

Bharati Vidyapeeth’s

**Institute of Management & Information Technology**

C.B.D. Belapur, Navi Mumbai 400614

**Vision:**

Providing high quality, innovative and value-based education in information technology to build

competent professionals.

**Mission**

M1. Technical Skills: To provide solid technical foundation theoretically as well as practically

capable of providing quality services to industry.

M2. Development: Department caters to the needs of students through comprehensive educational

programs and promotes lifelong learning in the field of computer Applications.

M3. Ethical leadership: Department develops ethical leadership insight in the students to succeed in

industry, government and academia

**CERTIFICATE**

This is to certify that the journal is the work of

**Mr. Mayur Vilas Bhujbal** Roll No. **34** of MCA

(Sem: - \_3\_ Div: -\_A\_) For the Academic Year 2021-2023

Subject Code: - MCAL35

Subject Name: Software Testing and Quality Assurance Lab

Subject-in-charge Principal

Date:

External Examiner

Date:

# Bharati Vidyapeeth's Institute of Management & Information Technology

**Academic Year – 2021-23**

**MCA Sem III Division A and B**

**PRACTICAL INDEX**

**MCAL35 SOFTWARE TESTING AND QUALITY ASSURANCE LAB**

**Name: Mayur Vilas Bhujbal Roll No.: 34 Div: A**

|  |  |  |
| --- | --- | --- |
| **Practical No** | **Problem Statement** | **Sign** |
| 1 | Write test cases for any known application Amazon. |  |
| 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) |  |
| 7 | Demonstrate the Locator (css selector) |  |
| 8 | Demonstrate the Locator (path) |  |
| 9 | Demonstrate synchronization in selenium |  |
| 10 | Demonstrate different types of alerts |  |
| 11 | Demonstrate: Handling Drop Down |  |
| 12 | Demonstrate: Handling List Boxes |  |
| 13 | Demonstrate Command Button, Radio buttons & text boxes. |  |
| 14 | Demonstrate Annotations using TestNG |  |
| 15 | Demonstrate Assertions in TestNG framework |  |
| 16 | Demonstrate data driven Framework. |  |
| 17 | Demonstrate ImplicitWait (using HRM\_Menu.Java) |  |

**Practical No.: 01**

**Aim: Write test cases for any known application Amazon.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case**  **No** | **Test Scenarios** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test Status** |
| tc\_01 | Check weather  the site is loading | Load the site. | Site=”https://w  ww  .amazon.in/” | Site should be loaded. | Site is Loaded. | Pass |
| tc\_02 | To search for a particular Item in the Search bar | Enter item name in the search bar. | Item name-  “Camera stand” | Related items Should displayed on the screen | Related items are displayed on the screen | Pass |
| tc\_03 | To check weather user is add able to same items to | Add item in the cart. | item name = "Item name" and site =  "https://www.a mazon  .in/gp/cart/view. html  ?ref\_=nav\_cart" | Items should be added in the cart and user | Items is be added in the cart and user continue d navigatin g another items | Pass |
| tc\_04 | To check whether item count in the  cart is  incrementing and their totals are displayed in the cart | Navigate to the same item Click on Add to Cart button | site =  "https://www.a mazon  .in/gp/cart/view. html  ?ref\_=nav\_cart"    Item – “Camera stand” Qty = 4 | After clicking on Add to cart button of same product the cart count should increment by 1 and price should get added to existing cart price | cart  count is incremen  t by 1 and price get added to existing cart price | Pass |
|  |  |  |  | Product should show inclusive of all taxes | Show's inclusive of all taxes |  |
|  |  |  | item = "camera stand" | Related items should be removed from the cart and item count | item removes from the cart and item |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | should decrement by 1 | count decreme  nted by 1 |  |
| tc\_05 | Verify weather tax as per location should applied | Click on any item in the cart and try to change the delievery location | **-** | Tax should be displayed with amount. | Amount  is  displayed | Pass |
| tc\_06 | To remove one of the item from the cart | Open cart and Remove one item from the cart. | Item – “Camera stand” Qty = 2 | Removed item should delete from cart | Removed  item is deleted from cart | Pass |
| tc\_07 | to check whether we can  proceed to buy | Open cart and click on Proceed to  Buy button | Link of cart = "https://www.a mazon  .in/gp/cart/view. html  /ref=dp\_atch\_ds s\_cart?” | The site should navigate user to sign-in | after clicking on Proceed to Buy button sign in option appers |  |
| tc\_08 | check different payment options don't checkout, close the site & come back later. the site should retain the items in the cart | close the site and reopen it | site =  "https://www.a mazon .in/" | The cart should retain the number of items | Cart symbol on th top right cornor shows same number  of items as previos session |  |
| tc\_09 | check if allowing checkout as guest, simply finish the purchase and provide an option to register. | Click on checkout button |  | The cart should retain the number of items | Cart symbol on the top right corner shows same number  of items as previous session |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| tc\_10 | To check whether sign in with mobile no. or email input  field appears |  |  | should allow to checkout as guest On the sign in page sign in with mobile no. or email input field should appear | Didn't allow to checkout as guest and redirecte d to sign in. |  |
|  | check whether input field is able to validate email id | Enter email id in the input  field | email =  "abc@gmail.co m" | Email should get validated | Email gets validated |  |
|  |  | Enter email id in the input  field without  @ symbol | email =  "abcgmail.com" | Email should not get validated and should show error message | Email doesn't get validated and shows message "Invalid mail id"  " |  |
|  |  | Enter email id in the input  field without  '.' | phone = 9632587415 | Phone number should get validate | Phone number validated |  |
|  | check whether input field is able to validate mobile number | Enter mobile number exactly 10 digit long in input field | phone = 9632587415 | Phone number should get validate | Phone number validated |  |
|  |  | Enter mobile number less than 10 digit in input field | phone = 9632587415 | Phone number should not get validate | Phone number didn't validate |  |
|  |  | Enter mobile number  greater than 10 digit in input field | phone = 78945623698 | Phone number should not get validate | Phone number  didn't validate |  |
|  | check whether continue button is working | Click on continue button |  |  |  |  |
| tc\_11 | check signup | On homepage ,from | site =  "https://www.a | Sign up page should open | Sign up page |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | accounts and  lists  dropdown  select start here | mazon.i n/" |  | opens |  |
|  | Check your name field | Enter your name | name = "abc" | field should not show any  error | Didn't show  error |  |
|  |  |  | name="abc 123" | field should not accept numbers as input | Field accepts numbers also as name |  |
|  | check whether input field is able to validate email id | Enter email id in there input field | email =  "abc077@gmail. com" | Email should get validated | Email gets validated |  |
|  |  | Enter email id in the input field without  @ symbol | email =  "abcgmail.com" | Email should not get validated and should show error message | Email doesn't get validated and shows message "Invalid mail id" |  |
|  | check whether input field is able to validate mobile number | Enter mobile number in input field | phone = 9632587415 | Phone number should get validate | Phone number validated |  |
|  |  | Enter mobile number in input field 9 numbers long | phone = 96325875 | Phone number should not get validate | Phone number didn't validate |  |
|  | Enter password  with 6 characters | Enter password with 6 characters | password=1234 56 | field should accept the password | field accepts  the password |  |
|  |  | Enter password greater than 6 characters | password=1234 56789 | field should accept the password | field accepts the password |  |
|  |  | Enter password less than 6 characters | password=1234 | field should not accept the password | field didnt accepts the |  |
|  |  |  |  |  | password |  |
|  |  | leave password field blank | password = " " | field should show error as required field | field gives error as required field |  |
|  | check password field Password again | Enter exact password as entered in Password  field |  | field should accept the password | field accepts the password |  |
|  |  | Enter password different than Password  field |  | Enter password different than Password field | field didn't accept  the password |  |

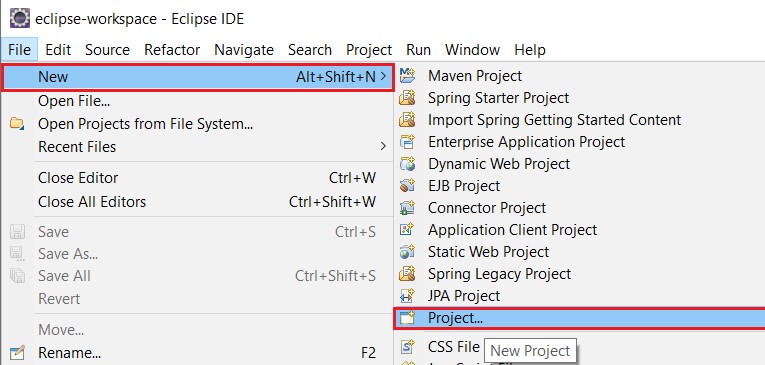
# Practical No.: 02

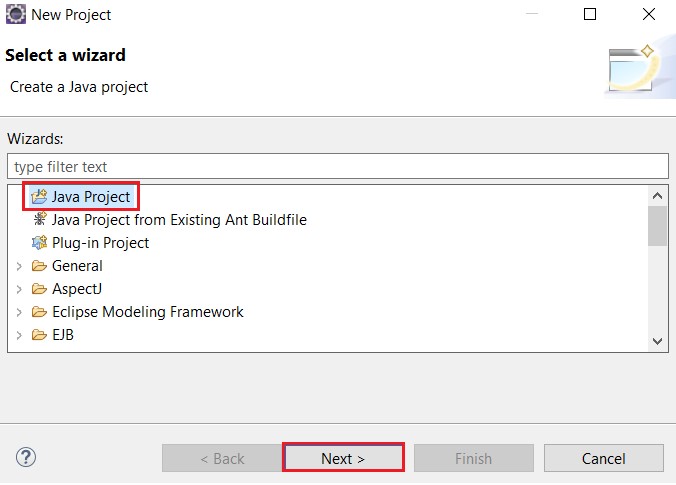
**Aim: Implement Web Drivers on Chrome & Firefox Browsers**.

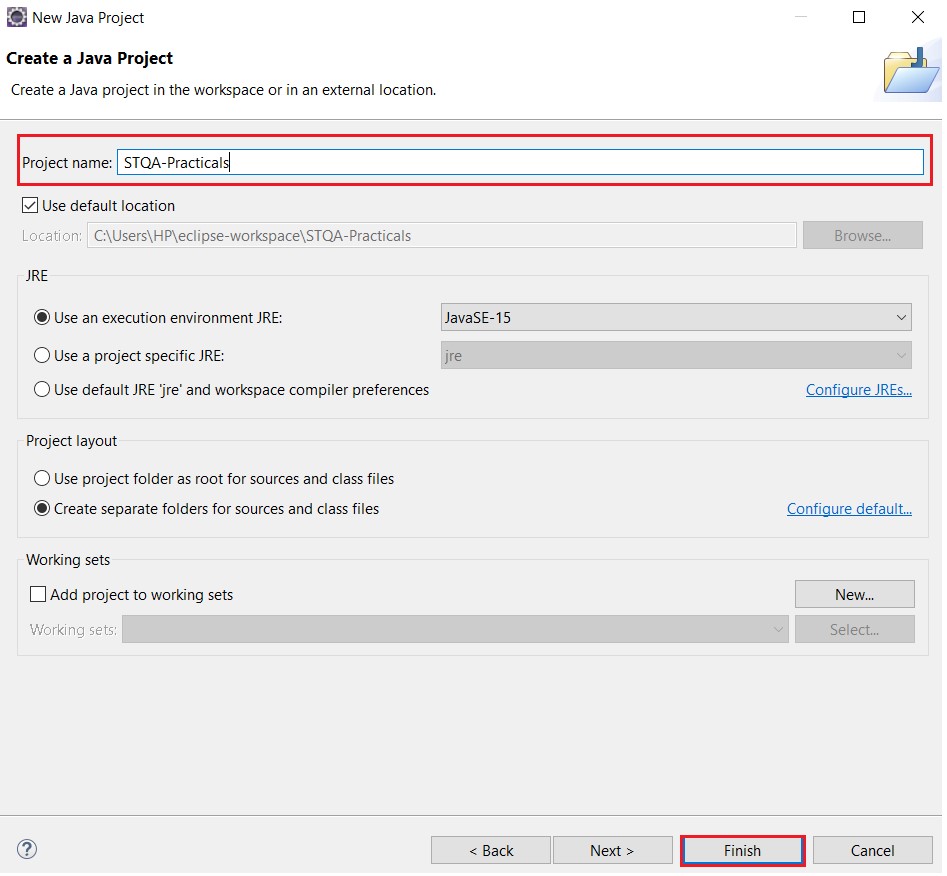
You need to download the following drivers to work with different browsers.

1. Firefox- Mozilla GeckoDriver
2. Selenium- Webdriver jar’s
3. Chrome- ChromeDriver

Open Eclipse→ Create Project→ give name→



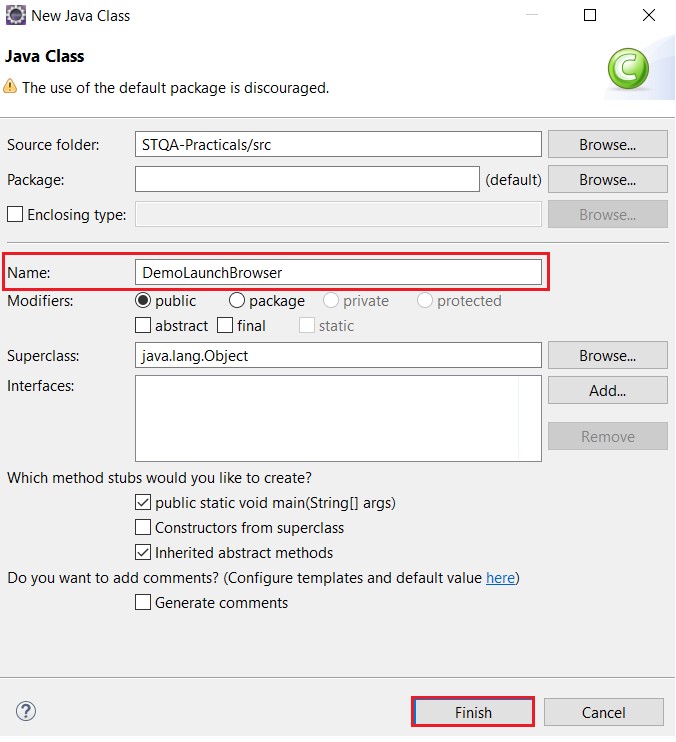




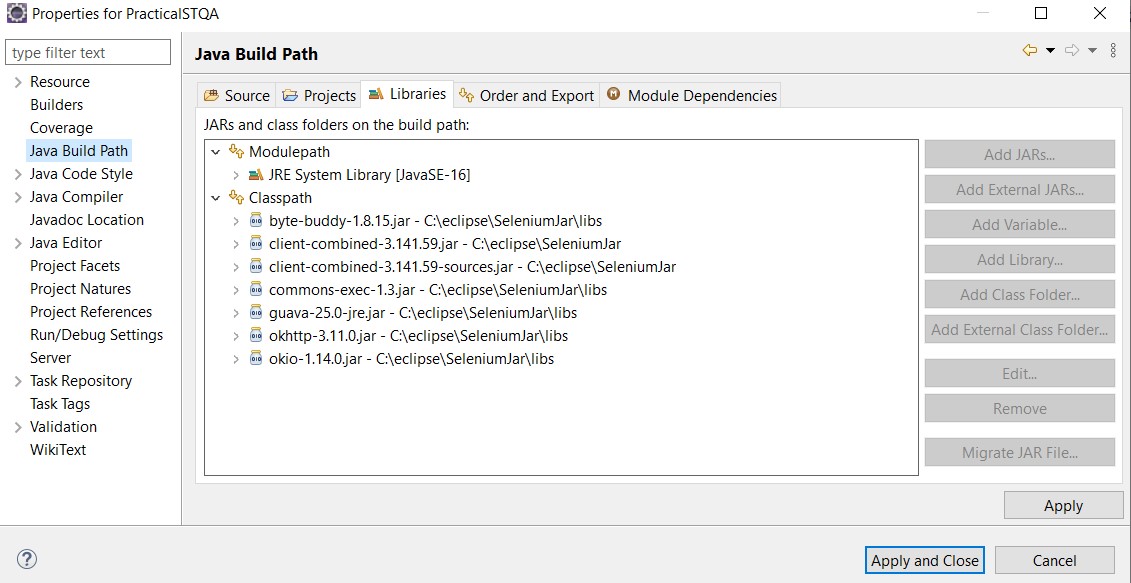
**Step1**. Right click on the "src" folder and create a new Class File from New > Class.



