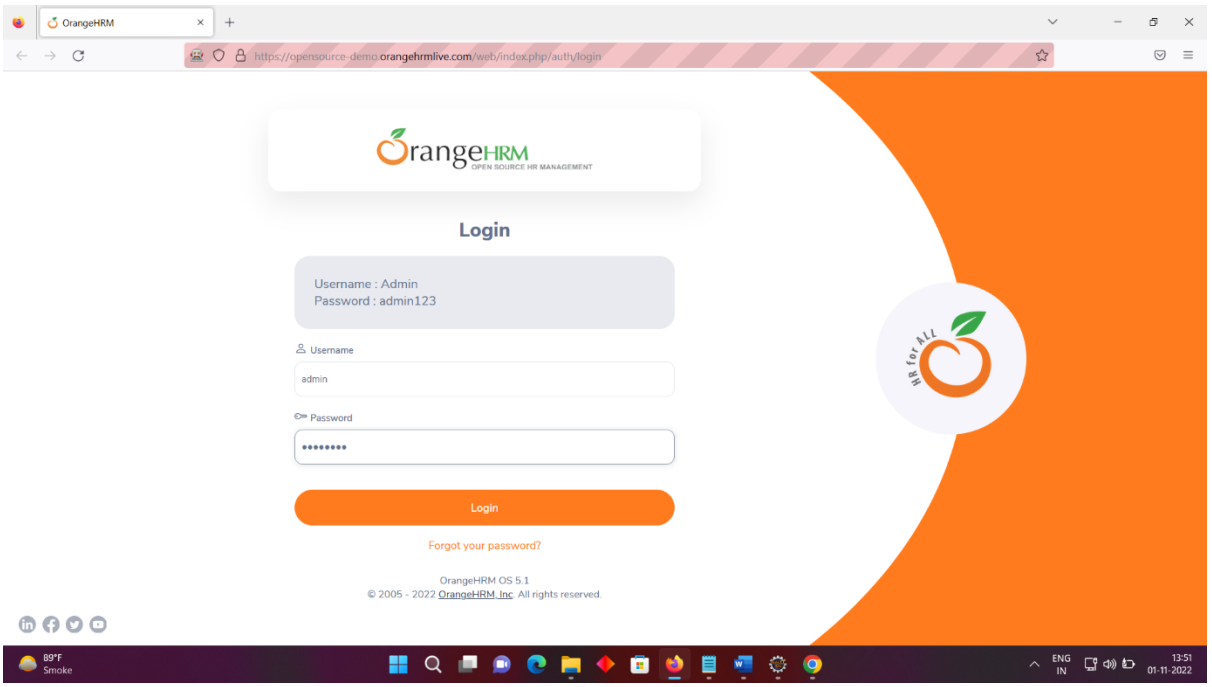
Step2 :- Write the code inside the java file

Sync.java

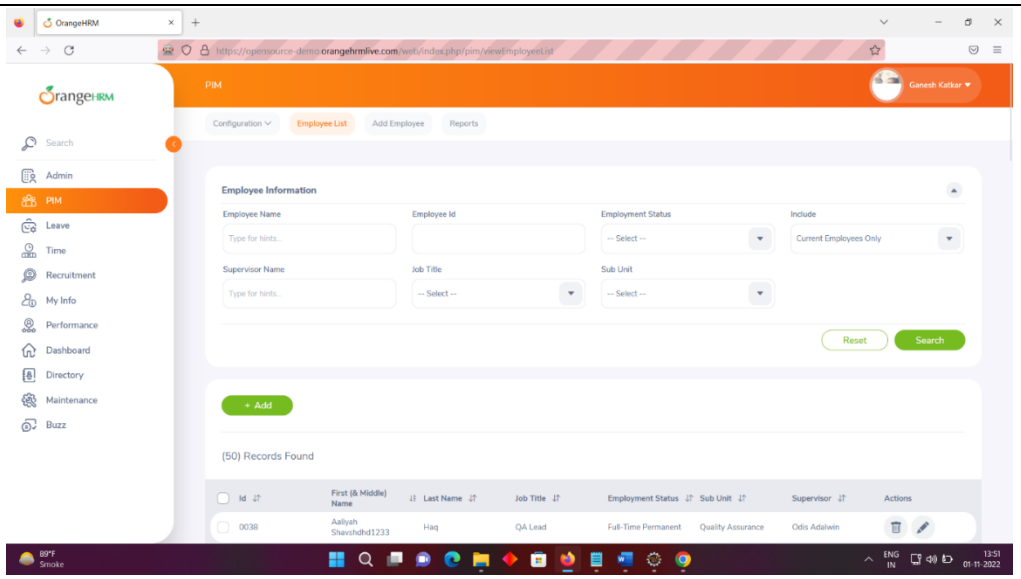
```
package synchronization;

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

public class Sync {

	public static void main(String[] args)
	{
		//step 1 webdriver launch
		System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser

		//step 2 getting the url
		driver.get("https://opensource-demo.orangehrmlive.com/");

		//step 3 finding the element through id,name,classname,linktext,xpath
		driver.findElement(By.id("txtUsername")).sendKeys("admin");//locator id
		driver.findElement(By.name("txtPassword")).sendKeys("admin123");
		driver.findElement(By.className("button")).click();
```

