

## PRACTICAL 1

---

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

Test Case ID	Component Name	Prerequisites	Steps	Input	Actual Output	Expected Output	Test Remark
TC001	User Registration	None	1. Navigate to the registration page. 2. Fill in the details. 3. Submit the form.	Valid user details (name, email, password)	User registered successfully	Registration success message	Pass
TC002	User Registration	None	1. Navigate to the registration page. 2. Enter an existing email. 3. Submit the form.	Existing email address	Error message displayed	"Email already in use"	Pass
TC003	User Login	User must be registered	1. Navigate to the login page. 2. Enter valid credentials. 3. Submit the form.	Valid username and password	User logged in	Redirected to homepage	Pass
TC004	User Login	User must be registered	1. Navigate to the login page. 2. Enter invalid credentials. 3. Submit the form.	Invalid username and/or password	Error message displayed	"Invalid credentials"	Pass
TC005	Flight Search	User is logged in	1. Navigate to the flight search page. 2. Enter departure and arrival cities. 3. Select travel dates. 4. Submit the search.	Departure: "New York", Arrival: "London"	Flight options displayed	List of available flights	Pass
TC006	Flight Search	User is logged in	1. Navigate to the flight search page. 2. Enter invalid cities. 3. Submit the search.	Departure: "XYZ", Arrival: "ABC"	Error message displayed	"No flights found"	Pass

## PRACTICAL 2

**AIM: Implement web drivers on chrome and Firefox browser(chrome, Firefox,choose browser)**

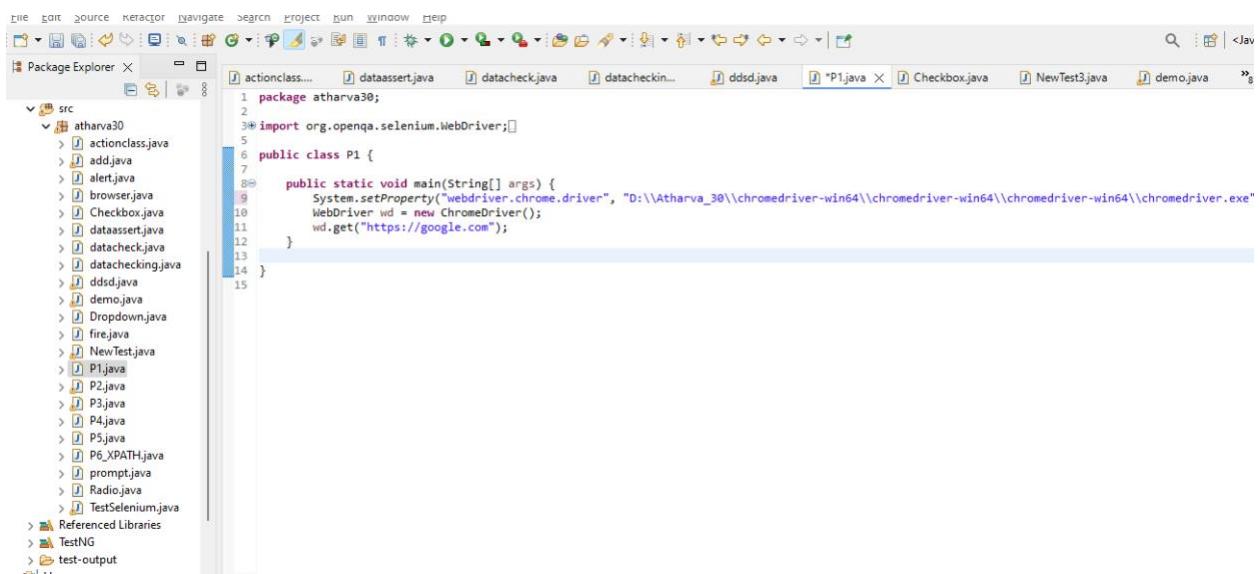
### A) Chrome

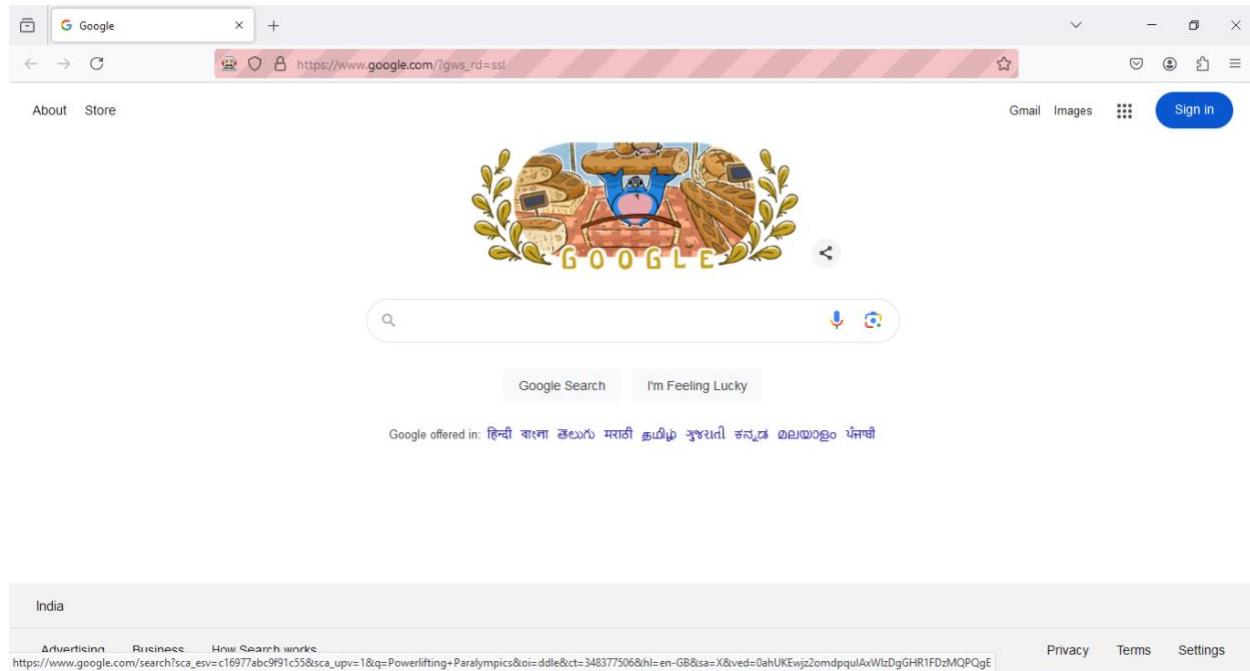
### B) CODE:

```
package atharva30;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class P1 {
public static void main(String[] args)
{System.setProperty("webdriver.chrome.driver","D:\\\\Atharva_30\\\\chromedriver-
win64\\\\chromedriver-win64\\\\chromedriver-win64\\\\chromedriver.exe");
    WebDriver wd = new FirefoxDriver();
    wd.get("http://google.com");
}}
```

### OUTPUT:





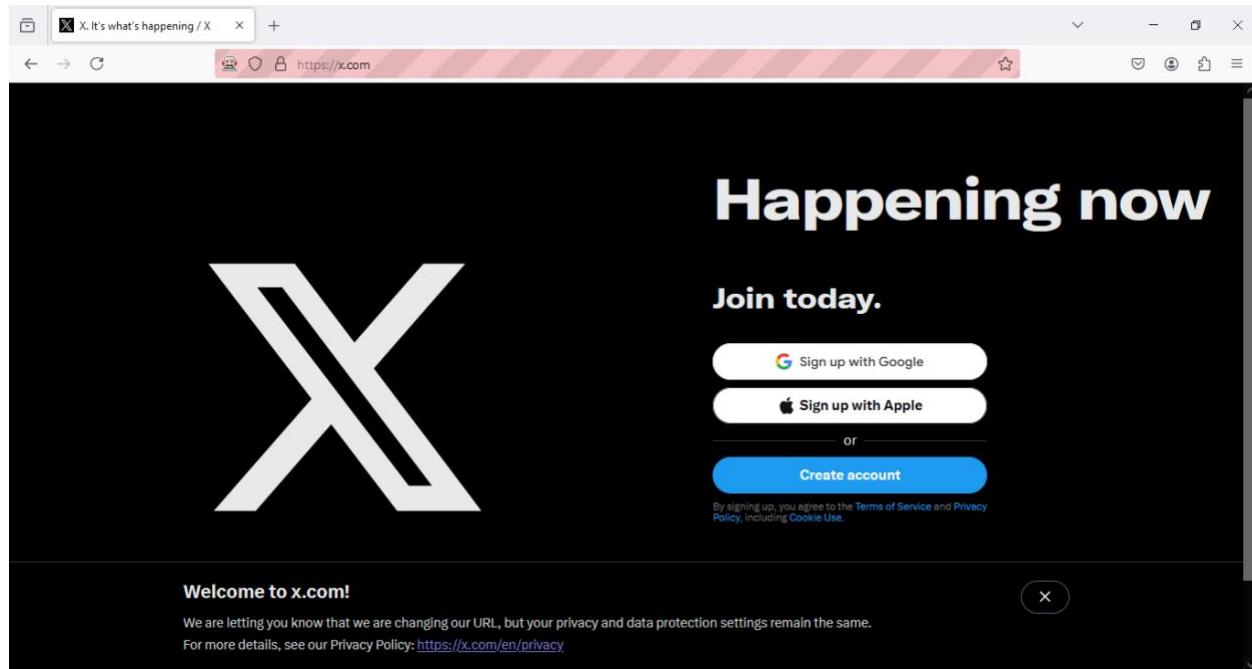
## B) FireFox

### CODE:

```
package atharva30;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class P2 {
    public static void main(String[] args)
        {System.setProperty("webdriver.chrome.driver","C:\\selenium web driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("http://x.com");
    }
}
```