Give your Class name as "DemoLaunchBrowser" and click on "Finish" button.



Step2: Add jar files



**Program:**

**DemoLaunchBrowser.java**

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class DemoLaunchBrowser {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","

C:\\Eclipse\\Drivers\\chromedriver.exe ");

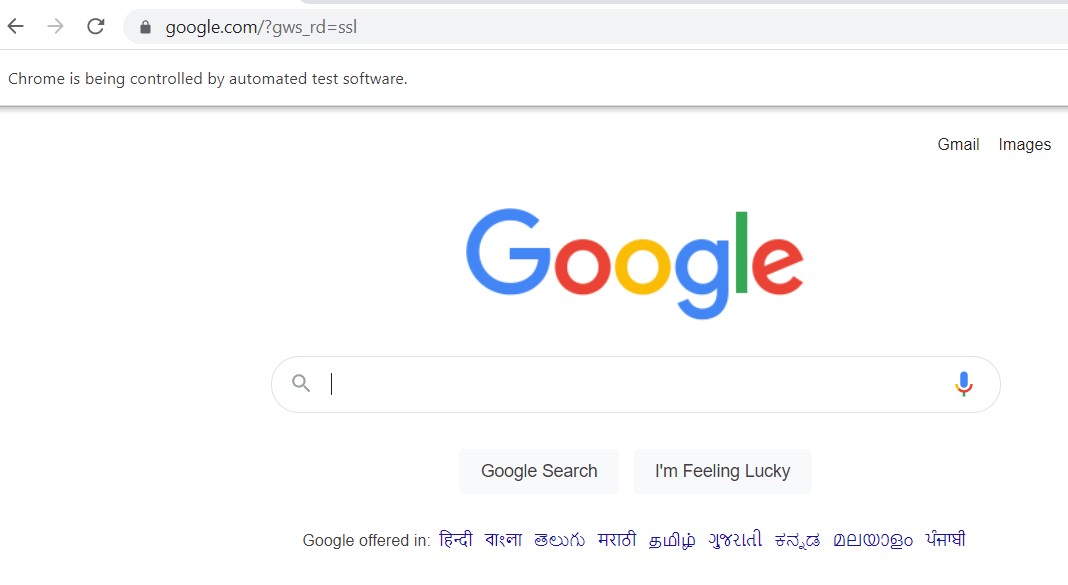
WebDriver wd = new ChromeDriver();

wd.get("http://google.com");

}

}

**Output:**



# Practical No.: 03

## Aim: Demonstrate handling multiple frames in selenium

**Program:**

package stqa;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import java.util.concurrent.TimeUnit;

public class MultipleFrameHandling {

public static void main(String[] args) {

System.*setProperty*("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe ");

WebDriver driver = new ChromeDriver();

String url = "https://the-internet.herokuapp.com/frames”;

driver.get(url);

driver.manage().timeouts().implicitlyWait(5, TimeUnit.*SECONDS*);

// identify element

driver.findElement(By.*linkText*("Nested Frames")).click();

// switch to frame with frame name and identify inside element driver.switchTo().frame("frame-bottom");

WebElement l = driver.findElement(By.*cssSelector*("body"));

System.*out*.println("Bottom frame text: " +l.getText());

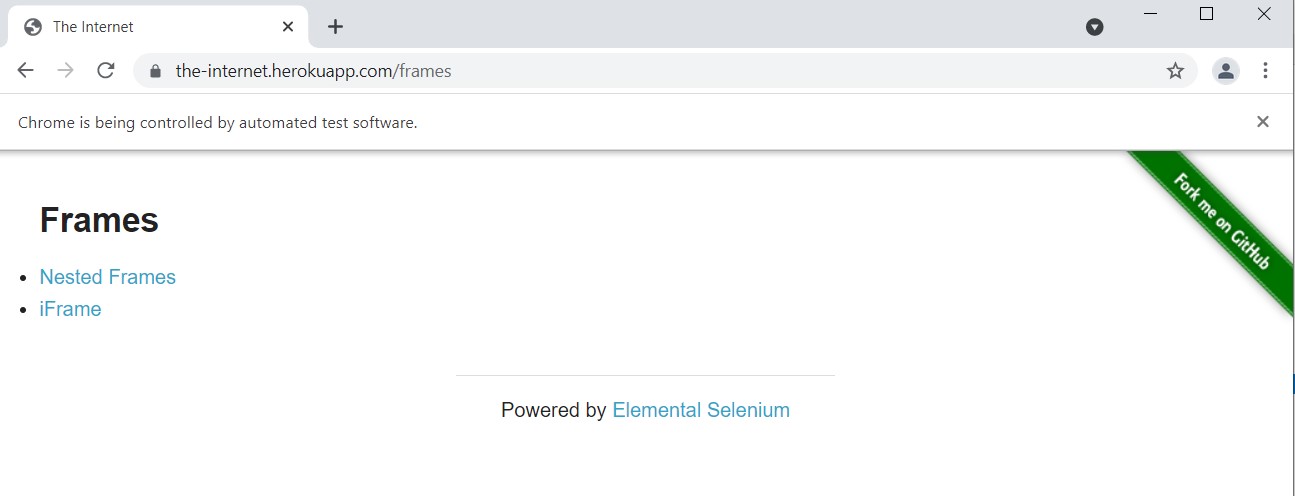
// switch to main page driver.switchTo().defaultContent();

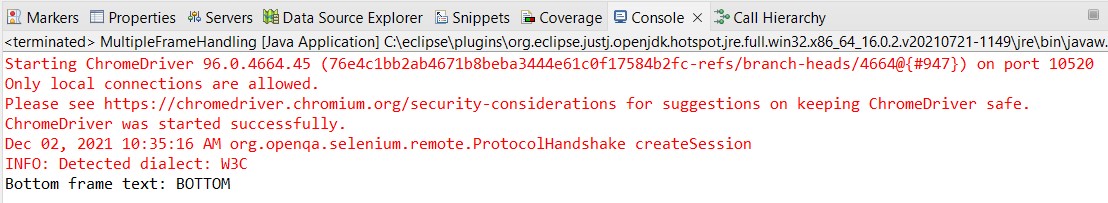
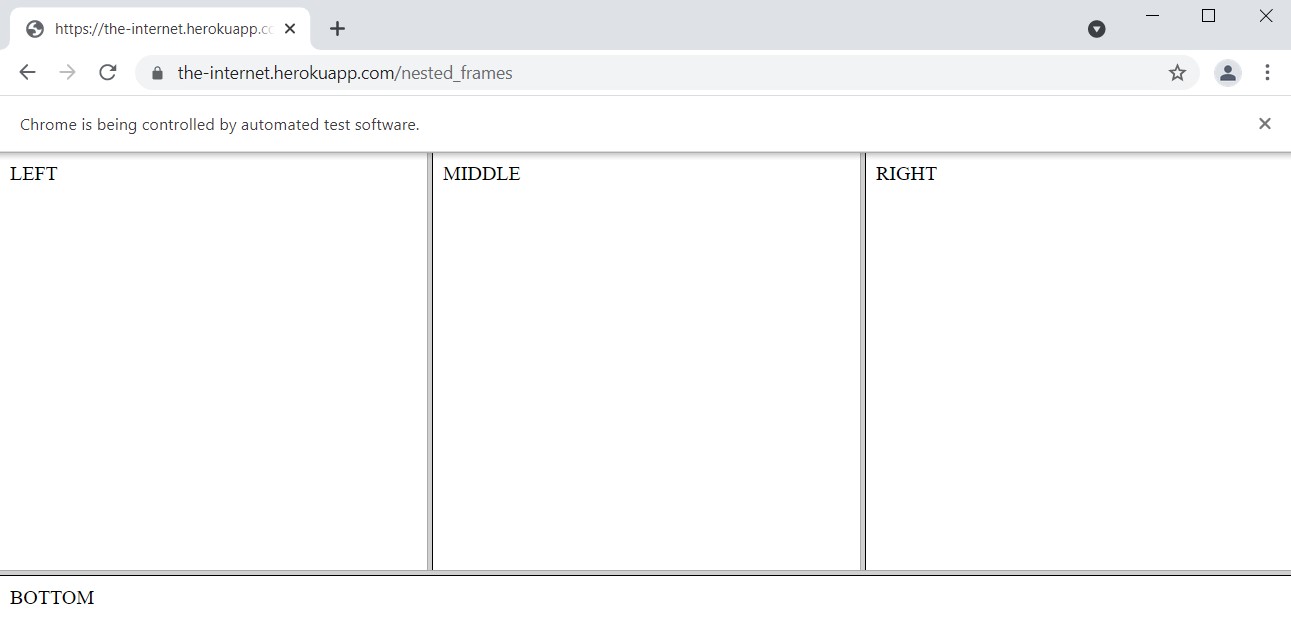
driver.quit();

}

}

**Output:**





# Practical No.: 04

**Aim: Implement Browser command and navigation Commands.**

**Program:**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.firefox.FirefoxDriver;

public class BrowserNavigationCommands {

public static void main(String[] args) {

// set path of the crome driver

System.setProperty("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe");

//Launch Firefox Broweser

WebDriver wd = new ChromeDriver();

//open URL of website

wd.get("https://www.techlistic.com");

//Maximize window

wd.manage().window().maximize();

//Navigate directly to some URL

wd.navigate().to("https://www.techlistic.com/p/java.html");

//Navigate Back

wd.navigate().back();

//Navigate Forward

wd.navigate().forward();

//Refresh Page

wd.navigate().refresh();

//close Browser

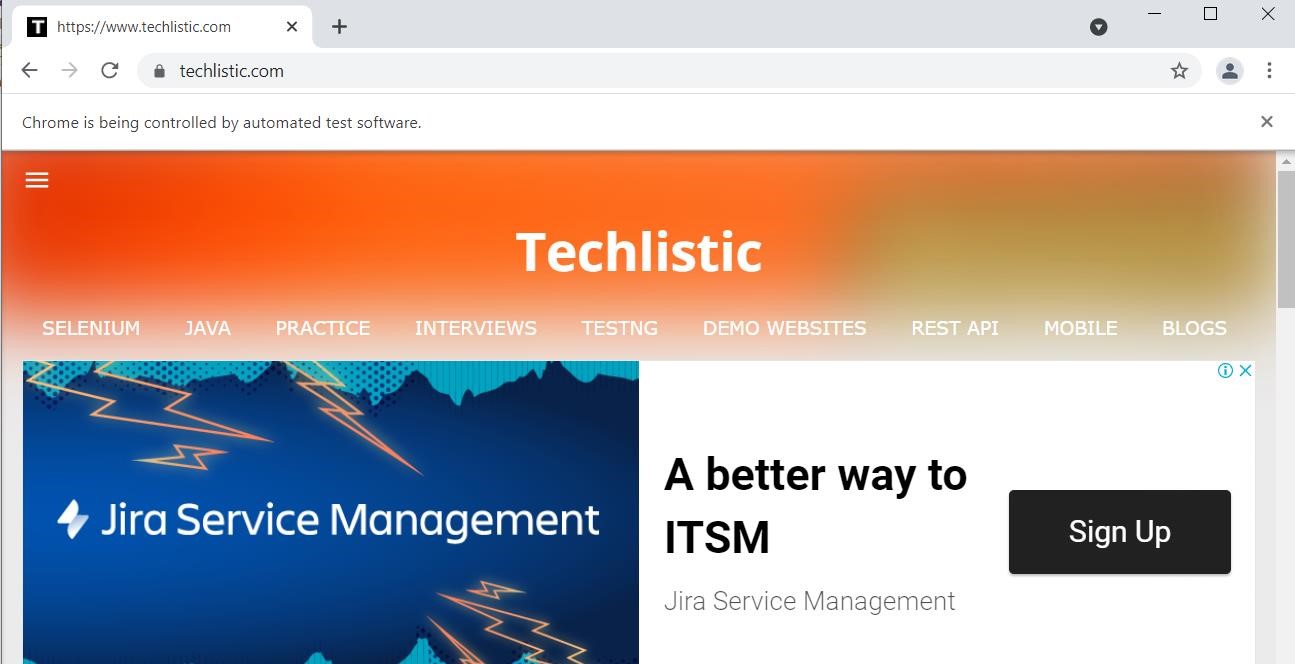
wd.close();