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



        }

}
```

Step3:-Run the java file it will redirect to a website https://opensource-demo.orangehrmlive.com/



Step 4:-After entering username and password (username:-admin,password:-admin123) it will redirect to this dashboard



b)

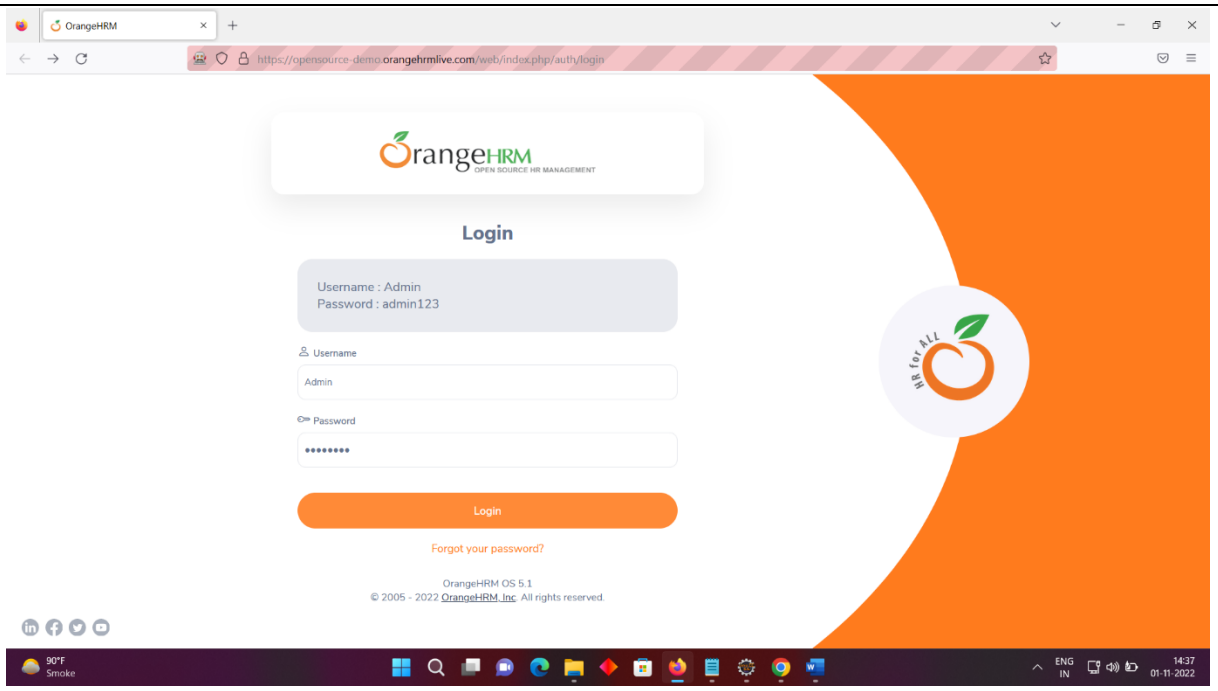
Step1:-Create java project select java SE 1.8 and add jar file of selenium and create package and inside it java class.

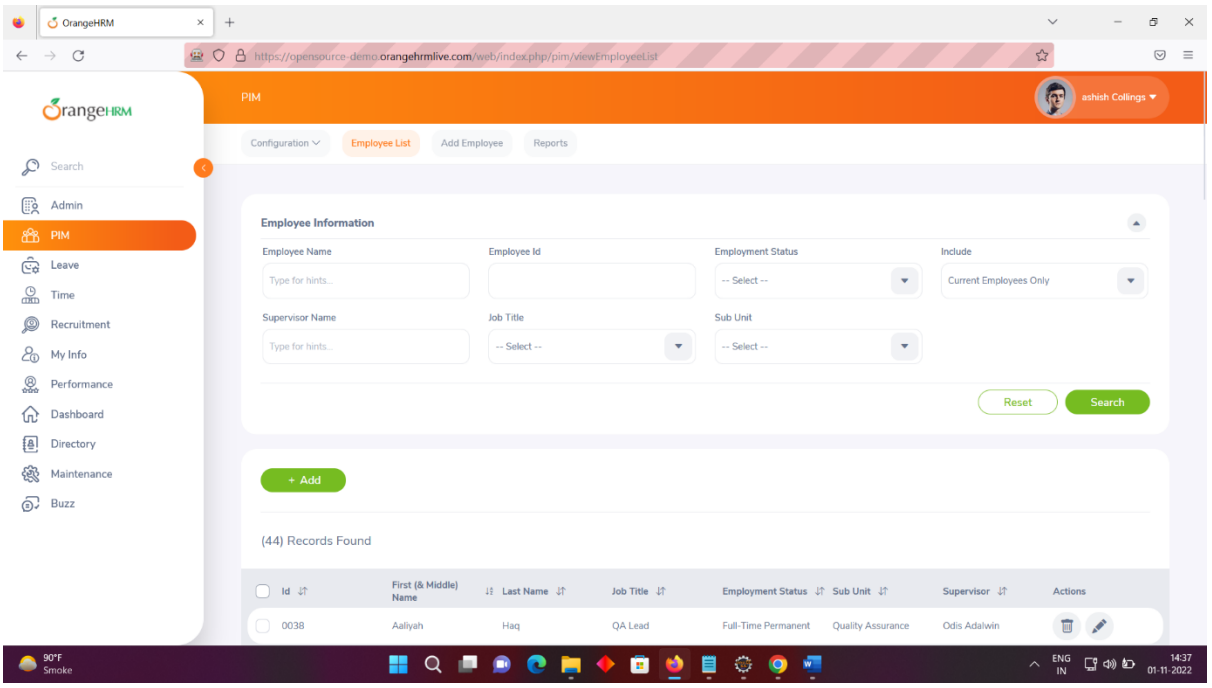
Step2 :- Write the code inside the java file

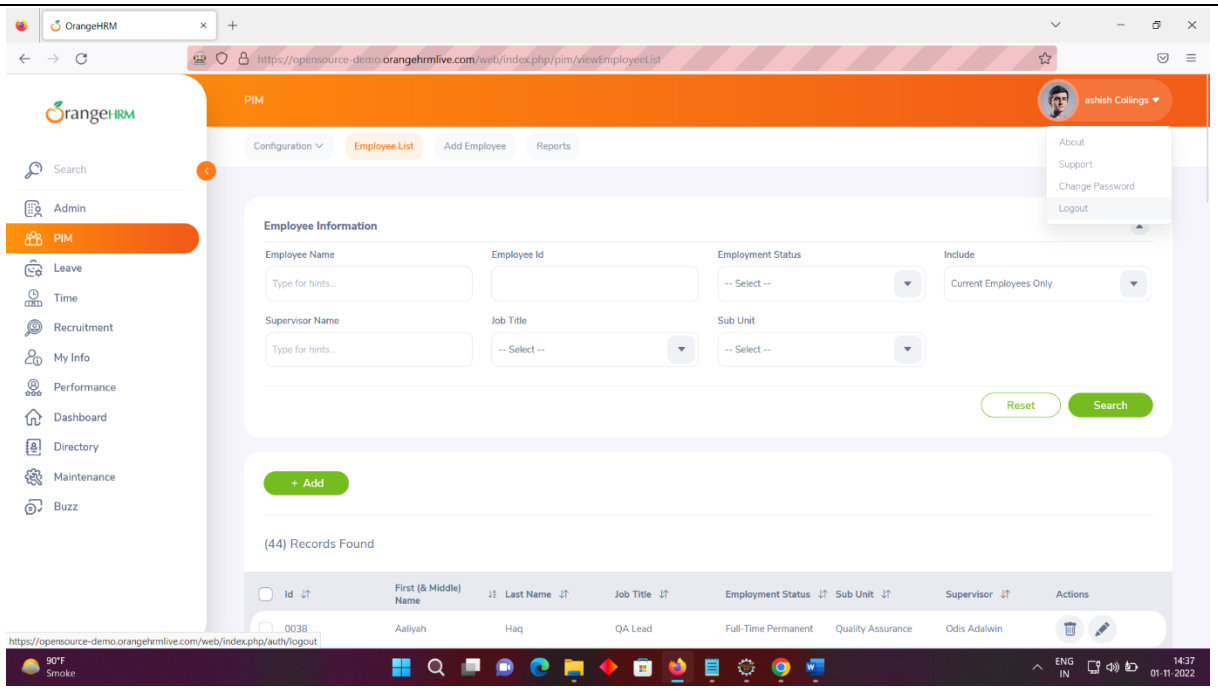
Step3:-

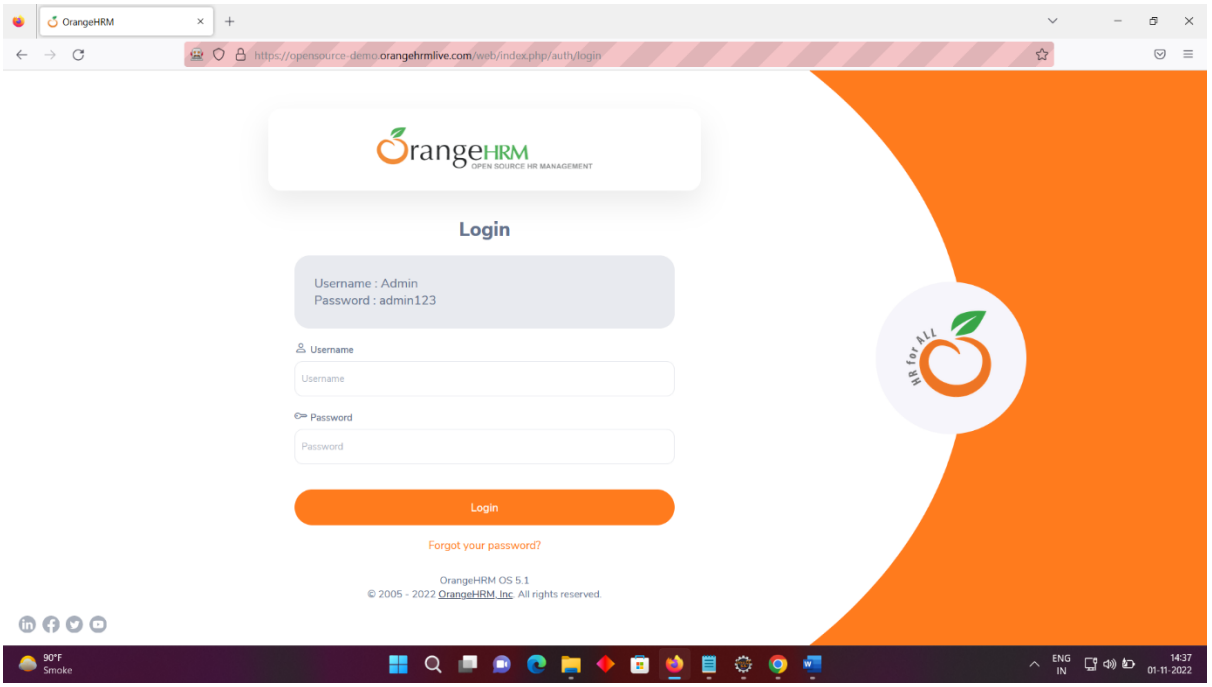
```
package ImplicitExplicit;