### OUTPUT:

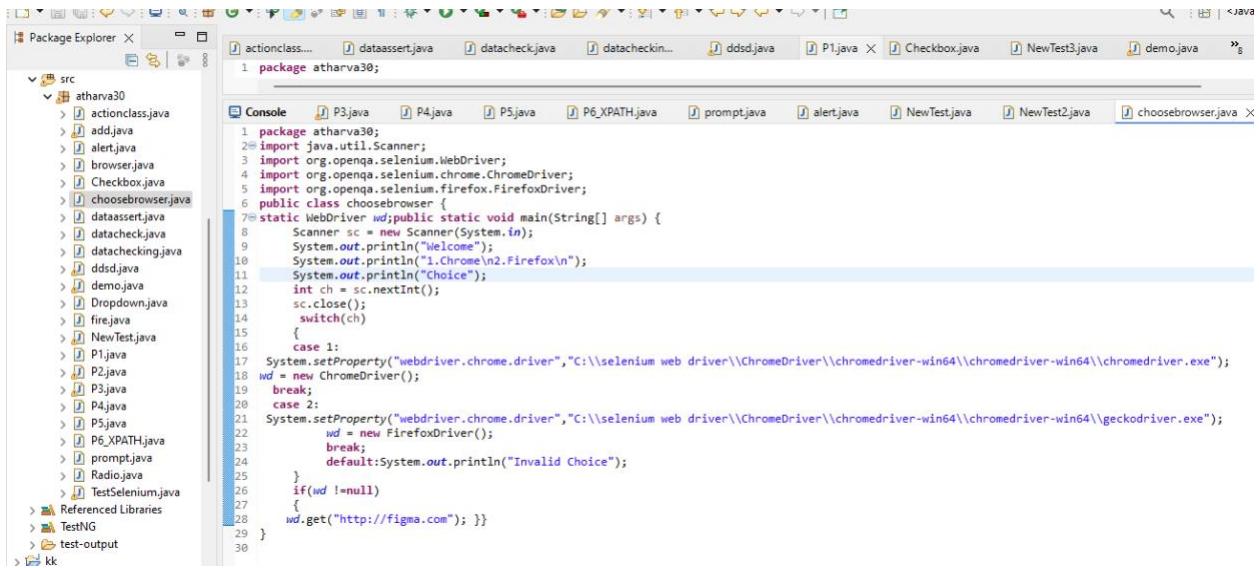


## C) Choose Browser

### CODE:

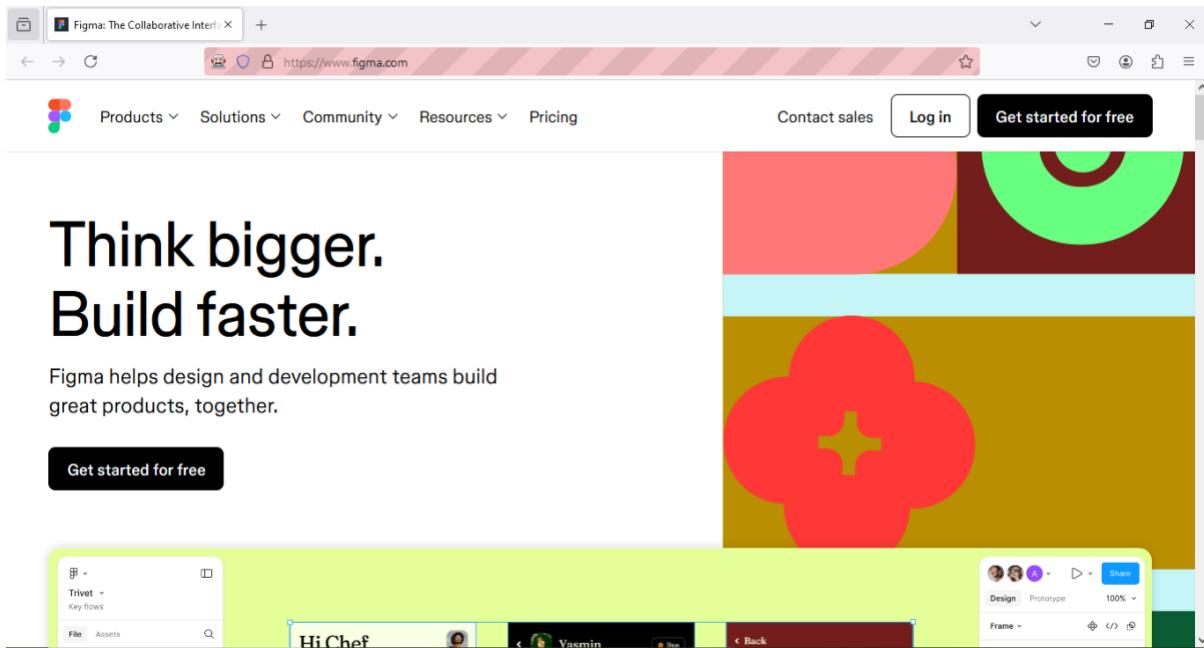
```
package Atharva30;
import java.util.Scanner;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class chooseBrowser {
    static WebDriver wd;public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Welcome");
        System.out.println("1.Chrome\n2.Firefox\n");
        System.out.println("Choice");
        int ch = sc.nextInt();
        sc.close();
        switch(ch)
        {
            case 1:
                System.setProperty("webdriver.chrome.driver","C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\chromedriver.exe");
                wd = new ChromeDriver();
                break;
            case 2:
                System.setProperty("webdriver.chrome.driver","C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\geckodriver.exe");
                wd = new FirefoxDriver();
                break;
            default:System.out.println("Invalid Choice");
        }
        if(wd !=null)
        {
            wd.get("http://figma.com"); }}
```

### OUTPUT:



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer view displays a project structure under 'src' with a package named 'atharva30' containing various Java files like actionclass.java, add.java, alert.java, browser.java, Checkbox.java, choosebrowser.java, dataassert.java, datacheck.java, datachecking.java, ddsd.java, demo.java, Dropdown.java, fire.java, NewTest.java, P1.java, P2.java, P3.java, P4.java, P5.java, P6\_XPATH.java, prompt.java, alert.java, NewTest3.java, NewTest2.java, and choosebrowser.java. The right side shows the Java Editor with the code for 'choosebrowser.java'. The code imports various Selenium WebDriver classes and uses Scanner to get user input to determine which browser to open.

```
1 package atharva30;
2
3 import java.util.Scanner;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6 import org.openqa.selenium.firefox.FirefoxDriver;
7
8 public class choosebrowser {
9     static WebDriver wd;
10    public static void main(String[] args) {
11        Scanner sc = new Scanner(System.in);
12        System.out.println("Welcome");
13        System.out.println("1.Chrome\n2.FIREFOX");
14        int ch = sc.nextInt();
15        sc.close();
16        switch(ch) {
17            case 1:
18                System.setProperty("webdriver.chrome.driver", "C:\\selenium web driver\\ChromeDriver\\chromedriver-win64\\chromedriver.exe");
19                wd = new ChromeDriver();
20            break;
21            case 2:
22                System.setProperty("webdriver.firefox.driver", "C:\\selenium web driver\\FirefoxDriver\\geckodriver.exe");
23                wd = new FirefoxDriver();
24            break;
25            default:System.out.println("Invalid Choice");
26        }
27        if(wd !=null)
28            wd.get("http://figma.com");
29    }
30 }
```



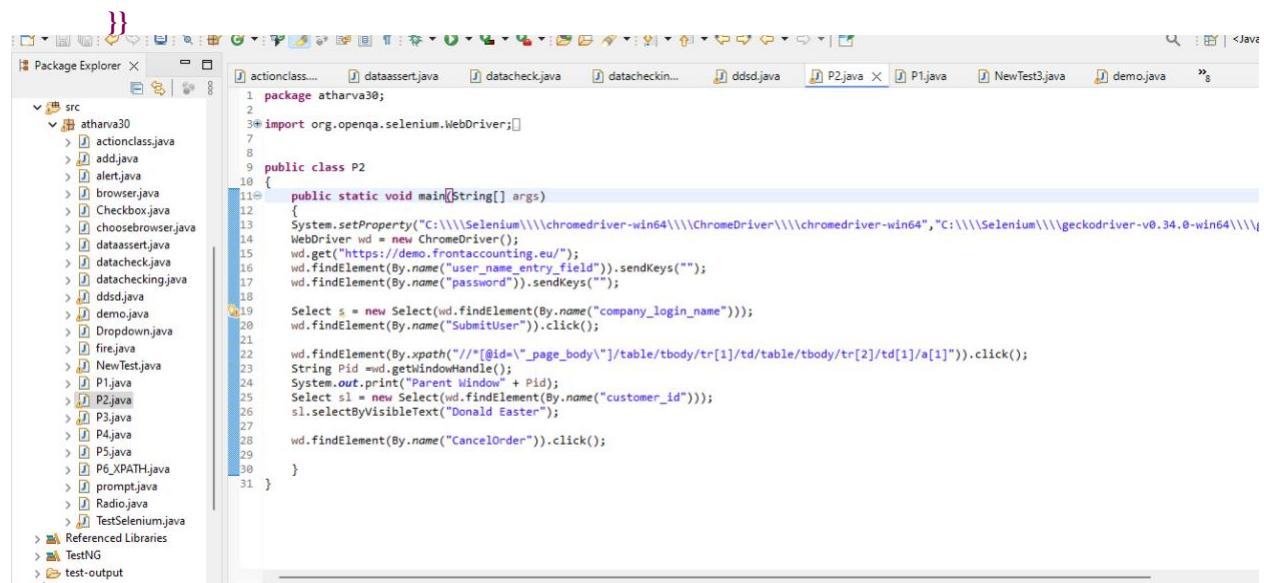
## PRACTICAL 3

**Aim:- Demonstrate Handling multiple window in selenium.**

### 1) Order Placed

```
package atharva30;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.*;
import org.openqa.selenium.support.ui.Select;
public class P2{
    public static void main(String[] args){
        System.setProperty("C:\\\\Selenium\\\\chromedriver-win64\\\\ChromeDriver\\\\chromedriver-win64","C:\\\\Selenium\\\\geckodriver-v0.34.0-win64\\\\geckodriver.exe");
        WebDriver wd = new ChromeDriver();
        wd.get("https://demo.frontaccounting.eu/");
        wd.findElement(By.name("user_name_entry_field")).sendKeys("");
        wd.findElement(By.name("password")).sendKeys("");
        Select s = new Select(wd.findElement(By.name("company_login_name")));
        wd.findElement(By.name("SubmitUser")).click();
        wd.findElement(By.xpath("//*[@id='\\_page_body\\']/table/tbody/tr[1]/td/table/tbody/tr[2]/td[1]/a[1]")).click();
        String Pid =wd.getWindowHandle();
        System.out.print("Parent Window" + Pid);
        Select sl = new Select(wd.findElement(By.name("customer_id")));
        sl.selectByVisibleText("Donald Easter");

        wd.findElement(By.name("CancelOrder")).click();
    }
}
```



FrontAccounting 2.4.12 - Login + https://demo.frontaccounting.eu FrontAccounting 2.4.12 - Login

**FrontAccounting**  
Version 2.4.12 Build 13.07.2022 - Login

User name: demouser  
Password: \*\*\*\*\*  
Company: Training Co.

**Invalid password or username. Please, try again.**

Login -->

09/21/2024 | 07:46 am

FrontAccounting 2.4.12 - Theme: default  
FrontAccounting

New Sales Quotation Entry + https://demo.frontaccounting.eu/sales/sales\_order\_entry.php?NewQuotation=Yes

Sales Purchases Items and Inventory Manufacturing Fixed Assets Dimensions Banking and General Ledger Setup  
Training Co. | demo.frontaccounting.eu | Demo User Dashboard Preferences Change password Help Logout

New Sales Quotation Entry

Customer: Donald Easter Current Credit: **-998,970.00** Payment: Cash Only Quotation Date: 12/31/2022  
Branch: abced Customer Discount: 0% Price List: Administrative  
Reference: 98982511/2022

Sales Quotation Items						
Item Code	Item Description	Quantity	Unit	Price after Tax	Discount %	Total
	Samsung	1	each	0.00	0.0	0.00
<b>Add Item</b>						
Shipping Charge 0.00						
Sub-total 0.00						
Amount Total 0.00 <b>Update</b>						

**Cash payment**

Deliver from Location: Default  
Cash account: Petty Cash account

Comments:

Activate Windows  
Go to Settings to activate Windows.

Place Quotation Cancel Quotation

The screenshot shows two windows side-by-side. The left window is the 'FrontAccounting 2.4.12 - Login' screen, displaying a login form with fields for 'User name' (demouser), 'Password' (redacted), and 'Company' (Training Co.). A red error message 'Invalid password or username. Please, try again.' is visible below the form. The right window is the 'New Sales Quotation Entry' screen, showing a header with navigation links like Sales, Purchases, and Banking and General Ledger. Below the header is a form with fields for Customer (Donald Easter), Current Credit (-998,970.00), Payment (Cash Only), Quotation Date (12/31/2022), Branch (abced), Customer Discount (0%), and Price List (Administrative). A reference number (98982511/2022) is also present. The main area contains a table for 'Sales Quotation Items' with one row for 'Samsung' at 1 unit, 0.00 price, and 0.00 total. Below the table are sections for 'Cash payment' (Deliver from Location: Default, Cash account: Petty Cash account) and 'Comments'. At the bottom are 'Place Quotation' and 'Cancel Quotation' buttons.

The screenshot shows a web browser window for 'New Sales Quotation Entry' at the URL [demo.frontaccounting.eu/sales/sales\\_order\\_entry.php?AddedQU=1241](http://demo.frontaccounting.eu/sales/sales_order_entry.php?AddedQU=1241). The browser status bar indicates 'Chrome is being controlled by automated test software.'

The page header includes links for Sales, Purchases, Items and Inventory, Manufacturing, Fixed Assets, Dimensions, Banking and General Ledger, and Setup. It also shows the company name 'Training Co.' and user information 'demo.frontaccounting.eu | Demo User'. The top right has links for Dashboard, Preferences, Change password, Help, and Logout.

The main content area displays a green message bar stating 'Quotation # 1241 has been entered.' Below this are several buttons and links:

- [View This Quotation](#)
- [Print This Quotation](#) (highlighted in red)
- [Email This Quotation](#)
- [Make Sales Order Against This Quotation](#)
- [Enter a New Quotation](#)

A 'Back' link is located at the bottom left. The footer contains the date '09/19/2024 | 05:10 pm' and the text 'FrontAccounting 2.4.12 - Theme: default - FrontAccounting'.

