Roll No: 12 **Exam Seat No:**

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

Hashu Advani Memorial Complex, Collector's Colony, R. C. Marg, Chembur, Mumbai – 400074. Contact No. 02261532532

Since 1962

CERTIFICATE

Certified that Mr./Miss **DESHMUKH PRANAV DINESH** of **SYMCA** has satisfactorily completed a course of the necessary experiments in **Software Testing and Quality Assurance (STQA)** under the supervision of **Dr. Meenakshi Garg** in the Institute of Technology in the academic year 2023-2024.

Principal

Head of Department
(Dr. Shiv kumar goel)

External Examiner

Subject Teacher
Dr. Meenakshi Garg

V.E.S. Institute of Technology, Collector Colony, Chembur, Mumbai

Department of M.C.A

MCAL35 – STQA LAB INDEX

Sr. No	Contents	Marks	Sign
1	Take a review and write test cases for any known application.		
2	Implement Web Drivers on Chrome & Firefox Browsers.		
3	Demonstrate handling multiple frames in selenium		
4	Implement Browser command and navigation Commands		
5	Implement the find element command		
6	Demonstrate the Locator(id,css selector, path)		
7	Demonstrate synchronization in selenium		
8	Demonstrate different types of alerts		
9	Demonstrate: Handling Drop Down, List Boxes q		
10	Demonstrate : Command Button, Radio buttons & text boxes. Waits command in selenium		

11	**Demonstrate action classes in Selenium**		
12	**Installation of TestNg , running testNg and TestNg annotations**		
13	**Demonstrate data driven Framework**		

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 1</td></tr>
<tr><td colspan="4">Title of LAB Assignment : Take a review and write test cases for any known application</td></tr>
<tr><td colspan="2">DOP : 08/09/2023</td><td colspan="2">DOS : 15/09/2023</td></tr>
<tr><td>CO Mapped :
CO1</td><td>PO Mapped:
PO2, PO3, PO7, PO9 & PSO1</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 1

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

Theory:

Test cases are a fundamental component of software testing, and they play a crucial role in ensuring the quality, functionality, and reliability of software systems. The theory behind test cases revolves around systematically designing and executing tests to identify defects, verify the correctness of software features, and validate that the software meets its intended requirements. Here's a comprehensive overview of the theory behind test cases:

Definition of Test Case:

A test case is a detailed specification that describes a specific scenario or condition to be tested within a software application. It consists of input data, expected outcomes, and a set of execution steps.

Purpose of Test Cases:

- Validate that the software functions correctly and meets its requirements.
- Identify defects and bugs in the software.
- Ensure that changes or updates do not introduce new issues.
- Provide documentation for testing processes and results.
- Support regression testing to ensure that previously working features remain intact.

Characteristics of Good Test Cases:

Reproducibility: Test cases should be repeatable to consistently verify the same behavior.

Independence: Test cases should not depend on each other, ensuring isolated testing.

Comprehensiveness: Test cases should cover a wide range of scenarios and edge cases.

Clarity: Test cases should be easy to understand and follow.

Traceability: Test cases should be linked to specific requirements or user stories.

Maintainability: Test cases should be easy to update when the software changes.

Test Case Design Techniques:

Equivalence Partitioning: Dividing input data into equivalent classes to test representative values from each class.
Boundary Value Analysis: Testing values at the boundaries of input ranges.

Decision Table Testing: Creating tables to represent combinations of inputs and expected outcomes.

State Transition Testing: Examining how the system transitions between states.

Use Case Testing: Testing scenarios based on user interactions and system responses.

Components of a Test Case:

Test Case ID: A unique identifier for each test case.
Test Case Name/Description: A brief description of what the test is intended to achieve.
Preconditions: Any necessary conditions or prerequisites for the test.
Input Data: The data or parameters to be used in the test.
Expected Results: The anticipated outcomes or behavior after executing the test.
Test Steps: A detailed sequence of actions to execute the test.
Post-conditions: Any conditions or state expected after the test execution.

Test Case Execution:

- Test cases are executed systematically by following the defined steps and providing the input data.
- Actual results are recorded during execution.
- Actual results are compared with expected results to determine if the test case passed or failed.
- Any discrepancies are reported as defects.

Test Case Management:

Test cases should be organized, prioritized, and stored in a test case repository.
Test case management tools help track the status of test cases and their execution.

Regression Testing: After modifications or updates to the software, previously executed test cases are re-run to ensure that existing functionality remains intact.

Continuous Improvement: Test cases and testing processes should be continually reviewed and improved to enhance test coverage and effectiveness.

Documentation and Reporting: Detailed records of test cases, their execution, and defect reports should be maintained for traceability and analysis.

Output:

Sr. No	Test Steps	Test Data	Expected Result	Actual Result	Status
1	Visit Website	-	Login Dialog	Login Dialog	Pass
2	User login using valid credentials	Github/Google Account Credentials	Login Successfully	Login Successfully	Pass
3	User login using invalid credentials	Wrong Credentials	Login Failed	Login Failed	Pass
4	User Selects categories	-	Category Page	Category Page	Pass
5	User Selects Settings	-	Settings Page	Settings Page	Pass
6	User Creates Store	Store name	Store created successfully	Store created successfully	Pass
7	User Creates Category	Category name and Billboard	Category created successfully	Category created successfully	Pass
8	User fails to enter store name to create it	-	Store creation failed	Store creation failed	Pass
9	User fails to select billboard while creating category	-	Category creation failed	Category creation failed	Pass
10	Change store's main billboard	Go to Settings, select main billboard	Main Billboard changed	Main Billboard changed	Pass
11	Rename store	Go to Settings, enter a new name.	Store name changed successfully	Store name changed successfully	Pass

Conclusion: With the help of this practical, I have Successfully implemented test case for Multi-Store CMS

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 2</td></tr>
<tr><td colspan="4">Title of LAB Assignment : Implement Web Drivers on Chrome & Firefox Browsers</td></tr>
<tr><td colspan="2">DOP : 15/09/2023</td><td colspan="2">DOS : 22/09/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 2

Aim: Implement Web Drivers on Chrome & Firefox Browsers.

Theory:

WebDriver

WebDriver drives a browser natively, as a user would, either locally or on a remote machine using the Selenium server, marks a leap forward in terms of browser automation. Selenium WebDriver refers to both the language bindings and the implementations of the individual browser controlling code. This is commonly referred to as just WebDriver.

Selenium WebDriver is a W3C Recommendation

- WebDriver is designed as a simple and more concise programming interface.
- WebDriver is a compact object-oriented API.
- It drives the browser effectively.

Download URLs of Drivers/Libs:

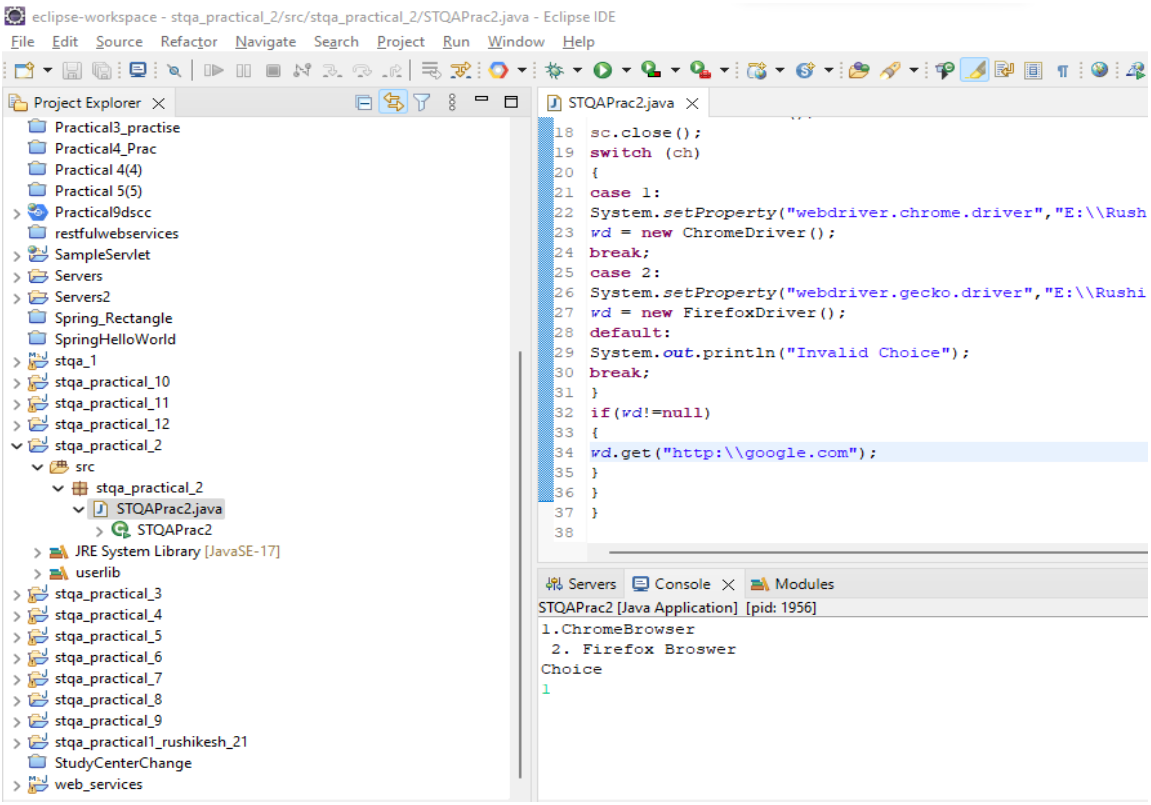
- **Selenium Driver:**https://www.selenium.dev/downloads/
- **ChromeDriver:**https://googlechromelabs.github.io/chrome-for-testing/
- **Geckodriver:**https://github.com/mozilla/geckodriver/releases

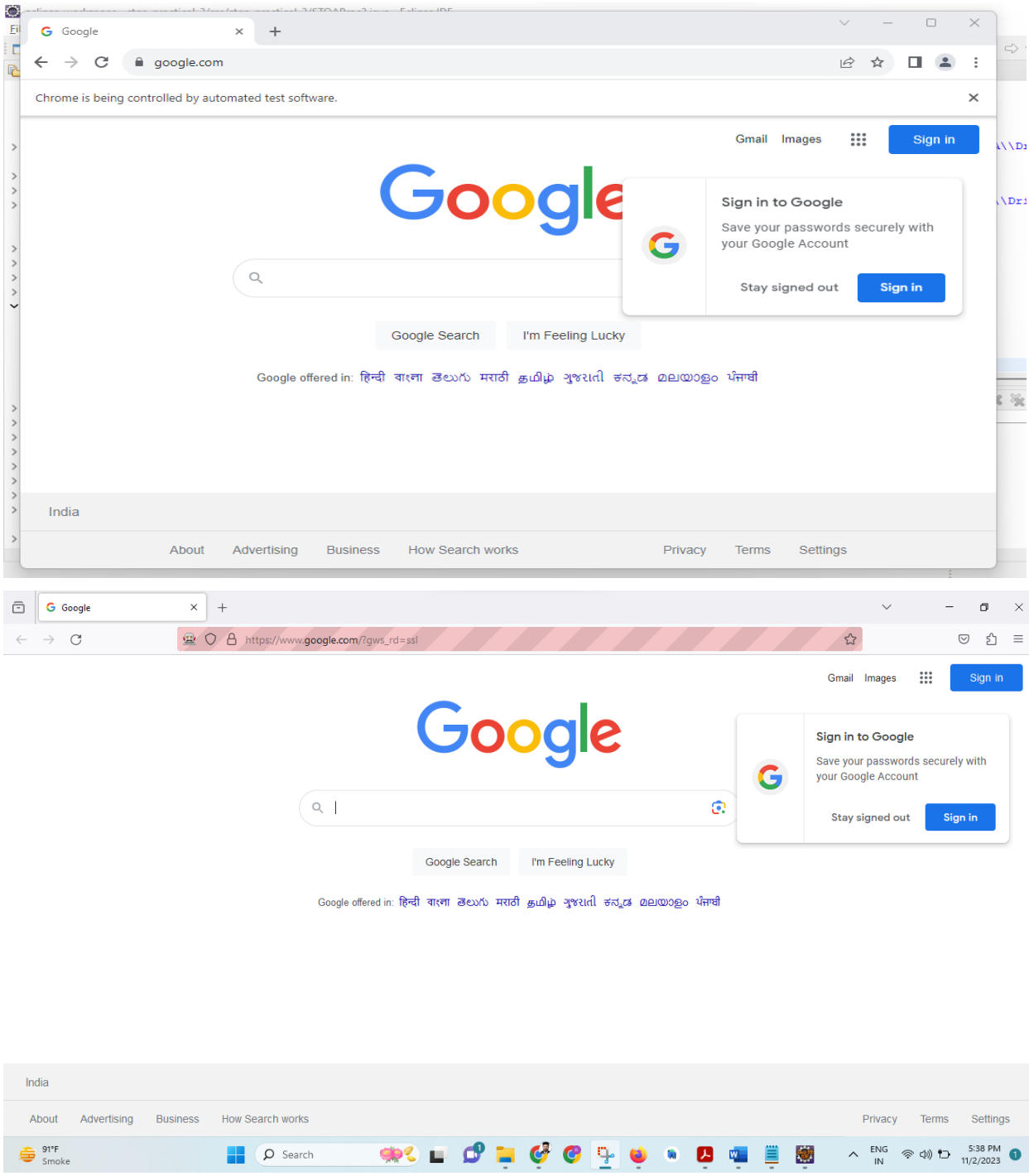
Source Code:

```
package stqa_practical_2;

import java.util.Scanner;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class STQAPrac2 {

	static WebDriver wd;

	public static void main(String[] args) {
		System.setProperty("webdriver.chrome.driver",

"D:\\Newfolder\\chromedriver-win64\\chromedriver-win64\\chromedriver.exe");
```

```
		Scanner sc = new Scanner(System.in);
		System.out.println("1.ChromeBrowser \n 2. Firefox Broswer");
		System.out.println("Choice");
		int ch = sc.nextInt();
		sc.close();
		switch (ch) {
		case 1:
			System.setProperty("webdriver.chrome.driver",

"E:\\PRANAV_MCA_SEM_3\\STQA\\Drivers\\chromedriver-win64\\chromedriver-win64\\chro
medriver.exe");
			wd = new ChromeDriver();
			break;
		case 2:
			System.setProperty("webdriver.gecko.driver",

"E:\\Rushikesh_MCA_SEM_3\\STQA\\Drivers\\geckodriver-v0.33.0-win64\\geckodriver.exe");
			wd = new FirefoxDriver();
		default:
			System.out.println("Invalid Choice");
			break;
		}
		if (wd != null) {
			wd.get("http:\\google.com");
		}
	}
}
```

OUTPUT:


eclipse-workspace - stqa_practical_2/src/stqa_practical_2/STQAPrac2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Practical3_practise
Practical4_Prac
Practical 4(4)
Practical 5(5)
Practical9dscc
restfulwebservices
SampleServlet
Servers
Servers2
Spring_Rectangle
SpringHelloWorld
stqa_1
stqa_practical_10
stqa_practical_11
stqa_practical_12
stqa_practical_2
src
stqa_practical_2
STQAPrac2.java
STQAPrac2
JRE System Library [JavaSE-17]
userlib
stqa_practical_3
stqa_practical_4
stqa_practical_5
stqa_practical_6
stqa_practical_7
stqa_practical_8
stqa_practical_9
stqa_practical1_rushikesh_21
StudyCenterChange
web_services
STQAPrac2.java
18 sc.close();
19 switch (ch)
20 {
21 case 1:
22 System.setProperty("webdriver.chrome.driver","E:\\Rush
23 wd = new ChromeDriver();
24 break;
25 case 2:
26 System.setProperty("webdriver.gecko.driver","E:\\Rushi
27 wd = new FirefoxDriver();
28 default:
29 System.out.println("Invalid Choice");
30 break;
31 }
32 if(wd!=null)
33 {
34 wd.get("http:\\google.com");
35 }
36 }
37 }
38
Servers Console Modules
STQAPrac2 [Java Application] [pid: 1956]
1.ChromeBrowser
2. Firefox Broswer
Choice
1



Conclusion: With the help of this Practical, I have successfully implemented Web Drivers on Chrome & Firefox Browsers are implemented.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 3</td></tr>
<tr><td colspan="4">Title of LAB Assignment : Demonstrate Handling Multiple Frames in Selenium</td></tr>
<tr><td colspan="2">DOP : 22/09/2023</td><td colspan="2">DOS : 29/09/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 03

Aim: Demonstrate Handling Multiple Frames in Selenium

Theory:

Theory:

iFrame in Selenium Webdriver

iFrame in Selenium Webdriver is a web page or an inline frame which is embedded in another web page or an HTML document embedded inside another HTML document. The iframe is often used to add content from other sources like an advertisement into a web page. The iframe is defined with the <iframe> tag.

We can identify the frames in Selenium using methods given below:

● Right click on the element, If you find the option like 'This Frame' then it is an iframe.(Please refer the below steps)

● Right click on the page and click 'View Page Source' and Search with the 'iframe', if you can find any tag name with the 'iframe' then it means to say the page consists of an iframe.

In the above diagram, you can see that '**This Frame**' option is available upon right clicking, so we are now sure that it is an iframe.

We can even identify the total number of iframes by using the below code:

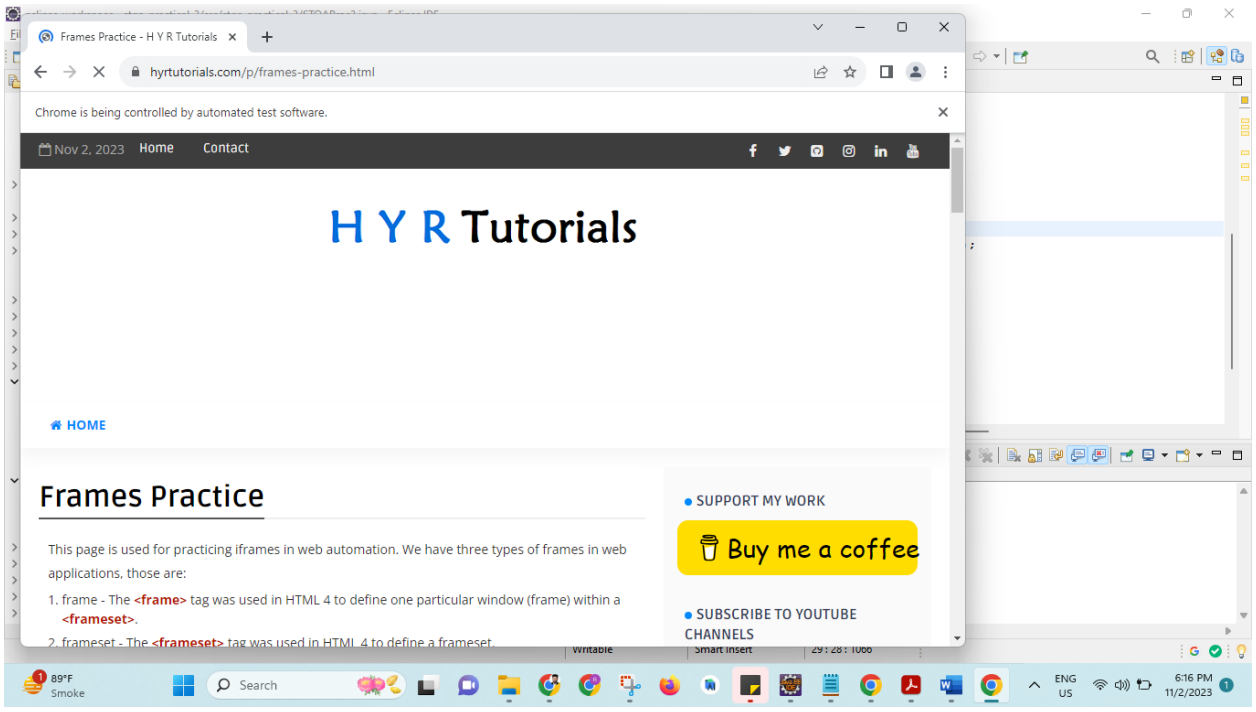
Int size = driver.findElements(By.tagName("iframe")).size();

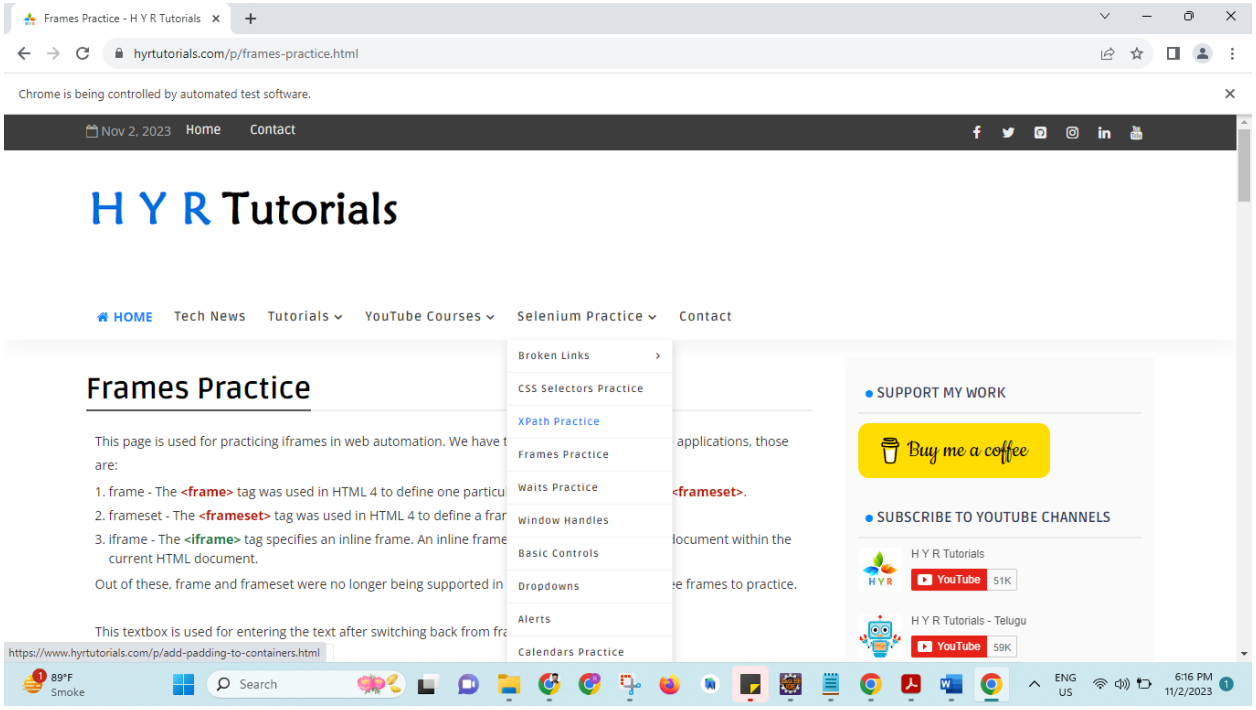
Code:

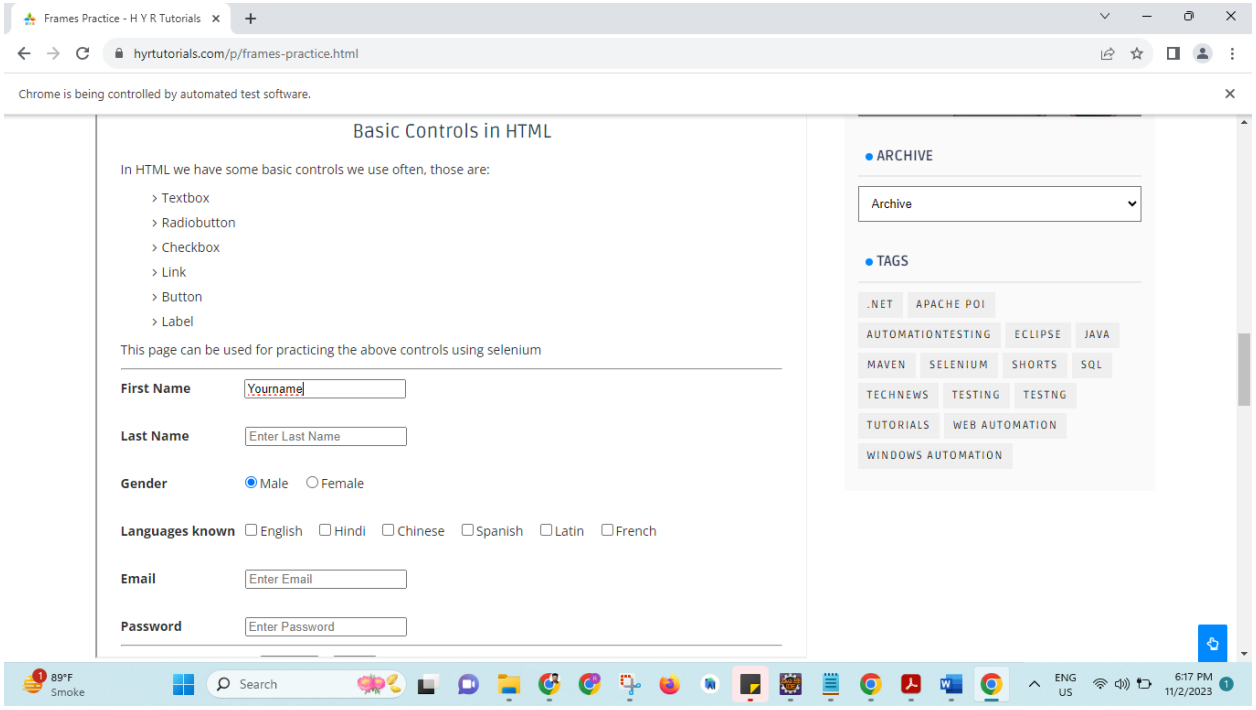
```
package stqa_practical_3;
import java.util.List;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.Select;
public class STQAPrac3 {
static WebDriver driver;
public static void main(String[] args) throws InterruptedException {
System.setProperty("webdriver.chrome.driver","E:\\PRANAV_MCA_SEM_3\\STQA\\Drive
rs\\chromedriver-win64\\chromedriver-win64\\chromedriver.exe");
driver = new ChromeDriver();
driver.get("https://www.hyrtutorials.com/p/frames-practice.html");
```

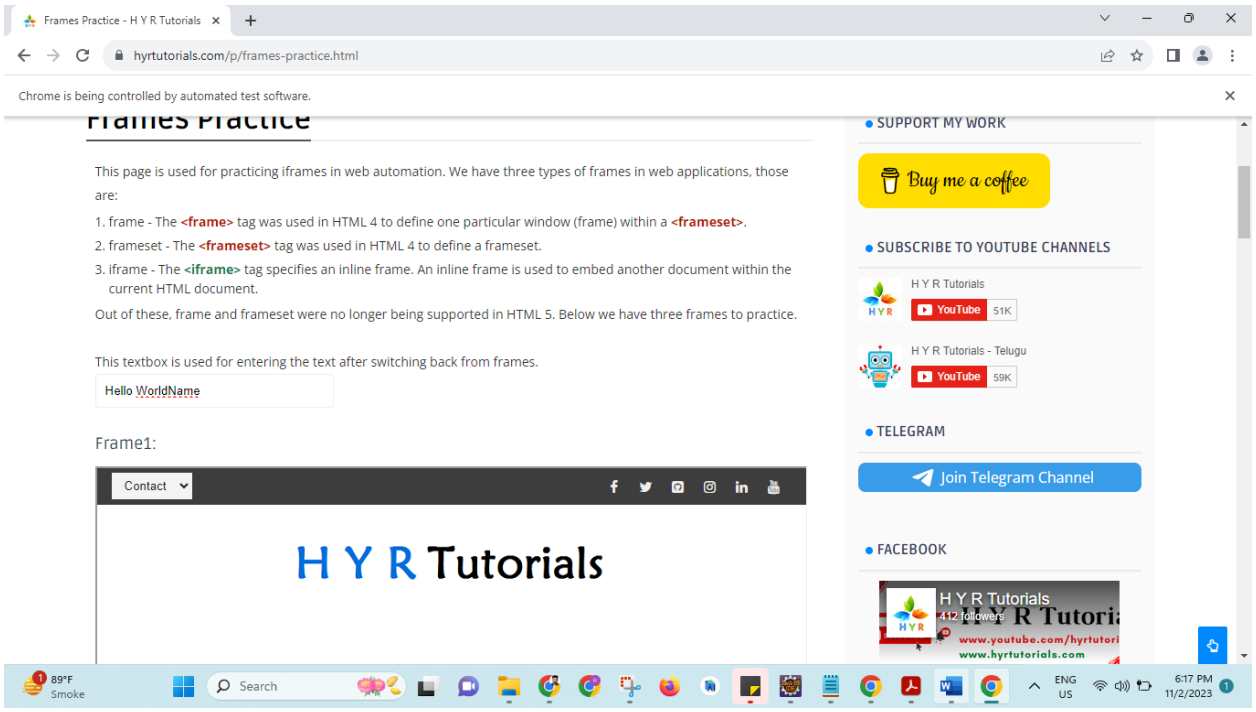
```
// navigates to the page consisting an iframe
driver.manage().window().maximize();
Thread.sleep(3000);
driver.findElement(By.id("name")).sendKeys("Hello");
driver.switchTo().frame(driver.findElement(By.id("frm1")));
Thread.sleep(3000);
Select courseDD=new Select(driver.findElement(By.id("selectnav1")));
courseDD.selectByVisibleText("Contact");
driver.switchTo().defaultContent();
Thread.sleep(3000);
driver.findElement(By.id("name")).sendKeys(" World");
driver.switchTo().frame(driver.findElement(By.id("frm2")));
driver.findElement(By.id("firstName")).sendKeys("Yourname");
driver.switchTo().defaultContent();
Thread.sleep(3000);
driver.findElement(By.id("name")).sendKeys("Name");
}
}
```

OUTPUT:




Frames Practice - H Y R Tutorials
hyrtutorials.com/p/frames-practice.html
Chrome is being controlled by automated test software.
Nov 2, 2023 Home Contact
H Y R Tutorials
HOME Tech News Tutorials YouTube Courses Selenium Practice Contact
Broken Links
CSS Selectors Practice
XPath Practice
Frames Practice
Waits Practice
Window Handles
Basic Controls
Dropdowns
Alerts
Calendars Practice
Frames Practice
This page is used for practicing iframes in web automation. We have t applications, those are:
1. frame - The <frame> tag was used in HTML 4 to define one particu <frameset>.
2. frameset - The <frameset> tag was used in HTML 4 to define a frar
3. iframe - The <iframe> tag specifies an inline frame. An inline frame document within the current HTML document.
Out of these, frame and frameset were no longer being supported in e frames to practice.
This textbox is used for entering the text after switching back from fra
SUPPORT MY WORK
Buy me a coffee
SUBSCRIBE TO YOUTUBE CHANNELS
H Y R Tutorials
YouTube 51K
H Y R Tutorials - Telugu
YouTube 59K
https://www.hyrtutorials.com/p/add-padding-to-containers.html


Frames Practice - H Y R Tutorials
hyrtutorials.com/p/frames-practice.html
Chrome is being controlled by automated test software.
Basic Controls in HTML
In HTML we have some basic controls we use often, those are:
› Textbox
› Radiobutton
› Checkbox
› Link
› Button
› Label
This page can be used for practicing the above controls using selenium
First Name Yourname
Last Name Enter Last Name
Gender Male Female
Languages known English Hindi Chinese Spanish Latin French
Email Enter Email
Password Enter Password
ARCHIVE
Archive
TAGS
.NET APACHE POI
AUTOMATIONTESTING ECLIPSE JAVA
MAVEN SELENIUM SHORTS SQL
TECHNEWS TESTING TESTNG
TUTORIALS WEB AUTOMATION
WINDOWS AUTOMATION



Conclusion: Demonstration of multiple frames has been implemented using selenium.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 4</td></tr>
<tr><td colspan="4">Title of LAB Assignment : Implement Browser command and navigation Commands.</td></tr>
<tr><td colspan="2">DOP : 22/09/2023</td><td colspan="2">DOS : 29/09/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 04

AIM: Implement Browser command and navigation Commands.

Theory:

Selenium WebDriver - Browser Commands

The very basic browser operations of WebDriver include opening a browser; perform few tasks and then closing the browser.
Given are some of the most commonly used Browser commands for Selenium WebDriver.

1. Get Command

Method:
get(String arg0) : **void**
In WebDriver, this method loads a new web page in the existing browser window. It accepts *String* as parameter and returns
void. Implementation:

2.Get TitleCommand

Method:
getTitle(): String
In WebDriver, this method fetches the title of the current web page. It accepts no parameter and returns a String.
Implementation:
Altaf Chaudhry ROLL NO. 2 6

```
driver.getCurrentUrl();
OR
String CurrentUrl = driver.getCurrentUrl();
driver.close();
driver.getTitle();
OR
String Title = driver.getTitle();
```

3. Get Current URL Command

Method:
getCurrentUrl(): String
In WebDriver, this method fetches the string representing the Current URL of the current web page. It accepts nothing as parameter and returns a String value.
Implementation:

4.Close Command

Method: close():

void

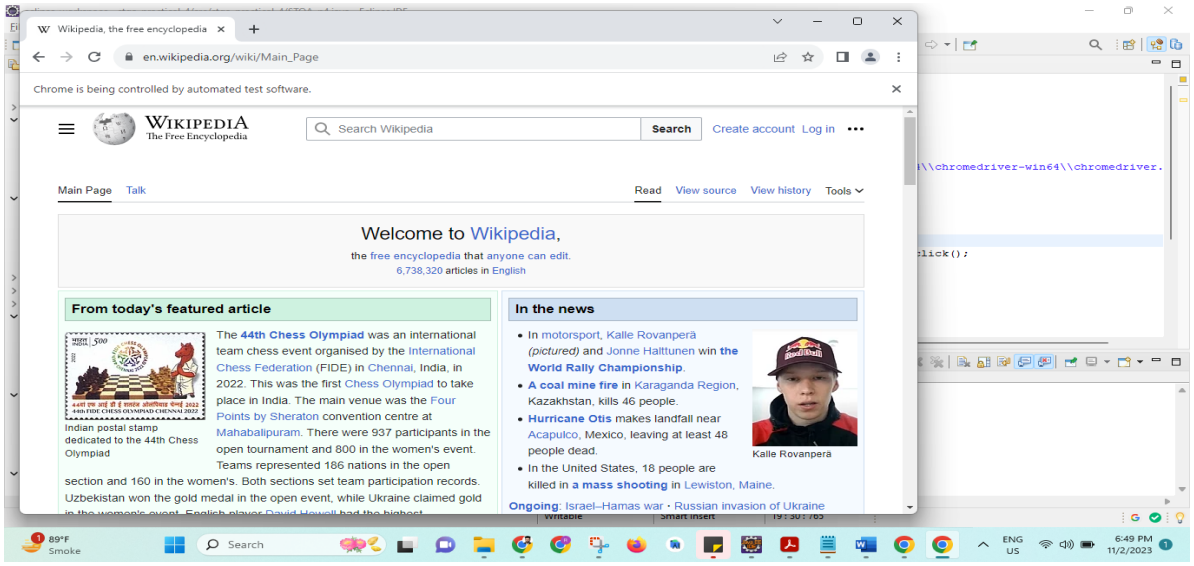
This method terminates the current browser window operating by WebDriver at the current time. If the current window is the only window operating by WebDriver, it terminates the browser as well. This method accepts nothing as parameter and returns *void*.

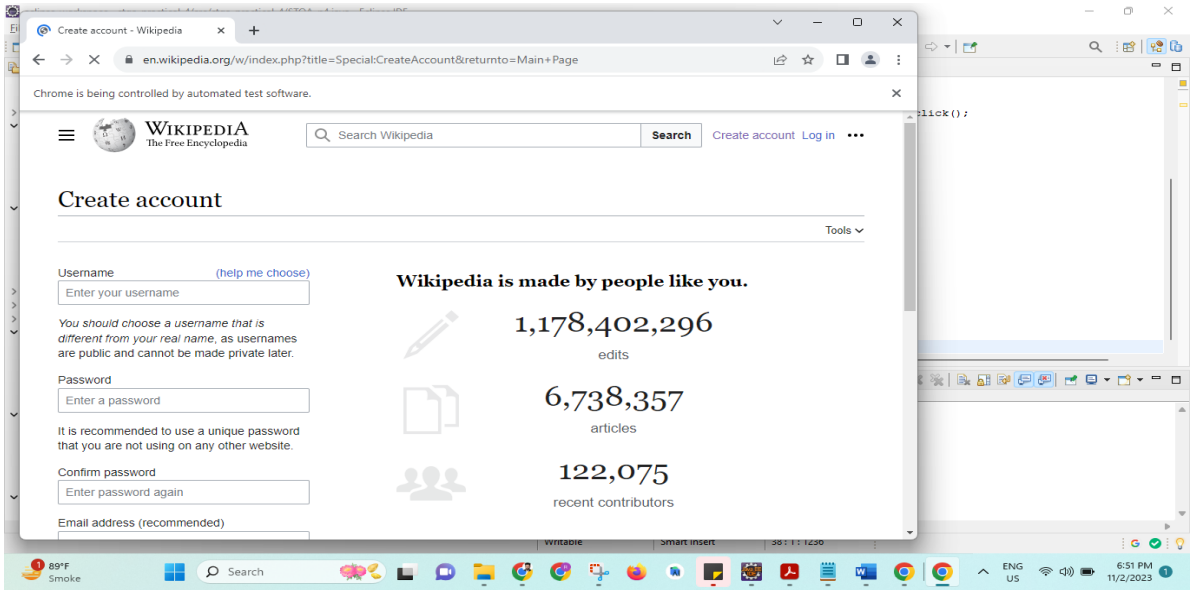
Implementation:

Source Code:

```
package stqa_practical_4;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.WebDriverWait;

public class STQA_p4 {
	public static void main(String[] args) throws InterruptedException {
		System.setProperty("webdriver.chrome.driver",

"E:\\PRANAV_MCA_SEM_3\\STQA\\Drivers\\chromedriver-win64\\chromedriver-win64\\c
hromedriver.exe");
		WebDriver driver = new ChromeDriver();
		driver = new ChromeDriver();
		String appUrl = "https://en.wikipedia.org/wiki/Main_Page";
		driver.get(appUrl);
		Thread.sleep(2000);
// Click on Registration link

driver.findElement(By.xpath("//*[@id=\"pt-createaccount-2\"]/a")).click();
		Thread.sleep(2000);
// Go back to Home Page
		driver.navigate().back();
		Thread.sleep(2000);
// Go forward to Registration page
		driver.navigate().forward();
		Thread.sleep(2000);
// Go back to Home page
		driver.navigate().to(appUrl);
		Thread.sleep(2000);
// Refresh browser
		driver.navigate().refresh();
		Thread.sleep(2000);
// Close browser
		driver.close();
	}
}
```

OUTPUT:


Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Main_Page
Chrome is being controlled by automated test software.
WIKIPEDIA
The Free Encyclopedia
Search Wikipedia
Search
Create account
Log in
Main Page
Talk
Read
View source
View history
Tools
Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
6,738,320 articles in English
From today's featured article
The 44th Chess Olympiad was an international team chess event organised by the International Chess Federation (FIDE) in Chennai, India, in 2022. This was the first Chess Olympiad to take place in India. The main venue was the Four Points by Sheraton convention centre at Mahabalipuram. There were 937 participants in the open tournament and 800 in the women's event. Teams represented 186 nations in the open section and 160 in the women's. Both sections set team participation records. Uzbekistan won the gold medal in the open event, while Ukraine claimed gold
Indian postal stamp dedicated to the 44th Chess Olympiad
In the news
In motorsport, Kalle Rovanperä (pictured) and Jonne Halttunen win the World Rally Championship.
A coal mine fire in Karaganda Region, Kazakhstan, kills 46 people.
Hurricane Otis makes landfall near Acapulco, Mexico, leaving at least 48 people dead.
In the United States, 18 people are killed in a mass shooting in Lewiston, Maine.
Kalle Rovanperä
Ongoing: Israel–Hamas war · Russian invasion of Ukraine
Search
ENG US
6:49 PM 11/2/2023


Create account - Wikipedia
en.wikipedia.org/w/index.php?title=Special:CreateAccount&returnto=Main+Page
Chrome is being controlled by automated test software.
WIKIPEDIA
The Free Encyclopedia
Search Wikipedia
Search
Create account
Log in
Create account
Tools
Username
(help me choose)
Enter your username
You should choose a username that is different from your real name, as usernames are public and cannot be made private later.
Password
Enter a password
It is recommended to use a unique password that you are not using on any other website.
Confirm password
Enter password again
Email address (recommended)
Wikipedia is made by people like you.
1,178,402,296
edits
6,738,357
articles
122,075
recent contributors
Search
ENG US
6:51 PM 11/2/2023

Conclusion:

Successfully Implemented Browser and navigation commands are implemented.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 5</td></tr>
<tr><td colspan="4">Title of LAB Assignment : Implement the find element command
.</td></tr>
<tr><td colspan="2">DOP : 22/09/2023</td><td colspan="2">DOS : 29/09/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 05

AIM: Implement the find element command

Theory:

FindElement:

Selenium Find Element command takes in the By object as the parameter and returns an object of type list WebElement in Selenium. By object in turn can be used with various locator strategies such as find element by ID Selenium, Name, Class Name, XPATH etc.
Below is the syntax of FindElement command in Selenium web driver:
Locator
Strategy can be any of the following values.

- ID
- Selenium find element by Name
- Class Name
- Tag Name
- Link Text
- Partial Link Text
- XPATH

Locator Value is the unique value using which a web element can be identified. It is the responsibility of developers and testers to make sure that web elements are uniquely identifiable using certain properties such as ID or name.