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

public class ImplicitExplicit {

	public static void main(String[] args) {
		System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser
		driver.get("https://opensource-demo.orangehrmlive.com/");

		//step 2 getting the url
		WebDriverWait wt = new WebDriverWait(driver, 10);

		//step 3 finding the element through id,name,classname,linktext,xpath
		driver.findElement(By.id("txtUsername")).sendKeys("admin");//locator id
		driver.findElement(By.name("txtPassword")).sendKeys("admin123");
		driver.findElement(By.className("button")).click();
```

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

                //step 4

        wt.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("logout")));

                driver.findElement(By.linkText("logout")).click();

        }

}
```


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/auth/login
OrangeHRM
OPEN SOURCE HR MANAGEMENT
Login
Username : Admin
Password : admin123
Username
Admin
Password
Login
Forgot your password?
OrangeHRM OS 5.1
© 2005 - 2022 OrangeHRM, Inc All rights reserved.
HR for ALL


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/pim/viewEmployeeList
PIM
ashish Collings
Configuration
Employee List
Add Employee
Reports
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Buzz
Employee Information
Employee Name
Employee Id
Employment Status
Include
Current Employees Only
Supervisor Name
Job Title
Sub Unit
Reset
Search
+ Add
(44) Records Found
Id
First (& Middle) Name
Last Name
Job Title
Employment Status
Sub Unit
Supervisor
Actions
0038
Aaliyah
Haq
QA Lead
Full-Time Permanent
Quality Assurance
Odis Adalwin


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/pim/viewEmployeeList
PIM
ashish Collings
Configuration
Employee List
Add Employee
Reports
About
Support
Change Password
Logout
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Buzz
Employee Information
Employee Name
Employee Id
Employment Status
Include
Type for hints...
-- Select --
Current Employees Only
Supervisor Name
Job Title
Sub Unit
Reset
Search
+ Add
(44) Records Found
Id
First (& Middle) Name
Last Name
Job Title
Employment Status
Sub Unit
Supervisor
Actions
0038
Aaliyah
Haq
QA Lead
Full-Time Permanent
Quality Assurance
Odis Adalwin
https://opensource-demo.orangehrmlive.com/web/index.php/auth/logout


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/auth/login
OrangeHRM
OPEN SOURCE HR MANAGEMENT
Login
Username : Admin
Password : admin123
Username
Password
Login
Forgot your password?
OrangeHRM OS 5.1
© 2005 - 2022 OrangeHRM, Inc All rights reserved.
HR for ALL

Practical 8

Aim:-Alert Message in selenium with alert method.

Step1:- Create java project and add jar file of selenium and create package and inside it create java class.

Step2:- Now “copy as path” of the driver (Shift + Right Click)

In this we have to pass two parameters one is driver name and other is driver path.

```
System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");

WebDriver driver = new FirefoxDriver(); //navigates to the browser
```

Step3:- To get the URL we to write

```
driver.get(“http://demo.guru99.com/test/delete_customer.php”);
```

Step4:- find element by using findElement

```
driver.findElement(By.name("cusid")).sendKeys("53920");
driver.findElement(By.name("submit")).submit();
```

Step5:-to give alert popup message we have to use “alert()”;

```
driver.switchTo().alert();
driver.switchTo().alert().getText();
```