## 2) Multiple Window Handling.

```
package atharva30;
import java.util.Set;
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 multiplehandling{
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://demo.frontaccounting.eu/");
        wd.findElement(By.name("user_name_entry_field")).sendKeys("your_username"); // Add username
        wd.findElement(By.name("password")).sendKeys("your_password"); // Add password
        Select s = new Select(wd.findElement(By.name("company_login_name")));
        wd.findElement(By.name("SubmitUser")).click();

        wd.findElement(By.xpath("/html/body/table[1]/tbody/tr/td/table[1]/tbody/tr/td/div[2]/
table/tbody/tr[1]/td/table/tbody/tr[2]/td[1]/a")).click();
        String Pid = wd.getWindowHandle();
        System.out.println("Parent Window > " + Pid);

        wd.findElement(By.xpath("/html/body/table[1]/tbody/tr/td/table[1]/tbody/tr/td/div[2]/
form/center[1]/table/tbody/tr/td[2]/table/tbody/tr[1]/td[2]/a")).click();
        Set<String> allwin = wd.getWindowHandles();
        for (String W : allwin) {
            wd.switchTo().window(W);
            System.out.println(wd.getTitle());
            if (!W.equals(Pid)) {
                Select ss = new Select(wd.findElement(By.id("customer_id"))); // Ensure
the ID is correct
                ss.selectByVisibleText("Donald Easter");
                wd.manage().window().maximize();
                wd.close(); // Close this window after processing
            }
        }
        wd.switchTo().window(Pid);
        wd.findElement(By.linkText("Logout")).click(); // Corrected syntax
    }
}
```

File Edit Source Refactor Navigate Project Run Window Help

Package Explorer X

src

- atharva30
  - actionclass.java
  - add.java
  - alert.java
  - browser.java
  - checkbox.java
  - choosebrowser.java
  - dataassert.java
  - datacheck.java
  - datachecking.java
  - ddsd.java
  - demo.java
  - dropdown.java
  - fire.java
  - multiplehandling.java
  - NewTest.java
  - P1.java
  - P2.java
  - P3.java
  - P4.java
  - P5.java
  - P6\_XPATH.java
  - prompt.java
  - Radio.java
  - TestSelenium.java
  - TestNG
  - test-output
- kk
- Referenced Libraries
- TestNG
- test-output

dataassert.java datacheck.java datacheckin... ddsd.java P1.java NewTest.java demo.java \*multiplehan... X

```
1 package atharva30;
2 import java.util.Set;
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6 import org.openqa.selenium.support.ui.Select;
7
8 public class multiplehandling{
9     public static void main(String[] args) {
10         System.setProperty("webdriver.chrome.driver", "C:\\selenium web driver\\ChromeDriver\\chromedriver-win64\\geckodriver.exe");
11         WebDriver wd = new FirefoxDriver();
12         wd.get("https://demo.frontaccounting.eu/");
13         wd.findElement(By.name("user_name_entry_field")).sendKeys("your_username"); // Add username
14         wd.findElement(By.name("password")).sendKeys("your_password"); // Add password
15         Select s = new Select(wd.findElement(By.name("company_login_name")));
16         wd.findElement(By.name("SubmitUser")).click();
17         wd.findElement(By.xpath("//html/body/table[1]/tbody/tr/td/table[1]/tbody/tr[2]/td/div[1]/a")).click();
18         String Pid = wd.getWindowHandle();
19         System.out.println("Parent Window > " + Pid);
20         wd.findElement(By.xpath("//html/body/table[1]/tbody/tr/td/div[1]/table/tbody/tr[2]/form/center[1]/table/tbody/tr[1]/td[2]"));
21         Set<String> allwin = wd.getWindowHandles();
22         for (String W : allwin) {
23             wd.switchTo().window(W);
24             System.out.println(wd.getTitle());
25             if (!W.equals(Pid)) {
26                 Select ss = new Select(wd.findElement(By.id("customer_id"))); // Ensure the ID is correct
27                 ss.selectByVisibleText("Donald Easter");
28                 wd.manage().window().maximize();
29                 wd.close(); // Close this window after processing
30             }
31         }
32     }
33     wd.switchTo().window(Pid);
34 }
```

Console P3.java P4.java P5.java P6\_XPATH.java prompt.java alert.java NewTest.java NewTest2.java choosebrowser.java

FrontAccounting 2.4.12 - Login X

FrontAccounting 2.4.12 - Login

User name: demouser

Password: \*\*\*\*\*

Company: Training Co.

Invalid password or username. Please, try again.

Login -->

09/21/2024 | 07:46 am

FrontAccounting 2.4.12 - Theme: default

FrontAccounting

The screenshot shows the 'Customer Transactions' page of the FrontAccounting software. At the top, there is a navigation bar with links for Sales, Purchases, Items and Inventory, Manufacturing, Fixed Assets, Dimensions, Banking and General Ledger, and Setup. Below the navigation bar, the URL is https://demo.frontaccounting.eu/sales/inquiry/customer\_inquiry.php?.

The main content area is titled 'Customer Transactions'. It features a search bar with fields for 'Select a customer' (set to 'Jash'), 'All Types' (set to 'All'), 'From' (09/23/2024), 'To' (10/23/2024), 'Zero values' (unchecked), and a 'Search' button. Below the search bar is a table with columns: Currency, Terms, Current, 1-30 Days, 31-60 Days, Over 60 Days, and Total Balance. A single row is shown: USD, Cash Only, 0.00, 0.00, 0.00, 0.00, and 0.00. Below the table, a message says 'No records' and 'Marked items are overdue.' There is a 'Back' link at the bottom of the table area.

At the bottom of the page, the date and time are listed as 10/23/2024 | 08:50 am, and the page title is F2 - Customers. A watermark for 'FrontAccounting 2.4.12 - Theme: default - FrontAccounting' is visible in the center. In the bottom right corner, there is a message: 'Activate Windows' and 'Go to Settings to activate Windows.'

The screenshot shows the 'Logout' page of the FrontAccounting software. At the top, there is a navigation bar with links for Sales, Purchases, Items and Inventory, Manufacturing, Fixed Assets, Dimensions, Banking and General Ledger, and Setup. Below the navigation bar, the URL is https://demo.frontaccounting.eu/access/logout.php?.

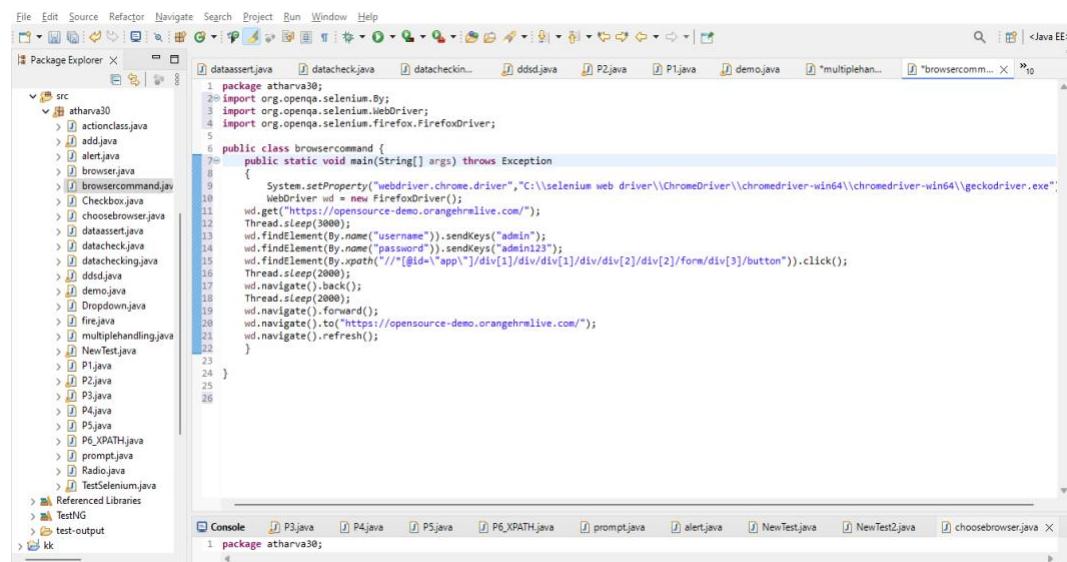
The main content area is titled 'FrontAccounting'. It displays a message: 'Thank you for using FrontAccounting 2.4.12' and a link 'Click here to Login Again.' Below this message, the date and time are listed as 10/23/2024 | 08:52 am, and the page title is F2 - Customers. A watermark for 'FrontAccounting 2.4.12 - Theme: default - FrontAccounting' is visible in the center.

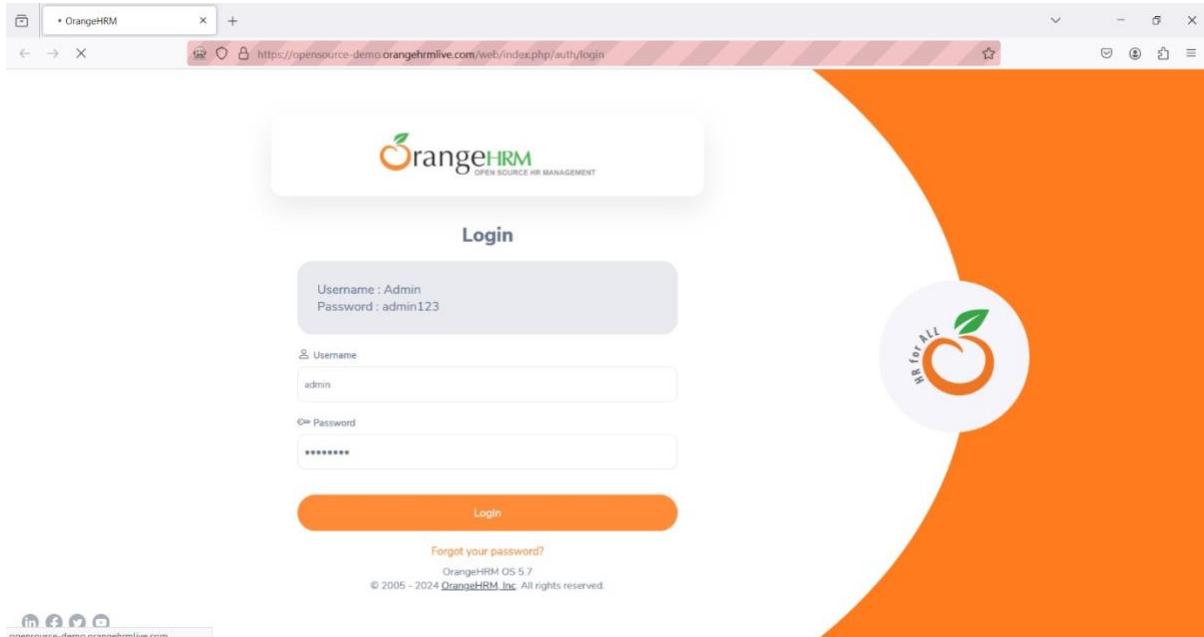
## PRACTICAL 4

### Aim:- Implement Browser command and navigation commands

```
package Atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class practical {
    public static void main(String[] args) throws Exception
    {
        System.setProperty("webdriver.chrome.driver","C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://opensource-demo.orangehrmlive.com/");
        Thread.sleep(3000);
        wd.findElement(By.name("username")).sendKeys("admin");
        wd.findElement(By.name("password")).sendKeys("admin123");
        wd.findElement(By.xpath("//*[@id='app']//div[1]//div//div[2]//div[2]//form//div[3]//button")).click()
    }
    Thread.sleep(2000);
    wd.navigate().back();
    Thread.sleep(2000);
    wd.navigate().forward();
    wd.navigate().to("https://opensource-demo.orangehrmlive.com/");
    wd.navigate().refresh();
}
}
```