}

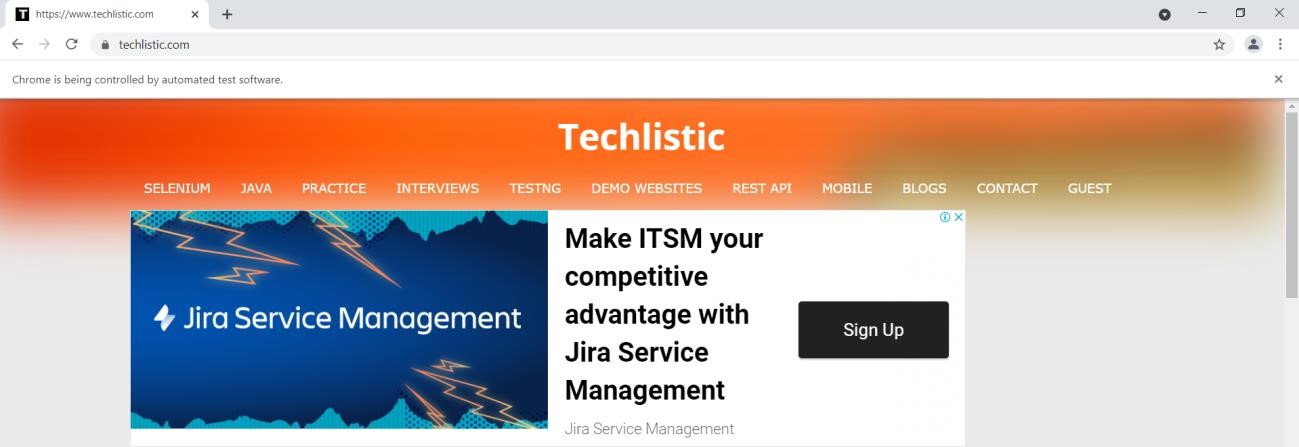
}

**Output:**

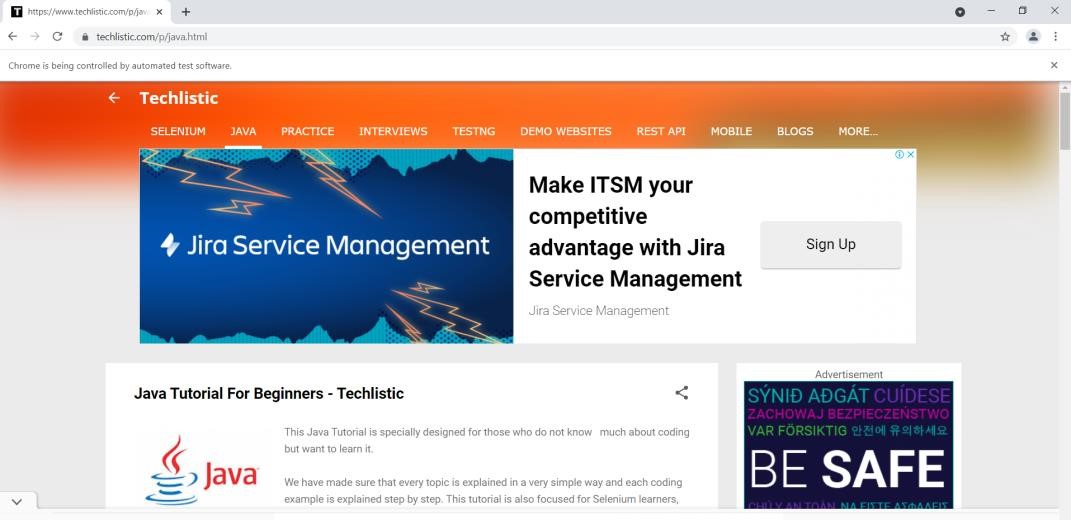
**Open URL of website**



**Maximize window**



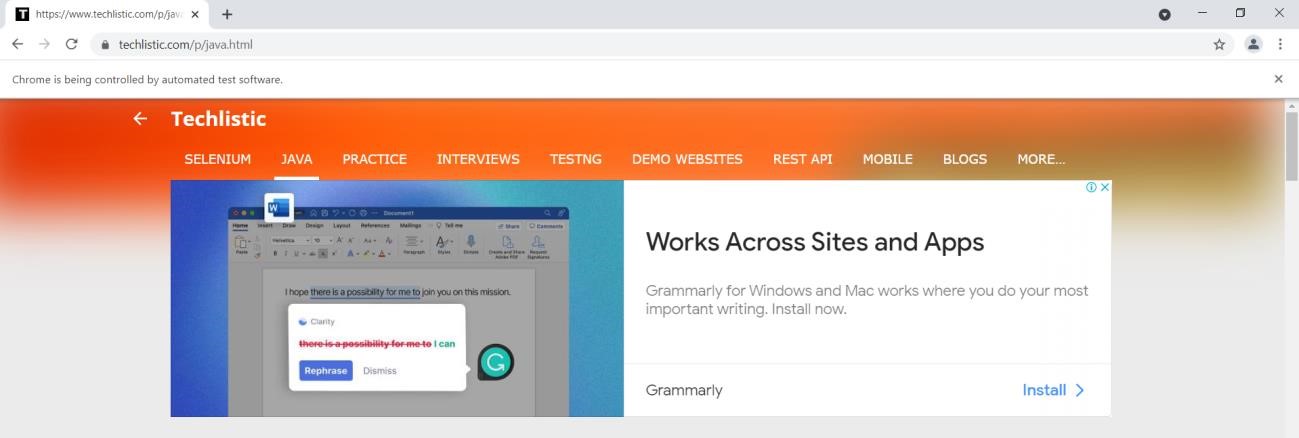
**Navigate directly to some URL – Java.html**



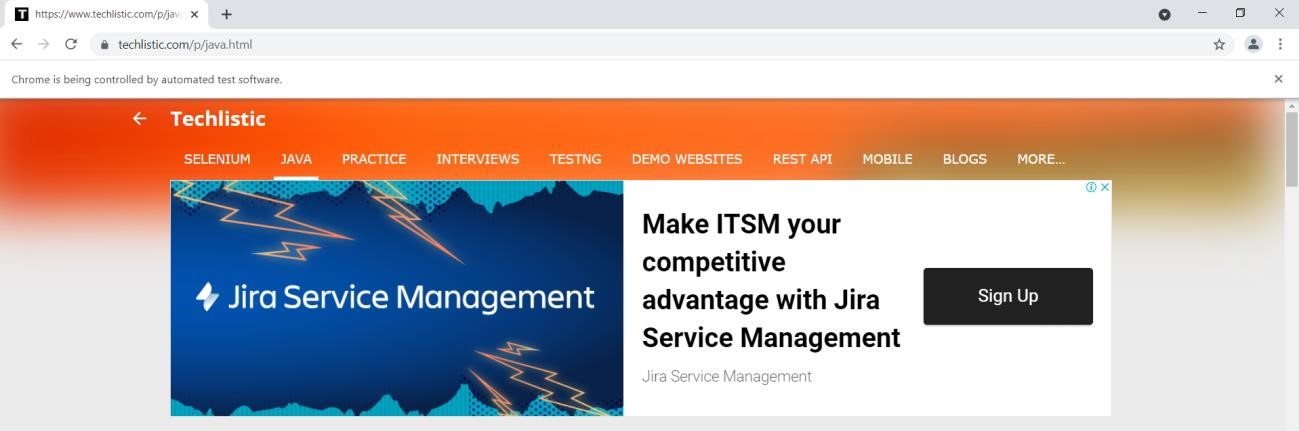
**Navigate Back**



**Navigate Forward**



**Refresh Page**



**Close Browser using** → **wd.close(); Browser automatically closed.**

## Practical No.: 05 Aim: Implement the find element command

**Program:**

**package** stqa;

**import** java.util.List;

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

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

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

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

**public** **class** FindElement {

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

System.*setProperty*("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver(); driver.get("https://demoqa.com/text-box/");

// Find elements using tag name

List<WebElement> allInputElements =

driver.findElements(By.*tagName*("input"));

**if**(allInputElements.size() != 0)

{

System.***out***.println(allInputElements.size() + " Elements found by TagName as input \n");

**for**(WebElement inputElement : allInputElements)

{

System.***out***.println(inputElement.getAttribute("placeholder"));

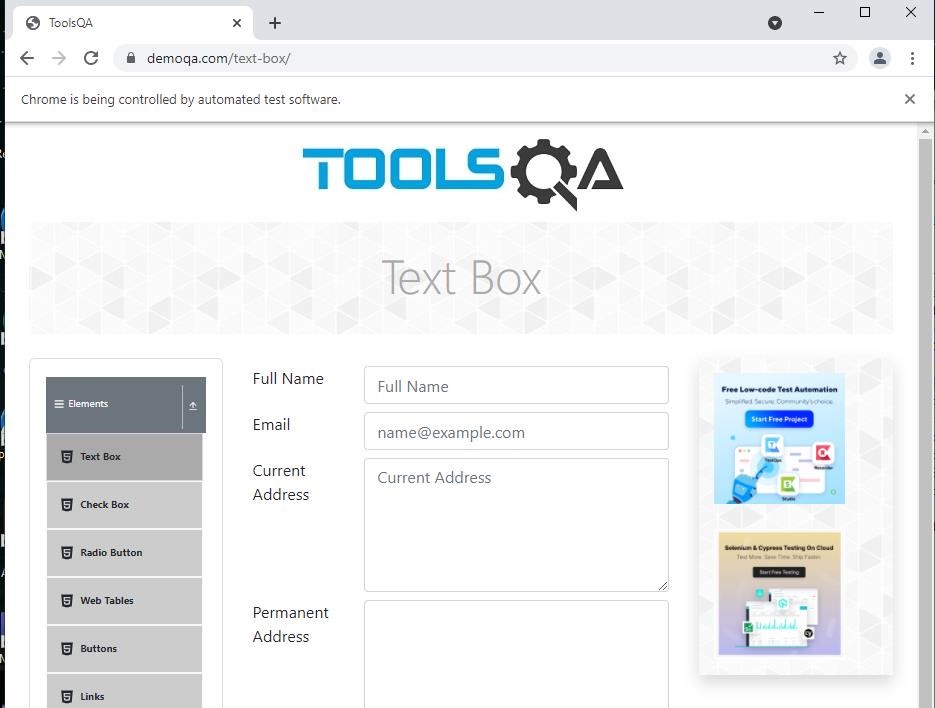
}

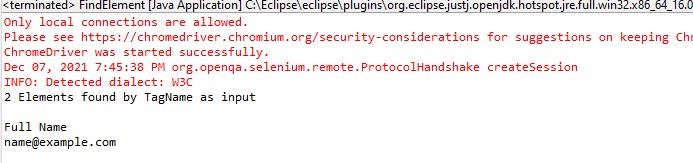
}

}

}

**Output:**





## Practical No.: 06 Demonstrate the Locator (id)

**Program:**

package stqa;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Locator {

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

{

//Locator---->id , name classname

System.setProperty("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe")

;

WebDriver wd = new ChromeDriver();

wd.get("https://opensource-demo.orangehrmlive.com/");

wd.findElement(By.id("txtUsername")).sendKeys("admin");//locator id wd.findElement(By.name("txtPassword")).sendKeys("admin123");//locater name wd.findElement(By.className("button")).click();//locator className

wd.findElement(By.partialLinkText("Welcome")).click();//locator partiallinkText

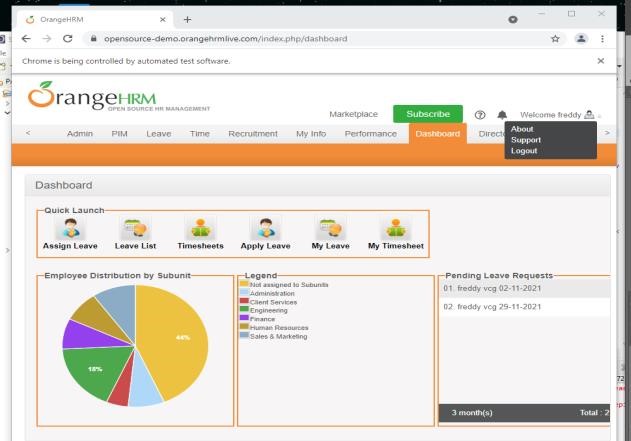
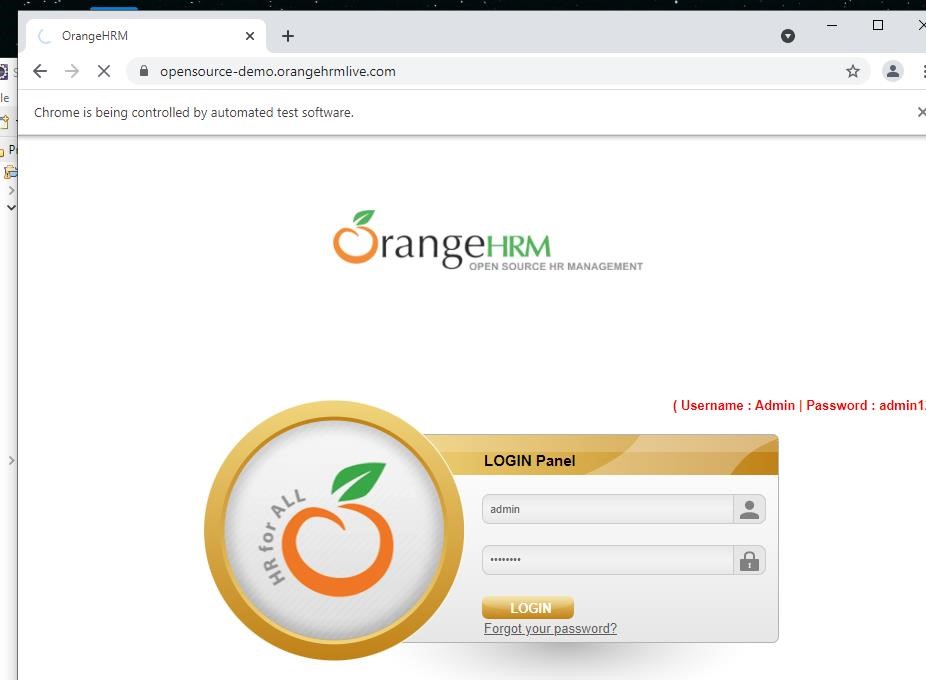
Thread.sleep(2000);

wd.findElement(By.linkText("Logout")).click();//locator linkText

}

}

**Output:**



## Practical No.: 07 Demonstrate the Locator (css selector)

**Program:**

package stqa;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class CssSelector {

public static void main(String[] args) {

//Demo cssSElector

System.setProperty("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe")

;

WebDriver wd = new ChromeDriver();

wd.get("https://opensource-demo.orangehrmlive.com/");

//tagname#id

wd.findElement(By.cssSelector("input#txtUsername")).sendKeys("admin"); //tagname[attribute=value of attribute]

wd.findElement(By.cssSelector("input[type=password]")).sendKeys("admin123");

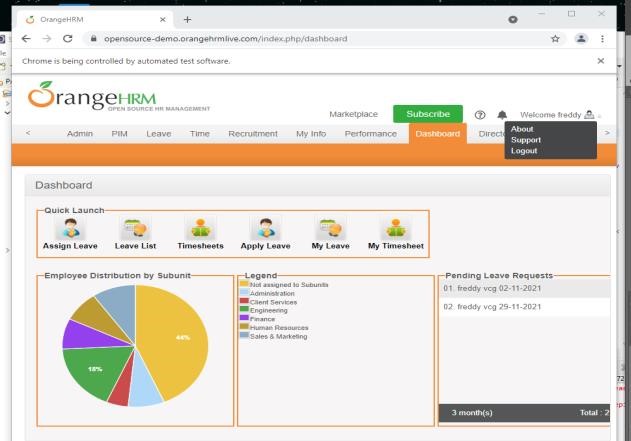
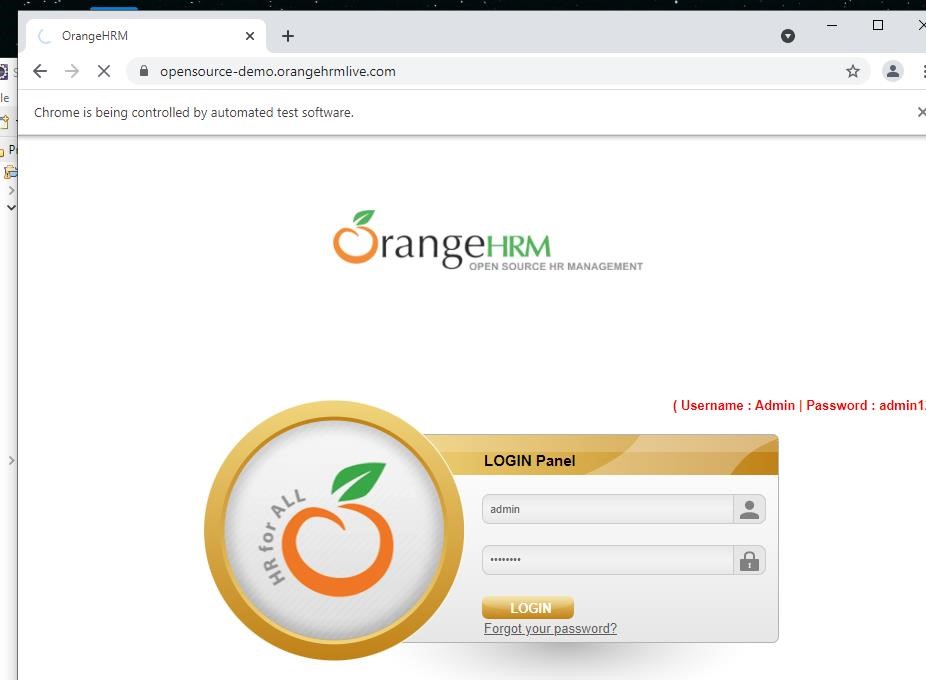
//tabname.classname

wd.findElement(By.cssSelector("input.button")).click();

}

}

**Output:**



## Practical No.: 08 Demonstrate the Locator (path)

**Right click**→**copy**→ **Copy full XPath**

/html/body/div[1]/div/div[3]/div[2]/div[2]/form/div[2]/input

**Right click**→**copy**→ **Copy XPath**

//\*[@id="txtPassword"]

**Program:**

package stqa;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class XPath {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe")

;

WebDriver wd = new ChromeDriver();

wd.get("https://opensource-demo.orangehrmlive.com/");

wd.findElement(By.xpath("/html/body/div[1]/div/div[3]/div[2]/div[2]/form/div[2]/input") ).sendKeys("admin");