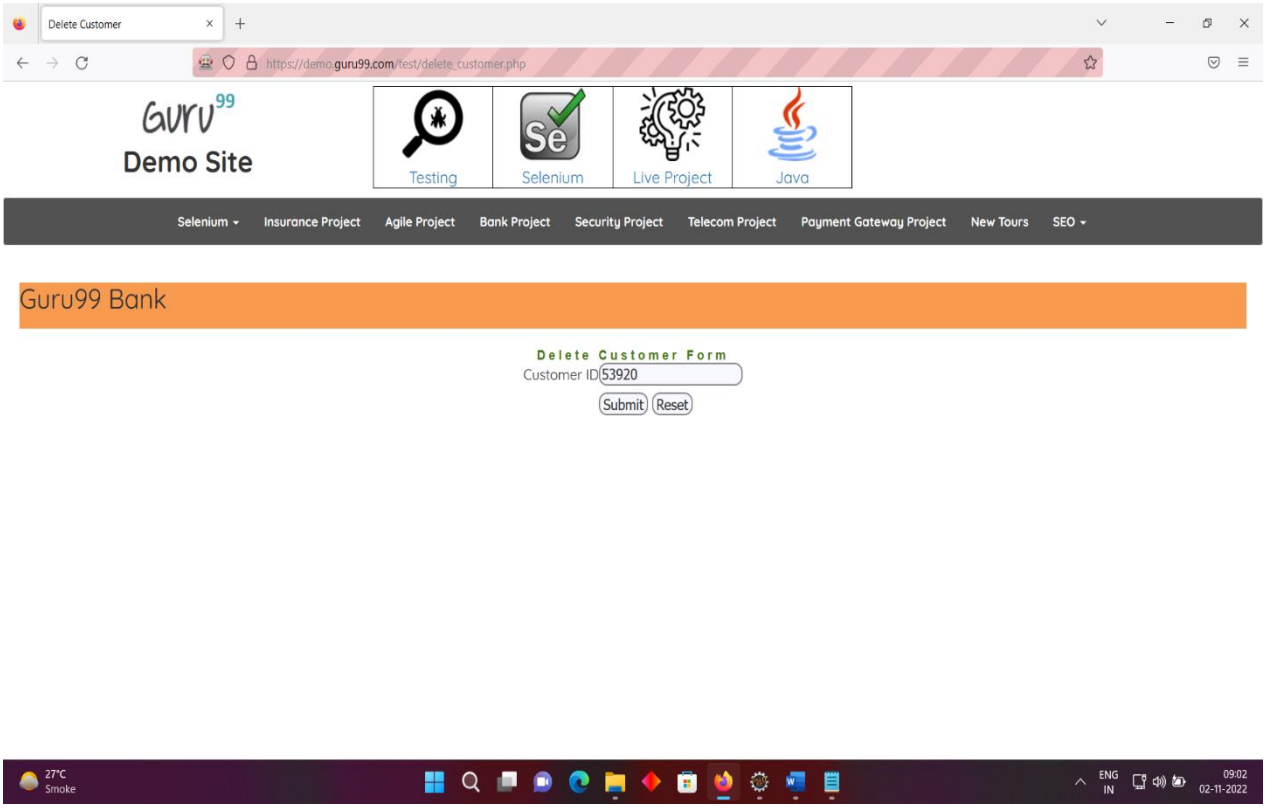
Full code

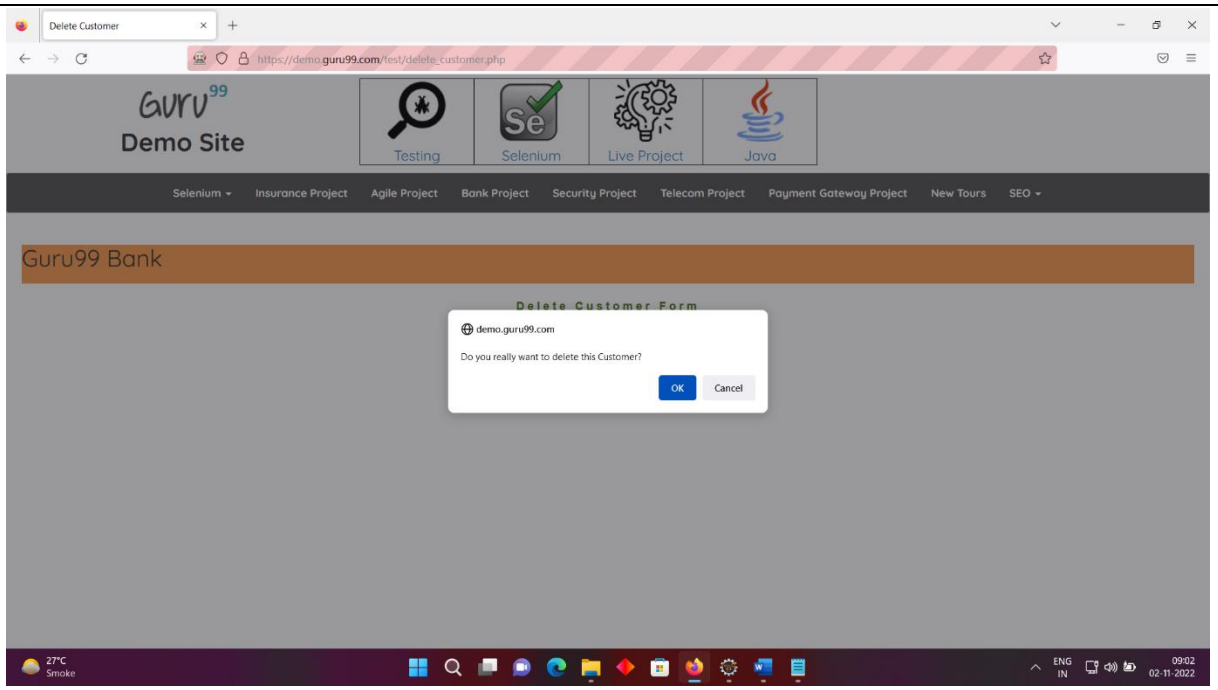
alert.java

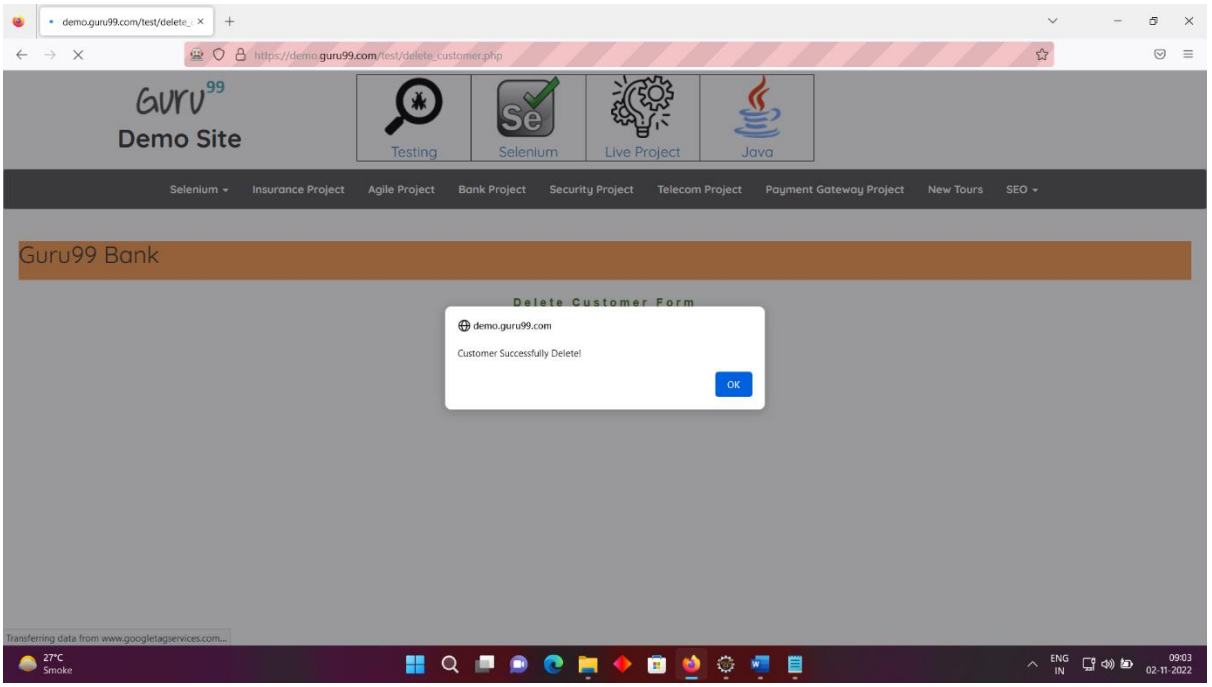
```
package Alert;

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

public class alert {

	public static void main(String[] args)
	{
		System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser

		driver.get("http://demo.guru99.com/test/delete_customer.php");
```

```
            driver.findElement(By.name("cusid")).sendKeys("53920");
            driver.findElement(By.name("submit")).submit();

            driver.switchTo().alert();
            driver.switchTo().alert().getText();

        }

}
```