FindElements:

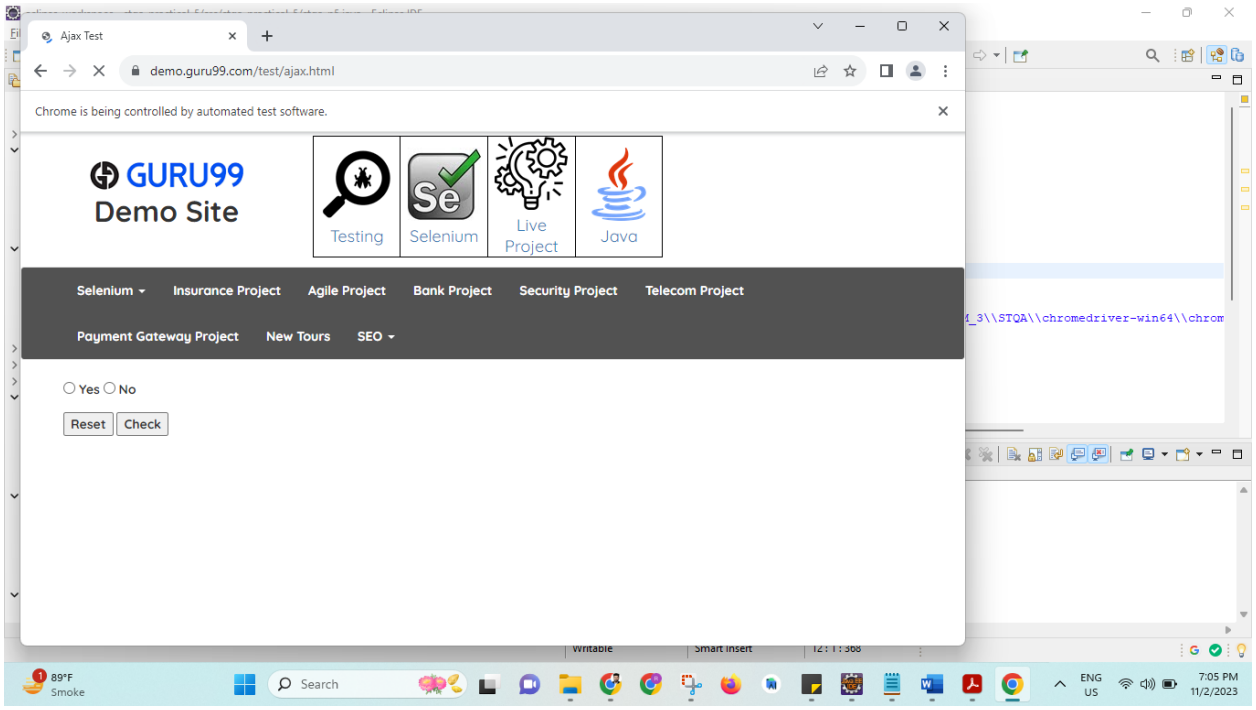
FindElements in Selenium command takes in By object as the parameter and returns a list of web elements. It returns an empty list if there are no elements found using the given locator strategy and locator value.
Below is the syntax of find elements command.

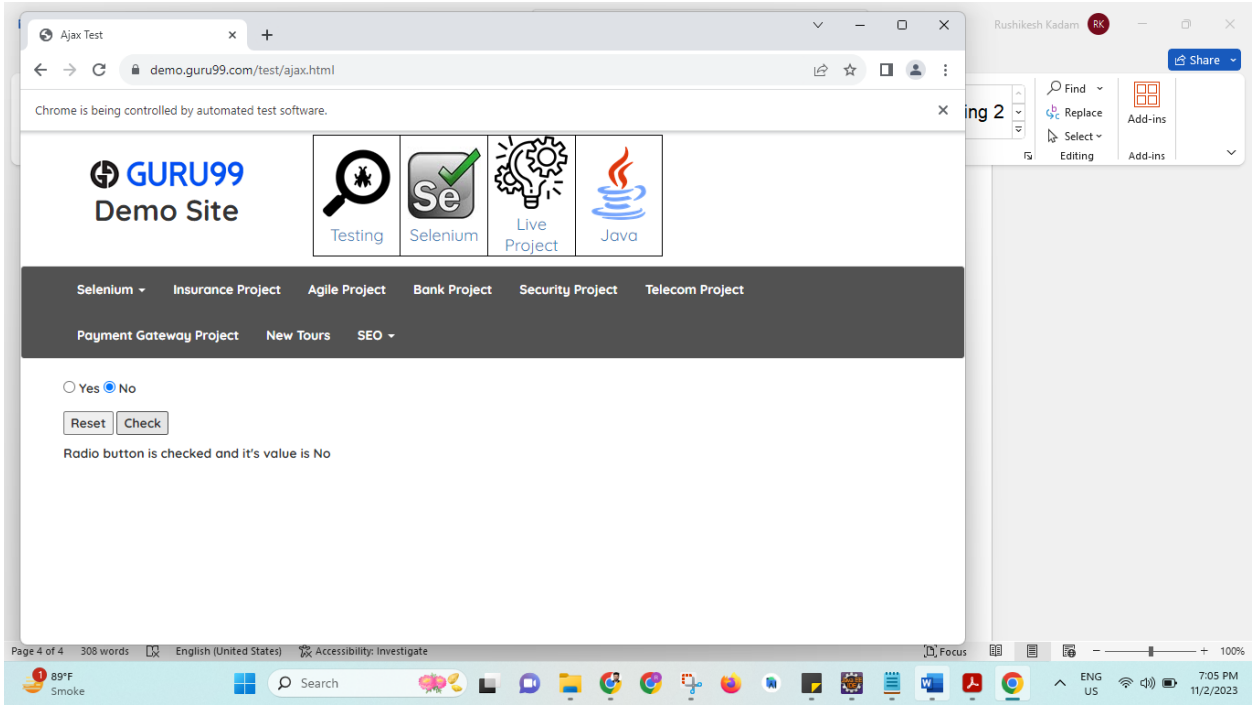
```
List<WebElement> elementName =
driver.findElements(By.LocatorStrategy("LocatorValue"));
```

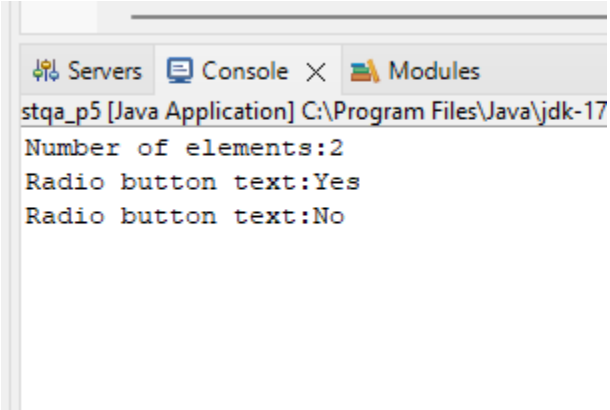
SOURCE CODE:

```
package stqa_practical_5;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
public class stqa_p5 {
public static void main(String[] args) throws InterruptedException {
System.setProperty("webdriver.chrome.driver","E:\\PRANAV_MCA_SEM_3\\STQA\\chrom
edriver-win64\\chromedriver-win64\\chromedriver.exe");
WebDriver wd=new ChromeDriver();
wd.get("http://demo.guru99.com/test/ajax.html");
Thread.sleep(2000);
wd.findElement(By.id("no")).click();
Thread.sleep(2000);
wd.findElement(By.id("buttoncheck")).click();
Thread.sleep(2000);
List<WebElement> elements = wd.findElements(By.name("name"));
System.out.println("Number of elements:" +elements.size());
Thread.sleep(2000);
for (int i=0; i<elements.size();i++){
System.out.println("Radio button text:" +
elements.get(i).getAttribute("value"));
Thread.sleep(2000);
}
}
}
```

OUTPUT:


Ajax Test
demo.guru99.com/test/ajax.html
Chrome is being controlled by automated test software.
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
Yes
No
Reset
Check


Ajax Test
demo.guru99.com/test/ajax.html
Chrome is being controlled by automated test software.
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
Yes
No
Reset
Check
Radio button is checked and it's value is No



Conclusion: Find element command has been successfully implemented.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 6</td></tr>
<tr><td colspan="4">Title of LAB Assignment :Demonstrate the Locator(id,css selector, path)</td></tr>
<tr><td colspan="2">DOP : 29/09/2023</td><td colspan="2">DOS : 06/10/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 06

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

Theory:

What are Locators?

Locator is a command that tells Selenium IDE which GUI elements (say Text Box, Buttons, Check Boxes etc) its needs to operate on. Identification of correct GUI elements is a prerequisite to creating an automation script. But accurate identification of GUI elements is more difficult than it sounds. Sometimes, you end up working with incorrect GUI elements or no elements at all! Hence, Selenium provides a number of Locators to precisely locate a GUI element

Locator Strategy can be any of the following values.

- ID
- Selenium find element by Name
- Class Name
- Tag Name
- Link Text
- Partial Link Text
- XPATH

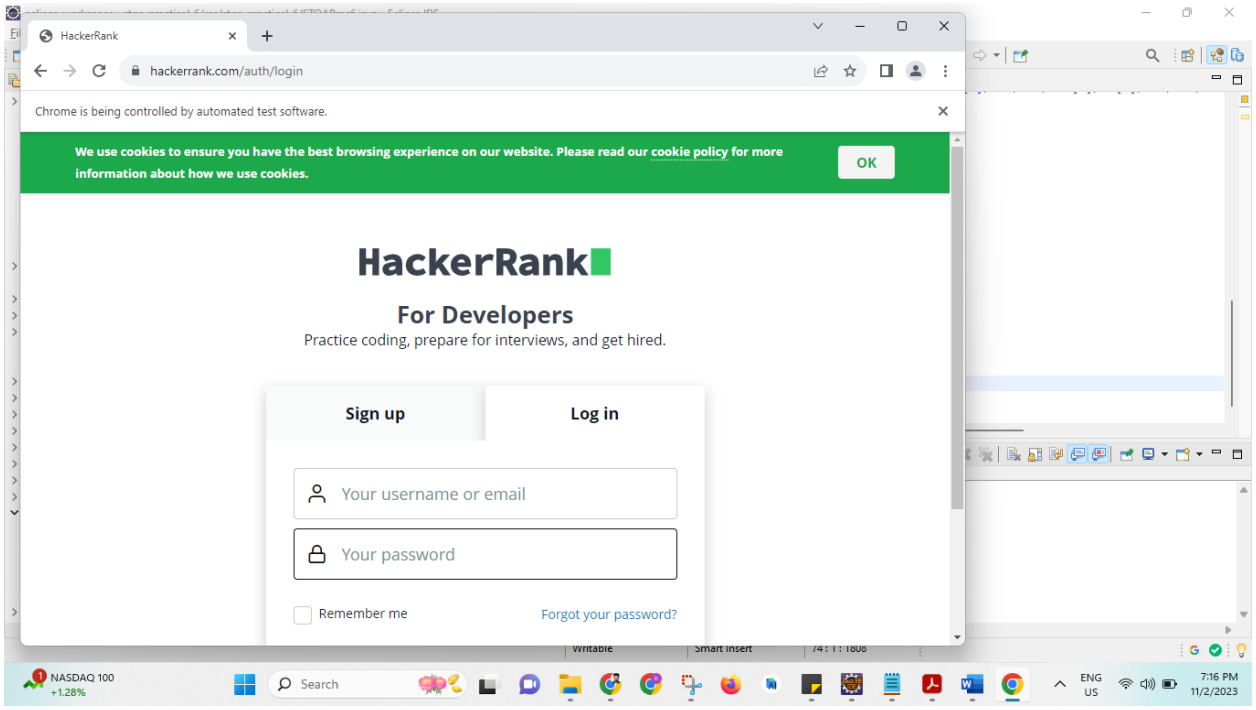
SOURCE CODE:

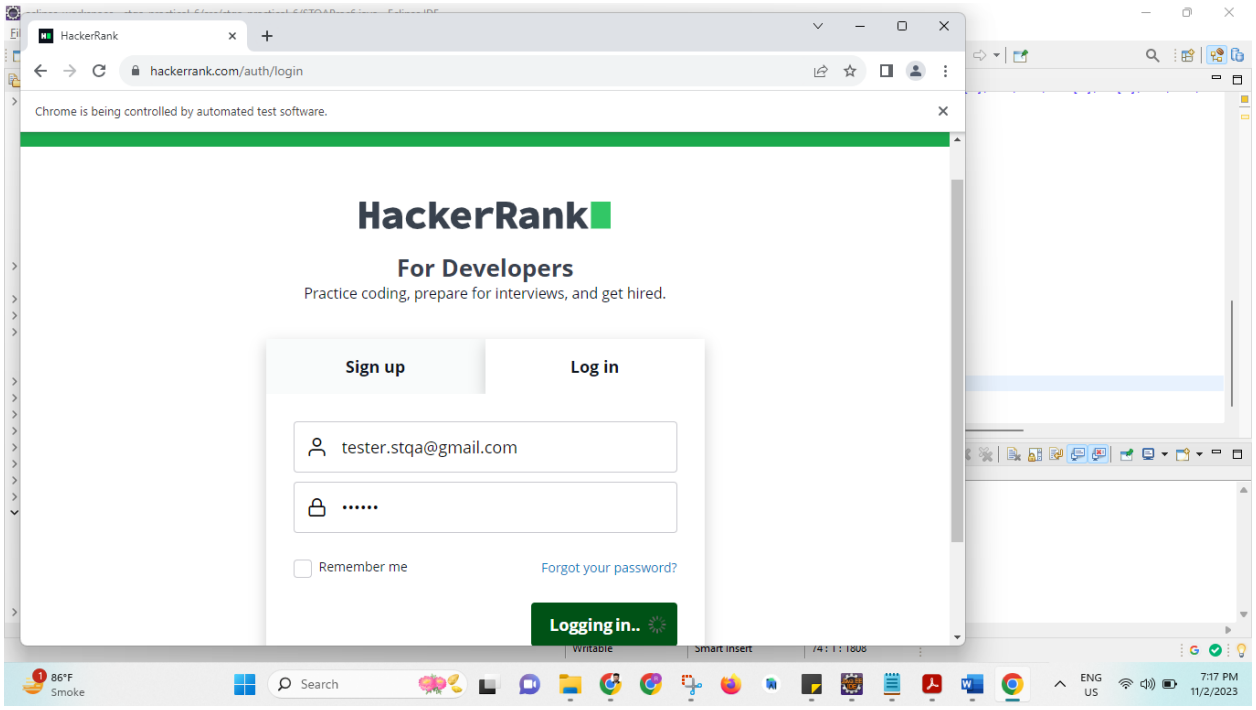
```
package stqa_practical_6;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
```

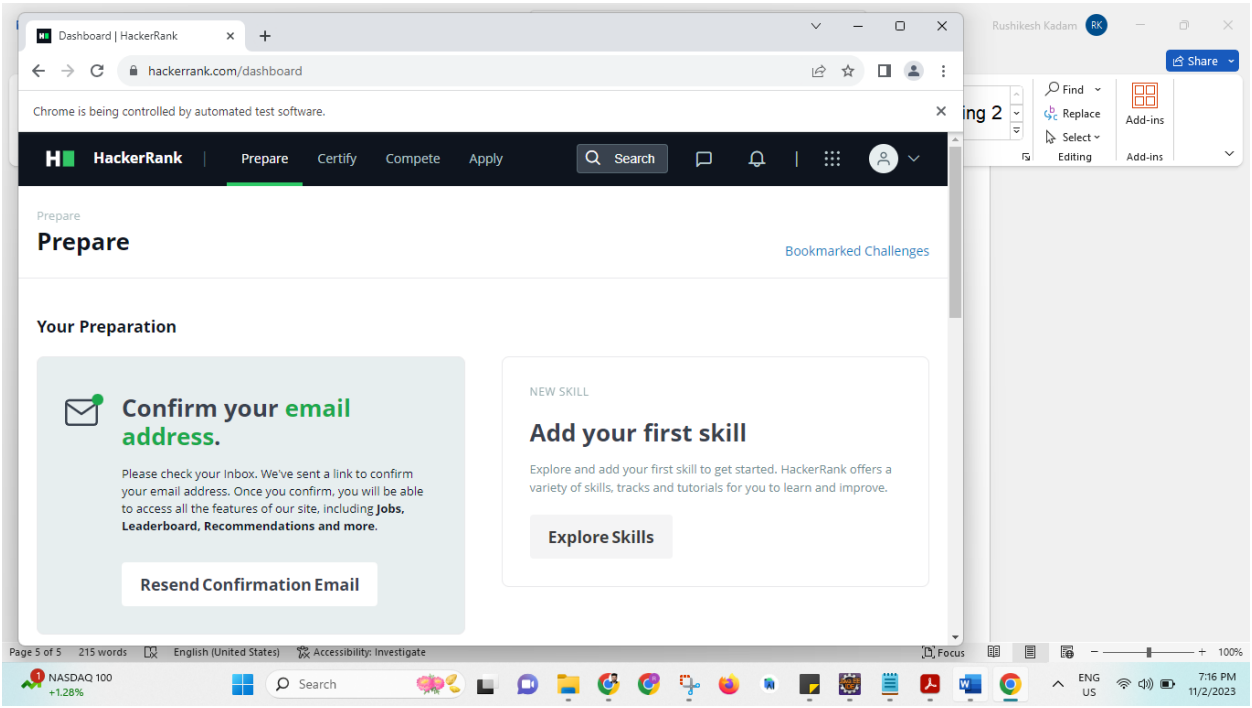
```
import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.WebDriverWait;

public class STQAPrac6 {
	public static void main(String[] args) throws InterruptedException {


System.setProperty("webdriver.chrome.driver","E:\\PRANAV_MCA_SEM_3\\STQA\\chromedrive
r-win64\\chromedriver-win64\\chromedriver.exe");

		WebDriver driver = new ChromeDriver();

		String url = "https://www.hackerrank.com/auth/login";

		driver.get(url);

		Thread.sleep(1000);

		System.out.println("HackerRank opened");

		driver.findElement(By.name("username")).sendKeys("tester.stqa@gmail.com");

		driver.findElement(By.cssSelector("input[type=password]")).sendKeys("tester");

		driver.findElement(By.cssSelector("button.auth-button")).click();

		System.out.println("Credentials Entered");

		Thread.sleep(6000);


		driver.findElement(


By.xpath("/html/body/div[4]/div/div/div/div/div[1]/nav/div/div[2]/ul[1]/li/div/div/div/div/input"))
.sendKeys("Tower Breakers");

		Thread.sleep(6000);
```

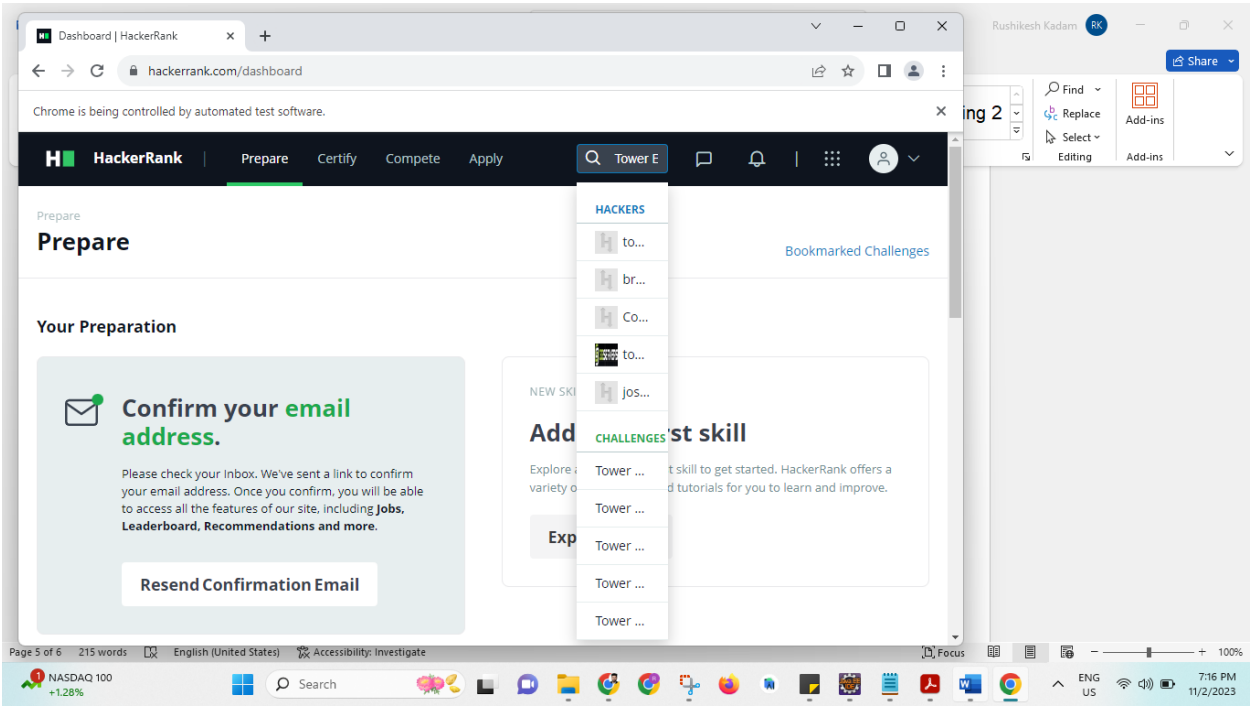
```
System.out.println("Question Searched");


driver.findElement(By.xpath("/html/body/div[4]/div/div/div/div/div[1]/nav/div/div[2]/ul[1]/
li/div/div/div/div[2]/ul/li[8]/div/div"))

.click();

Thread.sleep(6000);

driver.findElement(By.className("profile-menu")).click();

Thread.sleep(6000);

driver.findElement(By.className("logout-button")).click();

Thread.sleep(2000);

System.out.println("Logged out");

}

}
```