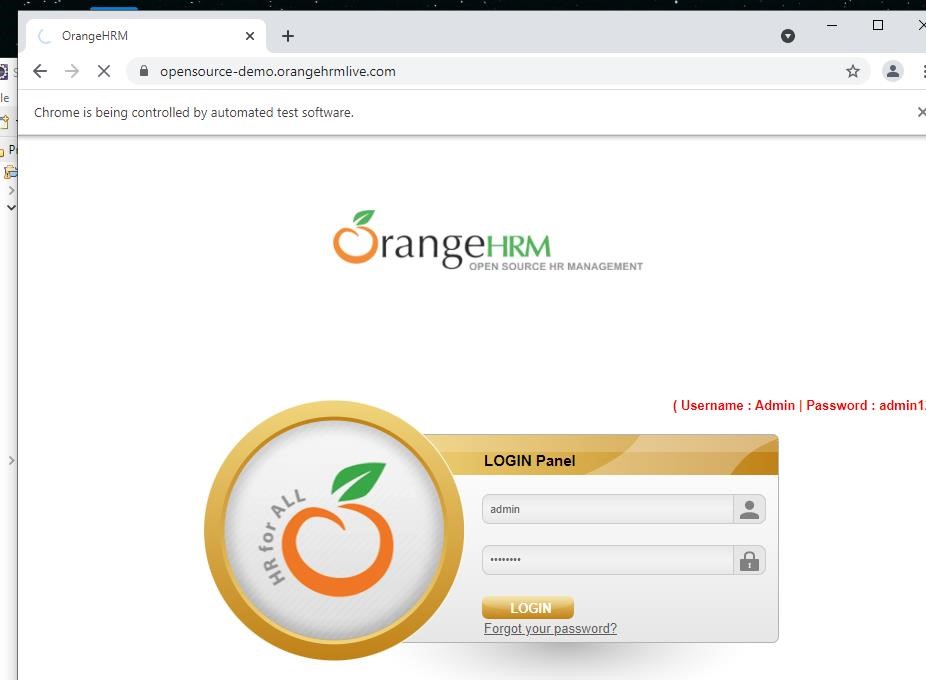
wd.findElement(By.xpath("//\*[@id=\"txtPassword\"]")).sendKeys("admin123");

//tagname[@attribute=value of attribute] wd.findElement(By.xpath("//input[@value='LOGIN']")).click();

}

}

**Output:**



## Practical No.: 09 Aim: Demonstrate synchronization in selenium

Synchronization can be classified into two categories:

1. **Unconditional**
2. **Conditional Synchronization**

**Unconditional:**

In this we just specify timeout value only. We will make the tool to wait until certain amount of time and then proceed further. Examples: Wait() and [Thread.Sleep();](https://docs.oracle.com/javase/tutorial/essential/concurrency/sleep.html)

**Program:**

package stqa;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Unconditional {

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

// Setting up browser executable using WebDriverManager

System.*setProperty*("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe")

;

WebDriver wd = new ChromeDriver();

// Loading a URL

wd.get("https://www.redbus.in/");

// Locating and typing in From text box.

WebElement fromTextBox = wd.findElement(By.*id*("src"));

fromTextBox.sendKeys("Ban");

// Put a sleep wait

Thread.*sleep*(5000);

// Clicking on first search result

wd.findElement(By.*xpath*("//li[@select-id='results[0]']")).click();

// Let's print the select value

String selectedCity = wd.findElement(By.*id*("src")).getAttribute("value");

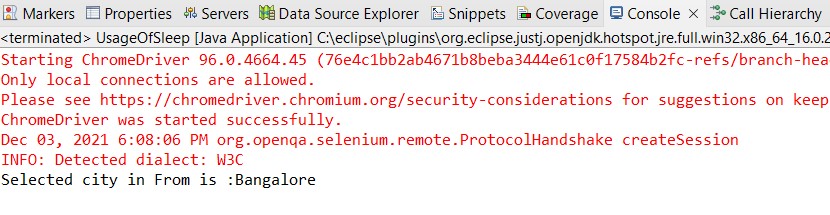
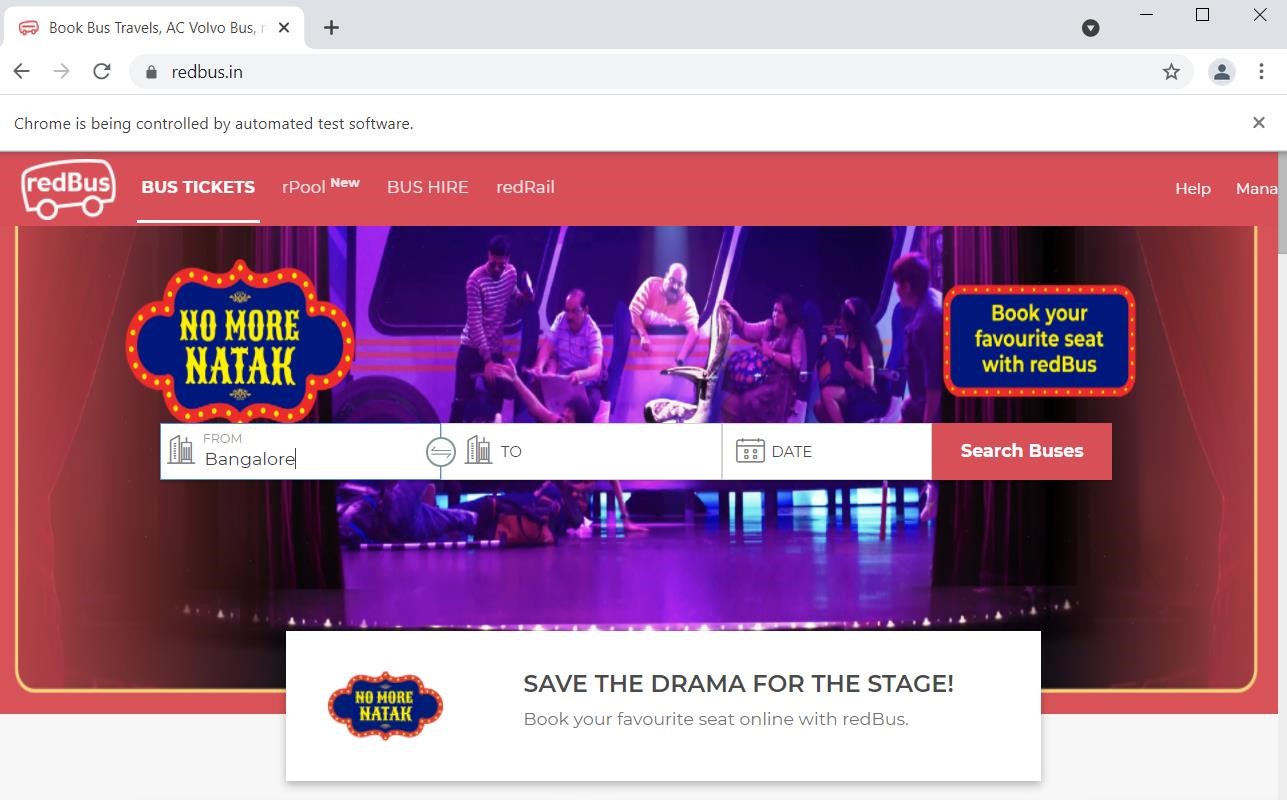
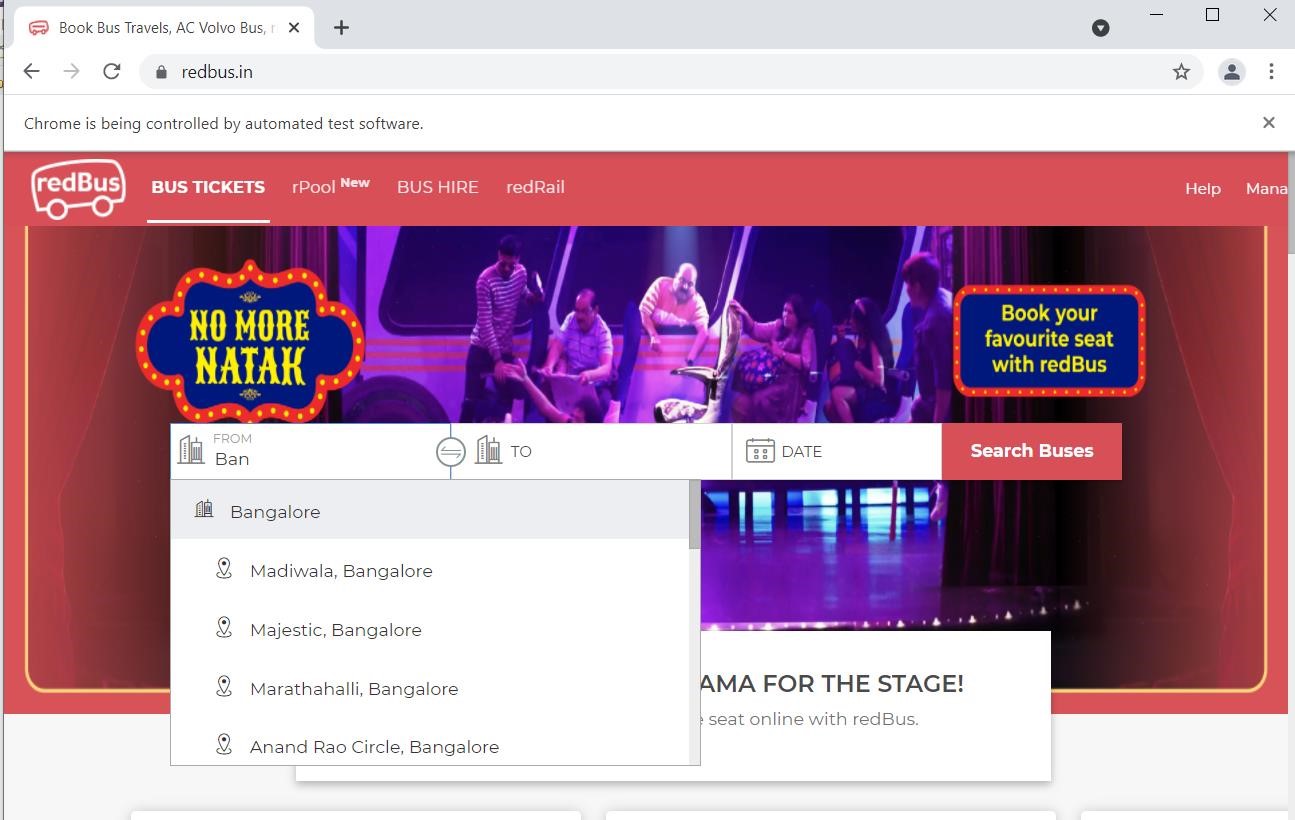
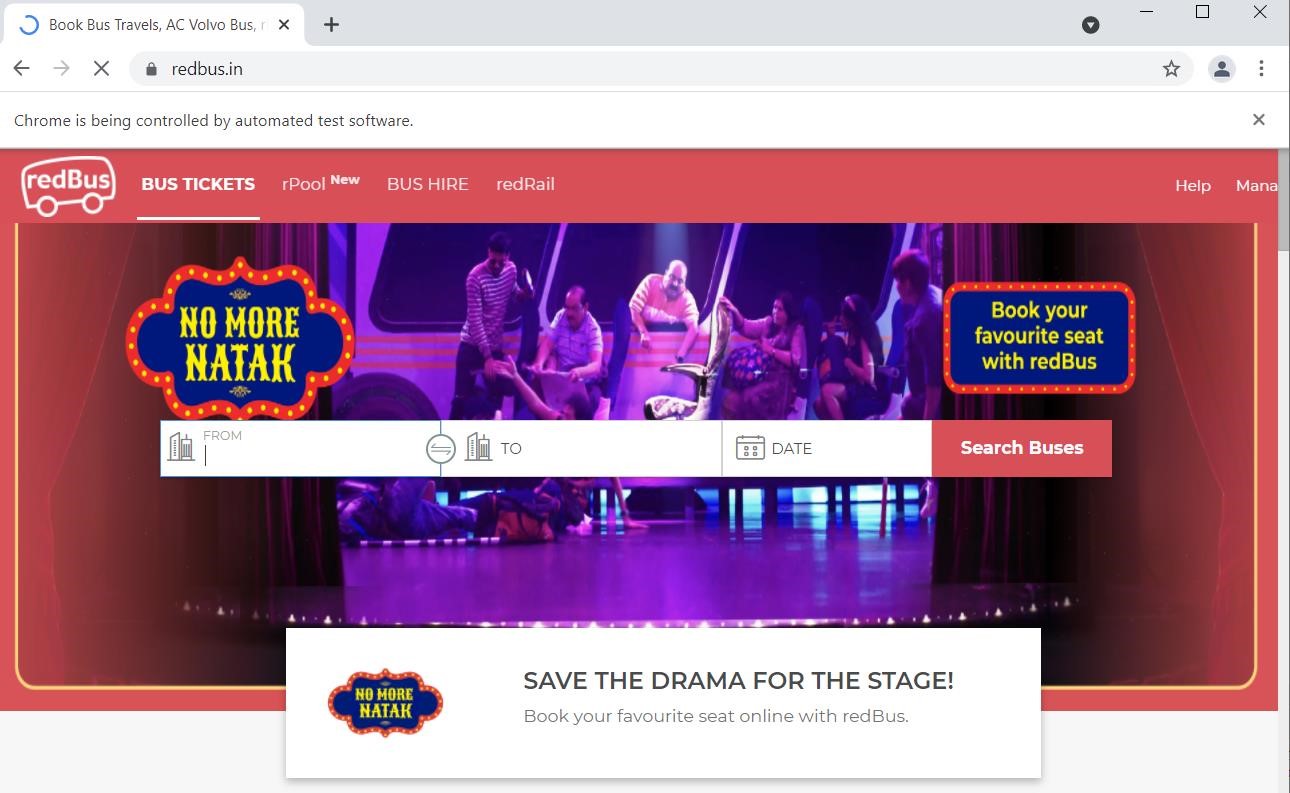
System.*out*.println("Selected city in From is :"+selectedCity);

// Closing browser

wd.quit();

}}

**Output:**



**Conditional Synchronization**

We specify a condition along with timeout value, so that tool waits to check for the condition and then come out if nothing happens.