Output




Delete Customer
https://demo.guru99.com/test/delete_customer.php
Guru99 Demo Site
Testing
Selenium
Live Project
Java
Selenium
Insurance Project
Agile Project
Bank Project
Security Project
Telecom Project
Payment Gateway Project
New Tours
SEO
Guru99 Bank
Delete Customer Form
demo.guru99.com
Do you really want to delete this Customer?
OK
Cancel


demo.guru99.com/test/delete_
https://demo.guru99.com/test/delete_customer.php
Guru99 Demo Site
Testing
Selenium
Live Project
Java
Selenium
Insurance Project
Agile Project
Bank Project
Security Project
Telecom Project
Payment Gateway Project
New Tours
SEO
Guru99 Bank
Delete Customer Form
demo.guru99.com
Customer Successfully Delete!
OK

PRACTICAL 9

Aim:- Demonstrate : Handling Drop Down, List Boxes

Step1:- Create java project and add jar file of selenium and create package and inside it create java class.

Step2:- Now “copy as path” of the driver (Shift + Right Click)

In this we have to pass two parameters one is driver name and other is driver path.

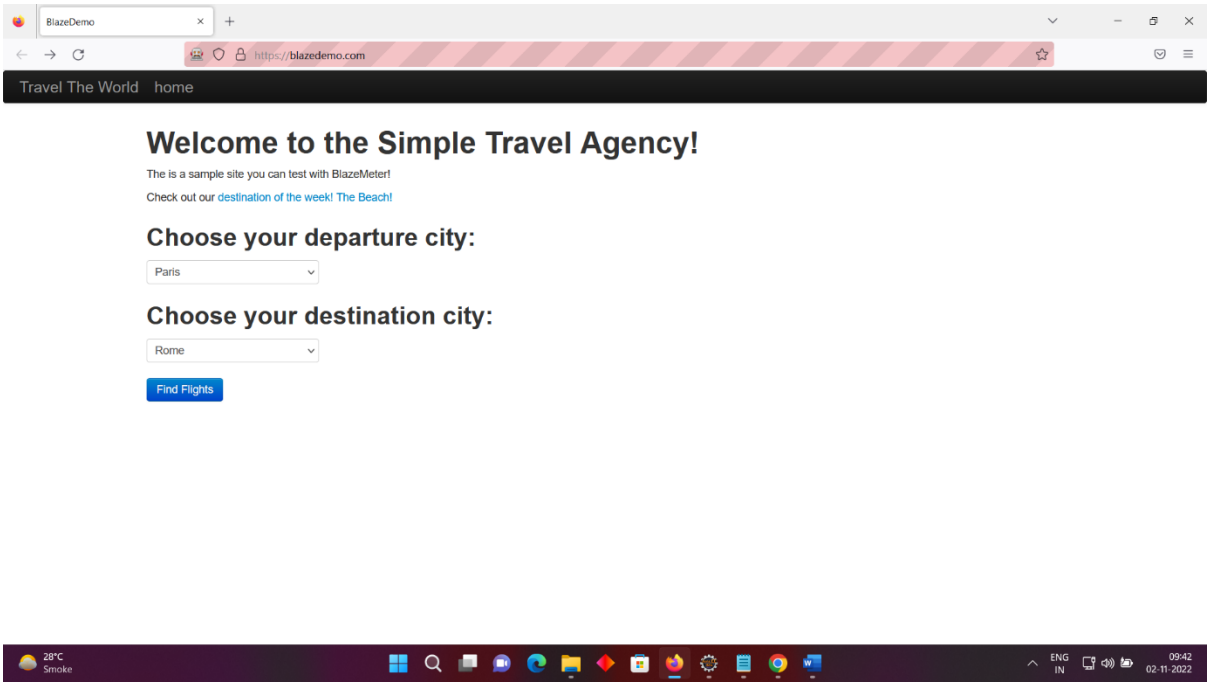
```
System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");

WebDriver driver = new FirefoxDriver(); //navigates to the browser
```

Step3:- Write full code

ClassFind.java

```
package Class;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class ClassFind
{
	public static void main(String args[])
	{
		System.setProperty("webdriver.gecko.driver", "D:\\MCA\\MCA sem 3\\STQA\\STQA_LAB\\Requirements\\STQA_Requirment\\geckodriver-v0.30.0-win64\\geckodriver.exe");
		WebDriver driver = new FirefoxDriver(); //navigates to the browser

		driver.get("https://blazedemo.com/");
		Select s = new Select(driver.findElement(By.name("fromPort")));
		Select t = new Select(driver.findElement(By.name("toPort")));

		s.selectByVisibleText("Paris");
		t.selectByVisibleText("Rome");
	}
}
```

Output:-


BlazeDemo
https://blazedemo.com
Travel The World home
Welcome to the Simple Travel Agency!
The is a sample site you can test with BlazeMeter!
Check out our destination of the week! The Beach!
Choose your departure city:
Paris
Choose your destination city:
Rome
Find Flights
28°C
Smoke
ENG
IN
09:42
02-11-2022

Practical No:10

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

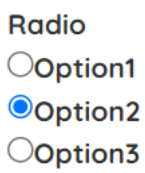
Description:

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:



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/tbody/t
r/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 12

Aim:

TestNG

What is Annotations in Selenium

What is TestNG?

TestNG is an automation testing framework in which NG stands for "Next Generation". TestNG is inspired by JUnit which uses the annotations (@). TestNG overcomes the disadvantages of JUnit and is designed to make end-to-end testing easy.

Using TestNG, you can generate a proper report, and you can easily come to know how many test cases are passed, failed, and skipped. You can execute the failed test cases separately.