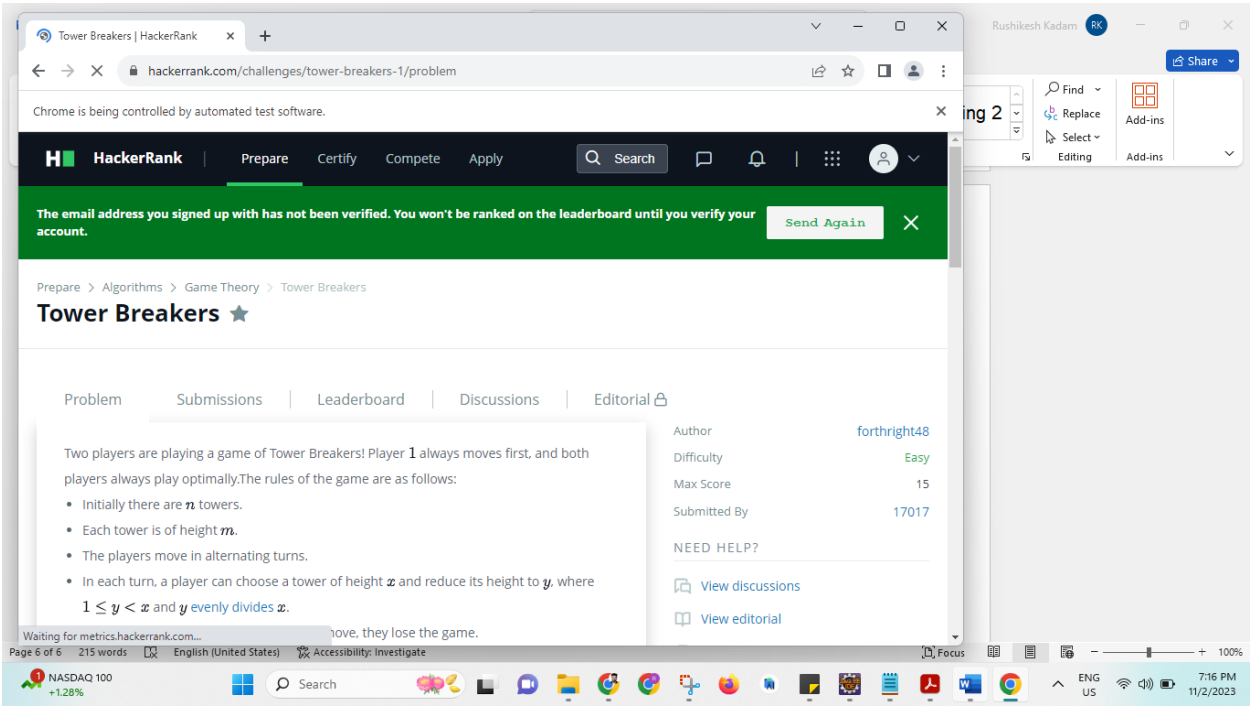
OUTPUT:

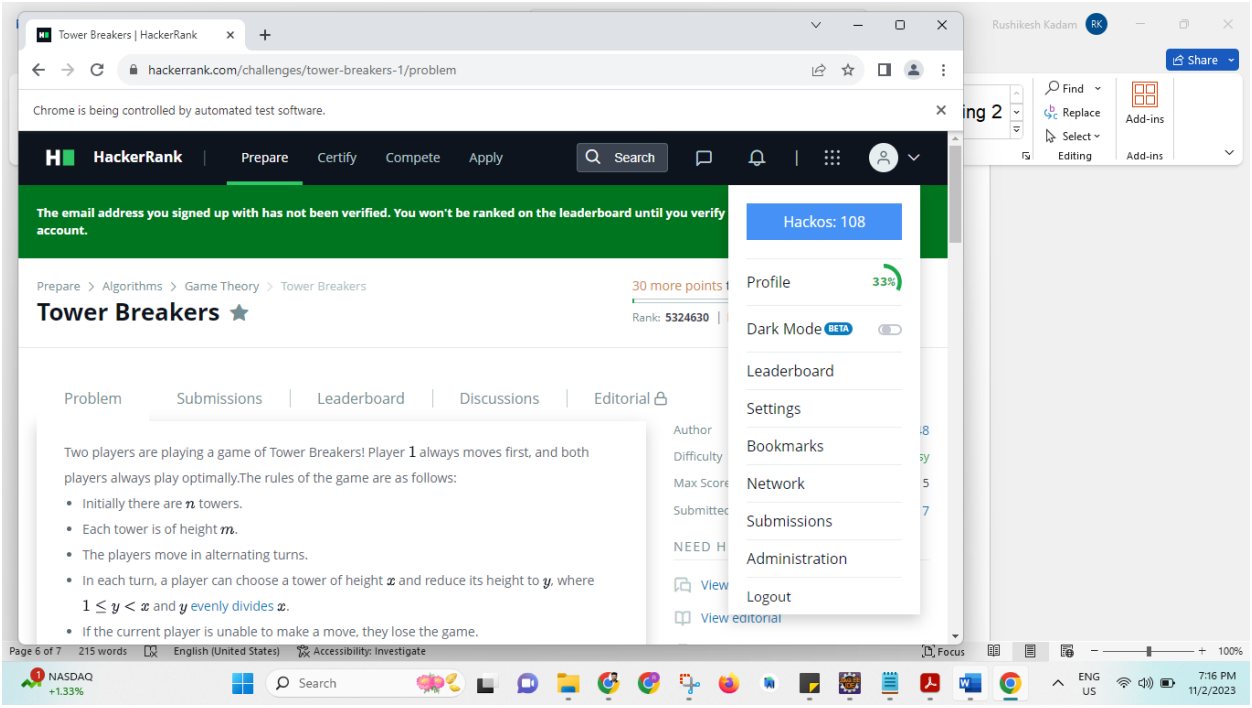

HackerRank
hackerrank.com/auth/login
Chrome is being controlled by automated test software.
We use cookies to ensure you have the best browsing experience on our website. Please read our cookie policy for more information about how we use cookies.
OK
HackerRank
For Developers
Practice coding, prepare for interviews, and get hired.
Sign up
Log in
Your username or email
Your password
Remember me
Forgot your password?

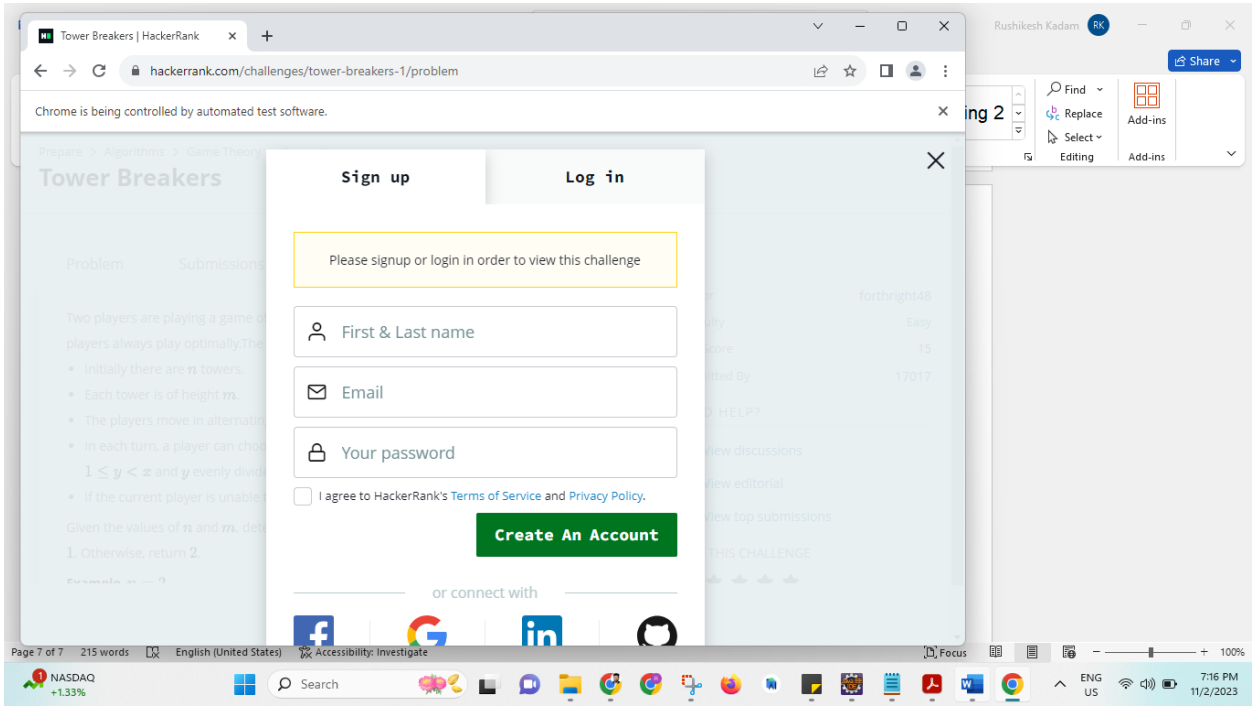

HackerRank
hackerrank.com/auth/login
Chrome is being controlled by automated test software.
HackerRank
For Developers
Practice coding, prepare for interviews, and get hired.
Sign up
Log in
tester.stqa@gmail.com
Remember me
Forgot your password?
Logging in..

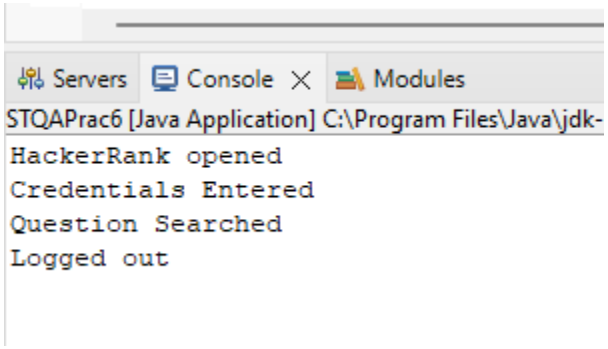

Dashboard | HackerRank
hackerrank.com/dashboard
Chrome is being controlled by automated test software.
HackerRank
Prepare
Certify
Compete
Apply
Search
Prepare
Prepare
Bookmarked Challenges
Your Preparation
Confirm your email address.
Please check your Inbox. We've sent a link to confirm your email address. Once you confirm, you will be able to access all the features of our site, including Jobs, Leaderboard, Recommendations and more.
Resend Confirmation Email
NEW SKILL
Add your first skill
Explore and add your first skill to get started. HackerRank offers a variety of skills, tracks and tutorials for you to learn and improve.
Explore Skills
Page 5 of 5
215 words
English (United States)
Accessibility: Investigate
Focus
100%
NASDAQ 100
+1.28%
Search
ENG
US
7:16 PM
11/2/2023


Dashboard | HackerRank
hackerrank.com/dashboard
Chrome is being controlled by automated test software.
HackerRank
Prepare
Certify
Compete
Apply
Tower E
HACKERS
CHALLENGES
Tower ...
Prepare
Prepare
Bookmarked Challenges
Your Preparation
Confirm your email address.
Please check your Inbox. We've sent a link to confirm your email address. Once you confirm, you will be able to access all the features of our site, including Jobs, Leaderboard, Recommendations and more.
Resend Confirmation Email
Page 5 of 6
215 words
English (United States)
Accessibility: Investigate
Focus
100%
NASDAQ 100
+1.28%
Search
ENG
US
7:16 PM
11/2/2023


Tower Breakers | HackerRank
hackerrank.com/challenges/tower-breakers-1/problem
Chrome is being controlled by automated test software.
HackerRank
Prepare
Certify
Compete
Apply
Search
The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify your account.
Send Again
Prepare > Algorithms > Game Theory > Tower Breakers
Tower Breakers
Problem
Submissions
Leaderboard
Discussions
Editorial
Two players are playing a game of Tower Breakers! Player 1 always moves first, and both players always play optimally.The rules of the game are as follows:
Initially there are n towers.
Each tower is of height m.
The players move in alternating turns.
In each turn, a player can choose a tower of height x and reduce its height to y, where 1 ≤ y < x and y evenly divides x.
Author forthright48
Difficulty Easy
Max Score 15
Submitted By 17017
NEED HELP?
View discussions
View editorial


Tower Breakers | HackerRank
hackerrank.com/challenges/tower-breakers-1/problem
Chrome is being controlled by automated test software.
HackerRank
Prepare
Certify
Compete
Apply
Search
The email address you signed up with has not been verified. You won't be ranked on the leaderboard until you verify account.
Hackos: 108
Profile 33%
Dark Mode BETA
Leaderboard
Settings
Bookmarks
Network
Submissions
Administration
Logout
Prepare > Algorithms > Game Theory > Tower Breakers
Tower Breakers
Problem
Submissions
Leaderboard
Discussions
Editorial
Two players are playing a game of Tower Breakers! Player 1 always moves first, and both players always play optimally.The rules of the game are as follows:
Initially there are n towers.
Each tower is of height m.
The players move in alternating turns.
In each turn, a player can choose a tower of height x and reduce its height to y, where 1 ≤ y < x and y evenly divides x.
If the current player is unable to make a move, they lose the game.





Conclusion: Locator based on id, css selector and xpath has been demonstrated.

Name of Student : DESHMUKH PRANAV DINESH			
Roll Number : 12		**LAB Assignment Number:** 7	
Title of LAB Assignment :Demonstrate synchronization in selenium			
DOP : 06/10/2023		**DOS :** 13/10/2023	
CO Mapped : C02	**PO Mapped:** PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2	**Faculty Signature:**	**Marks :**

Practical No: 07

Theory:

To synchronize between script execution and application, we need to wait after performing appropriate actions. Let us look at the ways to achieve the same.

Thread.Sleep

Thread.Sleep is a static wait and it is not a good way to use in scripts as it is sleep without condition.

```
Thread.Sleep(1000); //Will wait for 1 second.
```

Explicit Waits

An 'explicit wait,' waits for a certain condition to occur before proceeding further. It is
mainly used when we want to click or act on an object once it is visible.

```
WebDriver driver = new FirefoxDriver();
driver.get("Enter an URL"S);
WebElement DynamicElement =
(new WebDriverWait(driver,
10)).until(ExpectedConditions.presenceOfElementLocated(By.id("Dyn
a micElement")));
```

Implicit Wait

Implicit wait is used in cases where the WebDriver cannot locate an object immediately because of its unavailability. The WebDriver will wait for a specified implicit wait time and it will not try to find the element again during the specified time
period.
Once the specified time limit is crossed, the webDriver will try to search the element
once again for one last time. Upon success, it proceeds with the execution; upon failure, it throws exception.
It is a kind of global wait which means the wait is applicable for the entire driver. Hence, hardcoding this wait for longer time periods will hamper the execution time.

```
WebDriver driver = new FirefoxDriver();
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
driver.get("Enter an URL");
```

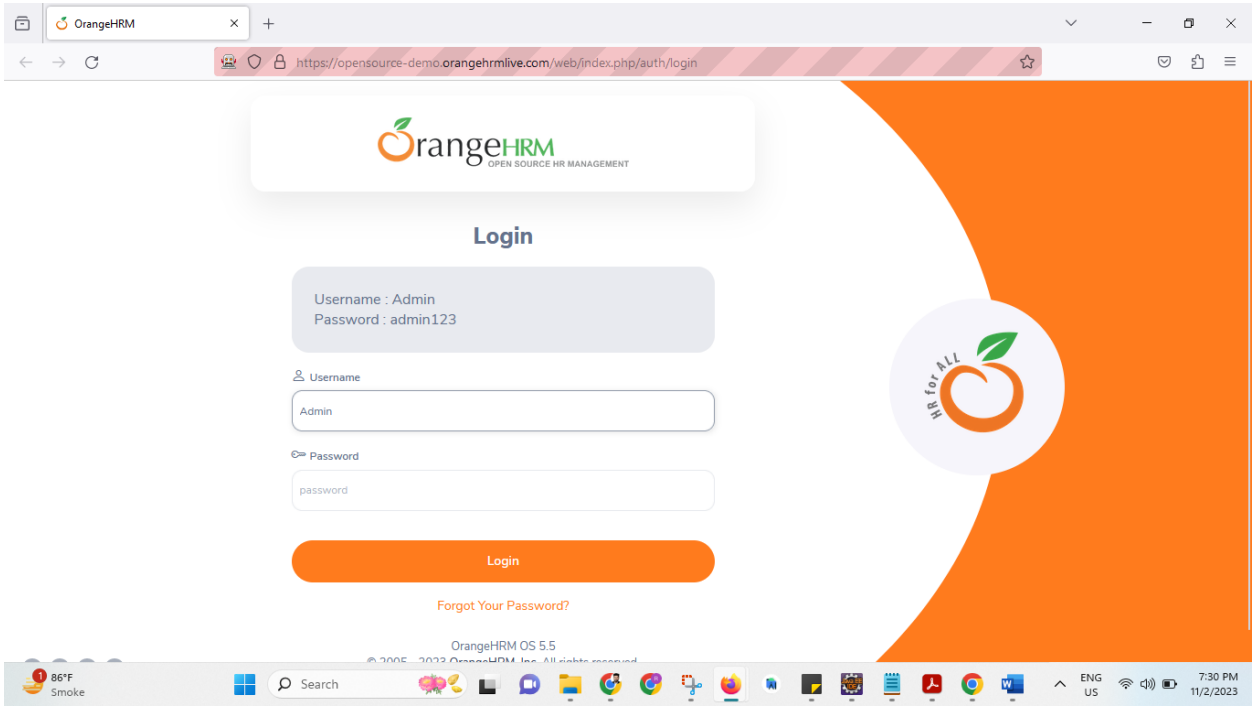
PRACTICAL 7A : IMPLICIT

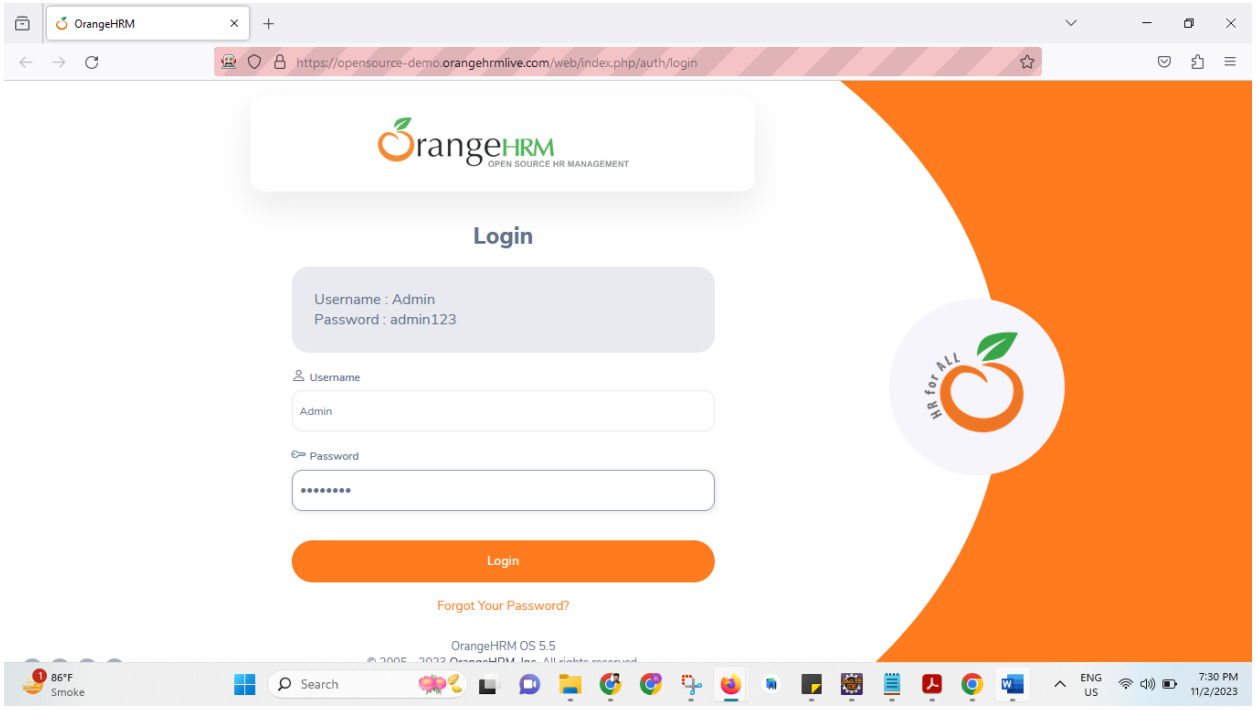
SOURCE CODE:

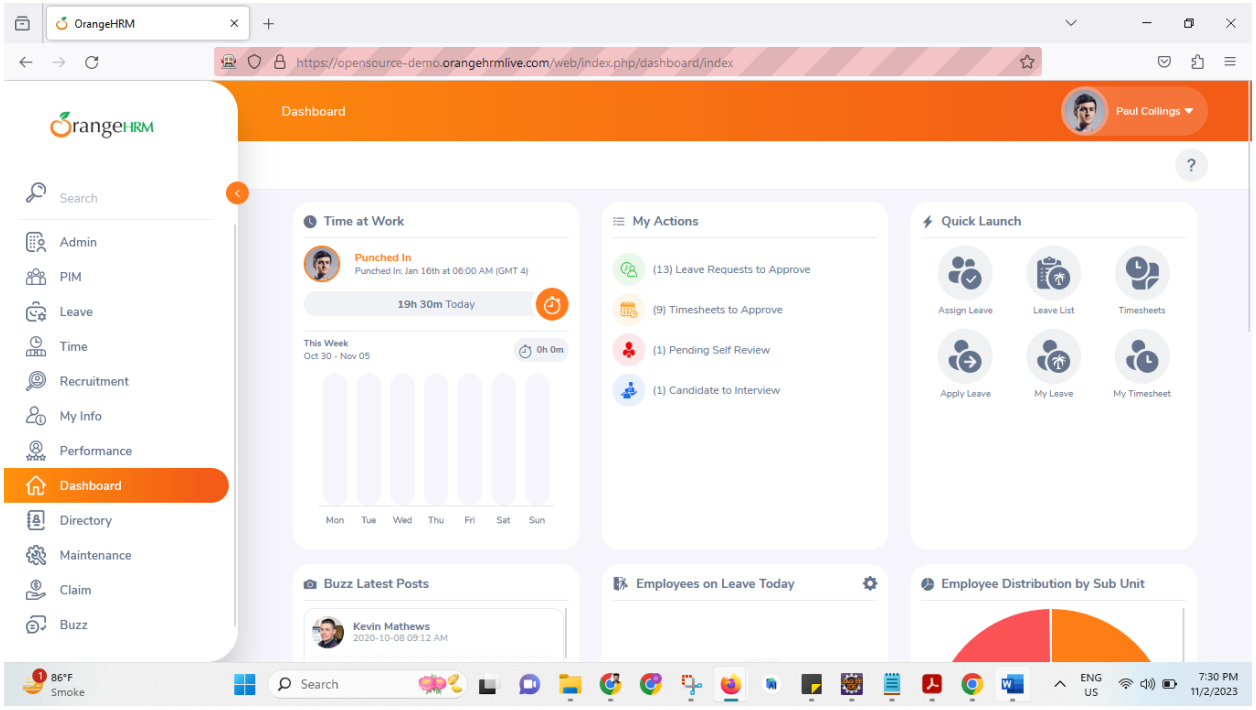
```
package stqa_practical_7;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.WebDriverWait;


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



System.setProperty("webdriver.gecko.driver","E:\\PRANAV_MCA_SEM_3\\STQA\\geckodriver-v
0.33.0-win64\\geckodriver.exe");


        WebDriver wd = new FirefoxDriver();

      wd.manage().timeouts().implicitlyWait(2000, TimeUnit.SECONDS); // Implicit wait

      wd.get("https://opensource-demo.orangehrmlive.com/");
      wd.findElement(By.name("username")).sendKeys("Admin"); // Locator id

      wd.findElement(By.name("password")).sendKeys("admin123");

      wd.findElement(By.cssSelector(".oxd-button")).click();  // Locator CssSelector


      wd.findElement(By.cssSelector(".oxd-userdropdown-name")).click(); // Locator
partialLinkText
```