It is very important to set the timeout value in conditional synchronization, because the tool should proceed further instead of making the tool to wait for a particular condition to satisfy

**Implicit Wait:**

**Program:**

package stqa;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class ImplicitWait {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe")

;

WebDriver wd = new ChromeDriver();

wd.get("https://opensource-demo.orangehrmlive.com/");

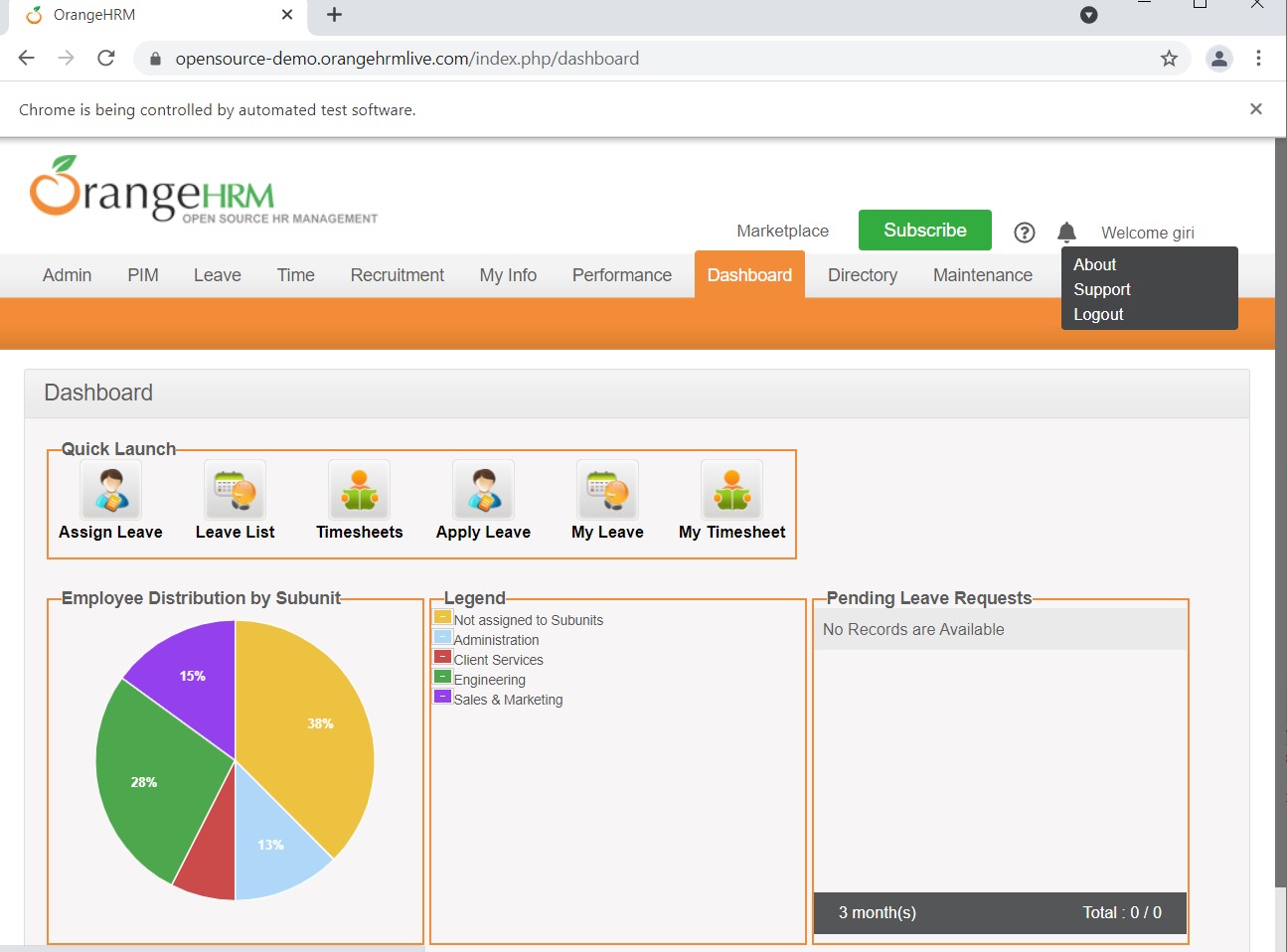
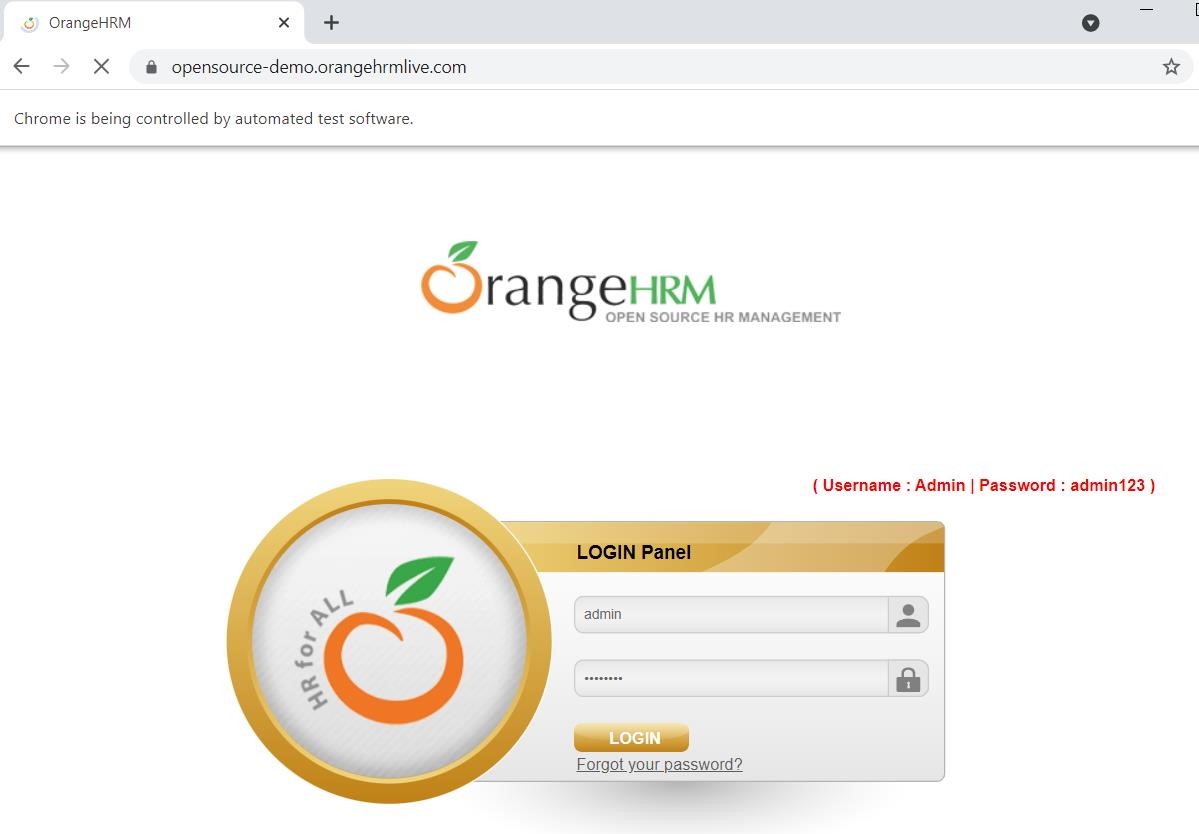
wd.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);//implicit wait wd.findElement(By.id("txtUsername")).sendKeys("admin");//locator id wd.findElement(By.name("txtPassword")).sendKeys("admin123");//locater name wd.findElement(By.className("button")).click();//locator className

wd.findElement(By.partialLinkText("Welcome")).click();//locator partiallinkText

wd.findElement(By.linkText("Logout")).click();//locator linkText

}

}



**Explicit Wait:**

**Program:**

package stqa;

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

public class ExplicitWait {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe")

;

WebDriver wd = new ChromeDriver();

wd.get("https://opensource-demo.orangehrmlive.com/");

WebDriverWait wt = new WebDriverWait(wd,10);

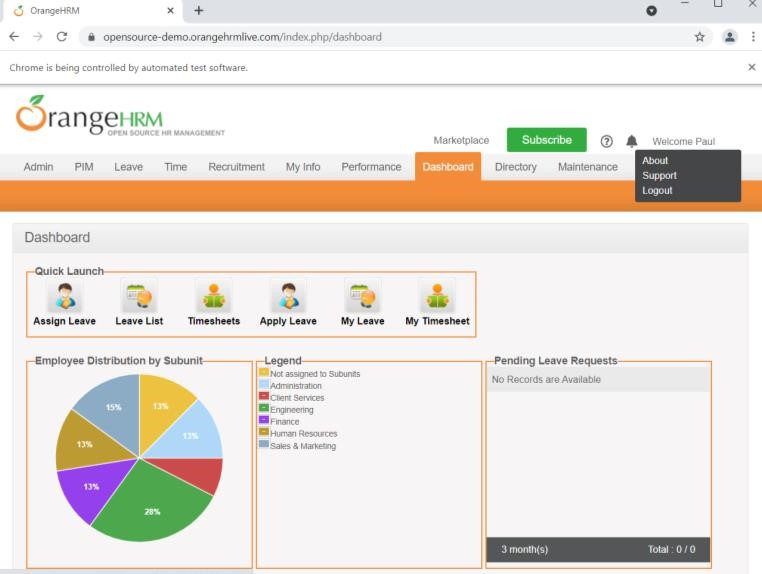
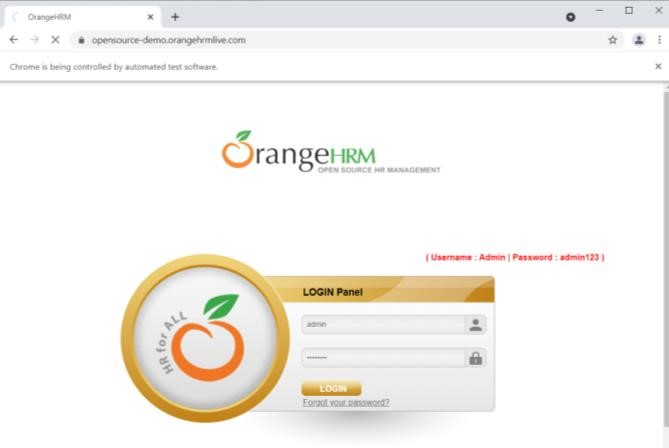
wd.findElement(By.id("txtUsername")).sendKeys("admin");//locator id wd.findElement(By.name("txtPassword")).sendKeys("admin123");//locater name wd.findElement(By.className("button")).click();//locator className

wd.findElement(By.partialLinkText("Welcome")).click();//locator partiallinkText

wt.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Logout"))); wd.findElement(By.linkText("Logout")).click();//locator linkText

}

}



**Logout-**



## Practical No.: 10

## Aim: Demonstrate different types of alerts

<https://www.toolsqa.com/selenium-webdriver/alerts-in-selenium/>

**Program:**

package stqa;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

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

public class prac10 {

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

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","C:\\Selenium WebDriver\\ChromeDriver\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("https://demoqa.com/alerts");

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

// This step will result in an alert on screen

driver.findElement(By.id("alertButton")).click();

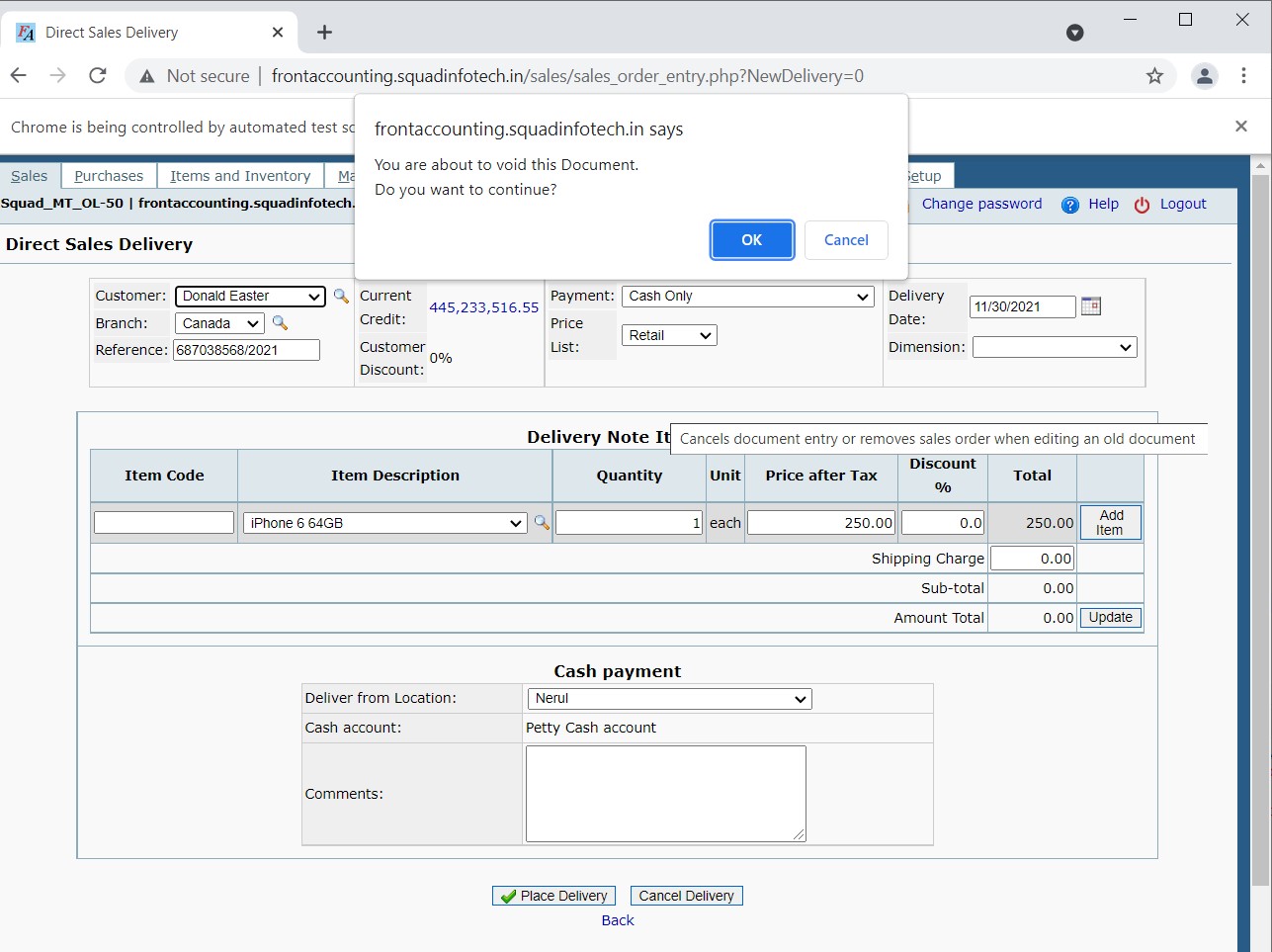
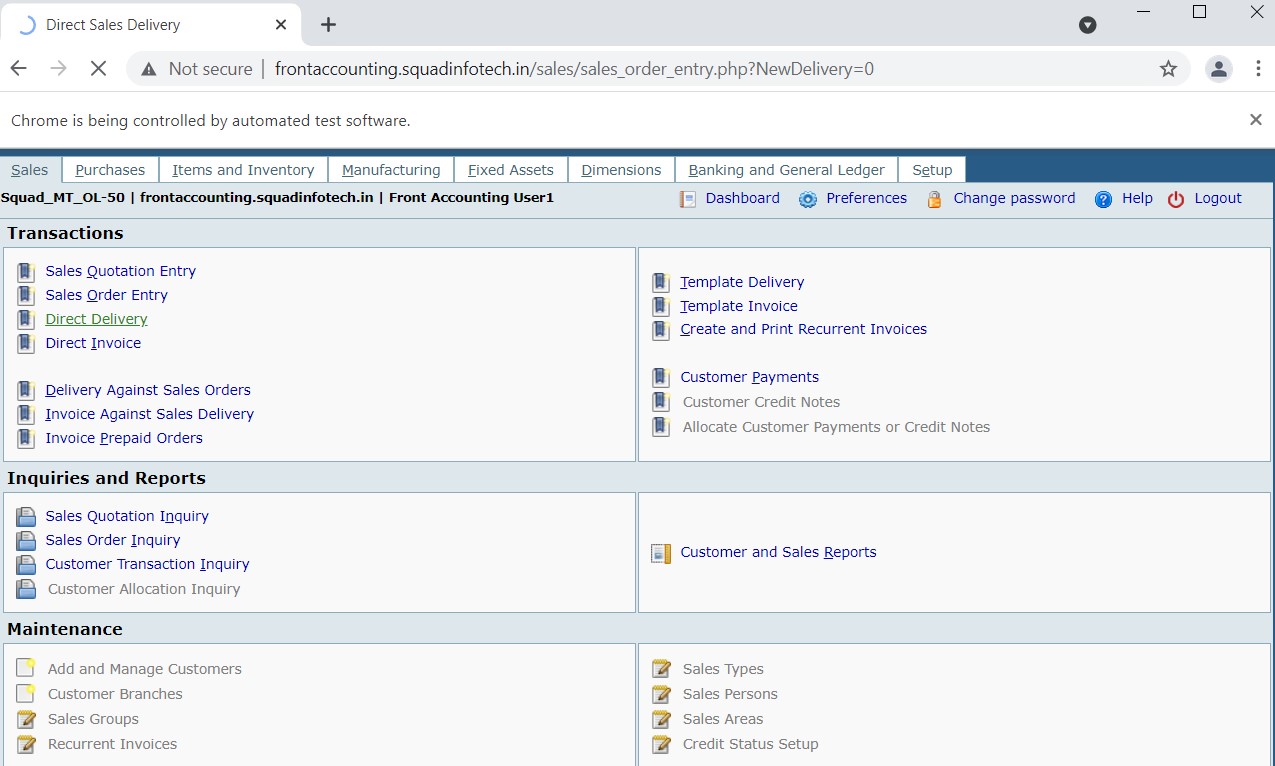
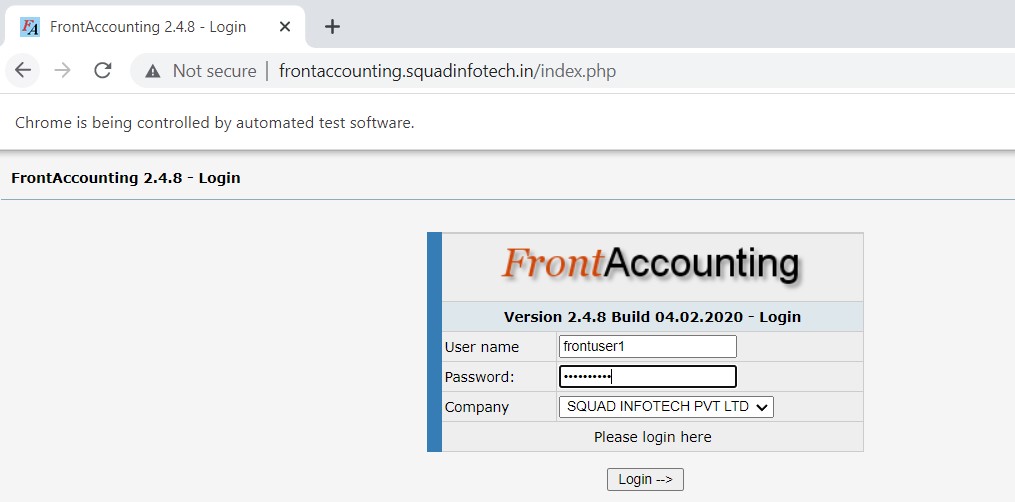
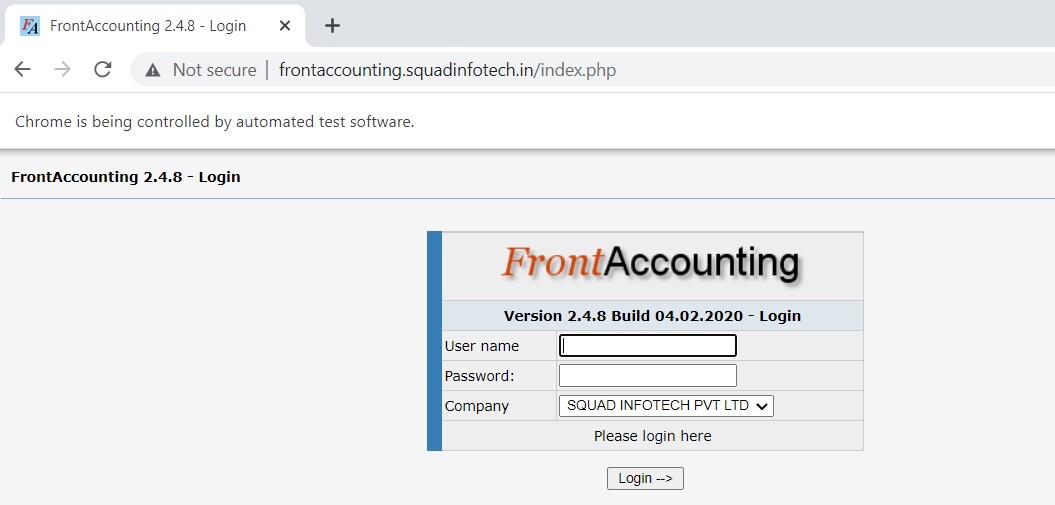
Alert simpleAlert = driver.switchTo().alert();