For example:

- Suppose, you have five test cases, one method is written for each test case (Assume that the program is written using the main method without using testNG). When you run this program first, three methods are executed successfully, and the fourth method is failed. Then correct the errors present in the fourth method, now you want to run only fourth method because first three methods are anyway executed successfully. This is not possible without using TestNG.
- The TestNG in Selenium provides an option, i.e., testng-failed.xml file in test-output folder. If you want to run only failed test cases means you run this XML file. It will execute only failed test cases.

Beside above concept, you will learn more on TestNG, like what are the Advantages of TestNG, how to create test methods using @test annotations, how to convert these classes into testing suite file and execute through the eclipse as well as from the command line.

Why Use TestNG with Selenium?

Default Selenium tests do not generate a proper format for the test results. Using TestNG in Selenium, we can generate test results.

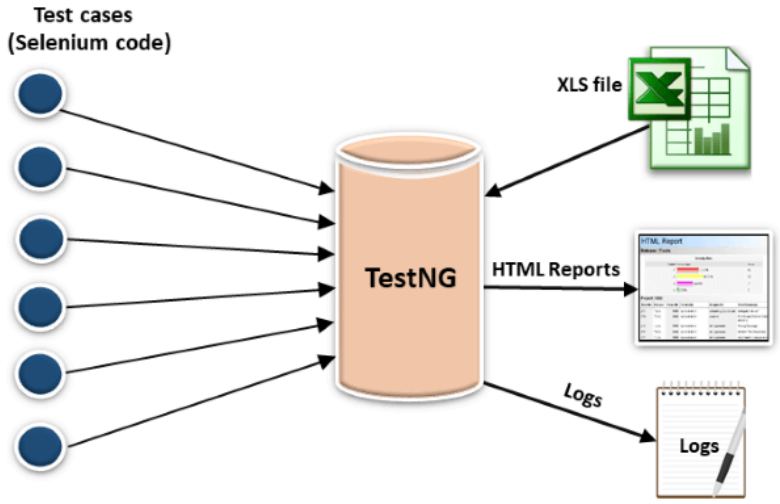
Most Selenium users use this more than Junit because of its advantages. There are so many features of TestNG, but we will only focus on the most important ones that we can use in Selenium. Following are the key features of Selenium TestNG:

- Generate the report in a proper format including a number of test cases runs, the number of test cases passed, the number of test cases failed, and the number of test cases skipped.
- Multiple test cases can be grouped more easily by converting them into testng.xml file. In which you can make priorities which test case should be executed first.
- The same test case can be executed multiple times without loops just by using keyword called 'invocation count.'
- Using testng, you can execute multiple test cases on multiple browsers, i.e., cross browser testing.
- The TestNG framework can be easily integrated with tools like TestNG Maven, Jenkins, etc.
- Annotations used in the testing are very easy to understand ex: @BeforeMethod, @AfterMethod, @BeforeTest, @AfterTest
- WebDriver has no native mechanism for generating reports. TestNG can generate the report in a readable format like the one shown below.

What is TestNG

- TestNG is a very important framework when you are actually developing the framework from scratch level.
- TestNG provides you full control over the test cases and the execution of the test cases. Due to this reason, TestNG is also known as a testing framework.
- Cedric Beust is the developer of a TestNG framework.
- If you want to run a test case A before that as a pre-request you need to run multiple test cases before you begin a test case A. You can set and map with the help of TestNG so that pre-request test cases run first and then only it will trigger a test case A. In such way, you can control the test cases.
- TestNG framework came after Junit, and TestNG framework adds more powerful functionality and easier to use.
- It is an open source automated TestNG framework. In TestNG, NG stands for "**Next Generation**".

- TestNG framework eliminates the limitations of the older framework by providing more powerful and flexible test cases with help of easy annotations, grouping, sequencing and parametrizing.



- TestNG simplifies the way the tests are coded. There is no more need for a static main method in our tests. The sequence of actions is regulated by easy-to-understand annotations that do not require methods to be static.

Annotations used in TestNG

@BeforeSuite: The annotated method will be run before all tests in this suite have run.

@AfterSuite: The annotated method will be run after all tests in this suite have run.

@BeforeTest: The annotated method will be run before any test method belonging to the classes inside the tag is run.

@AfterTest: The annotated method will be run after all the test methods belonging to the classes inside the tag have run.

@BeforeGroups: The list of groups that this configuration method will run before. This method is guaranteed to run shortly before the first test method that belongs to any of these groups is invoked.

@AfterGroups: The list of groups that this configuration method will run after. This method is guaranteed to run shortly after the last test method that belongs to any of these groups is invoked.

@BeforeClass: The annotated method will be run before the first test method in the current class is invoked.

@AfterClass: The annotated method will be run after all the test methods in the current class have been run.

@BeforeMethod: The annotated method will be run before each test method.

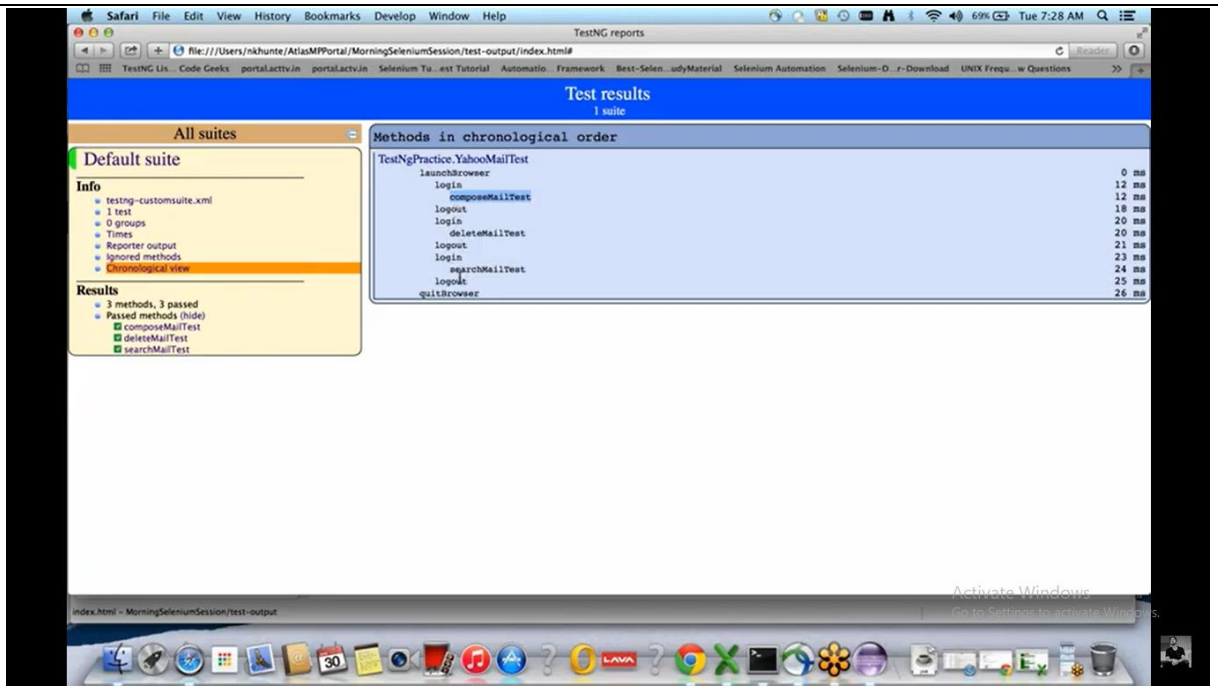
@AfterMethod: The annotated method will be run after each test method.