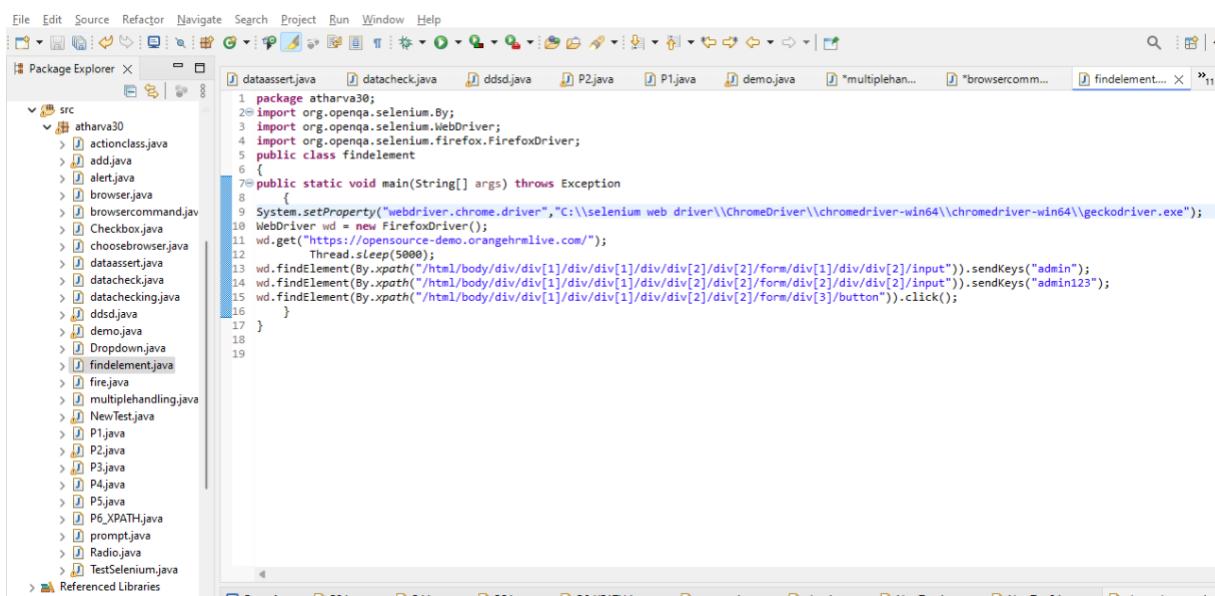
The screenshot shows the OrangeHRM dashboard. At the top, there's a header bar with the OrangeHRM logo and the text "Dashboard". On the right side of the header, there are buttons for "Upgrade", a user profile icon, and "First Name Employee". Below the header is a sidebar with a search bar and a list of menu items: Admin, PIM, Leave, Time, Recruitment, My Info, Performance, Dashboard (which is highlighted in orange), Directory, Maintenance, Claim, and Buzz. The main content area is divided into several cards: "Time at Work" (showing a "Punched Out" status for today at 08:39 PM), "My Actions" (listing "(1) Candidate to Interview"), "Quick Launch" (with icons for Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, and My Timesheet), "Buzz Latest Posts" (listing recent posts from users First Name akhil Employee and First Name Employee), "Employees on Leave Today" (listing First Name Employee on CAN - FMLA from 09:00 - 17:00), and "Employee Distribution by Sub Unit" (a pie chart showing employee distribution across sub units). The browser address bar shows "https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index".

## PRACTICAL 5

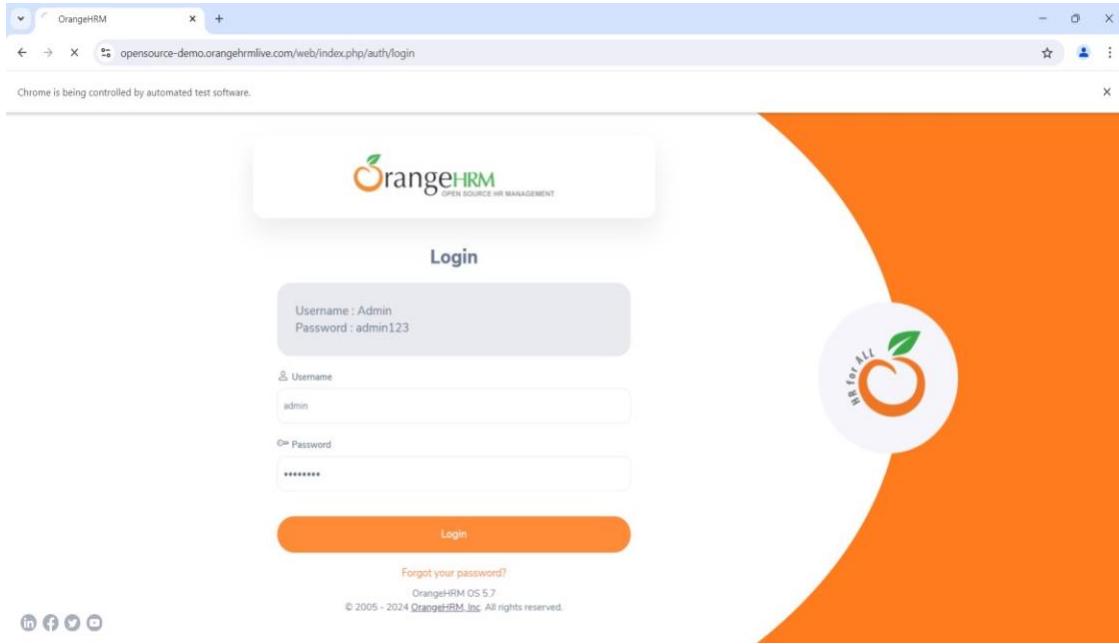
### AIM: Implement the find element command.

#### CODE:

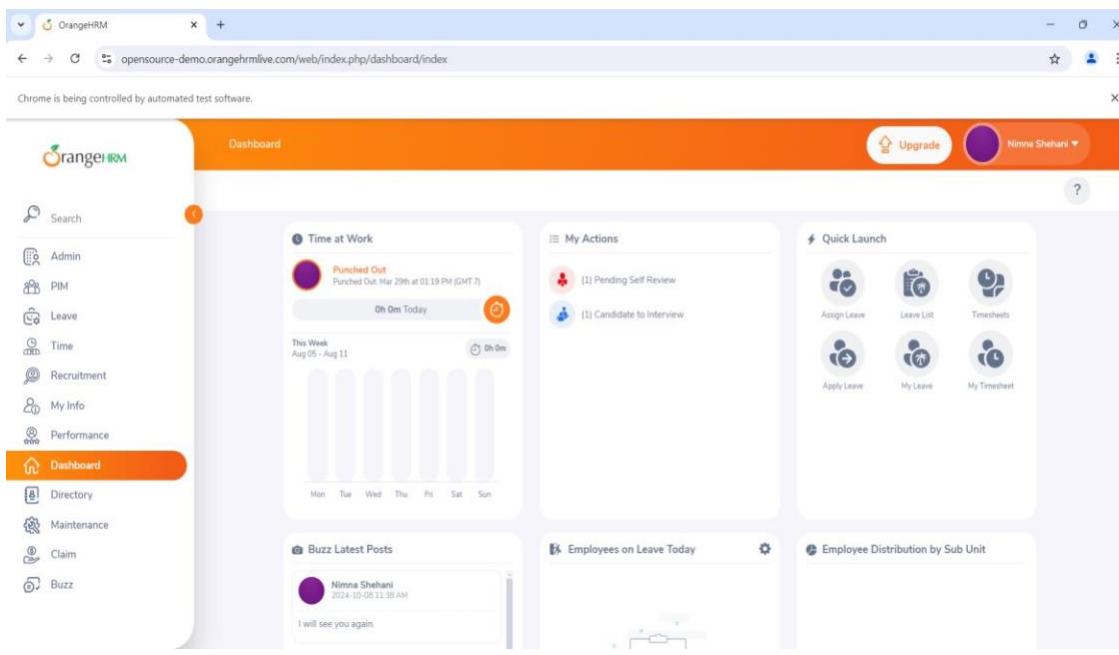
```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;public class findelement {
public static void main(String[] args) throws Exception {
System.setProperty("webdriver.chrome.driver","C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\geckodriver.exe");
WebDriver wd = new FirefoxDriver();
wd.get("https://opensource-demo.orangehrmlive.com/");
Thread.sleep(5000);
wd.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/form/div[
1]/div/div[2]/input")).sendKeys("admin");
wd.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/form/div[
2]/div/div[2]/input")).sendKeys("admin123");
wd.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/form/div[
3]/button")).click();}}
```



## OUTPUT:



The screenshot shows the OrangeHRM login interface. At the top, there's a header bar with the OrangeHRM logo and the text "OPEN SOURCE HR MANAGEMENT". Below it is a large orange background graphic featuring a stylized orange and leaf logo with the text "HR for ALL". The main form has fields for "Username" (admin) and "Password" (admin123). A "Login" button is at the bottom. Below the form are links for "Forgot your password?" and "OrangeHRM OS 5.7". The footer includes copyright information: "© 2005 - 2024 OrangeHRM Inc. All rights reserved." and social media icons for LinkedIn, Facebook, Twitter, and YouTube.

The screenshot shows the OrangeHRM dashboard. The left sidebar has a "Dashboard" tab selected, along with other options like Admin, PIM, Leave, Time, Recruitment, My Info, Performance, Directory, Maintenance, Claim, and Buzz. The main dashboard area displays several cards: "Time at Work" (Punched Out), "My Actions" (Pending Self Review, Candidate to Interview), "Quick Launch" (Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, My Timesheet), "Buzz Latest Posts" (Nimna Shehani, 2024-10-08 11:38 AM), "Employees on Leave Today" (I will see you again), and "Employee Distribution by Sub Unit". The top right corner shows a user profile for "Nimna Shehani" and an "Upgrade" button.

## PRACTICAL 6

**AIM: Demonstrate the locator(CSS Selector,Xpath,path)**

### A) cssSelector

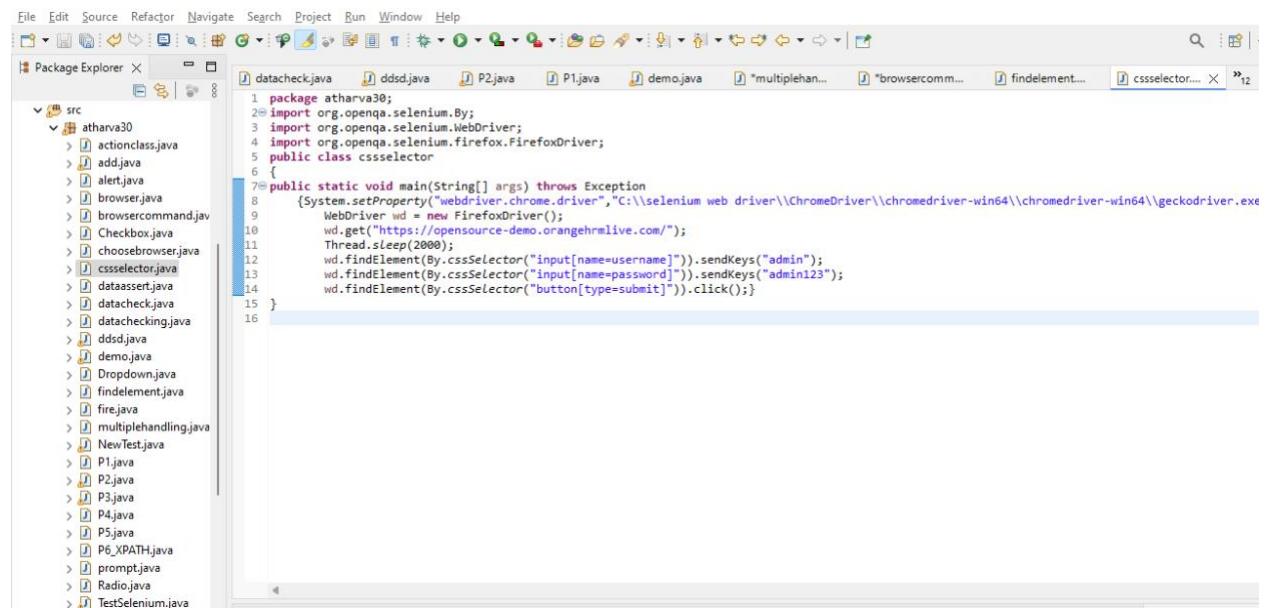
#### CODE:

```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class cssselector
{
    public static void main(String[] args) throws Exception
        {System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://opensource-demo.orangehrmlive.com/");
        Thread.sleep(2000);

        wd.findElement(By.cssSelector("input[name=username]")).sendKeys("admin");

        wd.findElement(By.cssSelector("input[name=password]")).sendKeys("admin123");
        wd.findElement(By.cssSelector("button[type=submit]")).click();}
}
```