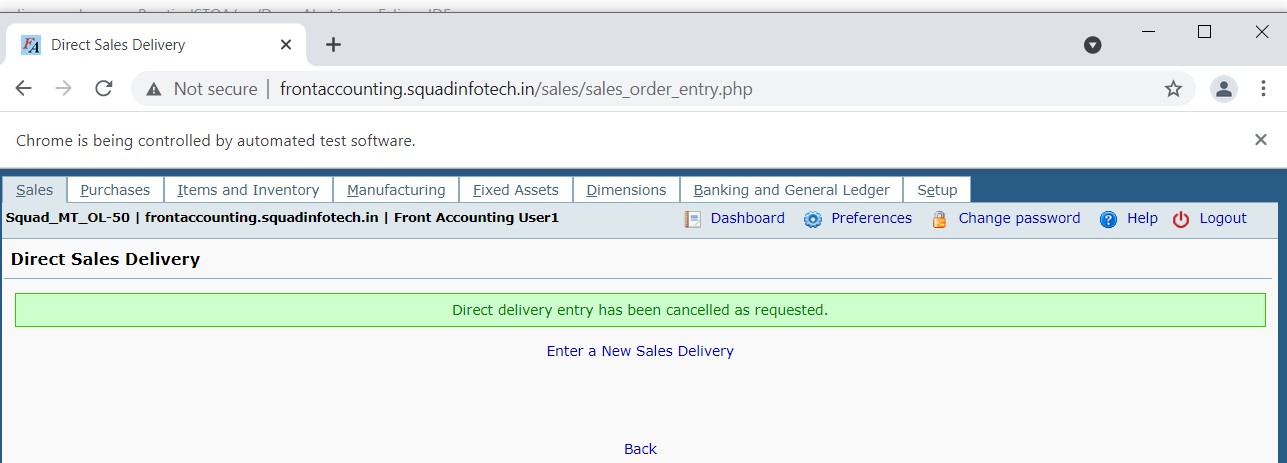
//simpleAlert.accept();

}

}

**Output:**







**Example 2:**

**Program:**

package stqa;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.support.ui.Select;

public class Alert2 {

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

// TODO Auto-generated method stub

System.*setProperty*("webdriver.chrome.driver",

"C:\\Eclipse\\Drivers\\chromedriver.exe"); WebDriver wd = new ChromeDriver();

wd.get("https://demoqa.com/alerts");

wd.findElement(By.*id*("alertButton")).click();

Thread.*sleep*(2000);

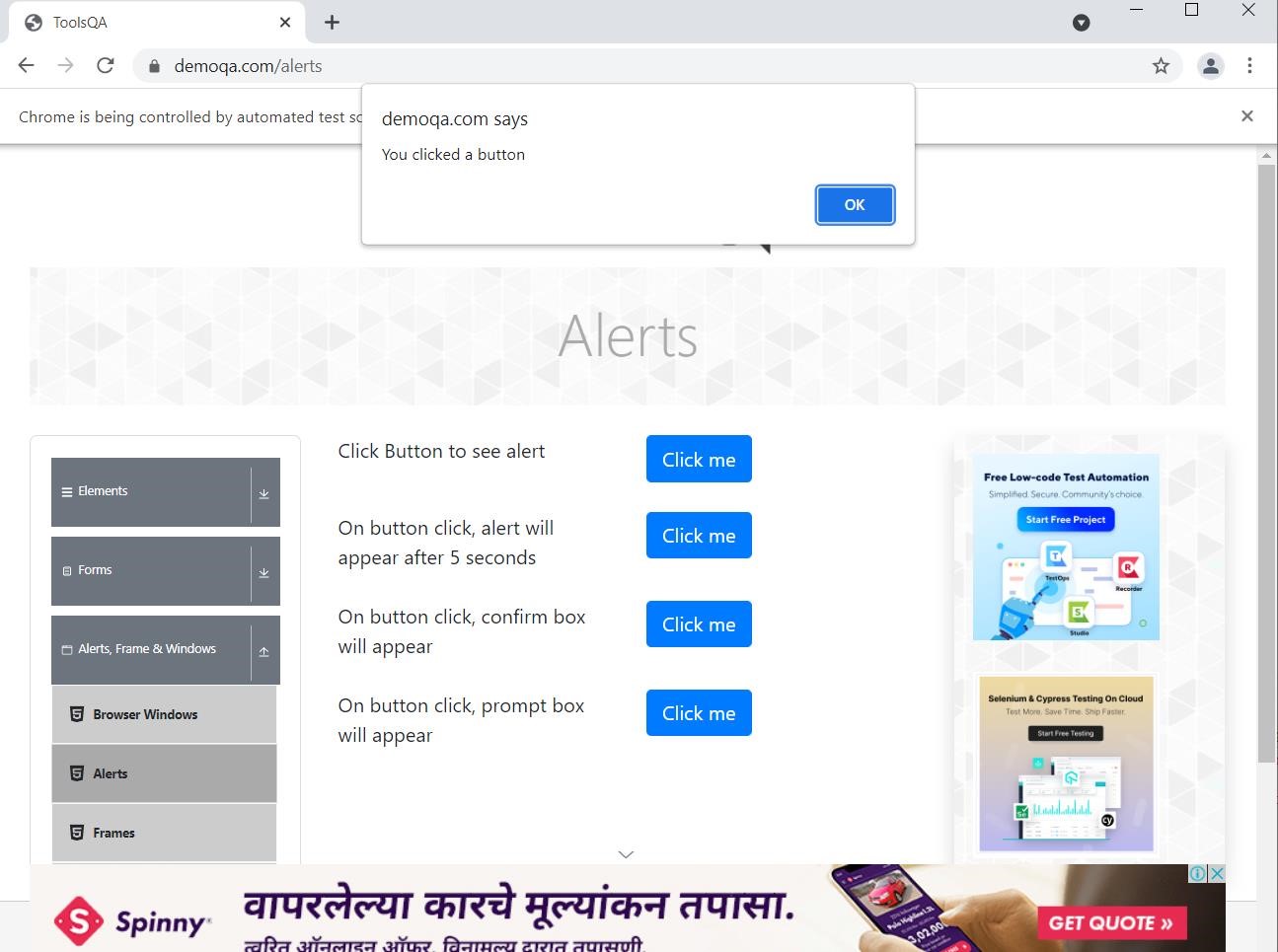
Alert alt = wd.switchTo().alert();

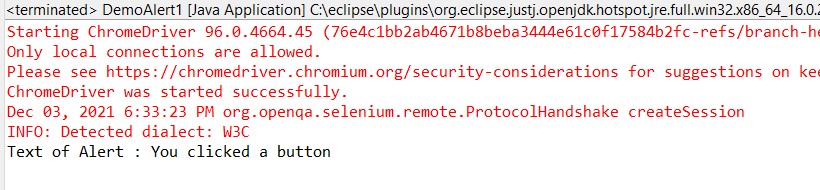
System.*out*.println("Text of Alert : " + alt.getText());//capture text alt.accept();//click on OK button //alt.dismiss(); click on cancel button

}

}

**Output:**





## Practical No.: 11 Aim: Demonstrate: Handling Drop Down

**Program:**

package stqa;

import org.openqa.selenium.By; import org.openqa.selenium.WebDriver; import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.support.ui.Select;

public class DropDown {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe")

;

WebDriver wd = new ChromeDriver();

wd.get("https://blazedemo.com/");

Select s = new Select(wd.findElement(By.name("fromPort")));

//

s.selectByIndex(1);

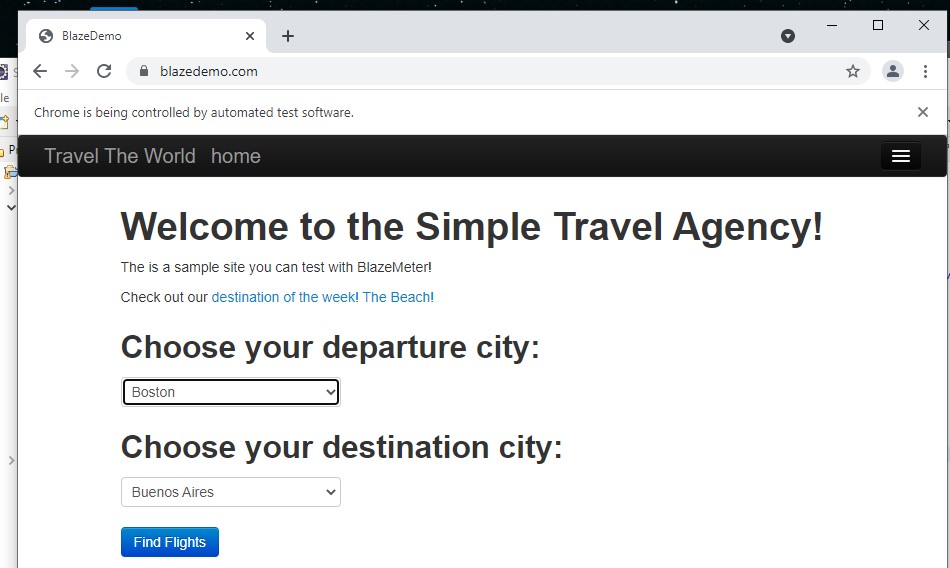
//s.selectByValue("Boston");

s.selectByVisibleText("Boston");

}

}

**Output:**



## Practical No.: 12 Aim: Demonstrate: Handling List Boxes

**Program:**

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver; import org.openqa.selenium.support.ui.Select; public class ListBoxes {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.*setProperty*("webdriver.chrome.driver","C:\\Selenium\\chromedriver.exe");

WebDriver wd = new ChromeDriver();

wd.get("file:///D:/SQLT/index.html");

wd.findElement(By.*name*("custname")).sendKeys("Shubham");

//Selecting the second checkbox using Xpath

//wd.findElement(By.xpath("/html/body/select/option[2]")).click();

Select s = new Select(wd.findElement(By.*name*("Cars")));

s.selectByIndex(1);

s.selectByVisibleText("Tesla");

}

}

<!DOCTYPE html>

<html>

<title>

Example of List Box

</title>

<body>

Customer Name:  <input type="text" Placeholder="Enter the Customer Name"/>

<br>

<br>

<select name="Cars" size="5">

    <option value="Merceders"> Merceders </option>

    <option value="BMW"> BMW </option>

    <option value="Jaguar"> Jaguar </option>

    <option value="Lamborghini"> Lamborghini </option>

    <option value="Ferrari"> Ferrari </option>

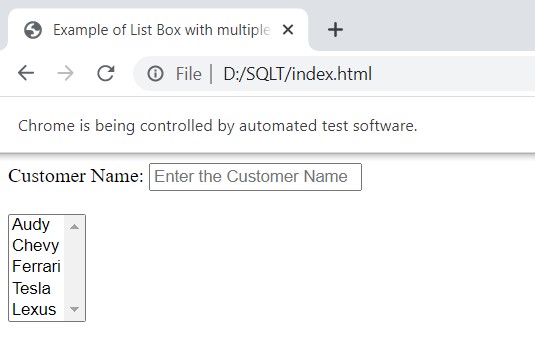
    <option value="Ford"> Ford </option>

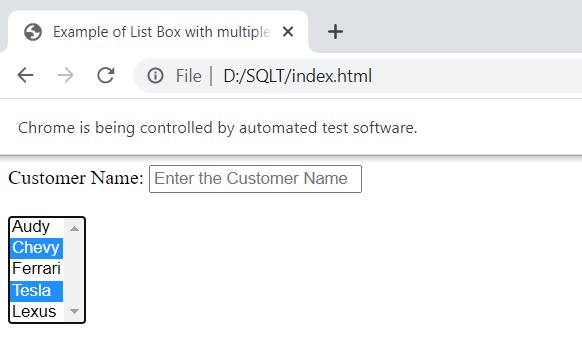
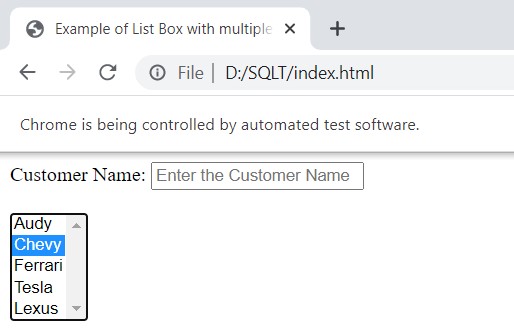
</select>