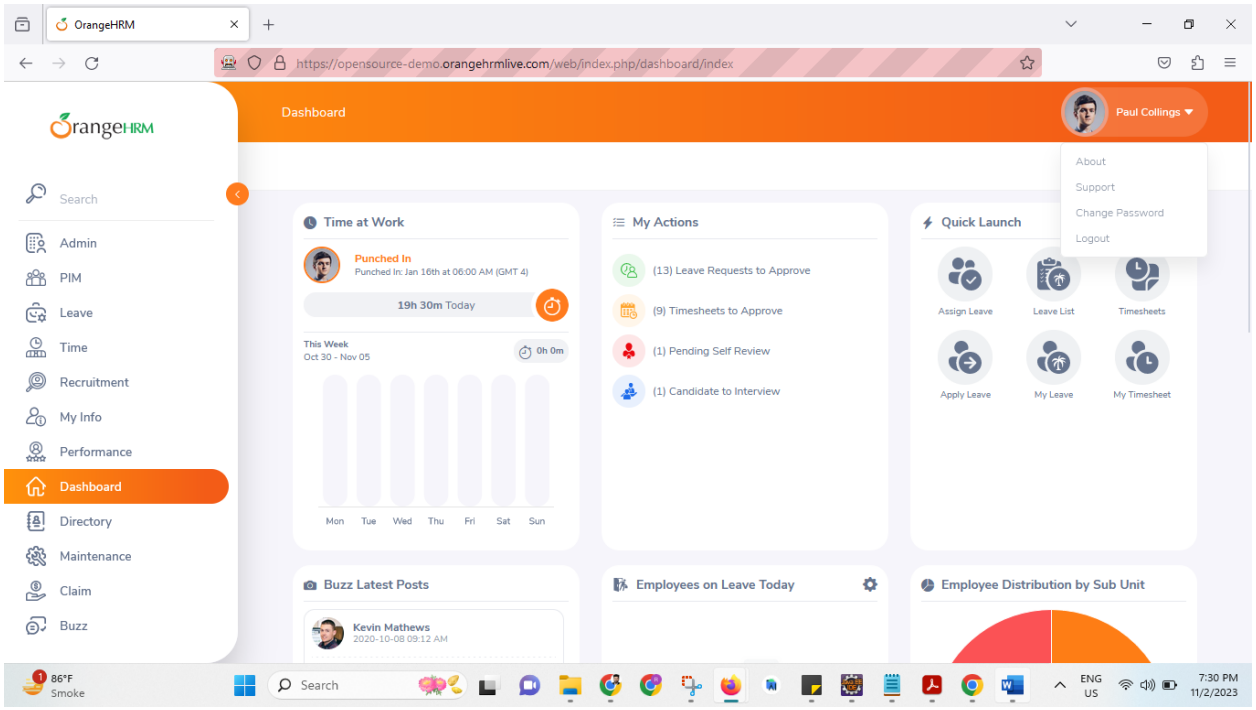
```
        wd.findElement(By.cssSelector(".oxd-dropdown-menu > li:nth-child(4) >
a:nth-child(1)")).click(); // Locator linkText
    }
}
```

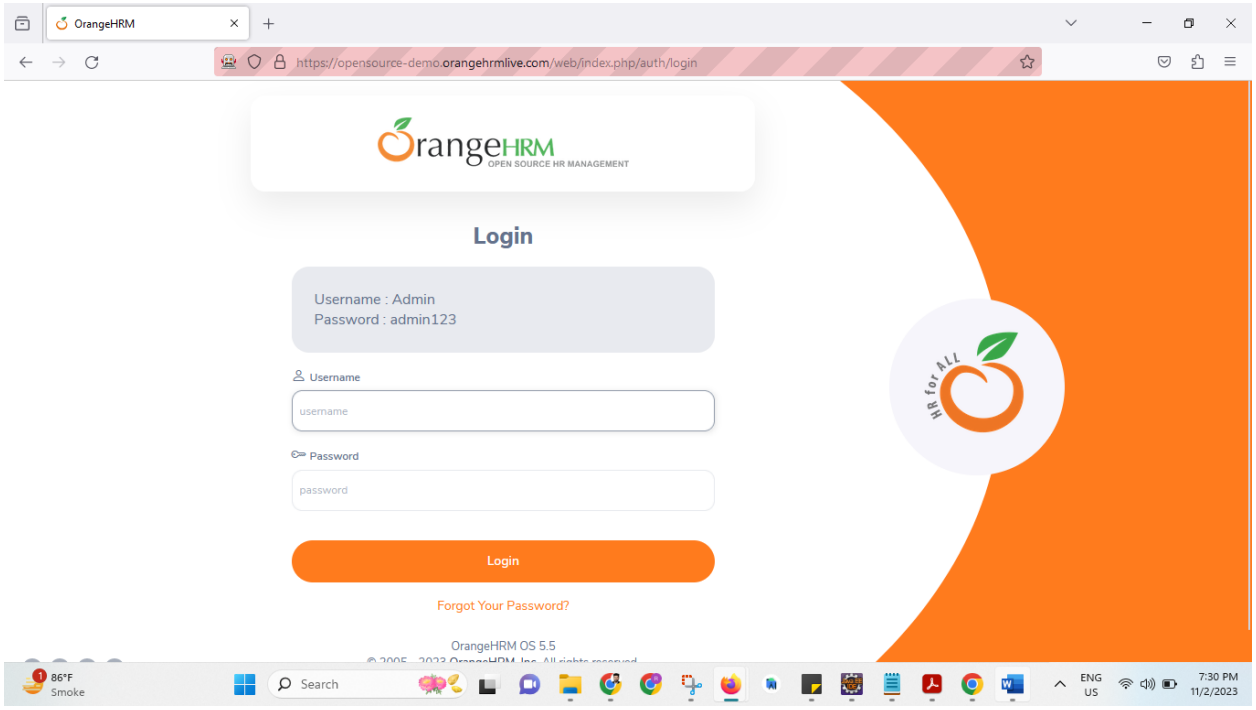
OUTPUT:


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/auth/login
OrangeHRM
OPEN SOURCE HR MANAGEMENT
Login
Username : Admin
Password : admin123
Username
Admin
Password
password
Login
Forgot Your Password?
OrangeHRM OS 5.5
HR for ALL
Search
ENG
US
7:30 PM
11/2/2023


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 5.5
HR for ALL
Search
ENG
US
7:30 PM
11/2/2023


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index
Dashboard
Paul Collings
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Claim
Buzz
Time at Work
Punched In
Punched In: Jan 16th at 06:00 AM (GMT 4)
19h 30m Today
This Week
Oct 30 - Nov 05
0h 0m
Mon Tue Wed Thu Fri Sat Sun
My Actions
(13) Leave Requests to Approve
(9) Timesheets to Approve
(1) Pending Self Review
(1) Candidate to Interview
Quick Launch
Assign Leave
Leave List
Timesheets
Apply Leave
My Leave
My Timesheet
Buzz Latest Posts
Kevin Mathews
2020-10-08 09:12 AM
Employees on Leave Today
Employee Distribution by Sub Unit
86°F
Smoke
Search
ENG
US
7:30 PM
11/2/2023


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index
Dashboard
Paul Collings
About
Support
Change Password
Logout
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Claim
Buzz
Time at Work
Punched In
Punched In: Jan 16th at 06:00 AM (GMT 4)
19h 30m Today
This Week
Oct 30 - Nov 05
0h 0m
Mon Tue Wed Thu Fri Sat Sun
My Actions
(13) Leave Requests to Approve
(9) Timesheets to Approve
(1) Pending Self Review
(1) Candidate to Interview
Quick Launch
Assign Leave
Leave List
Timesheets
Apply Leave
My Leave
My Timesheet
Buzz Latest Posts
Kevin Mathews
2020-10-08 09:12 AM
Employees on Leave Today
Employee Distribution by Sub Unit
86°F
Smoke
Search
ENG
US
7:30 PM
11/2/2023



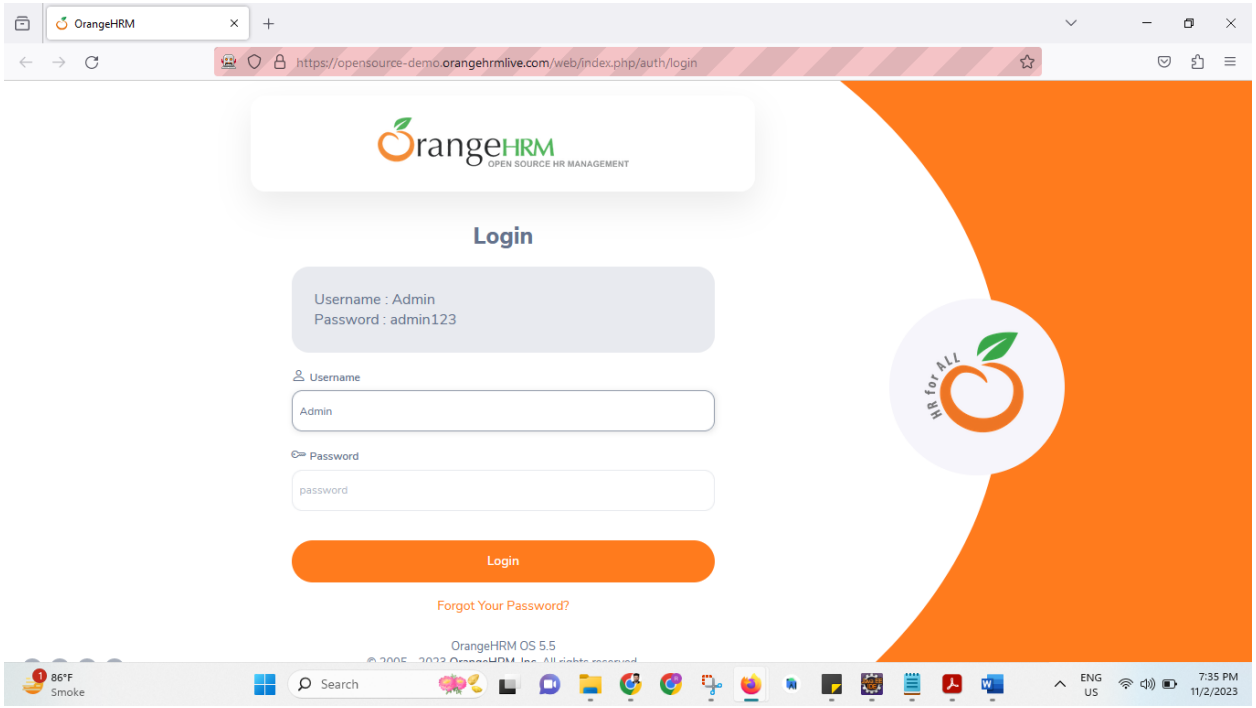
PRACTICAL 7B : EXPLICIT

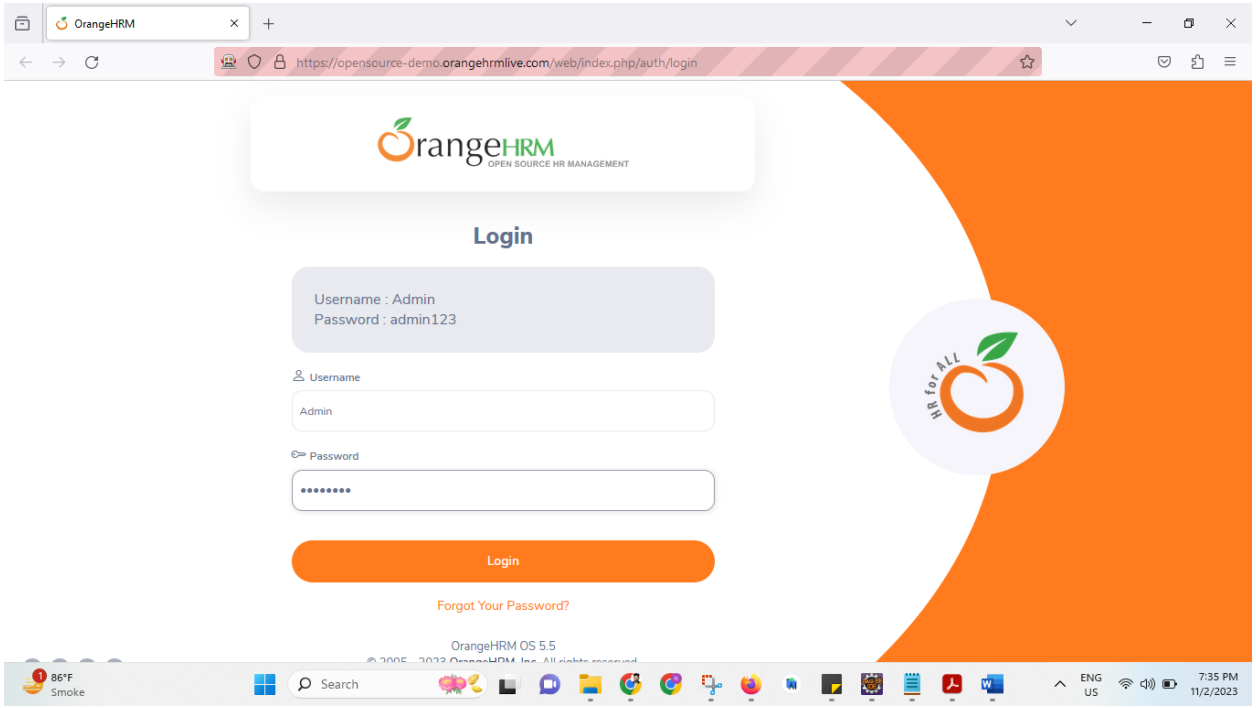
SOURCE CODE:

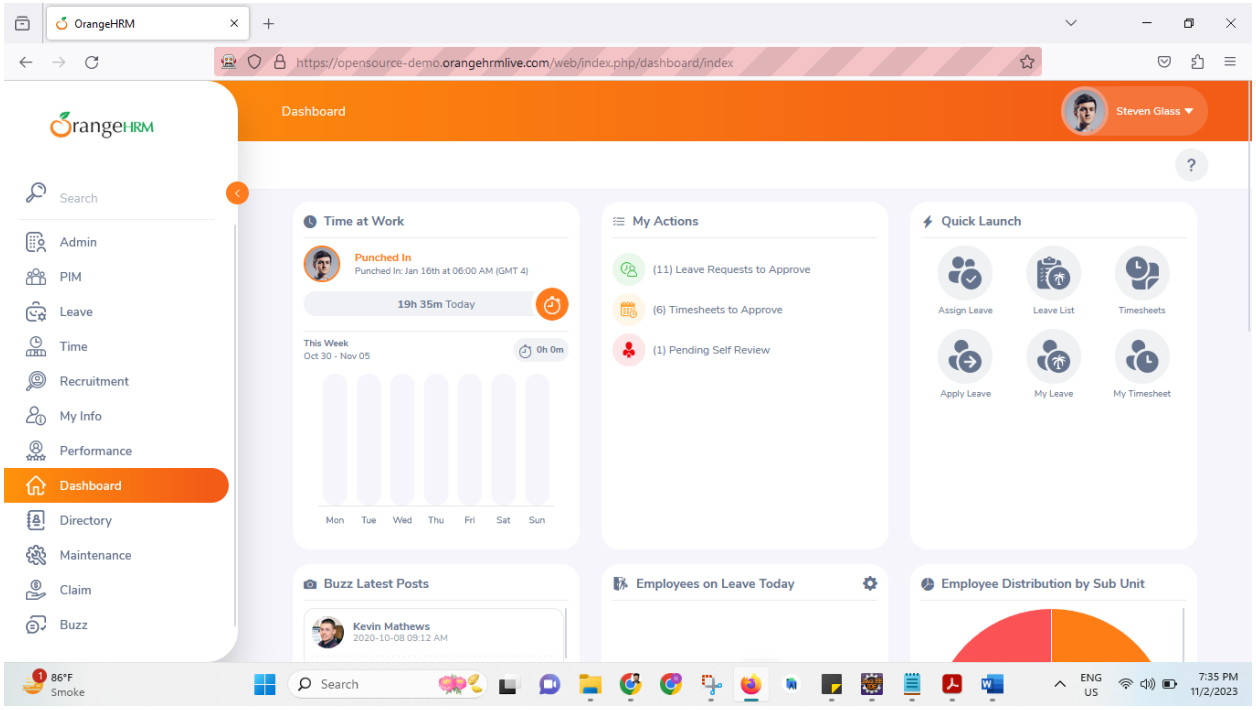
```
package stqa_practical_7;


import java.time.Duration;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class SevenB {
	public static void main(String[] args) throws InterruptedException {
```

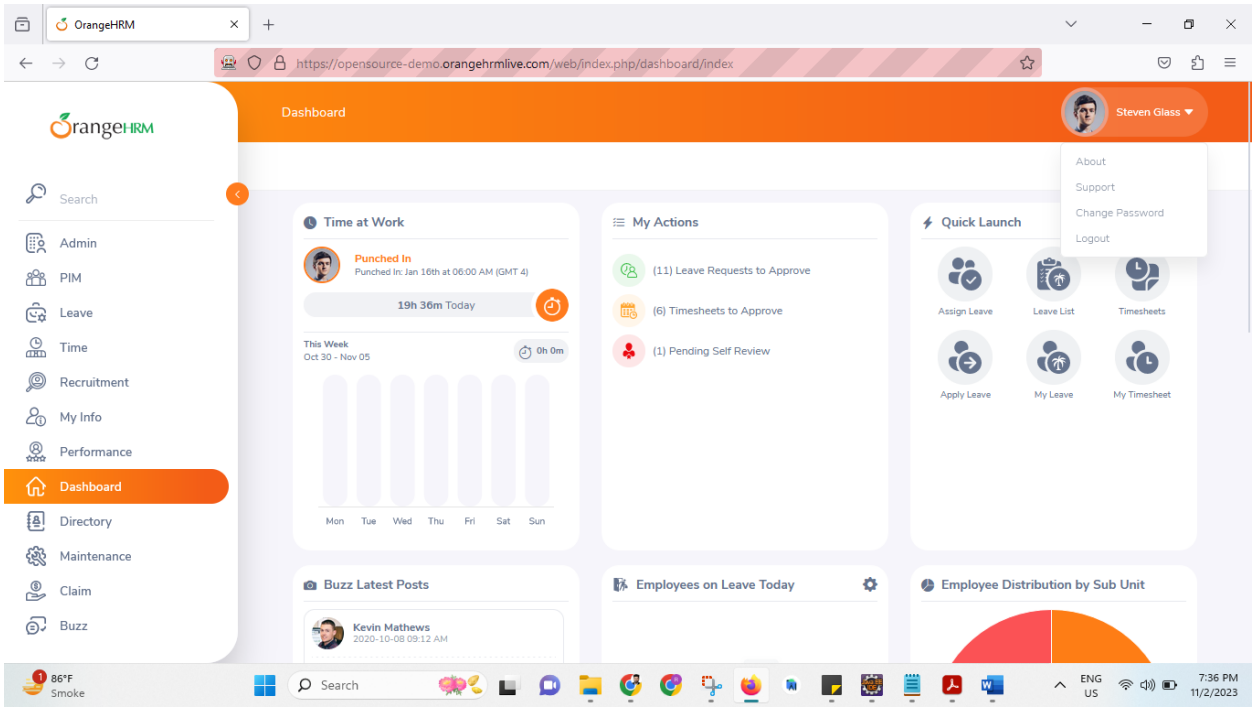
```
System.setProperty("webdriver.gecko.driver","E:\\PRANAV_MCA_SEM_3\\STQA\\geckodriver-v
0.33.0-win64\\geckodriver.exe");

                WebDriver wd = new FirefoxDriver();
                wd.get("https://opensource-demo.orangehrmlive.com/");

                WebDriverWait wt = new WebDriverWait(wd,Duration.ofSeconds(10));


wt.until(ExpectedConditions.presenceOfElementLocated(By.name("username")));
            Thread.sleep(5000);
                 wd.findElement(By.name("username")).sendKeys("Admin");
                  Thread.sleep(5000);



wt.until(ExpectedConditions.presenceOfElementLocated(By.name("password")));
                wd.findElement(By.name("password")).sendKeys("admin123");

                Thread.sleep(5000);
                wt.until(ExpectedConditions.elementToBeClickable(By.cssSelector(".oxd-button")));
                wd.findElement(By.cssSelector(".oxd-button")).click();
                Thread.sleep(5000);
               // Locator CssSelector



wt.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector(".oxd-userdropdown-na
me")));
                wd.findElement(By.cssSelector(".oxd-userdropdown-name")).click();
                Thread.sleep(5000);


wt.until(ExpectedConditions.visibilityOfElementLocated(By.cssSelector(".oxd-dropdown-menu >
li:nth-child(4) > a:nth-child(1)")));
                wd.findElement(By.cssSelector(".oxd-dropdown-menu > li:nth-child(4) >
a:nth-child(1)")).click();
                Thread.sleep(5000);
            }
```