@Test: The annotated method is a part of a test case

```
package UnderstandingTestNG;

import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

public class TestNGClass {
@BeforeClass
public void launchbrowser()
{
	System.out.println("launch FireFox browser");
}
@BeforeMethod
public void login()
{
	System.out.println("login to app");
}

@Test
public void composemail()
{
	System.out.println("composed mail test");
}
```

```
@Test
public void deletemail()
{
	System.out.println(" mail deleted test");
}
@Test
public void searchemail()
{
	System.out.println("search mail test");
}
@AfterMethod
public void loginout()
{
	System.out.println("login out to app");
}
@AfterClass
public void quitbrowser()
{
	System.out.println("quit FireFox browser");
}

}
```

Output



Practical 13

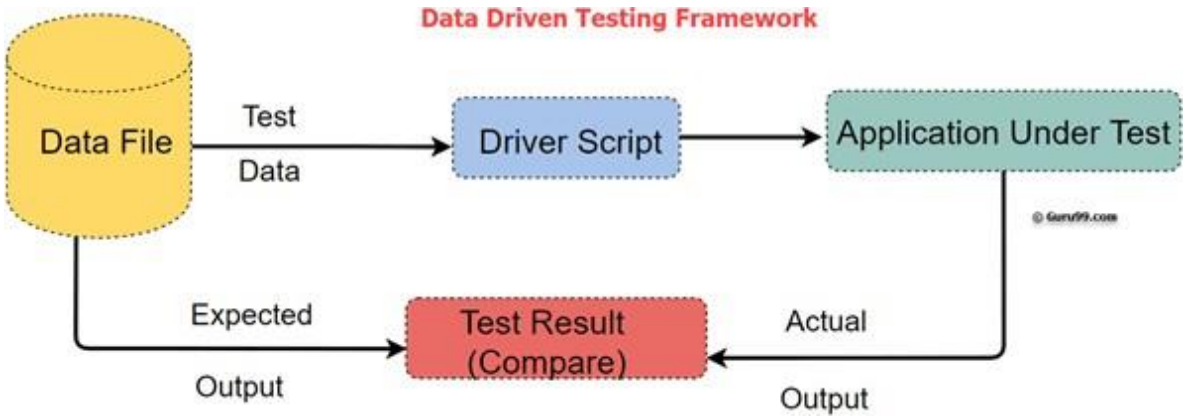
Aim:Demonstrate data driven Framework.

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");
	}
```

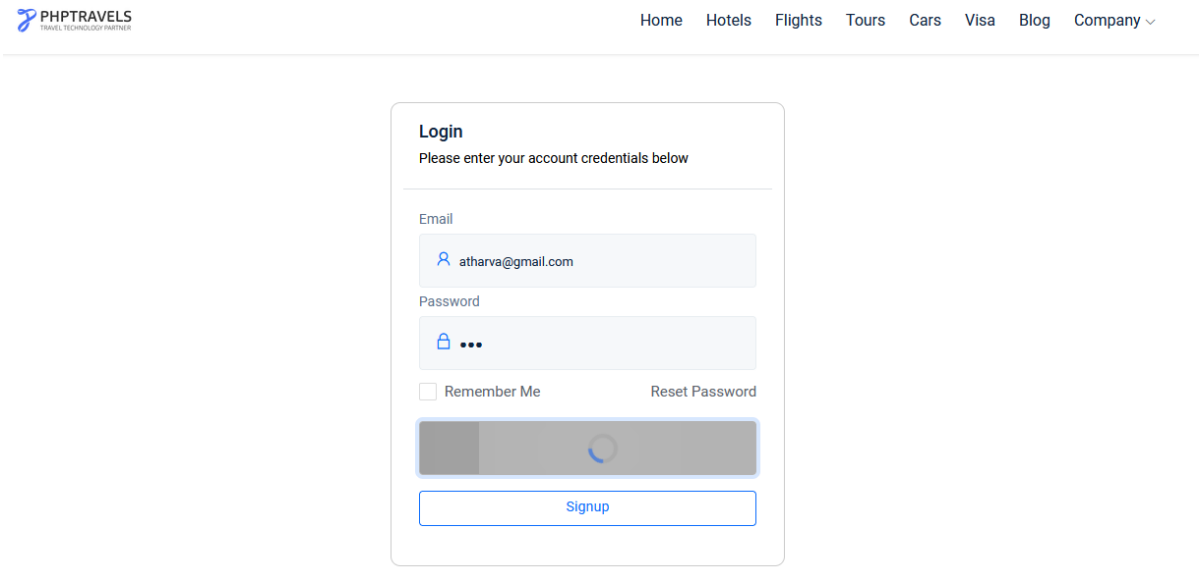
```
@AfterMethod
void ProgramTermination() {
driver.quit();
}