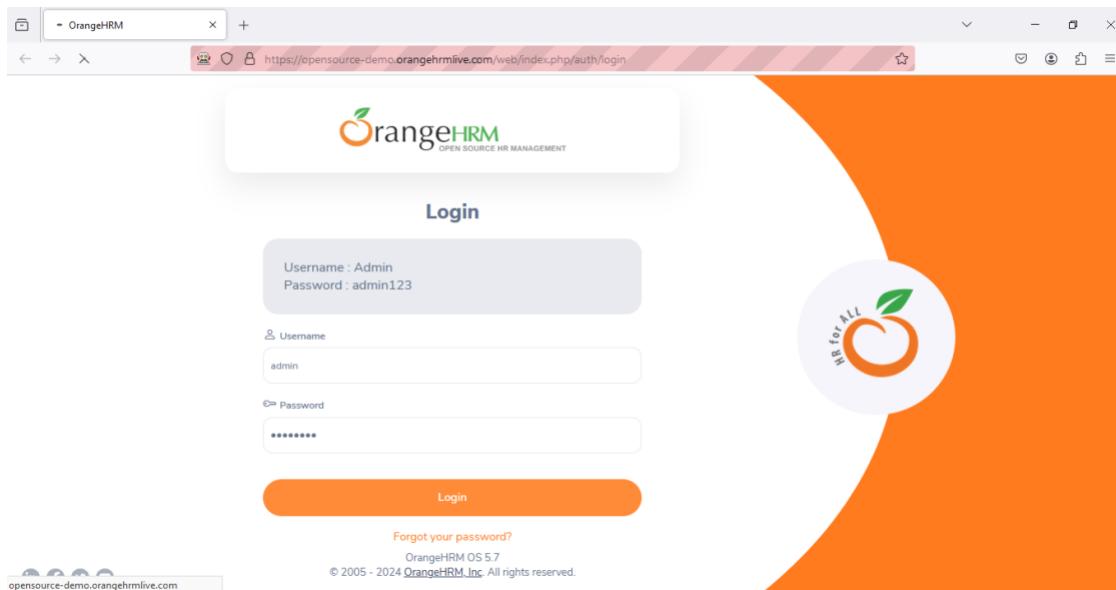
#### **OUTPUT:**



The screenshot shows the Eclipse IDE interface with the 'Package Explorer' view on the left and the 'Editor' view on the right. The 'Package Explorer' shows a project structure with a 'src' folder containing various Java files. The 'Editor' view displays the 'cssselector.java' file, which contains the provided Java code for demonstrating CSS selectors.

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X
src
  atharva30
    actionclass.java
    add.java
    alert.java
    browser.java
    browsercommand.java
    Checkbox.java
    choosebrowser.java
    cssselector.java
    dataassert.java
    datachecking.java
    ddsd.java
    demo.java
    Dropdown.java
    findelement.java
    fire.java
    multiplehandling.java
    NewTest.java
    P1.java
    P2.java
    P3.java
    P4.java
    P5.java
    P6_XPATH.java
    prompt.java
    Radio.java
    TestSelenium.java
cssselector.java X 12

1 package atharva30;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.firefox.FirefoxDriver;
5 public class cssselector
6 {
7     public static void main(String[] args) throws Exception
8         {System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
9 driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
10        WebDriver wd = new FirefoxDriver();
11        wd.get("https://opensource-demo.orangehrmlive.com/");
12        Thread.sleep(2000);
13        wd.findElement(By.cssSelector("input[name=username]")).sendKeys("admin");
14        wd.findElement(By.cssSelector("input[name=password]")).sendKeys("admin123");
15        wd.findElement(By.cssSelector("button[type=submit]")).click();}
16 }
```



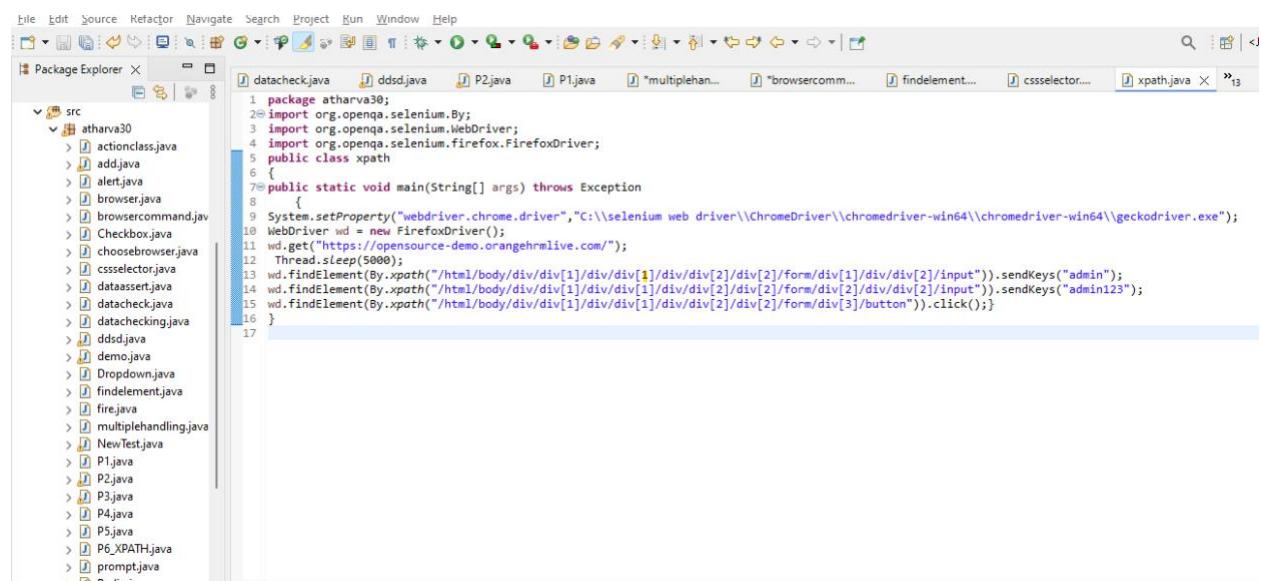
A screenshot of the OrangeHRM dashboard. The URL is https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index. The dashboard has an orange header with the OrangeHRM logo and a user profile for "Oliver Hofer". On the left is a sidebar with navigation links: Admin, PIM, Leave, Time, Recruitment, My Info, Performance, Dashboard (which is selected and highlighted in orange), Directory, Maintenance, Claim, and Buzz. The main content area contains several widgets: "Time at Work" showing a punch record for "Punched Out" at 12:56 PM (GMT 5.5), "My Actions" listing tasks like "Timesheet to Approve", "Pending Self Review", and "Candidate to Interview", and "Quick Launch" buttons for Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, and My Timesheet. Other widgets include "Buzz Latest Posts" (showing a post from Oliver Sven Hofer on Sep 08, 2020), "Employees on Leave Today" (empty), and "Employee Distribution by Sub Unit" (a donut chart).

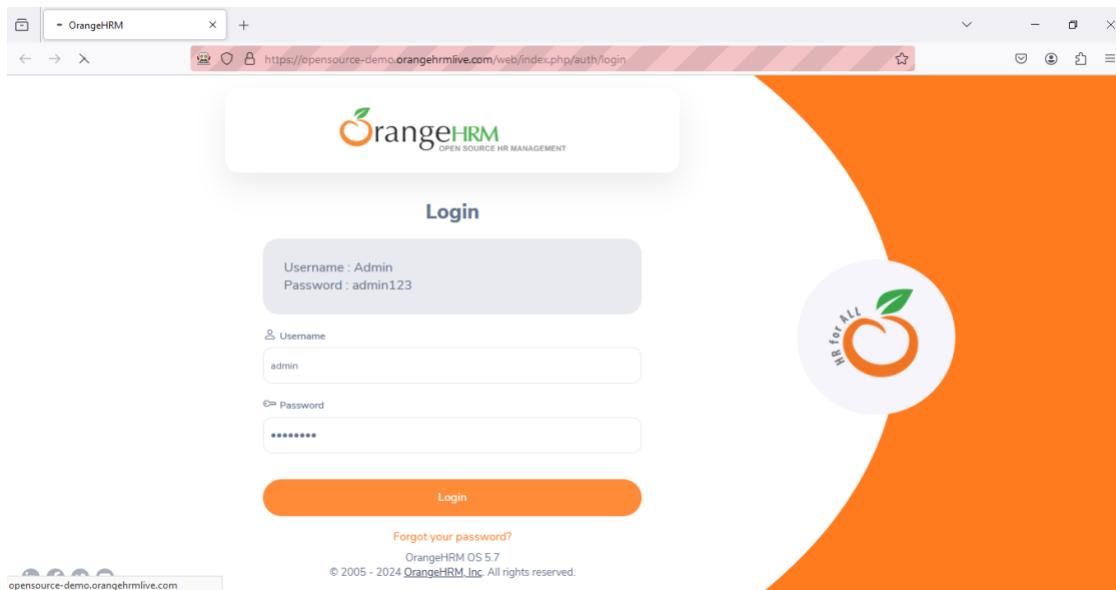
## B) XPath

### CODE:

```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;public class xpath
{public static void main(String[] args) throws Exception {
System.setProperty("webdriver.chrome.driver","C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\geckodriver.exe");
WebDriver wd = new FirefoxDriver();
wd.get("https://opensource-demo.orangehrmlive.com/");
Thread.sleep(5000);
wd.findElement(By.xpath("//html/body/div/div[1]/div/div[2]/div[2]/form/div[
1]/div/div[2]/input")).sendKeys("admin");
wd.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/div[2]/form/div[
2]/div/div[2]/input")).sendKeys("admin123");
wd.findElement(By.xpath("//html/body/div/div/div[1]/div/div[2]/div[2]/form/div[
3]/button")).click();}
}
```

### CODE:

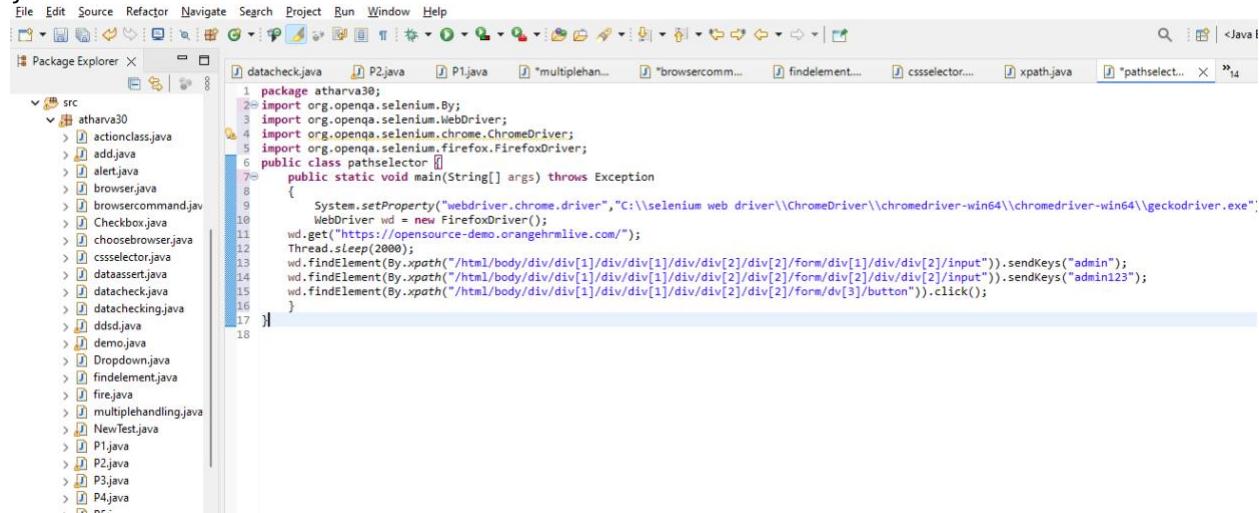




A screenshot of the OrangeHRM dashboard. The URL in the address bar is https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index. The dashboard has a light blue header with the OrangeHRM logo and a user profile for "Oliver Hofer". On the left is a sidebar with various menu items like Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and Dashboard (which is selected and highlighted in orange). The main area contains several cards: "Time at Work" showing a punch-in record for "Punched Out" at 12:56 PM (GMT 5.5), "My Actions" listing tasks like "Timesheet to Approve", "Pending Self Review", and "Candidate to Interview", and "Quick Launch" buttons for Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, and My Timesheet. Other cards include "Buzz Latest Posts" (with a post from Oliver Sven Hofer) and "Employees on Leave Today".

## C) PathSelector