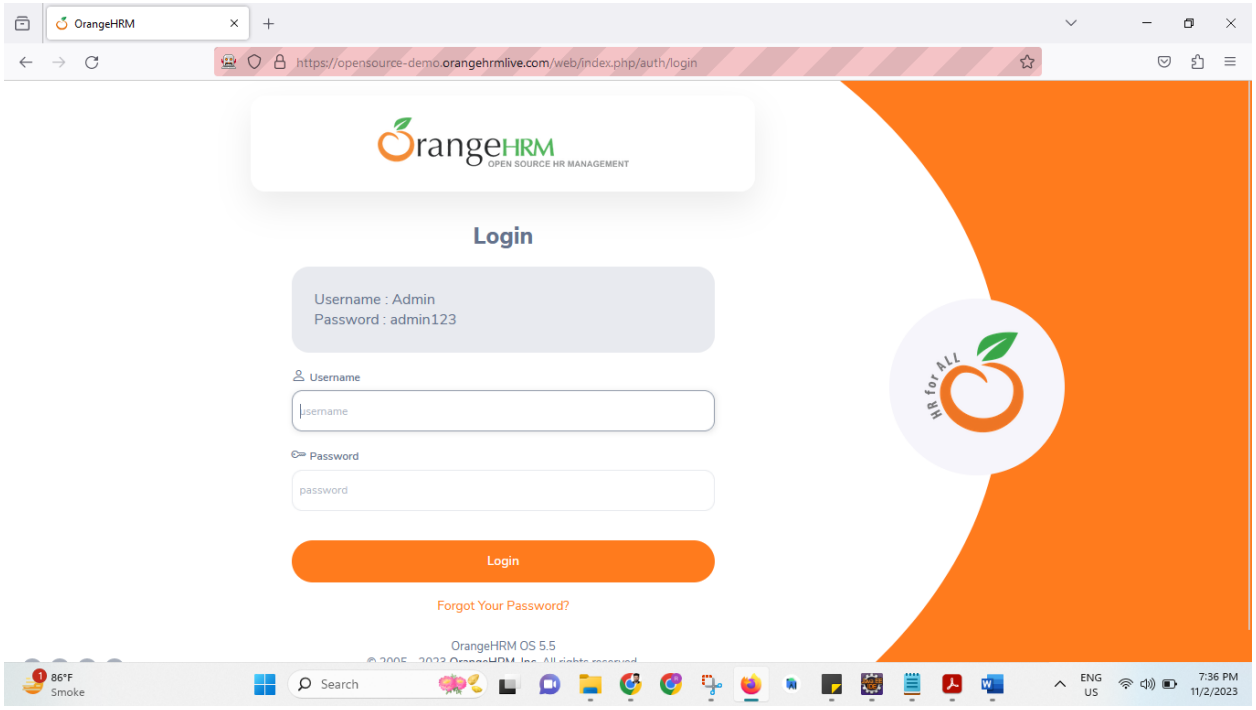
```
}
```


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/auth/login
OrangeHRM
OPEN SOURCE HR MANAGEMENT
Login
Username : Admin
Password : admin123
Username
Admin
Password
password
Login
Forgot Your Password?
OrangeHRM OS 5.5
HR for ALL
Search
ENG
US
7:35 PM
11/2/2023


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 5.5
HR for ALL
Search
ENG
US
7:35 PM
11/2/2023


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index
Dashboard
Steven Glass
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Claim
Buzz
Time at Work
Punched In
Punched In: Jan 16th at 06:00 AM (GMT 4)
19h 35m Today
This Week
Oct 30 - Nov 05
0h 0m
Mon Tue Wed Thu Fri Sat Sun
My Actions
(11) Leave Requests to Approve
(6) Timesheets to Approve
(1) Pending Self Review
Quick Launch
Assign Leave
Leave List
Timesheets
Apply Leave
My Leave
My Timesheet
Buzz Latest Posts
Kevin Mathews
2020-10-08 09:12 AM
Employees on Leave Today
Employee Distribution by Sub Unit
86°F
Smoke
Search
ENG
US
7:35 PM
11/2/2023


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index
Dashboard
Steven Glass
About
Support
Change Password
Logout
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Claim
Buzz
Time at Work
Punched In
Punched In: Jan 16th at 06:00 AM (GMT 4)
19h 36m Today
This Week
Oct 30 - Nov 05
0h 0m
Mon Tue Wed Thu Fri Sat Sun
My Actions
(11) Leave Requests to Approve
(6) Timesheets to Approve
(1) Pending Self Review
Quick Launch
Assign Leave
Leave List
Timesheets
Apply Leave
My Leave
My Timesheet
Buzz Latest Posts
Kevin Mathews
2020-10-08 09:12 AM
Employees on Leave Today
Employee Distribution by Sub Unit
86°F
Smoke
Search
ENG
US
7:36 PM
11/2/2023



Conclusion:

Implicit wait and explicit wait has been demonstrated.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 8</td></tr>
<tr><td colspan="4">Title of LAB Assignment : Demonstrate different types of alerts</td></tr>
<tr><td colspan="2">DOP : 06/10/2023</td><td colspan="2">DOS : 13/10/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 08

AIM: Demonstrate different types of alerts

Theory:

What is Alert in Selenium?

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.

How to handle Alert in Selenium WebDriver:

1) void dismiss() **// To click on the 'Cancel' button of the alert.**
driver.switchTo().alert().dismiss();
2) void accept() **// To click on the 'OK' button of the alert.**
driver.switchTo().alert().accept();
3) String getText**() // To capture the alert message.**
driver.switchTo().alert().getText();
4) void sendKeys(String stringToSend) **// To send some data to alert box.** driver.switchTo().alert().sendKeys("Text");

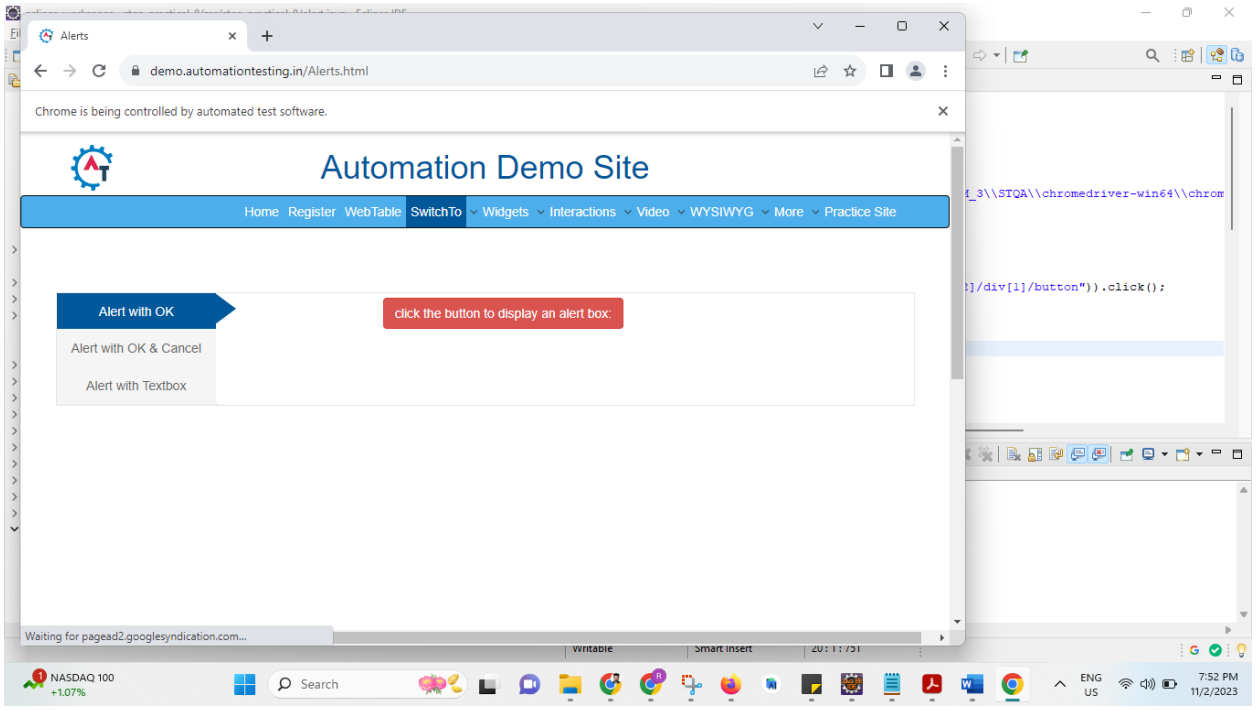
SOURCE CODE:

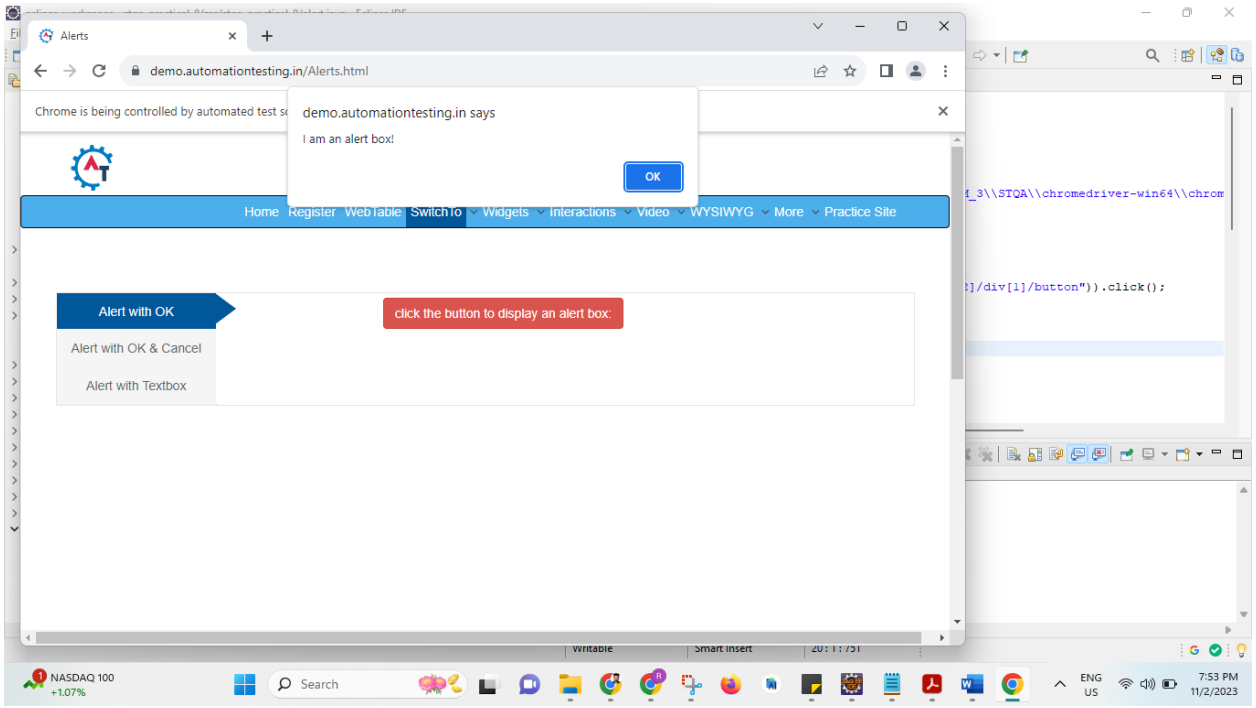
```
package stqa_practical_8;

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

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

System.setProperty("webdriver.chrome.driver","E:\\PRANAV_MCA_SEM_3\\STQA\\chromedriver-win64\\chromedriver-win64\\chromedriver.exe");
```

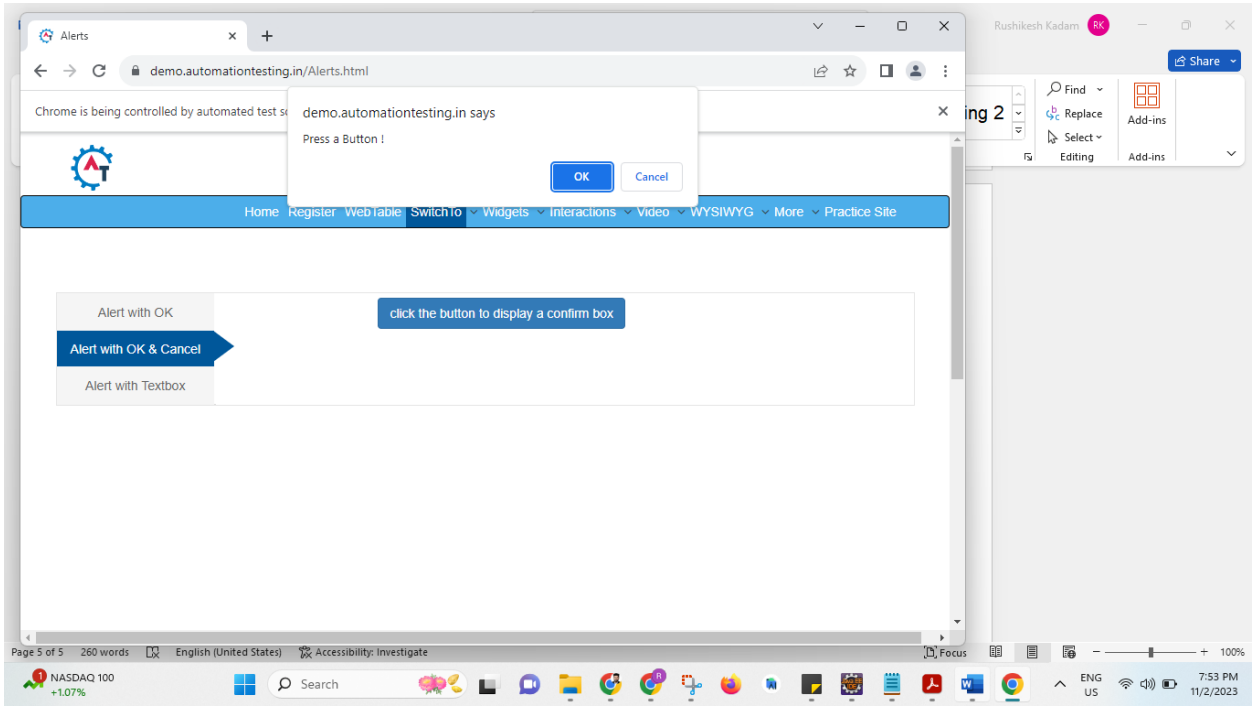
```
        WebDriver webDriver = new ChromeDriver();

        webDriver.get("https://demo.automationtesting.in/Alerts.html");
        Thread.sleep(2000);

webDriver.findElement(By.xpath("/html/body/div[1]/div/div/div/div[2]/div[1]/button")).click();
        Thread.sleep(2000);
        // Switching to Alert
        Alert alert = webDriver.switchTo().alert();

        // Capturing and printing alert message.
        System.out.println(webDriver.switchTo().alert().getText());

        alert.accept();
        Thread.sleep(2000);

webDriver.findElement(By.xpath("/html/body/div[1]/div/div/div/div[1]/ul/li[2]/a")).click();
        Thread.sleep(2000);

webDriver.findElement(By.xpath("/html/body/div[1]/div/div/div/div[2]/div[2]/button")).click();
        Thread.sleep(2000);
        // Switching to Alert
        alert = webDriver.switchTo().alert();

        // Capturing and printing alert message.
        System.out.println(webDriver.switchTo().alert().getText());
        alert.accept();


webDriver.findElement(By.xpath("/html/body/div[1]/div/div/div/div[1]/ul/li[3]/a")).click();
        Thread.sleep(2000);

webDriver.findElement(By.xpath("/html/body/div[1]/div/div/div/div[2]/div[3]/button")).click();
        Thread.sleep(2000);
        // Switching to Alert
        alert = webDriver.switchTo().alert();
        alert.sendKeys("ABC");
        Thread.sleep(2000);
        // Capturing and printing alert message.
        System.out.println(webDriver.switchTo().alert().getText());
        alert.accept();
```

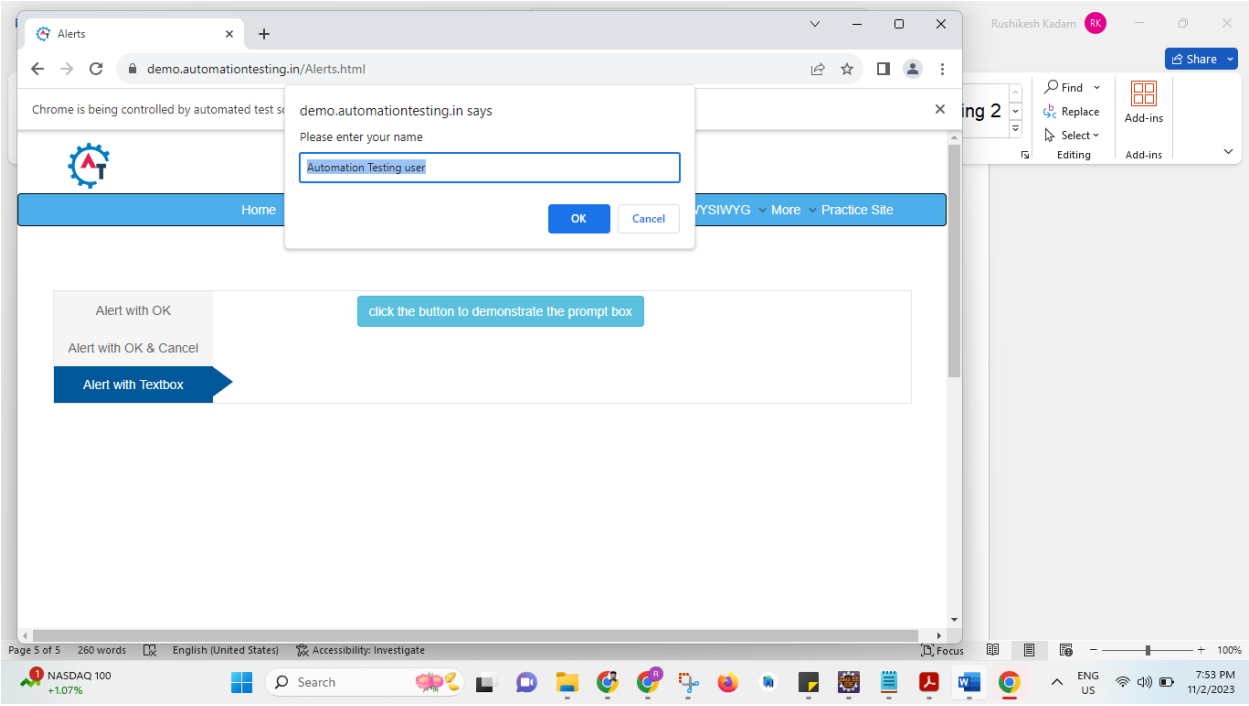
```
System.out.println(webDriver.findElement(By.xpath("/html/body/div[1]/div/div/div/div[2]/div[3
]/p")).getText());

                Thread.sleep(2000);
                webDriver.close();
                }

}
```

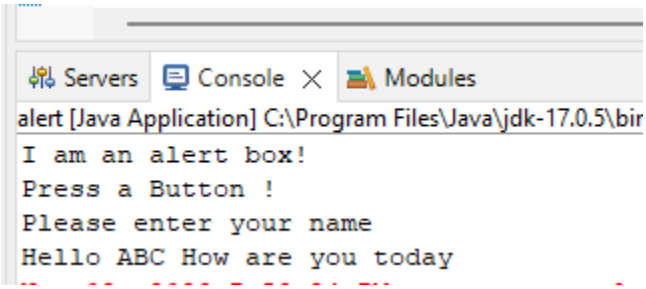
Output:




Alerts
demo.automationtesting.in/Alerts.html
Chrome is being controlled by automated test s
demo.automationtesting.in says
I am an alert box!
OK
Home Register WebTable SwitchTo Widgets Interactions Video WYSIWYG More Practice Site
Alert with OK
Alert with OK & Cancel
Alert with Textbox
click the button to display an alert box:
NASDAQ 100
+1.07%
Search
ENG
US
7:53 PM
11/2/2023


Alerts
demo.automationtesting.in/Alerts.html
Chrome is being controlled by automated test s
demo.automationtesting.in says
Press a Button !
OK
Cancel
Home Register WebTable SwitchTo Widgets Interactions Video WYSIWYG More Practice Site
Alert with OK
Alert with OK & Cancel
Alert with Textbox
click the button to display a confirm box
Share
Find
Replace
Select
Editing
Add-ins
Page 5 of 5
260 words
English (United States)
Accessibility: Investigate
Focus
100%
NASDAQ 100
+1.07%
Search
ENG
US
7:53 PM
11/2/2023





Conclusion: With the help of this practical, we have successfully Implemented the concept of Alert.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 9</td></tr>
<tr><td colspan="4">Title of LAB Assignment :Demonstrate Handling Drop Down and List Boxes</td></tr>
<tr><td colspan="2">DOP : 13/10/2023</td><td colspan="2">DOS : 20/10/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 09

AIM: Demonstrate different types of alerts

Theory:

Select Class in Selenium

The **Select Class in Selenium** is a method used to implement the HTML SELECT tag. The html select tag provides helper methods to select and deselect the elements. The Select class is an ordinary class so New keyword is used to create its object and it specifies the web element location.