@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;
}}
```

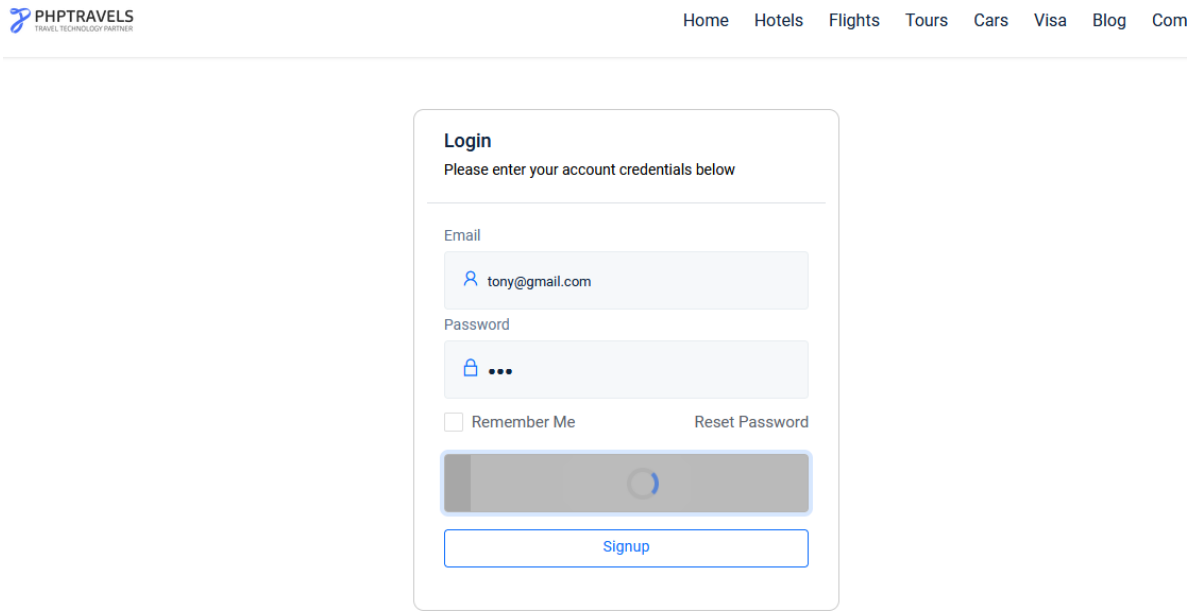
Output:

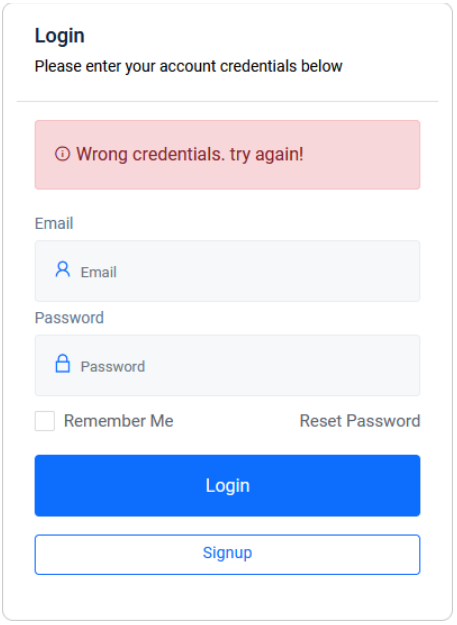
Data in Excel sheet:

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

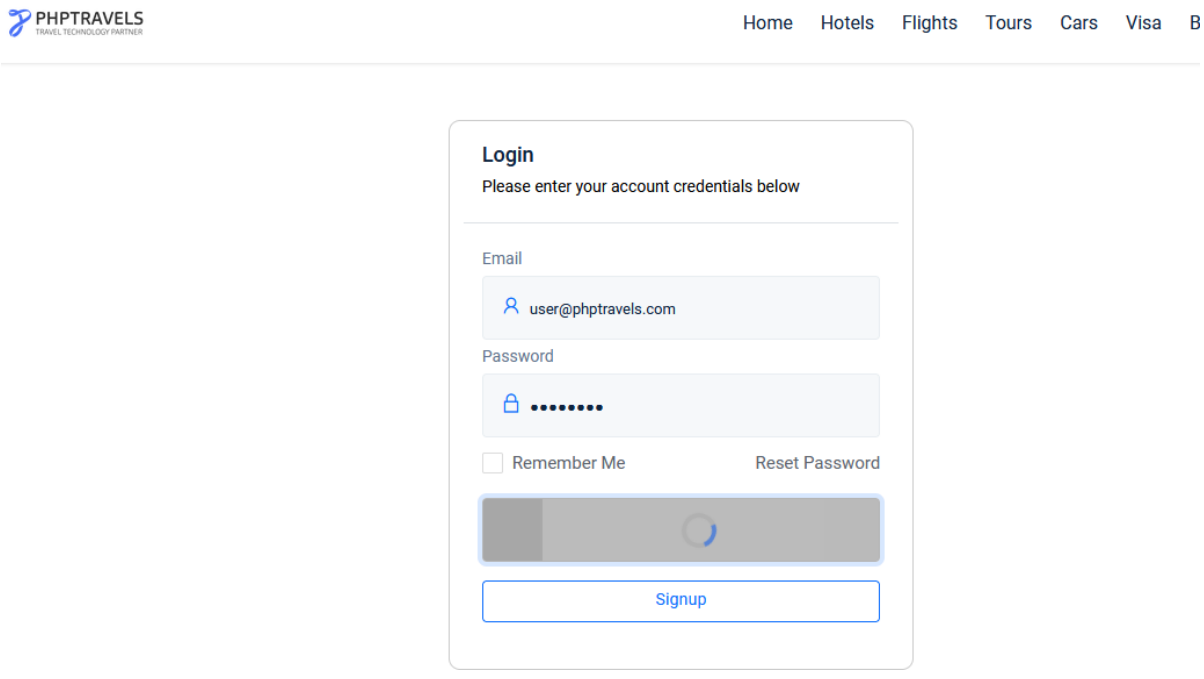
Wrong credentials

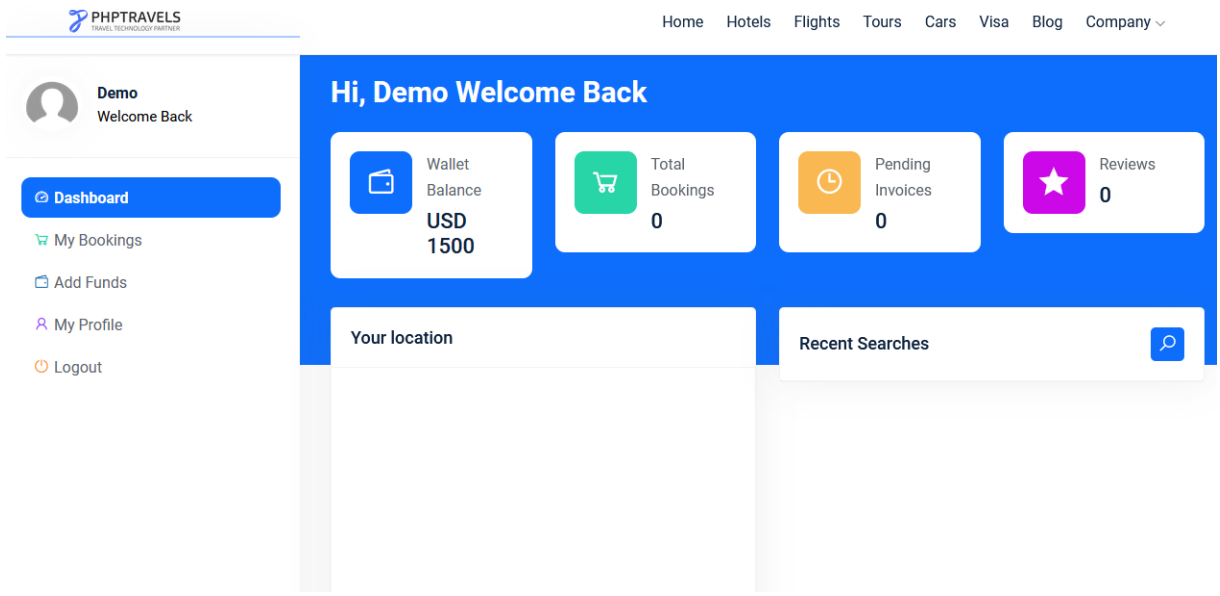






Correct credentials





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 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 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 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 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

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.