</body>

</html>

**Output:**





# Practical No.: 13

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

**Program:**

package stqa;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class prac13 {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("webdriver.chrome.driver","C:\\Selenium WebDriver\\ChromeDriver\\chromedriver\_win32\\chromedriver.exe");

WebDriver wd = new ChromeDriver();

wd.get("file:C:\\Users\\CAPTAIN\\OneDrive\\Desktop\\New folder\\Form.html");

// Maximize window

wd.manage().window().maximize();

wd.findElement(By.name("firstname")).sendKeys("mayur");

wd.findElement(By.name("middlename")).sendKeys("vilas");

wd.findElement(By.name("lastname")).sendKeys("bhujbal");

//wd.findElement(By.name("hobby[ ]")).click();

wd.findElement(By.name("Course")).click();

//Selecting the first checkbox using XPath

wd.findElement(By.xpath("/html/body/form/select/option[5]")).click();

//Selecting the second checkbox using Xpath

wd.findElement(By.name("male")).click();

wd.findElement(By.name("click")).click();

}

}

<Html>

    <head>

    <title>

    Registration Page

    </title>

    </head>

    <script>

        function showAlert() {

          var myText = "This can be whatever text you like!";

          alert (myText);

        }

        </script>

    <body>

    <br>

    <br>

    <form>

    <label> Firstname </label>

    <input type="text" name="firstname" size="15"/> <br> <br>

    <label> Middlename: </label>

    <input type="text" name="middlename" size="15"/> <br> <br>

    <label> Lastname: </label>

    <input type="text" name="lastname" size="15"/> <br> <br>

    <label>

    Course :

    </label>

    <select name="Course" size="6">

    <option value="BCA">BCA</option>

    <option value="BBA">BBA</option>

    <option value="B.Tech">B.Tech</option>

    <option value="MBA">MBA</option>

    <option value="MCA">MCA</option>

    <option value="M.Tech">M.Tech</option>

    </select>

    <br>

    <br>

    <label>

    Gender :

    </label><br>

    <input type="radio" name="male"/> Male <br>

    <input type="radio" name="female"/> Female <br>

    <input type="radio" name="other"/> Other

    <br>

    <br>

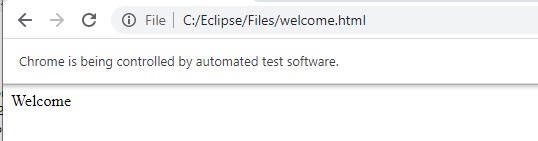
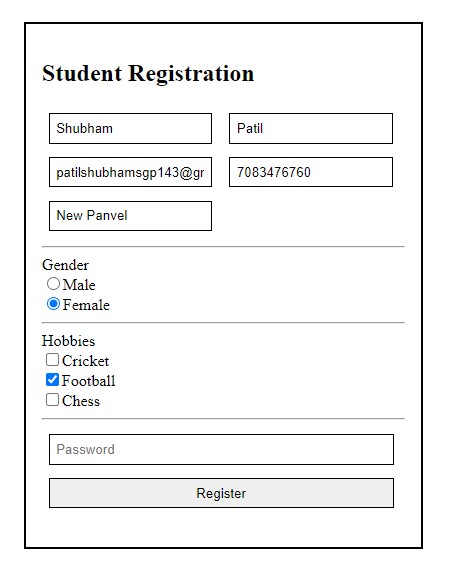
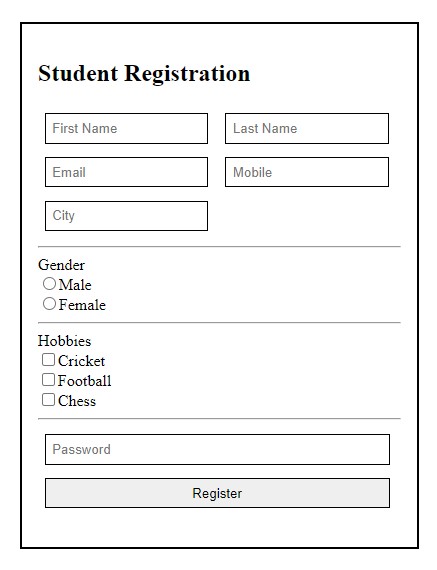
    <input type="button" value="Submit" name="click" onclick="showAlert()" />

    </form>

    </body>

    </html>

**Output:**



**Practical No.: 14 Aim: Demonstrate Annotations in TestNG framework**

**Program:**

package testNG;

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; import org.testng.annotations.BeforeTest; import org.testng.annotations.AfterTest; import org.testng.annotations.BeforeSuite;

import org.testng.annotations.AfterSuite;

public class Annotations { @Test(priority = 1) public void firstTest()

{

System.*out*.println("Test1");

}

@Test(priority = 0)

public void secondTest()

{

System.*out*.println("Test2");

}

@BeforeMethod

public void beforeMethod()

{

System.*out*.println("BeforeMethod");

}

@AfterMethod

public void afterMethod()

{

System.*out*.println("AfterMethod");

}

@BeforeClass

public void beforeClass()

{

System.*out*.println("Before Class");

}

@AfterClass public void afterClass()

{

System.*out*.println("After Class");

}

@BeforeTest public void beforeTest()

{

System.*out*.println("Before Test");

}

@AfterTest public void afterTest()

{

System.*out*.println("After Test");

}

@BeforeSuite public void beforeSuite()

{

System.*out*.println("Before Suite");

}

@AfterSuite public void afterSuite()

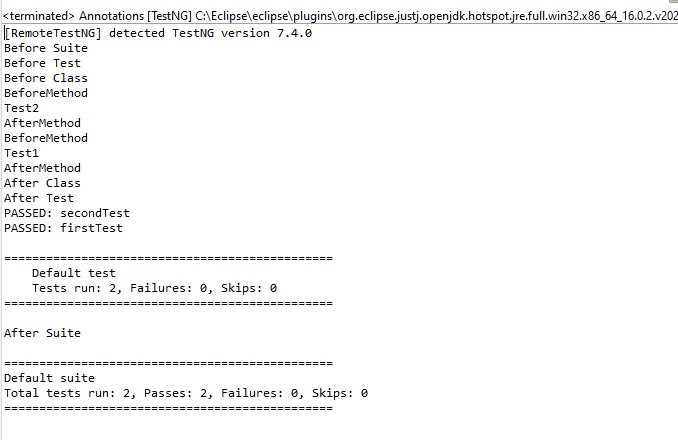
{

System.*out*.println("After Suite");

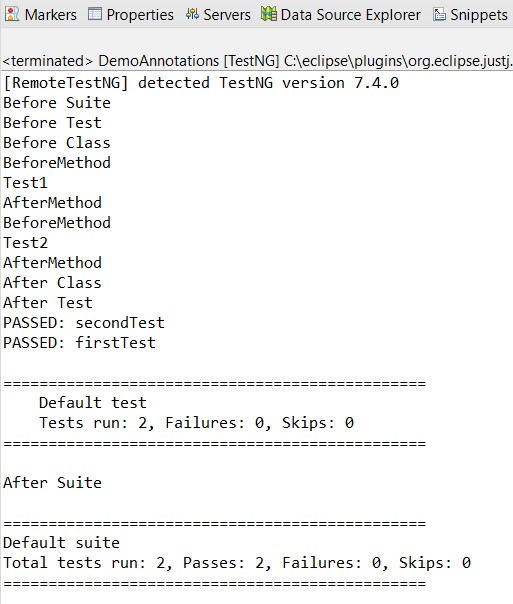
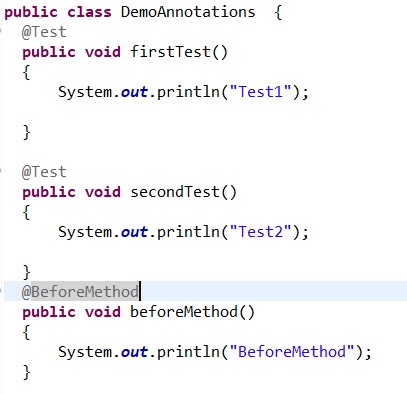
}

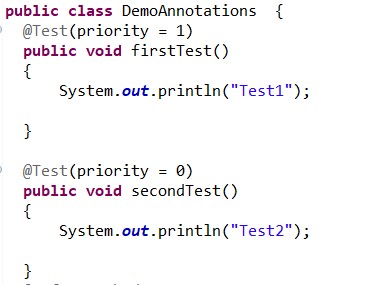
}

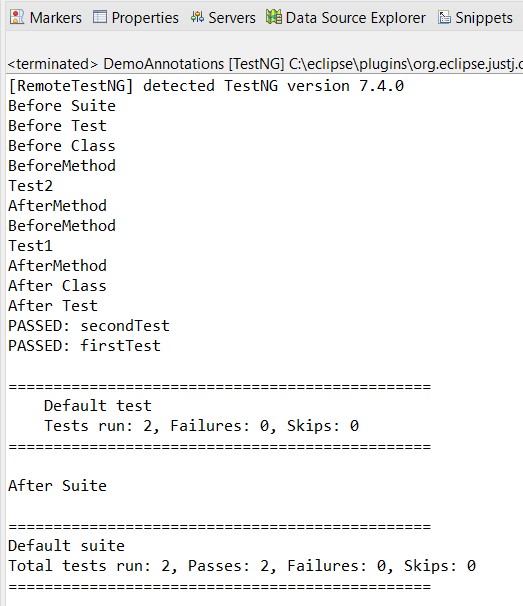
**Output:**



Add Second test and run







**Practical No.: 15 Aim: Demonstrate Assertions in TestNG framework**

**Program:**

package testNG;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Assert;

import org.testng.annotations.Test;

public class Assertion {

//Assertion ---> is to verify expected result and actual result are matching or not.

@Test

public void assertOrange() {

System.*setProperty*("webdriver.chrome.driver","C:\\Eclipse\\Drivers\\chromedriver.exe"

);

WebDriver wd= new ChromeDriver();

wd.get("https://www.google.co.in/");

String ExpectedTitle="Google"; //if there is a matching result then will get as PASS

//String ExpectedTitle= "Orange"; // if there is no matching result then will get FAIL.

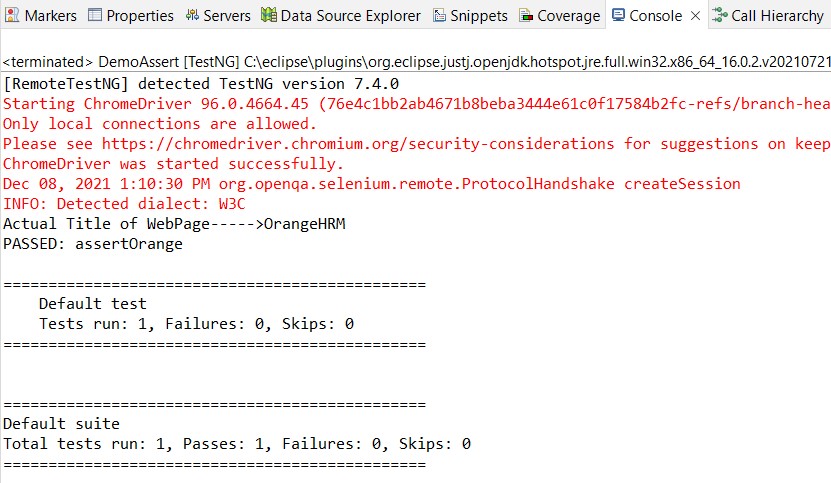
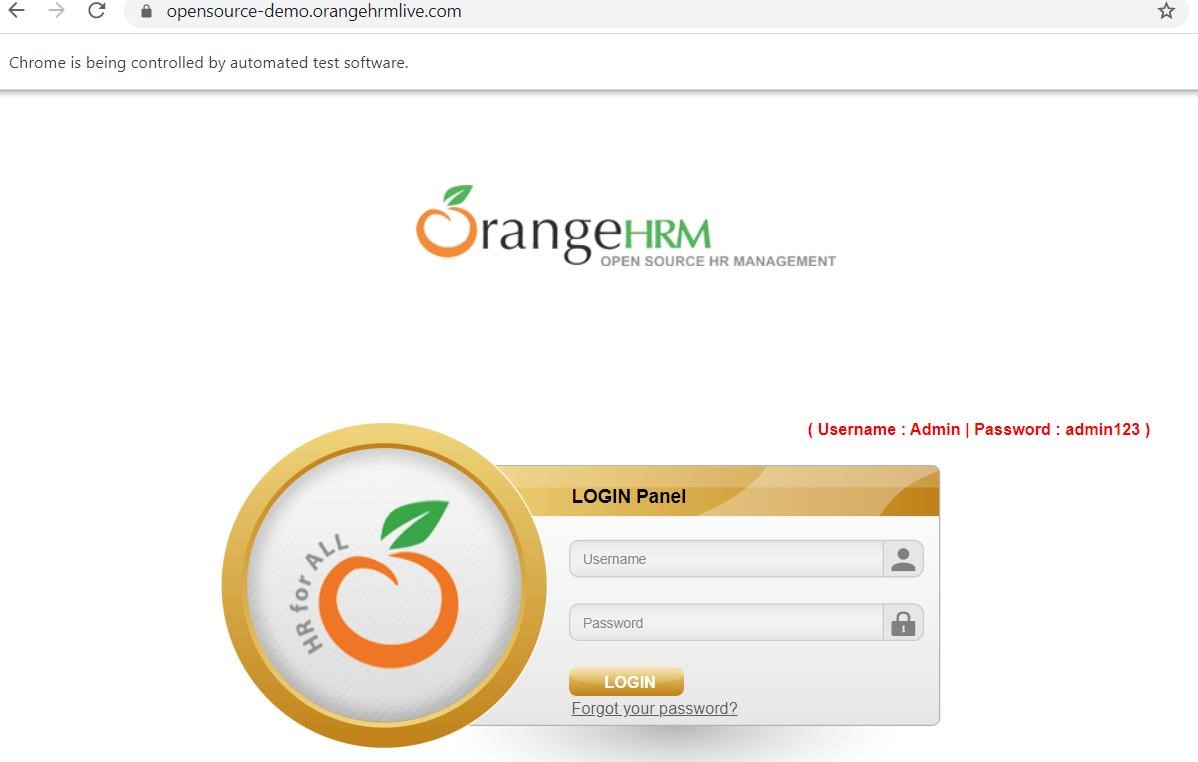
String ActualTitle=wd.getTitle();

Assert.*assertEquals*(ActualTitle, ExpectedTitle);

System.*out*.println("Actual Title of WebPage----->" + ActualTitle);