Select Option from Drop-Down Box

Following is a step by step process on how to select value from dropdown in Selenium:

Before handling dropdown in Selenium and controlling drop-down boxes, we must do following two things:

1. Import the package **org.openqa.selenium.support.ui.Select**

2. Instantiate the drop-down box as an object, Select in Selenium WebDriver

ListBox:

ListBox is an element where user can select/deselect one or more items from it.

To identify the ListBox on webpage, look for select tag and attribute should be ‘multiple’ to select multiple items and there will be option tag which contains each item in it.

Practical 9A : ListBox

SOURCE CODE:

package stqa_practical_9;

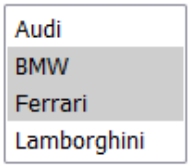
NineA.java

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



System.setProperty("webdriver.gecko.driver","E:\\PRANAV_MCA_SEM_3\\STQA\\geckodriver-v
0.33.0-win64\\geckodriver.exe");


			WebDriver wd = new FirefoxDriver();
			 wd.get("E:\\PRANAV_MCA_SEM_3\\STQA\\Practical_9\\Multi.html");

			Select s = new Select(wd.findElement(By.id("car")));
			if(s.isMultiple())
			{
			s.selectByIndex(1);
			s.selectByValue("3");
			s.selectByVisibleText("Ferrari");
			Thread.sleep(1000);
			//s.deselectByIndex(1);


			}

		}
	}
```

Multi.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Multi-Select List</title>
</head>
<body>
  <h1>Multi-Select List</h1>
  <form>
    <select id="car" multiple="multiple">
      <option value="1">Audi</option>
      <option value="2">BMW</option>
      <option value="3">Ferrari</option>
      <option value="4">Lamborghini</option>
      <option value="5">Mercedes-Benz</option>
    </select>
  </form>
</body>
</html>
```

OUTPUT:



Practical 9B: Drop Down :

Source Code:

```
package stqa_practical_9;

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

System.setProperty("webdriver.gecko.driver","E:\\PRANAV_MCA_SEM_
3\\STQA\\geckodriver-v0.33.0-win64\\geckodriver.exe");
      WebDriver wd = new FirefoxDriver();

wd.get("E:\\PRANAV_MCA_SEM_3\\STQA\\Practical_9\\DropDown.ht
ml");
```

```
        Select s = new Select(wd.findElement(By.id("fromPort")));
        Select t = new Select(wd.findElement(By.id("toPort")));

        s.selectByVisibleText("India");
        t.selectByVisibleText("Australia");
    }
}
```

DropDown.html

```
<!DOCTYPE html>
<html>
<head>
  <title>DropDowns</title>
</head>
<body>
  <h1>DropDowns Program by PRANAV_12</h1>
  <form>
    <label for="fromPort">From Port:</label>
    <select id="fromPort" name="fromPort">
      <option value="Paris">Paris</option>
      <option value="London">London</option>
      <option value="New York">New York</option>
      <option value="Sydney">Sydney</option>
      <option value="India">India</option>
    </select>
    <br><br>
    <label for="toPort">To Port:</label>
    <select id="toPort" name="toPort">
      <option value="Paris">Paris</option>
      <option value="London">London</option>
      <option value="New York">New York</option>
      <option value="Sydney">Sydney</option>
      <option value="Australia">Australia</option>
    </select>
  </form>
</body>
</html>
```

Output:

Conclusion: With the help of this practical, we have successfully Implemented the concept of ListBox and DropDown.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 10</td></tr>
<tr><td colspan="4">Title of LAB Assignment :Demonstrate Command Button, Radio buttons & text boxes</td></tr>
<tr><td colspan="2">DOP : 13/10/2023</td><td colspan="2">DOS : 20/10/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 10

AIM: Demonstrate Command Button, Radio buttons & text Boxes

Theory:

Selenium WebDriver - Navigation Commands

WebDriver provides some basic Browser Navigation Commands that allows the browser to move backwards or forwards in the browser's history.
Just like the browser methods provided by WebDriver, we can also access the navigation methods provided by WebDriver by typing driver.navigate() in the Eclipse panel.

1. Navigate To Command

Method:
to(String arg0) : **void**
In WebDriver, this method loads a new web page in the existing browser window. It accepts *String* as parameter and returns *void*.
The respective command to load/navigate a new web page can be written as: driver.navigate().to("www.javatpoint.com");

2. Forward Command

Method:
to(String arg0) : **void**
In WebDriver, this method enables the web browser to click on the **forward** button in the existing browser window. It neither accepts anything nor returns anything.
Implementation:
driver.navigate().forward();

3. Back Command

Method:
back() : **void**
In WebDriver, this method enables the web browser to click on the **back** button in the existing browser window. It neither accepts anything nor returns anything.
The respective command that takes you back by one page on the browser's history can be written as:
driver.navigate().back();

4. Refresh Command

Method:
refresh() : **void**
In WebDriver, this method refresh/reloads the current web page in the existing

browser window. It neither accepts anything nor returns anything.
The respective command that takes you back by one page on the browser's history can be written as:
driver.navigate().refresh();

Radio Button

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

Input Box

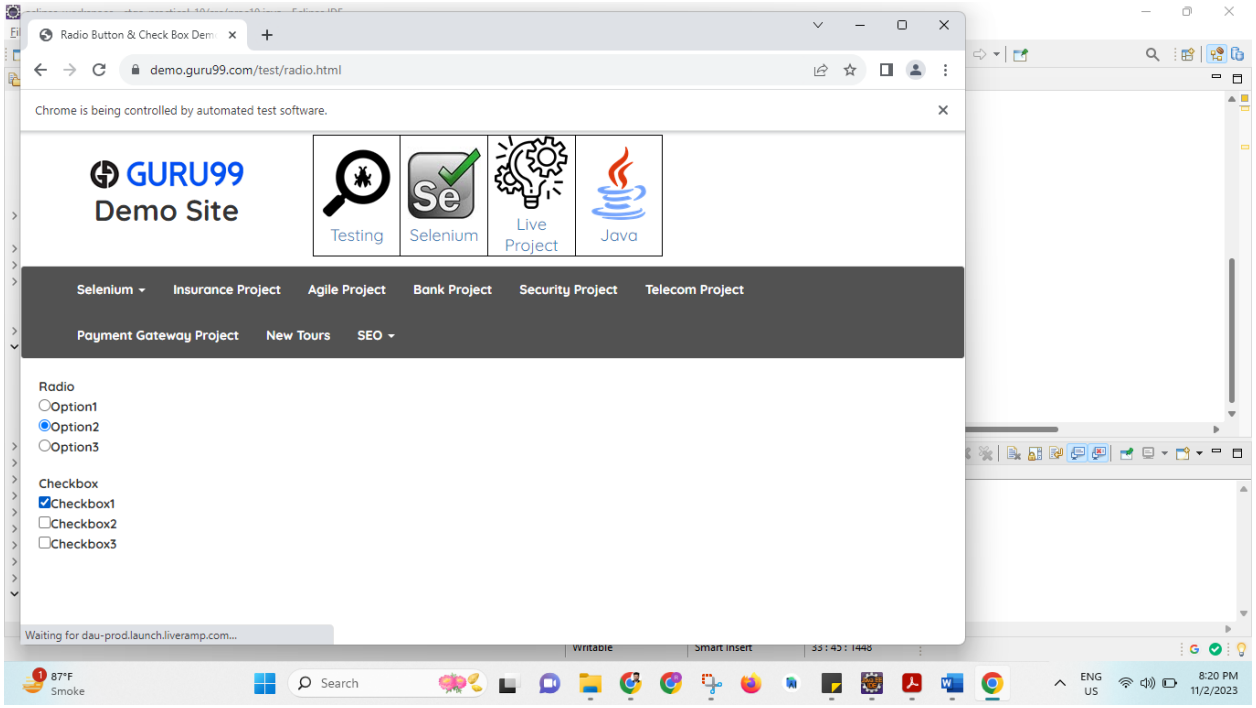
Input boxes refer to either of these two types:
1. **Text Fields**– Selenium input text boxes that accept typed values and show them as they are.
2. **Password Fields**– text boxes that accept typed values but mask them as a series of special characters (commonly dots and asterisks) to avoid sensitive values to be displayed

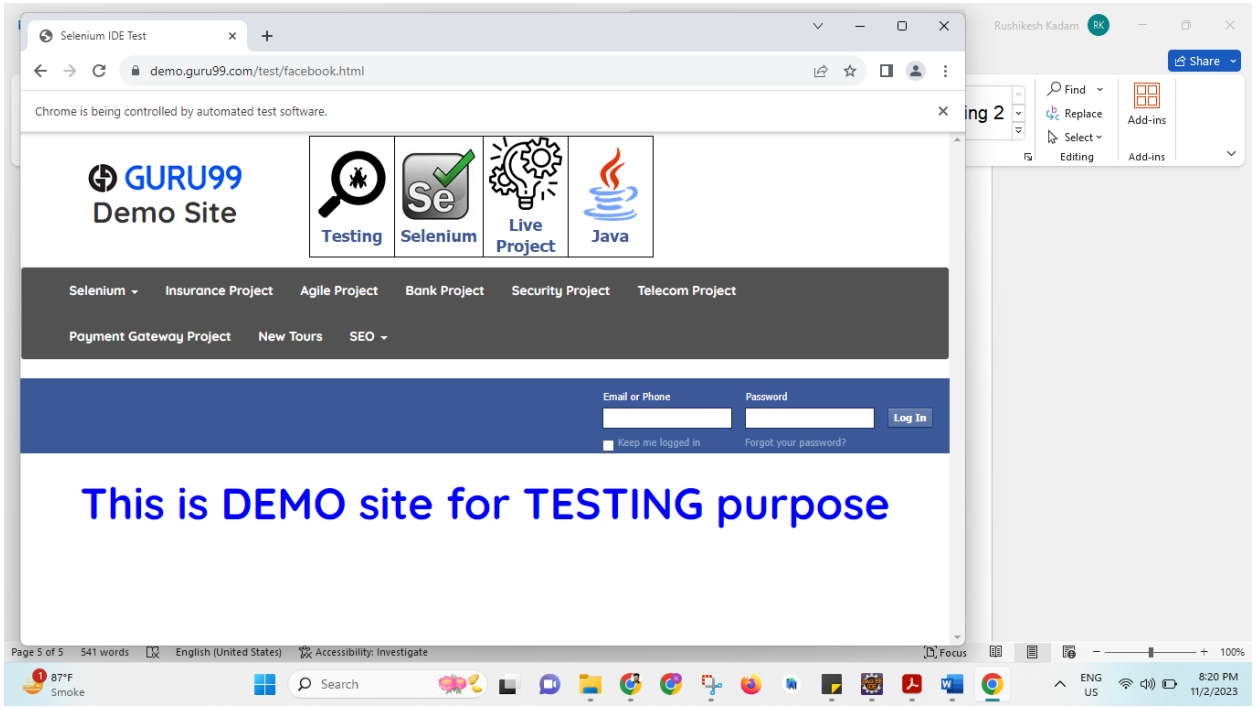
SOURCE CODE:

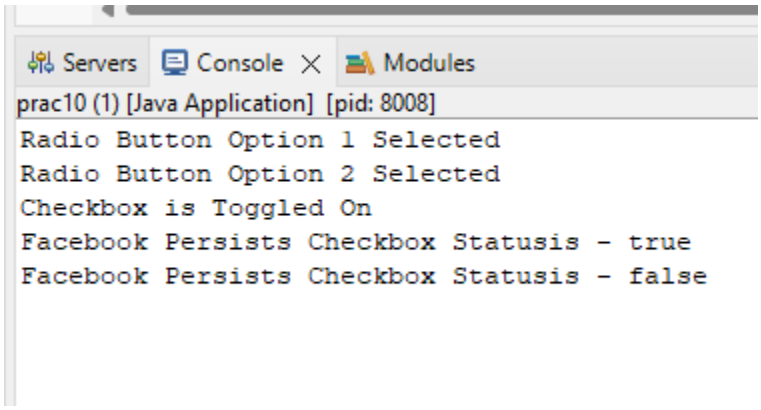
```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;
public class prac10 {
public static void main(String[] args) throws InterruptedException {

System.setProperty("webdriver.chrome.driver","E:\\PRANAV_MCA_SEM_3\\STQA
\\chromedriver-win64\\chromedriver-win64\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("https://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 deselected and Radio Button2 is selected
radio2.click();
System.out.println("Radio Button Option 2 Selected");
// Selecting CheckBox
WebElement option1 = driver.findElement(By.id("vfb-6-0"));
// This will Toggle the Check box
option1.click();
// Check whether the Check box is toggled on
if (option1.isSelected()) {
System.out.println("Checkbox is Toggled On");
} else {
System.out.println("Checkbox is Toggled Off");
}
Thread.sleep(5000);
// Selecting Checkbox and using isSelected Method
driver.get("http://demo.guru99.com/test/facebook.html");
WebElement chkFBPersist = driver.findElement(By.id("persist_box"));
for (int i = 0; i < 2; i++) {
chkFBPersist.click();

System.out.println("Facebook Persists Checkbox Statusis - " +

chkFBPersist.isSelected());
}
Thread.sleep(5000);
driver.close();

}
}
```

OUTPUT:


Radio Button & Check Box Demo
demo.guru99.com/test/radio.html
Chrome is being controlled by automated test software.
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
Radio
Option1
Option2
Option3
Checkbox
Checkbox1
Checkbox2
Checkbox3
Waiting for dau-prod.launch.liveramp.com...


Selenium IDE Test
demo.guru99.com/test/facebook.html
Chrome is being controlled by automated test software.
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
Email or Phone
Password
Log In
Keep me logged in
Forgot your password?
This is DEMO site for TESTING purpose



Conclusion: With the help of this practical, we have successfully Implemented the concept of CheckBox and RadioButton.

Name of Student : DESHMUKH PRANAV DINESH			
Roll Number : 12		**LAB Assignment Number:** 11	
Title of LAB Assignment :Demonstrate action classes in Selenium			
DOP : 19/10/2023		**DOS :** 27/10/2023	
CO Mapped : C02	**PO Mapped:** PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2	**Faculty Signature:**	**Marks :**

Practical No: 11

AIM: Demonstrate action classes in Selenium

Theory:

What is Action Class in Selenium?

Actions class is an ability provided by Selenium for handling keyboard and mouse events. In Selenium WebDriver, handling these events includes operations such as drag and drop, clicking on multiple elements with the control key, among others. These operations are performed using the advanced user interactions API. It mainly consists of *Actions* that are needed while performing these operations.
Action class is defined and invoked using the following syntax:
Actions action = new Actions(driver);
action.moveToElement(element).click().perform();

Methods of Action Class

Action class is useful mainly for mouse and keyboard actions. In order to perform such actions, Selenium provides various methods.
Mouse Actions in Selenium:
doubleClick(): Performs double click on the element
clickAndHold(): Performs long click on the mouse without releasing it
dragAndDrop(): Drags the element from one point and drops to another
moveToElement(): Shifts the mouse pointer to the center of the element
contextClick(): Performs right-click on the mouse **Keyboard Actions in Selenium:**
sendKeys(): Sends a series of keys to the element

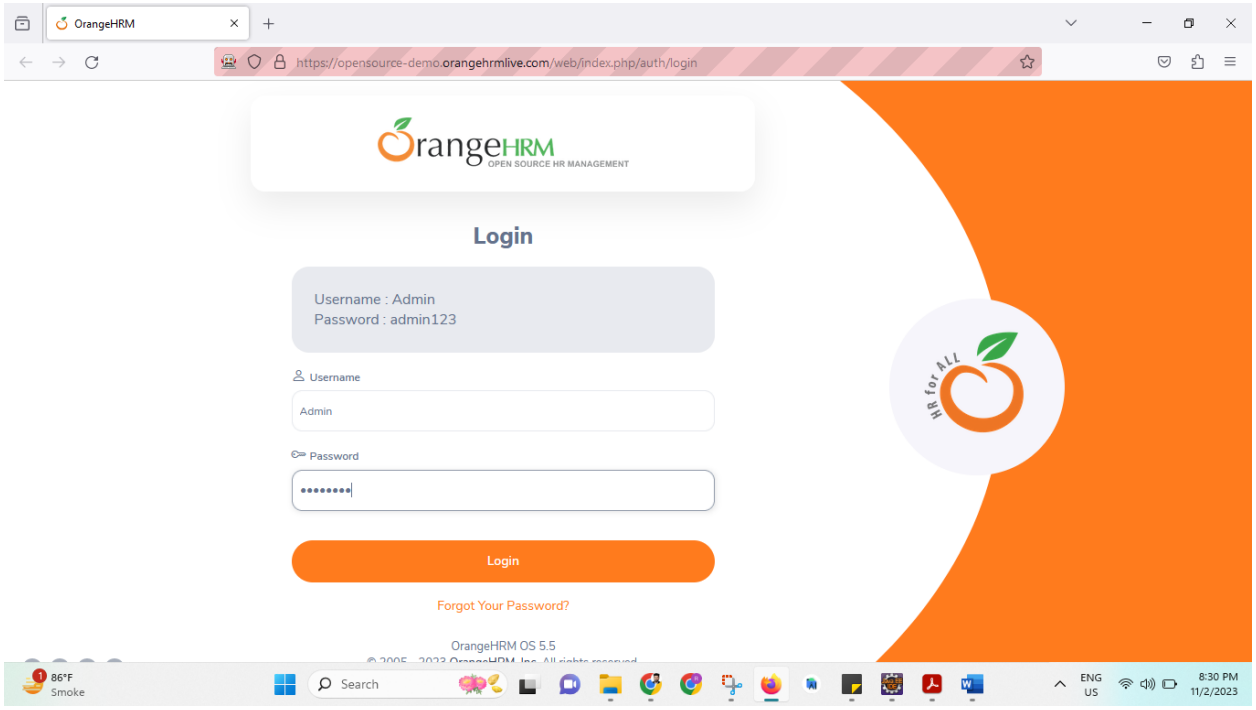
keyUp(): Performs key release keyDown():
Performs keypress without release

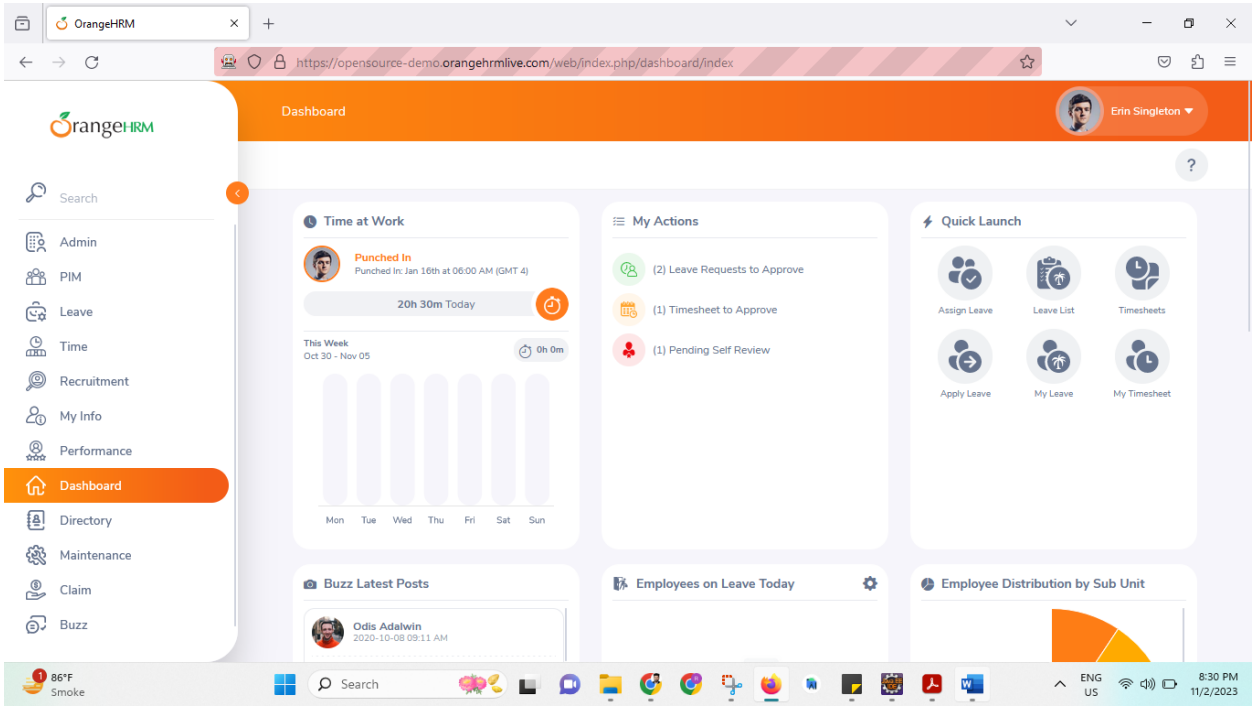
SOURCE CODE:

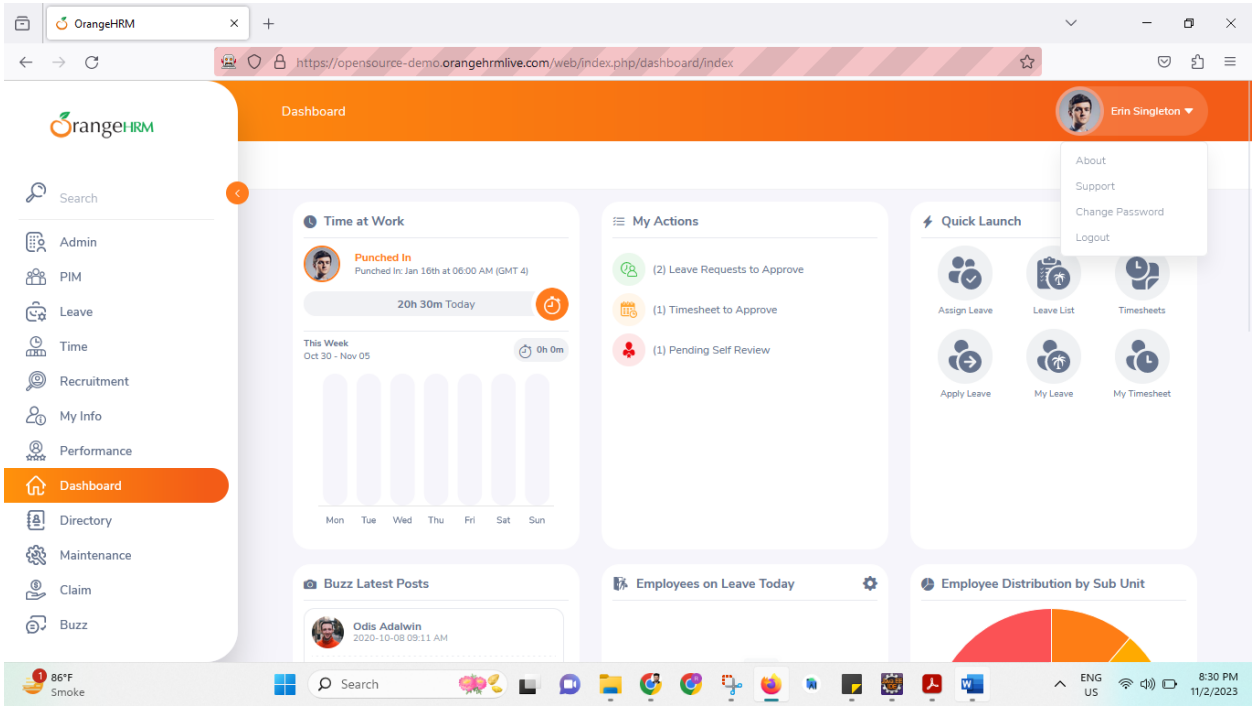
```
package stqa_practical_11;

import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;
public class Prac_11 {
	public static void main(String[] args) throws InterruptedException
	{
		System.setProperty("webdriver.chrome.driver",

"E:\\PRANAV_MCA_SEM_3\\STQA\\chromedriver-win64\\chromedriver-win64\\chrome
driver.exe");

		WebDriver wd = new FirefoxDriver();
		wd.get("https://opensource-demo.orangehrmlive.com/");

			wd.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);

			 wd.findElement(By.name("username")).sendKeys("Admin"); //
Locator id

			wd.findElement(By.name("password")).sendKeys("admin123");

			wd.findElement(By.cssSelector(".oxd-button")).click();

			Actions act = new Actions(wd);

			List<WebElement> menu
=wd.findElements(By.cssSelector("ul.oxd-main-menu"));

			for(int i=0;i<=menu.size()-1;i++)
			{
```

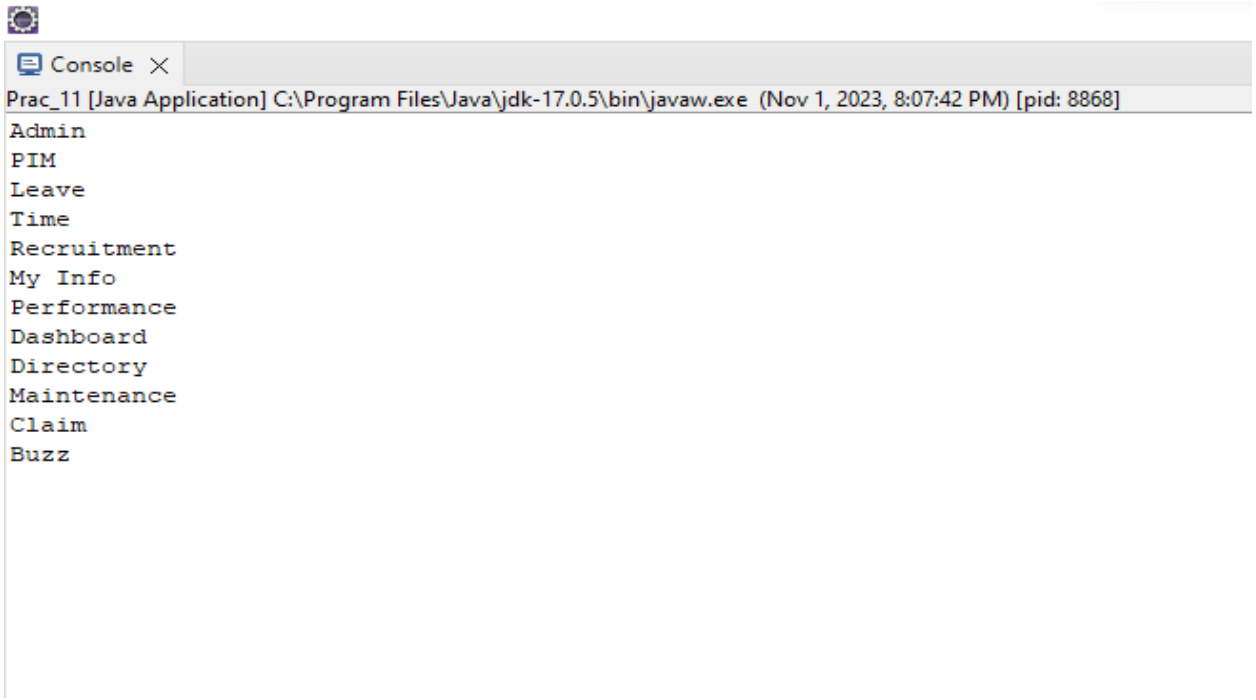
```
                    System.out.println(menu.get(i).getText());//print text ofall the
element on console
                    act.moveToElement(menu.get(i)).perform();//toperform
mousehover on all elements of list
                    Thread.sleep(5000);
                    }

//wd.navigate().to("https://practicetestautomation.com/practice-test-login/");

              wd.findElement(By.cssSelector(".oxd-userdropdown-name")).click(); //
Locator partialLinkText

            //  wd.findElement(By.cssSelector(".oxd-dropdown-menu > li:nth-child(4) >
a:nth-child(1)")).click(); // Locator linkText
                    //Thread.sleep(2000);
                    //System.out.println("Logout");
                    //wd.close();
                    }
                    }
```

OUTPUT:


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 5.5
HR for ALL


OrangeHRM
https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index
Dashboard
Erin Singleton
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Maintenance
Claim
Buzz
Time at Work
Punched In
20h 30m Today
My Actions
(2) Leave Requests to Approve
(1) Timesheet to Approve
(1) Pending Self Review
Quick Launch
Assign Leave
Leave List
Timesheets
Apply Leave
My Leave
My Timesheet
Buzz Latest Posts
Employees on Leave Today
Employee Distribution by Sub Unit





Conclusion: With the help of this practical, we have successfully Implemented the concept of Action Classes.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 12</td></tr>
<tr><td colspan="4">Title of LAB Assignment :Installation of TestNg , running testNg and TestNg annotations.</td></tr>
<tr><td colspan="2">DOP : 19/10/2023</td><td colspan="2">DOS : 27/10/2023</td></tr>
<tr><td>CO Mapped :
C03</td><td>PO Mapped:
PO2,PO3,PO4,PO5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 12

Aim: Installation of TestNg , running testNg and TestNg annotations.

Theory:

TestNG and TestNG Annotations:

TestNG is a testing framework for Java that makes it easy to perform unit testing, integration testing, and end-to-end testing of applications. It is inspired by JUnit but introduces new functionalities that make it more powerful and easier to use. Here are the key steps to install TestNG, run TestNG tests, and understand TestNG annotations:

- **Installation of TestNG:** You can install TestNG in several ways, but the most common method is to use a build tool like Maven or Gradle. For Maven, you need to add a dependency to your project's pom.xml file. For Gradle, you can add it to your build.gradle file. Alternatively, you can download the TestNG JAR and add it to your project's classpath.
- **Running TestNG Tests:** To run TestNG tests, you can use the TestNG XML configuration file or run tests programmatically. You can execute TestNG tests from your IDE or through the command line.
- **TestNG Annotations:** TestNG uses annotations to mark methods as test methods and to control the test execution flow. Some common TestNG annotations include @Test, @BeforeSuite, @AfterSuite, @BeforeTest, @AfterTest, @BeforeClass, @AfterClass, @BeforeMethod, and @AfterMethod. These annotations help you set up test conditions and manage the test lifecycle.

1. Installation of TestNg , running testNg and TestNg annotations.

Installation Steps:

Step 1: Go to the github repository:
https://github.com/testng-team/testng-eclipse

Step 2: Download the latest version zip file.

Step 3: Add all the jar files to the project classpath.
Note: Jar files are only need for Normal java projects, If you use Maven project, you need to add dependencies in pom.xml file.

Source Code:

WebAppLoginTest.java:

```
package org.example;

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

public class WebAppLoginTest {

    @BeforeClass
    public void setupWebDriver() {
        // Code to initialize the web driver, e.g., opening a browser.
        System.out.println("Setting up WebDriver and opening the browser.");
    }

    @BeforeMethod
    public void navigateToLoginPage() {
        // Code to navigate to the login page of the web application.
        System.out.println("Navigating to the login page.");
    }

    @Test
    public void testValidLogin() {
        // Code to perform a valid login.
        System.out.println("Test: Valid login.");
        // Assert statements to check if the login was successful.
    }

    @Test
    public void testInvalidLogin() {
        // Code to perform an invalid login.
        System.out.println("Test: Invalid login.");
        // Assert statements to check if the login failed as expected.
    }

    @AfterMethod
    public void logout() {
        // Code to perform logout after each test.
        System.out.println("Logging out.");
    }

    @AfterClass
    public void closeWebDriver() {
```

```
        // Code to close the web driver and clean up resources.
        System.out.println("Closing the browser and cleaning up WebDriver.");
    }
}
```

pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>Pract6</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>19</maven.compiler.source>
        <maven.compiler.target>19</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>7.8.0</version> <!-- Use the latest version available -->
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>ch.qos.logback</groupId>
            <artifactId>logback-classic</artifactId>
            <version>1.2.3</version> <!-- Use the desired version -->
        </dependency>  </dependencies>

</project>
```

webapp-test-suite.xml:

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="WebAppTestSuite">
    <test name="WebAppLoginTest">
        <classes>
            <class name="WebAppLoginTest" />
        </classes>
    </test>
</suite>
```

Output:

Run webapp-test-suite.xml file in Intellij ide

```
"C:\Program Files\Java\jdk-19\bin\java.exe" ...
21:48:17.949 [main] DEBUG org.testng.TestNG - suiteXmlPath: "C:\Users\ROHAN RATHOD\AppData\Local\JetBrains\IdeaIC2023.2\temp-testng-custo
21:48:18.142 [main] INFO org.testng.internal.Utils - [TestNG] Running:
  C:\Users\ROHAN RATHOD\AppData\Local\JetBrains\IdeaIC2023.2\temp-testng-customsuite.xml

Setting up WebDriver and opening the browser.
Navigating to the login page.
Test: Invalid login.
Logging out.
Navigating to the login page.
Test: Valid login.
Logging out.
Closing the browser and cleaning up WebDriver.

===============================================
Default Suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================


Process finished with exit code 0
```

Conclusion: Hence Successfully implemented and demonstrated the testing of different web applications using testNg.

<table>
<tr><td colspan="4">Name of Student : DESHMUKH PRANAV DINESH</td></tr>
<tr><td colspan="2">Roll Number : 12</td><td colspan="2">LAB Assignment Number: 13</td></tr>
<tr><td colspan="4">Title of LAB Assignment :Demonstrate data driven Framework.</td></tr>
<tr><td colspan="2">DOP : 19/10/2023</td><td colspan="2">DOS : 27/10/2023</td></tr>
<tr><td>CO Mapped :
C02</td><td>PO Mapped:
PO2,PO3,PO4,PO 5,PO7,P09, PSO1 & PSO2</td><td>Faculty Signature:</td><td>Marks :</td></tr>
</table>

Practical No: 13

Aim: Demonstrate data driven Framework.

Theory:

Data-Driven Framework:

A data-driven testing framework is a methodology that separates test scripts and test data. It allows you to execute the same test script with multiple sets of data. Here's how it works:

- **Test Data:** You maintain test data separately, typically in external files like Excel spreadsheets, CSV files, or a database.
- **Test Scripts:** You create test scripts that are parameterized to accept data from the external sources. Instead of hardcoding data within the script, you fetch the data dynamically.
- **Execution:** The test framework runs the same test script multiple times, each time with a different set of test data. This allows you to perform tests with different input values and verify the results.

Data-driven frameworks are useful for running tests with various data permutations, which is common in scenarios like form validation, login testing, and more.

Code:

pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>

   <groupId>org.example</groupId>
   <artifactId>Pract6_2</artifactId>
   <version>1.0-SNAPSHOT</version>

   <properties>
      <maven.compiler.source>19</maven.compiler.source>
      <maven.compiler.target>19</maven.compiler.target>
      <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
   </properties>

   <dependencies>
      <!-- https://mvnrepository.com/artifact/org.testng/testng -->
      <dependency>
         <groupId>org.testng</groupId>
         <artifactId>testng</artifactId>
```

```
            <version>7.8.0</version>
            <scope>test</scope>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-api -->
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-api</artifactId>
            <version>4.14.1</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-chrome-driver -->
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-chrome-driver</artifactId>
            <version>4.14.1</version>
        </dependency>

        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-api</artifactId>
            <version>2.0.0-alpha1</version> <!-- Use the desired version -->
        </dependency>
        <dependency>
            <groupId>ch.qos.logback</groupId>
            <artifactId>logback-classic</artifactId>
            <version>1.2.6</version> <!-- Use the desired version -->
        </dependency>

        <dependency>
            <groupId>org.apache.poi</groupId>
            <artifactId>poi</artifactId>
            <version>5.2.0</version>
        </dependency>
    </dependencies>
</project>
```

WebDriverSetup.java:

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

public class WebDriverSetup {
    public static WebDriver initializeWebDriver() {
        System.setProperty("webdriver.chrome.driver",
"C:/Drivers/chromedriver-win64/chromedriver.exe");
        return new ChromeDriver();
    }
}
```

web-login-test.xml:

```
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="WebLoginSuite">
  <test name="WebLoginTest">
    <classes>
      <class name="WebLoginTest" />
    </classes>
  </test>
</suite>
```

WebLoginTest.java:

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

public class WebLoginTest {
  WebDriver driver;

  @BeforeTest
  public void setUp() {
    System.setProperty("webdriver.chrome.driver",
"C:/Drivers/chromedriver-win64/chromedriver.exe");
    driver = new ChromeDriver();
    driver.get("C:/Example/login.html");
  }

  @Test(dataProvider = "loginData")
  public void testLogin(String username, String password, String expectedMessage) {
    WebElement usernameField = driver.findElement(By.id("username"));
    WebElement passwordField = driver.findElement(By.id("password"));
    WebElement loginButton = driver.findElement(By.xpath("//input[@value='Login']"));
    WebElement loginMessage = driver.findElement(By.id("loginMessage"));

    usernameField.sendKeys(username);
    passwordField.sendKeys(password);
    loginButton.click();
    // Add a delay of 20 seconds
    try {
      Thread.sleep(20000); // 20 seconds
    } catch (InterruptedException e) {
      e.printStackTrace();
```

```
        }
        // Add assertions for the expected message
        Assert.assertEquals(loginMessage.getText(), expectedMessage);
    }

    @DataProvider(name = "loginData")
    public Object[][] provideLoginData() {
        return new Object[][]{
                {"user", "password", "Login Successful"},
                {"invalidUser", "wrongPassword", "Incorrect Credentials"},
                // Add more test data sets as needed
        };
    }

    @AfterTest
    public void tearDown() {
        driver.quit();
    }
}
```

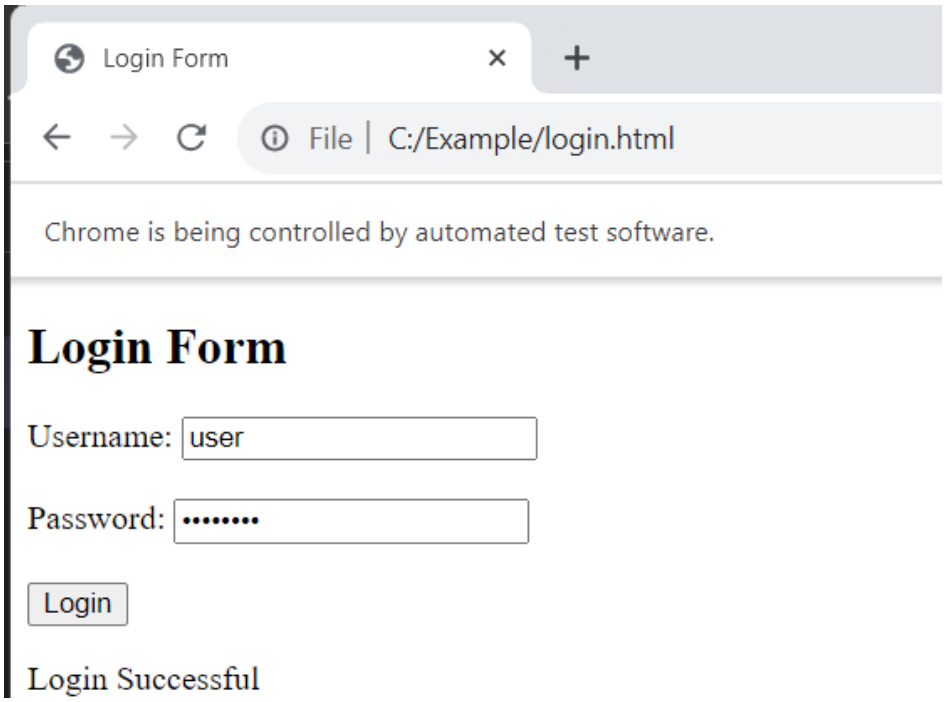
Login.html: (Path is given in the file WebLoginTest.java)

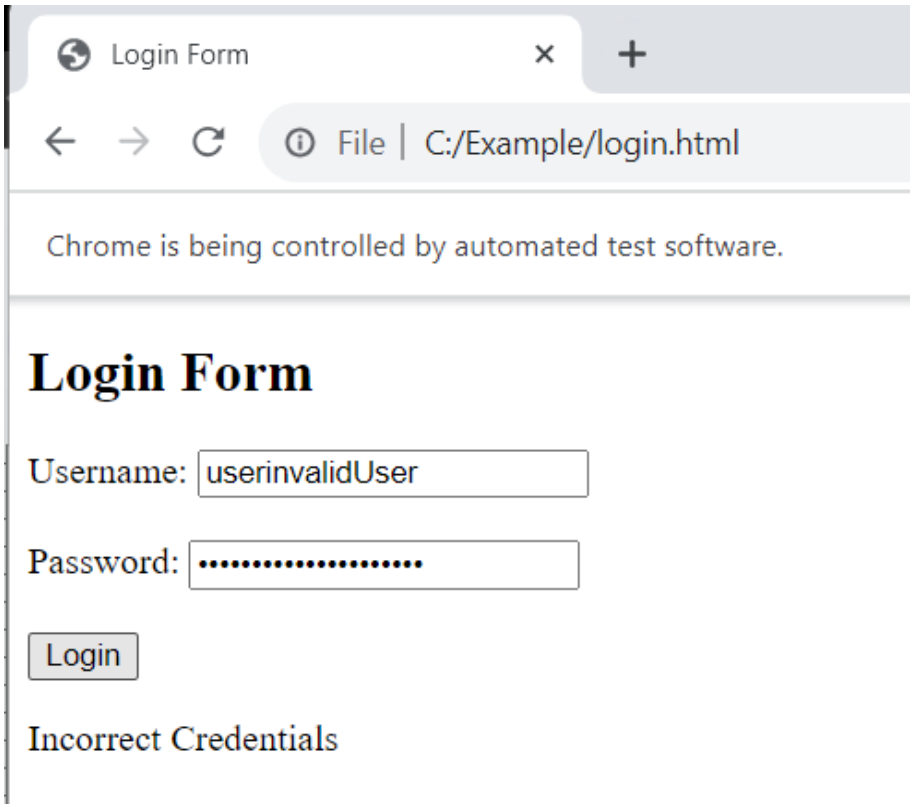
```
<!DOCTYPE html>
<html>
<head>
    <title>Login Form</title>
</head>
<body>
    <h2>Login Form</h2>
    <form id="loginForm" action="#">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username"><br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password"><br><br>
        <input type="button" value="Login" onclick="checkCredentials()">
    </form>
    <p id="loginMessage"></p>

    <script>
        function checkCredentials() {
            var username = document.getElementById("username").value;
            var password = document.getElementById("password").value;

            if (username === "user" && password === "password") {
                document.getElementById("loginMessage").innerHTML = "Login Successful";
            } else {
                document.getElementById("loginMessage").innerHTML = "Incorrect Credentials";
```

```
        }
      }
   </script>
</body>
</html>
```

Output:

To run the file add the file name in the class name in the web-login-test.xml file:






```
===============================================
Default Suite
Total tests run: 2, Passes: 2, Failures: 0, Skips: 0
===============================================


Process finished with exit code 0
```

Conclusion: Hence Successfully implemented and demonstrated the testing of different web applications using data driven framework using testng