```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class pathselector {
    public static void main(String[] args) throws Exception
    {
        System.setProperty("webdriver.chrome.driver", "C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://opensource-demo.orangehrmlive.com/");
        Thread.sleep(2000);
        wd.findElement(By.xpath("/html/body/div/div[1]/div/div[2]/div[2]/fo
rm/div[1]/div/div[2]/input")).sendKeys("admin");
        wd.findElement(By.xpath("/html/body/div/div[1]/div/div[1]/div/div[2]/div[2]/fo
rm/div[2]/div/div[2]/input")).sendKeys("admin123");
        wd.findElement(By.xpath("/html/body/div/div[1]/div/div[1]/div/div[2]/div[2]/fo
rm/div[3]/button")).click();
    }
}
```



The screenshot displays two windows of the OrangeHRM web application.

**Login Page:** The top window shows the 'Login' screen. It has a text input for 'Username' containing 'admin' and a password input containing 'admin123'. Below the inputs is an orange 'Login' button. To the right of the form is a circular logo with an orange fruit and the text 'HR for ALL'.

**Dashboard:** The bottom window shows the 'Dashboard' page. The left sidebar contains a navigation menu with items like Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and Dashboard (which is selected). The main dashboard area includes several modules: 'Time at Work' (showing a punch-in message and a weekly attendance chart), 'My Actions' (listing pending self-review and interview tasks), 'Quick Launch' (with icons for Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, and My Timesheet), 'Buzz Latest Posts' (a feed from user 'Nimra Shehani'), 'Employees on Leave Today' (a small chart), and 'Employee Distribution by Sub Unit' (another small chart).

## PRACTICAL 7

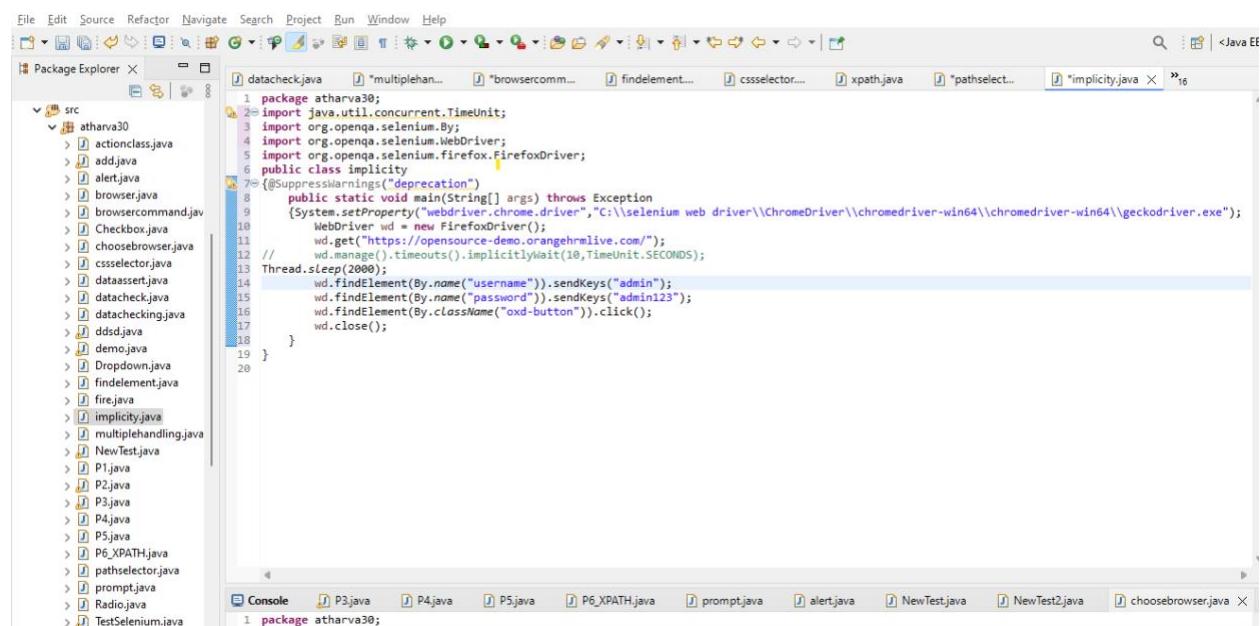
### AIM: Demonstrate Synchronisation in Selenium (Implicit wait, Explicitly wait)

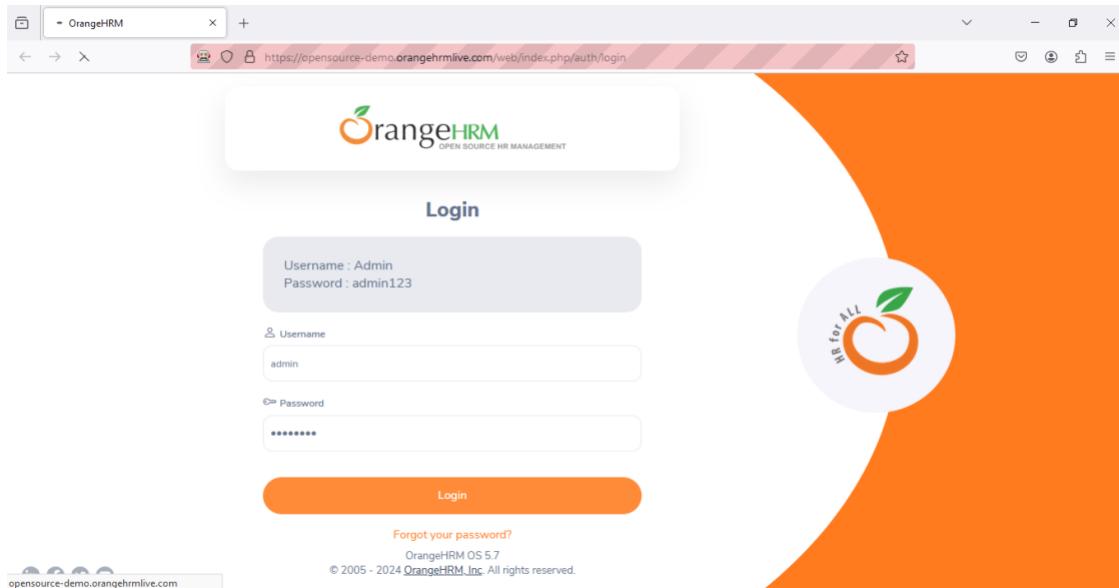
#### A) Implicit wait

#### CODE:

```
package atharva30;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class implicitity
{@SuppressWarnings("deprecation")
    public static void main(String[] args) throws Exception
        {System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://opensource-demo.orangehrmlive.com/");
        //        wd.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        Thread.sleep(2000);
        wd.findElement(By.name("username")).sendKeys("admin");
        wd.findElement(By.name("password")).sendKeys("admin123");
        wd.findElement(By.className("oxd-button")).click();
        wd.close();}}
```

#### OUTPUT:





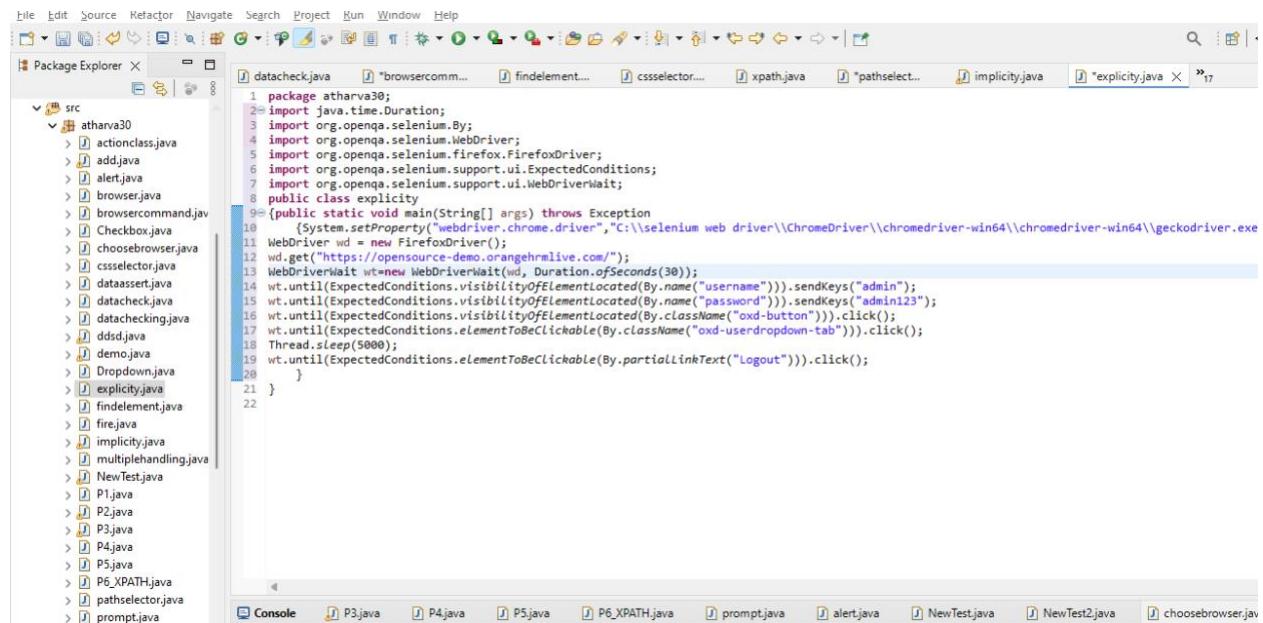
A screenshot of the OrangeHRM dashboard. The URL in the address bar is https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index. The dashboard has a header with the OrangeHRM logo, a search bar, and a user profile for 'Nimra Shahani'. On the left is a sidebar with navigation links: Admin, PIM, Leave, Time, Recruitment, My Info, Performance, Dashboard (which is selected and highlighted in orange), Directory, Maintenance, Claim, and Buzz. The main content area displays several cards: 'Time at Work' (Punched Out, 0h 15m Today, 1h 9m This Week), 'My Actions' (1 Pending Self Review, 1 Candidate to Interview), 'Quick Launch' (Assign Leave, Leave List, Timesheets, Apply Leave, My Leaves, My Timesheet), 'Buzz Latest Posts' (Nimra Shahani, 2024-08-08 11:53 AM, Hello world! :)), and 'Employees on Leave Today' and 'Employee Distribution by Sub Unit' (both partially visible).

## B) Explicitly wait

### CODE:

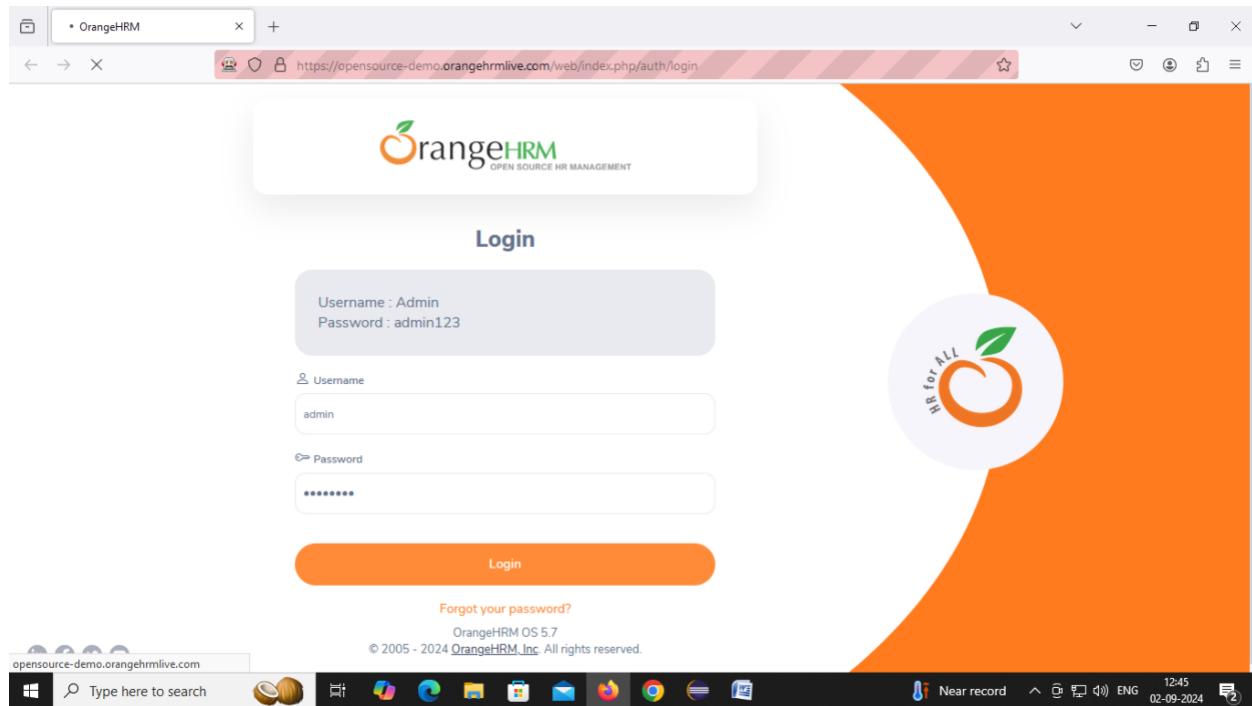
```
package atharva30;
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class explicity
{public static void main(String[] args) throws Exception
    {System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
WebDriver wd = new FirefoxDriver();
wd.get("https://opensource-demo.orangehrmlive.com/");
WebDriverWait wt=new WebDriverWait(wd, Duration.ofSeconds(30));
wt.until(ExpectedConditions.visibilityOfElementLocated(By.name("username"))).sendKeys
("admin");
wt.until(ExpectedConditions.visibilityOfElementLocated(By.name("password"))).sendKeys
("admin123");wt.until(ExpectedConditions.visibilityOfElementLocated(By.className("oxd
-button"))).click();
wt.until(ExpectedConditions.elementToBeClickable(By.className("oxd-userdropdown-
tab"))).click();
Thread.sleep(5000);
wt.until(ExpectedConditions.elementToBeClickable(By.partialLinkText("Logout"))).click
();    }}
```

### OUTPUT:



The screenshot shows the Eclipse IDE interface with the 'Package Explorer' view on the left and the 'Code Editor' view on the right. The 'Package Explorer' shows a package named 'atharva30' containing several Java files. The 'Code Editor' displays the Java code for the 'explicity' class, which implements an explicit wait for element visibility. The code uses the WebDriver and WebDriverWait classes from the Selenium library to interact with the OrangeHRM demo website.