}}

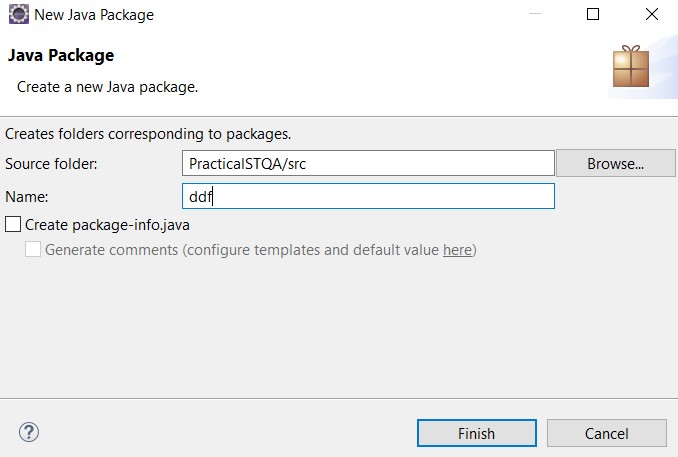
**Output:**



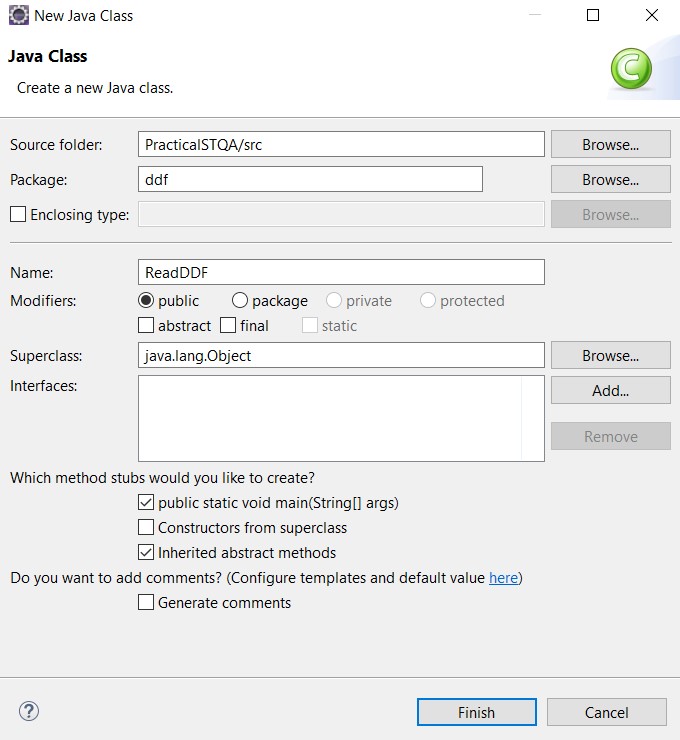
**Practical No.: 16 Aim: Demonstrate data driven Framework.**

**STEPS:**

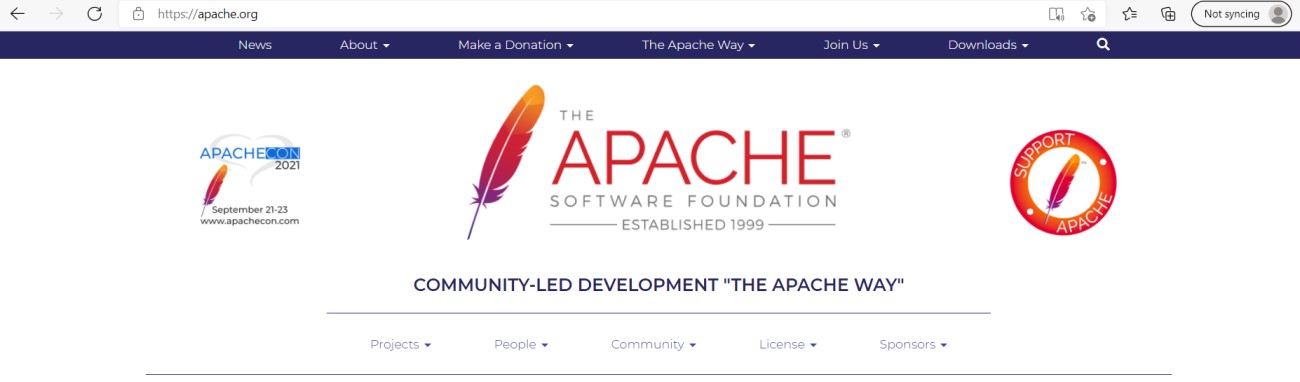
1.Right click on src folder→new →package(provide name ddf)→finish.



2.Now right click on package →new→class(provide the name ReadDDF)→select the main method→finish.



3.Go to the apache.org website and download the POI jar zip file. And then configured it with the project.

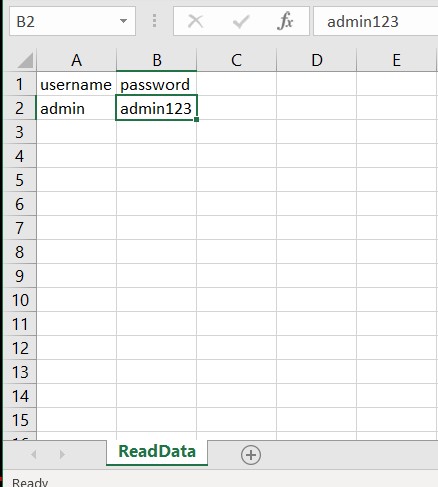








4.For data driven framework we need to have data file here we use Excel file as data. So create one excel file with some data.



**Program:**

package ddf;

import java.io.FileInputStream;

import org.apache.poi.xssf.usermodel.XSSFRow; import org.apache.poi.xssf.usermodel.XSSFCell; import org.apache.poi.xssf.usermodel.XSSFSheet; import org.apache.poi.xssf.usermodel.XSSFWorkbook; import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class ReadDDF

{

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

{

System.*setProperty*("webdriver.chrome.driver","C:\\Selenium\\chromedriver.exe"

);

WebDriver wd= new ChromeDriver();

wd.get("https://opensource-demo.orangehrmlive.com/");

FileInputStream fis = new FileInputStream("D:\\Selenium\\ExcelData.xlsx");

XSSFWorkbook wk = new XSSFWorkbook(fis);

XSSFSheet sh = wk.getSheet("ReadData");

for(int i=1; i<=sh.getLastRowNum(); i++)

{

XSSFRow rw = sh.getRow(i);

XSSFCell un = rw.getCell(0);

XSSFCell pw = rw.getCell(1);

System.*out*.println(un + " "+ pw);

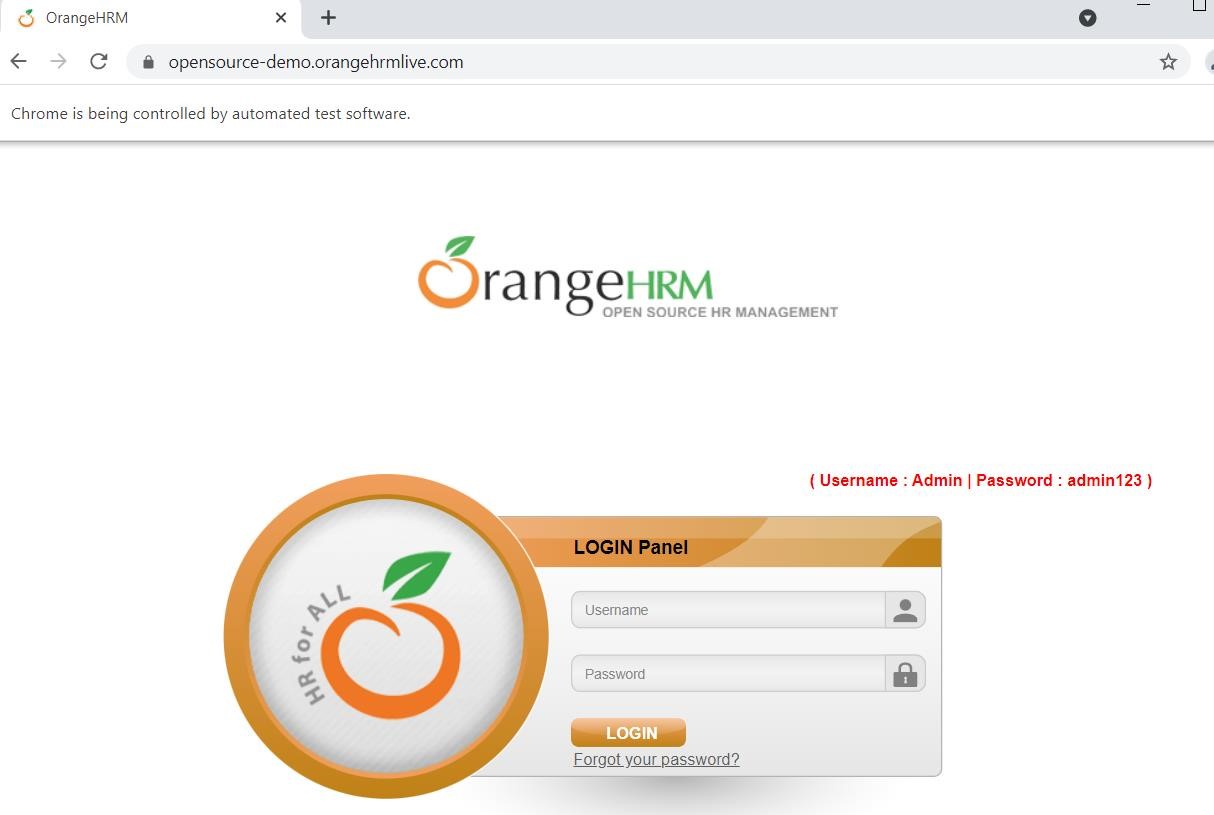
wd.findElement(By.*id*("txtUsername")).sendKeys(un.toString()); wd.findElement(By.*name*("txtPassword")).sendKeys(pw.toString()); wd.findElement(By.*className*("button")).click();

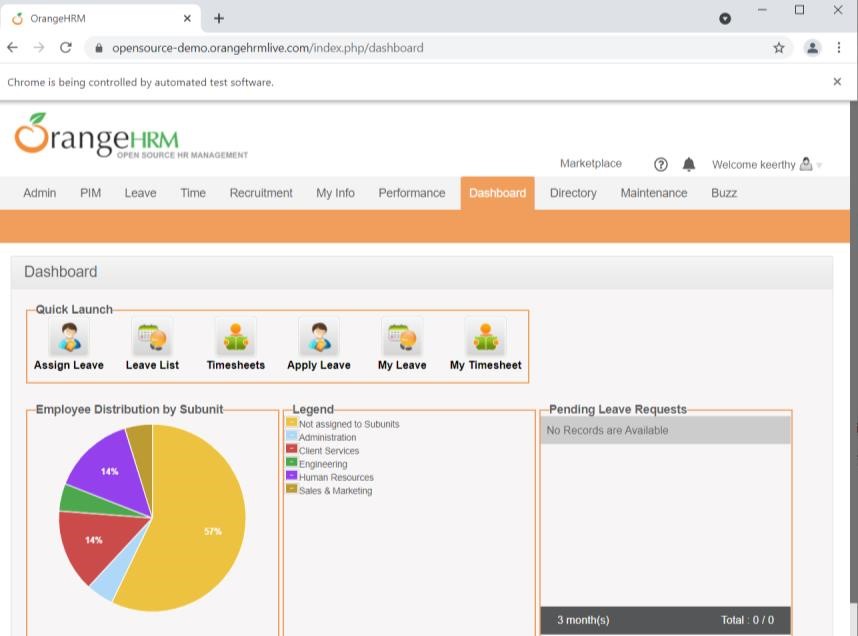
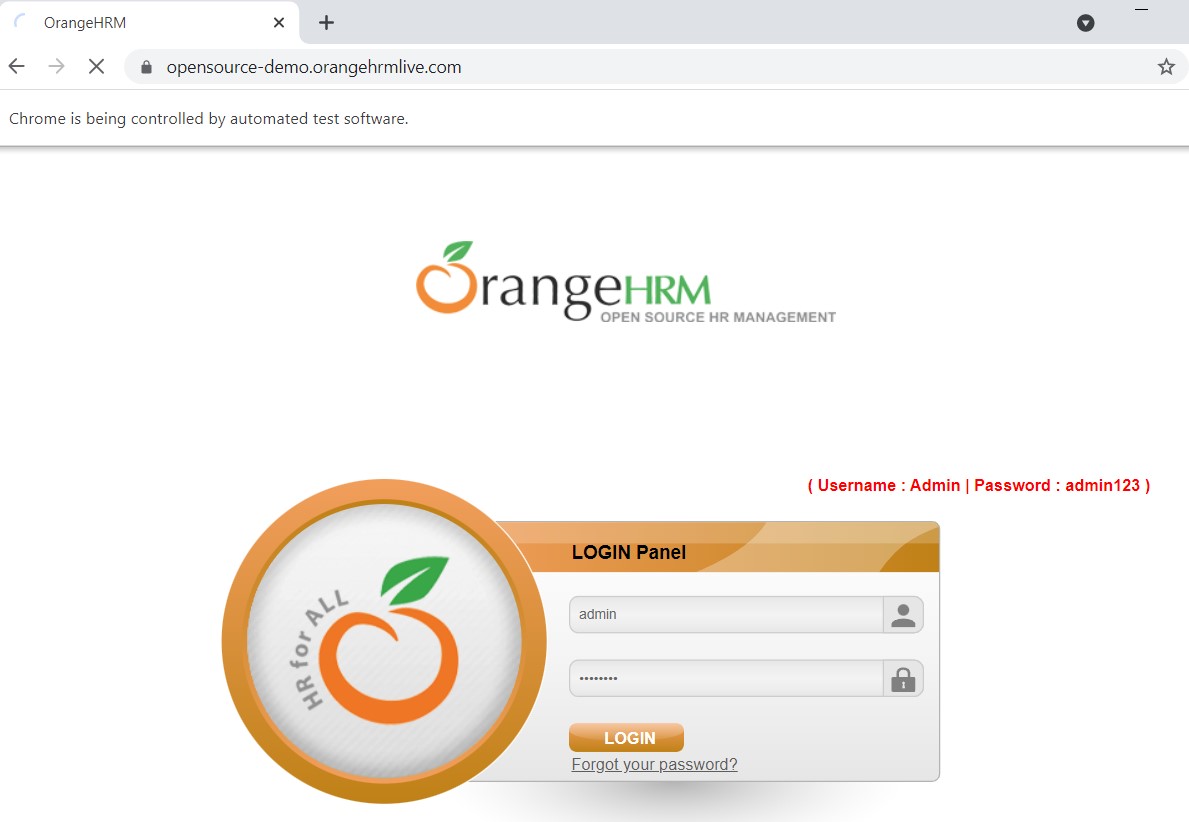
}

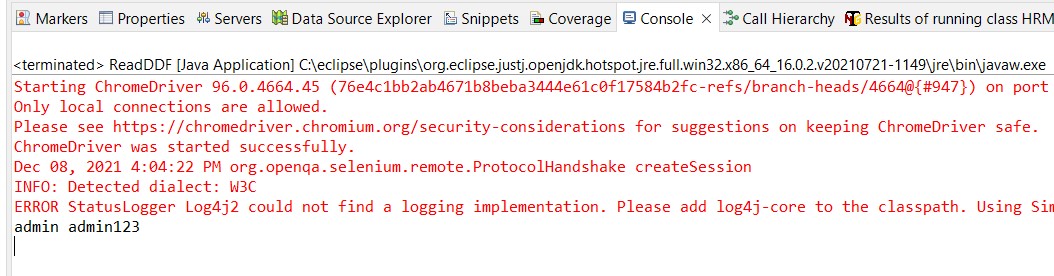
}

}

**Output:**







**Practical No.: 17 Aim: Demonstrate ImplicitWait (using HRM\_Menu.Java).**

**Program:**

**HRM\_Menu.java**

package testng;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver; i

mport org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass; import org.testng.annotations.AfterSuite;

import org.testng.annotations.BeforeClass; import org.testng.annotations.BeforeSuite;

import org.testng.annotations.Test;

public class HRM\_Menu

{

WebDriver wd;

@BeforeSuite

public void openBrowser()

{

System.*setProperty*("webdriver.chrome.driver","C:\\Selenium\\chromedriver.exe"

);

WebDriver wd = new wd ChromeDriver();

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

}

@BeforeClass

public void loginProcess()

{

wd.get("https://opensource-demo.orangehrmlive.com/");

wd.findElement(By.*id*("txtUsername")).sendKeys("admin");

wd.findElement(By.*id*("txtPassword")).sendKeys("admin123");

wd.findElement(By.*id*("btnLogin")).click();

}

@Test

public void myinfo()

{

wd.findElement(By.*id*("menu\_pim\_viewMyDetails")).click();

}

@Test

public void pim()

{

wd.findElement(By.*id*("menu\_pim\_viewPimModule")).click();

}

@AfterClass

public void logoutProcess()

{

wd.findElement(By.*id*("welcome")).click();

wd.findElement(By.*linkText*("Logout")).click();

}

@AfterSuite

public void closeBrowser()

{

wd.quit();

}

}

**Output:**