```
1 package atharva30;
2 import java.time.Duration;
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6 import org.openqa.selenium.support.ui.ExpectedConditions;
7 import org.openqa.selenium.support.ui.WebDriverWait;
8 public class explicity
9 {public static void main(String[] args) throws Exception
10     {System.setProperty("webdriver.chrome.driver","C:\\\\selenium web driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
11     WebDriver wd = new FirefoxDriver();
12     wd.get("https://opensource-demo.orangehrmlive.com/");
13     WebDriverWait wt=new WebDriverWait(wd, Duration.ofSeconds(30));
14     wt.until(ExpectedConditions.visibilityOfElementLocated(By.name("username"))).sendKeys("admin");
15     wt.until(ExpectedConditions.visibilityOfElementLocated(By.name("password"))).sendKeys("admin123");
16     wt.until(ExpectedConditions.visibilityOfElementLocated(By.className("oxd-button"))).click();
17     wt.until(ExpectedConditions.elementToBeClickable(By.className("oxd-userdropdown-tab"))).click();
18     Thread.sleep(5000);
19     wt.until(ExpectedConditions.elementToBeClickable(By.partialLinkText("Logout"))).click();
20     }
21 }
22 }
```

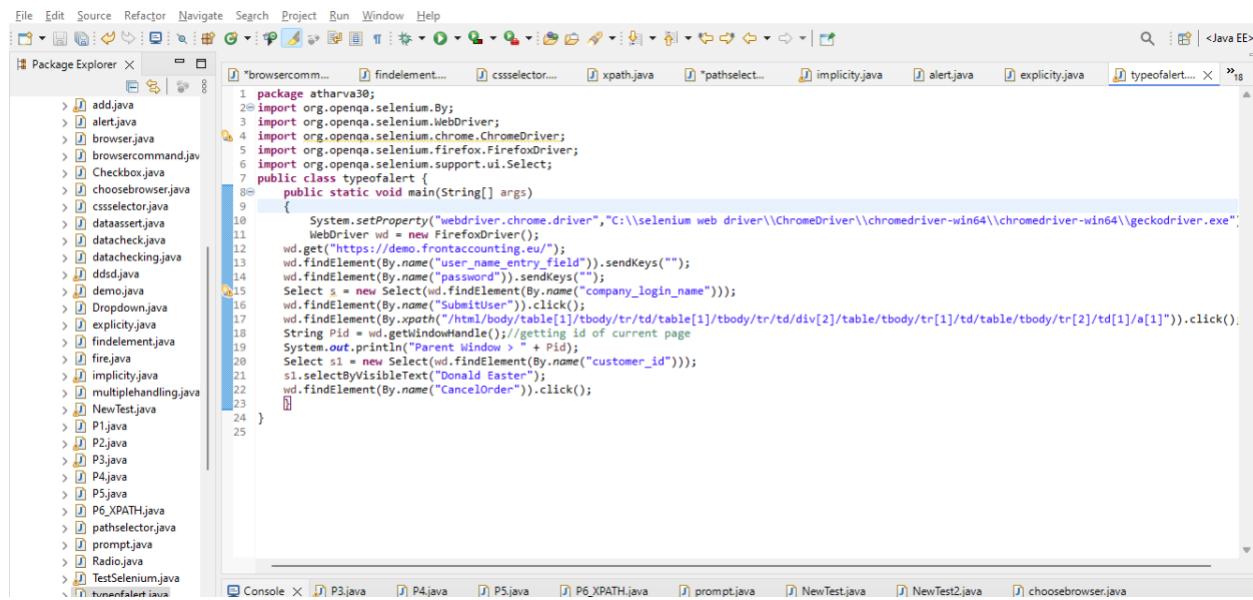


The screenshot shows the OrangeHRM dashboard. The left sidebar lists navigation items like Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and Dashboard (which is selected and highlighted in orange). The main dashboard area displays several cards: "Time at Work" (Punched Out), "My Actions" (Pending Self Review, Candidate to Interview), "Quick Launch" (Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, My Timesheet), "Buzz Latest Posts" (Nimna Shehan, 2024-10-08 12:26 PM), "Employees on Leave Today" (empty), and "Employee Distribution by Sub Unit" (empty). The top right corner shows user information (Nimna Shehan) and a dropdown menu with options like Upgrade, About, Support, Change Password, and Logout. The status bar at the bottom indicates "Near record" and the date "02-09-2024".

## PRACTICAL 8

### Aim:- Demonstrate Different types of alerts.

```
package atharva30;
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.Select;
public class typeofalert {
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver","C:\\selenium web
driver\\ChromeDriver\\chromedriver-win64\\chromedriver-win64\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://demo.frontaccounting.eu/");
        wd.findElement(By.name("user_name_entry_field")).sendKeys("");
        wd.findElement(By.name("password")).sendKeys("");
        Select s = new Select(wd.findElement(By.name("company_login_name")));
        wd.findElement(By.name("SubmitUser")).click();
        wd.findElement(By.xpath("/html/body/table[1]/tbody/tr/td/table[1]/tbody/tr/td/
div[2]/table/tbody/tr[1]/td/table/tbody/tr[2]/td[1]/a[1]")).click();
        String Pid = wd.getWindowHandle(); //getting id of current page
        System.out.println("Parent Window > " + Pid);
        Select s1 = new Select(wd.findElement(By.name("customer_id")));
        s1.selectByVisibleText("Donald Easter");
        wd.findElement(By.name("CancelOrder")).click();
    }
}
```



FrontAccounting 2.4.12 - Login X +

demo.frontaccounting.eu/index.php

Chrome is being controlled by automated test software. X

FrontAccounting 2.4.12 - Login

**FrontAccounting**  
Version 2.4.12 Build 13.07.2022 - Login

User name: demouser  
Password: \*\*\*\*\*  
Company: Training Co.

Login as user: demouser and password: password

Login →

09/19/2024 | 05:06 pm

FrontAccounting 2.4.12 - Theme: default

FrontAccounting

New Sales Quotation Entry X +

https://demo.frontaccounting.eu/sales/sales\_order\_entry.php?NewQuotation=Yes

Sales Purchases Items and Inventory Manufacturing Fixed Assets Dimensions Banking and General Ledger Setup

Training Co. | demo.frontaccounting.eu | Demo User Dashboard Preferences Change password Help Logout

New Sales Quotation Entry

Customer: Donald Easter Current Credit: -998,970.00 Payment: Cash Only Quotation Date: 12/31/2022

Branch: abcd Customer Discount: 0% Price List: Administrative

Reference: 98982479/2024

**Sales Quotation Items**

Item Code	Item Description	Discount %	Total
	Samsung	0.0	0.00
		Shipping Charge	0.00
		Sub-total	0.00
		Amount Total	0.00

Add Item OK Cancel

You are about to void this Document.  
Do you want to continue?

**Cash payment**

Deliver from Location: Default  
Cash account: Petty Cash account

Comments:

Place Quotation Cancel Quotation

Activate Windows  
Go to settings to activate Windows.

## PRACTICAL 9

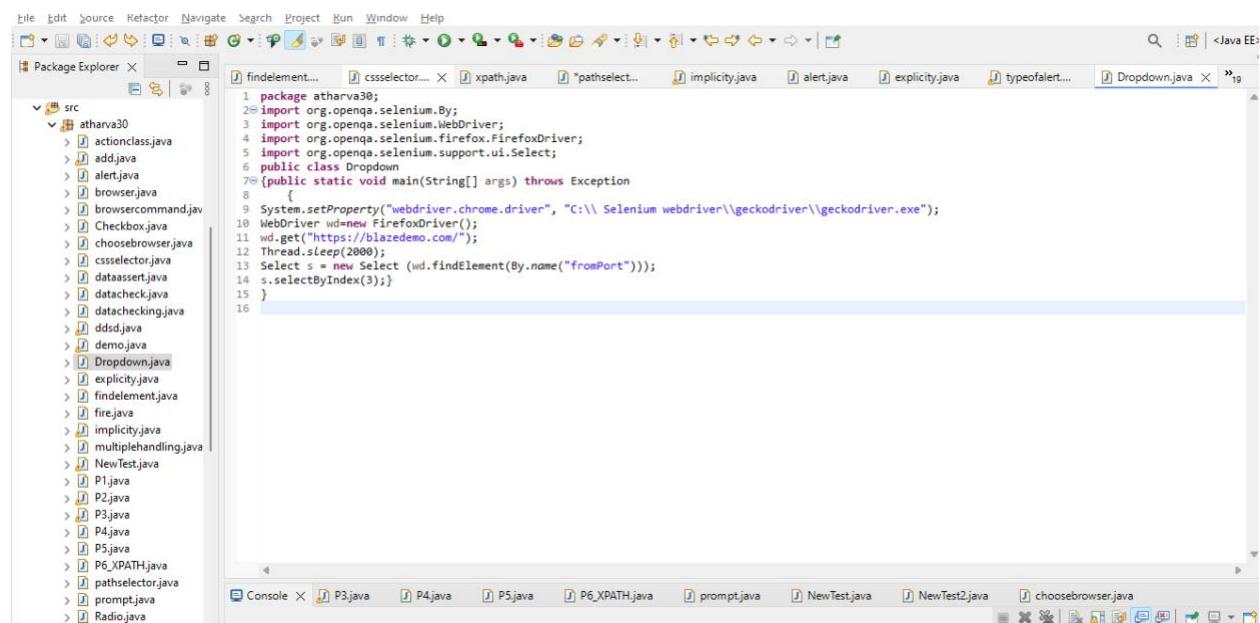
**AIM:**Demonstrate handling drop down list(ByIndex,ByValue,ByText).

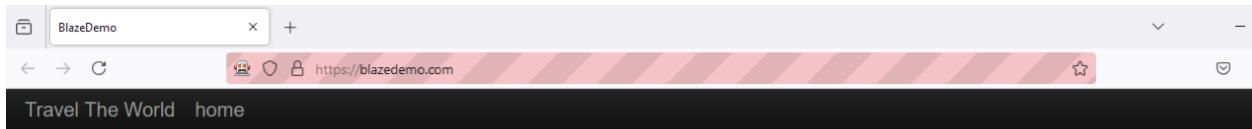
### A) Dropdown List ByIndex

#### CODE:

```
package atharva30;
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 Dropdown
{public static void main(String[] args) throws Exception
{
System.setProperty("webdriver.chrome.driver", "C:\\ Selenium
webdriver\\geckodriver\\geckodriver.exe");
WebDriver wd=new FirefoxDriver();
wd.get("https://blazemeter.com/");
Thread.sleep(2000);
Select s = new Select (wd.findElement(By.name("fromPort")));
s.selectByIndex(3);}
}
```

#### OUTPUT:





## Welcome to the Simple Travel Agency!

This is a sample site you can test with BlazeMeter!

Check out our [destination of the week! The Beach!](#)

**Choose your departure city:**

Portland

**Choose your destination city:**

Buenos Aires

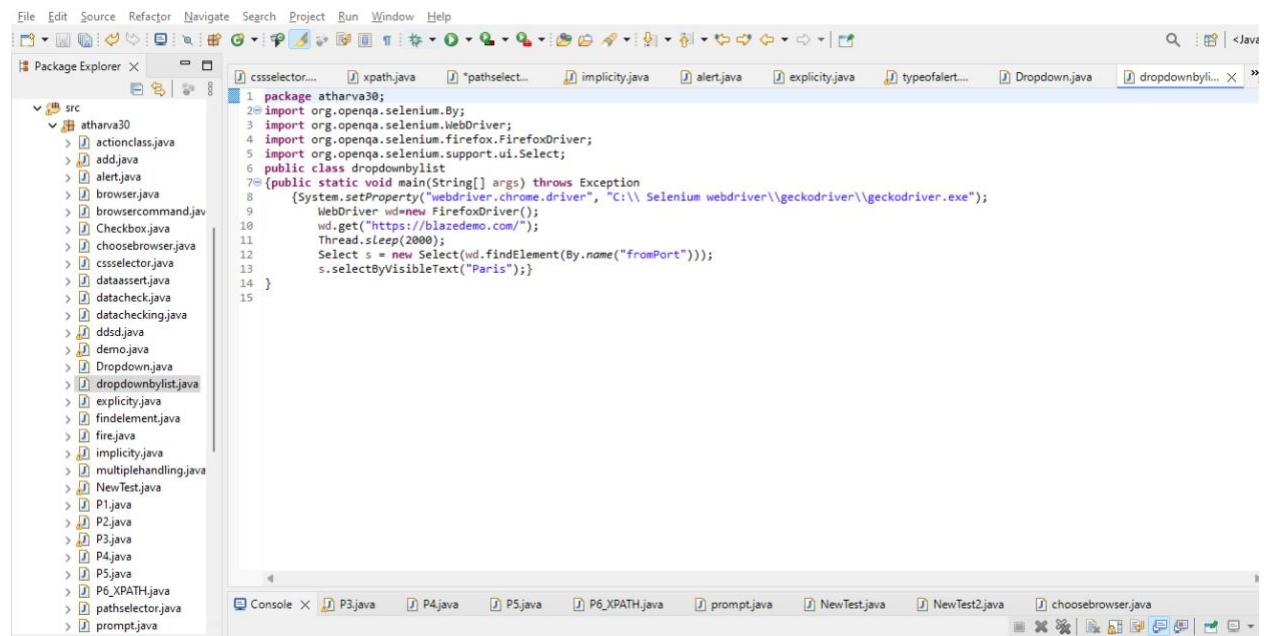
**Find Flights**

## A) DropDownList ByText

### CODE:

```
package atharva30;
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 dropdownbylist
{public static void main(String[] args) throws Exception
    {System.setProperty("webdriver.chrome.driver", "C:\\ Selenium
webdriver\\geckodriver\\geckodriver.exe");
        WebDriver wd=new FirefoxDriver();
        wd.get("https://blazemeter.com/");
        Thread.sleep(2000);
        Select s = new Select(wd.findElement(By.name("fromPort")));
        s.selectByVisibleText("Paris");}
}
```

### OUTPUT:





## Welcome to the Simple Travel Agency!

This is a sample site you can test with BlazeMeter!

Check out our [destination of the week! The Beach!](#)

**Choose your departure city:**

Paris

**Choose your destination city:**

Buenos Aires

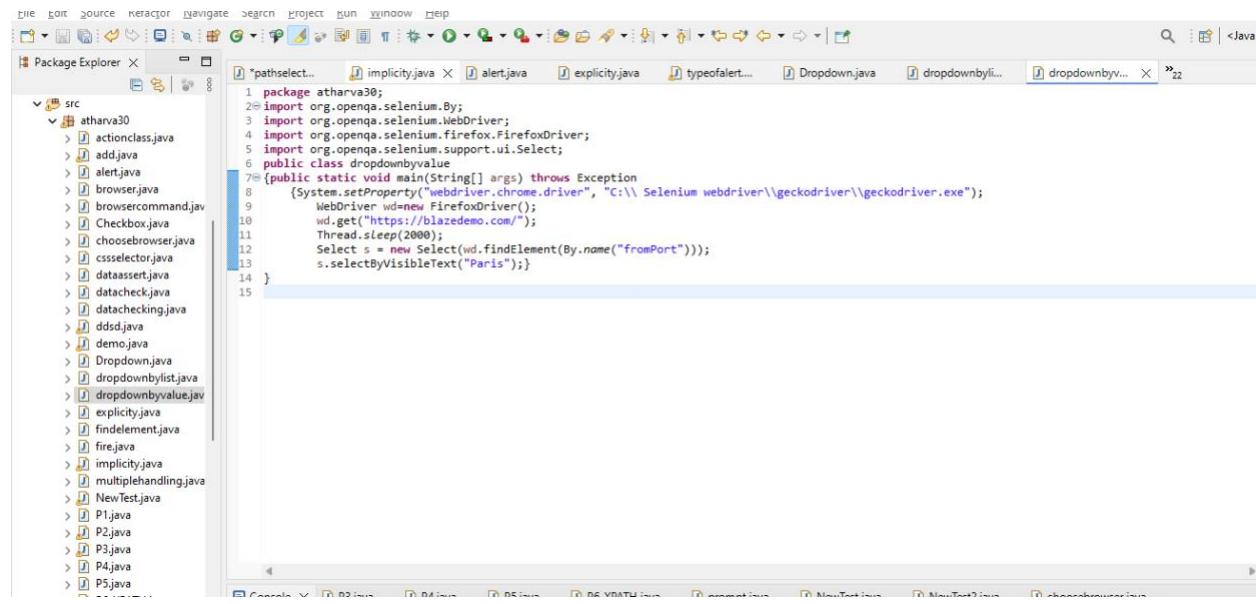
**Find Flights**

## A)DropDownByValue

### CODE:

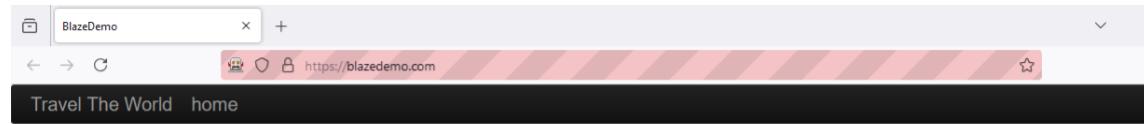
```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
public class dropdownbyvalue
{public static void main(String[] args) throws Exception
    {System.setProperty("webdriver.chrome.driver", "C:\\ Selenium
webdriver\\geckodriver\\geckodriver.exe");
    WebDriver wd=new FirefoxDriver();
    wd.get("https://blazemeter.com/");
    Thread.sleep(2000);
    Select s = new Select(wd.findElement(By.name("fromPort")));
    s.selectByVisibleText("Paris");}
}
```

## OUTPUT:



The screenshot shows the Eclipse IDE interface. The Package Explorer view on the left displays a project structure under 'src' with various Java files. The central editor window shows a Java code snippet for selecting a value from a dropdown using Selenium WebDriver. The code imports org.openqa.selenium.\* and org.openqa.selenium.support.ui.Select, and uses System.setProperty to set the browser driver path. It then creates a FirefoxDriver instance, navigates to blazedemo.com, sleeps for 2 seconds, finds the 'fromPort' dropdown element, and selects 'Paris' by visible text.

```
1 package atharva30;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.firefox.FirefoxDriver;
5 import org.openqa.selenium.support.ui.Select;
6 public class dropdownbyvalue
7 {
8     public static void main(String[] args) throws Exception
9     {
10         System.setProperty("webdriver.chrome.driver", "C:\\ Selenium webdriver\\geckodriver\\geckodriver.exe");
11         WebDriver wd=new FirefoxDriver();
12         wd.get("https://blazedemo.com/");
13         Thread.sleep(2000);
14         Select s = new Select(wd.findElement(By.name("fromPort")));
15         s.selectByVisibleText("Paris");
16     }
17 }
```



### Welcome to the Simple Travel Agency!

The is a sample site you can test with BlazeMeter!

Check out our destination of the week! The Beach!

**Choose your departure city:**

**Choose your destination city:**

**Find Flights**

## B) Listbox(byindex,bytext,byvalue,multiplevalue)

### listbox.html

```
<!DOCTYPE html>
<html>

<head>
    <title>Multiple Listboxes</title>
    <style>
        .listbox {
            width: 200px;
            height: 100px;
            border: 1px solid #ccc;
            overflow-y: scroll;
        }

        .listboxoption {
            padding: 5px;
            cursor: pointer;
        }

        .listboxoption:hover {
            background-color: #f0f0f0;
        }

        label {
            display: block;
        }
    </style>
</head>

<body>

    <!-- Listbox for Cars -->
    <label for="cars">Choose a car:</label>
    <select name="cars" id="cars" class="listbox" size="4">
        <option id="volvo" value="volvo">Volvo</option>
        <option id="saab" value="saab">Saab</option>
        <option id="mercedes" value="mercedes">Mercedes</option>
        <option id="audi" value="audi">Audi</option>
    </select>

    <br><br>

    <!-- Listbox for Anime -->
    <label for="anime">Choose an anime:</label>
    <select name="anime" id="anime" class="listbox" size="4">
```

```
<optionid="naruto" value="naruto">Naruto</option>
<optionid="onepiece" value="onepiece">One Piece</option>
<optionid="bleach" value="bleach">Bleach</option>
<optionid="attackontitan" value="attackontitan">Attack on Titan</option>
</select>

<br><br>

<!-- Listbox for Songs -->
<label for="songs">Choose a song:</label>
<select name="songs" id="songs" class="listbox" size="4">
    <optionid="song1" value="Senorita">Senorita</option>
    <optionid="song2" value="Shape of You">Shape of You</option>
    <optionid="song3" value="BigDawgs">Big Dawgs</option>
    <optionid="song4" value="Dandelious">Dandelious</option>
</select>

</body>

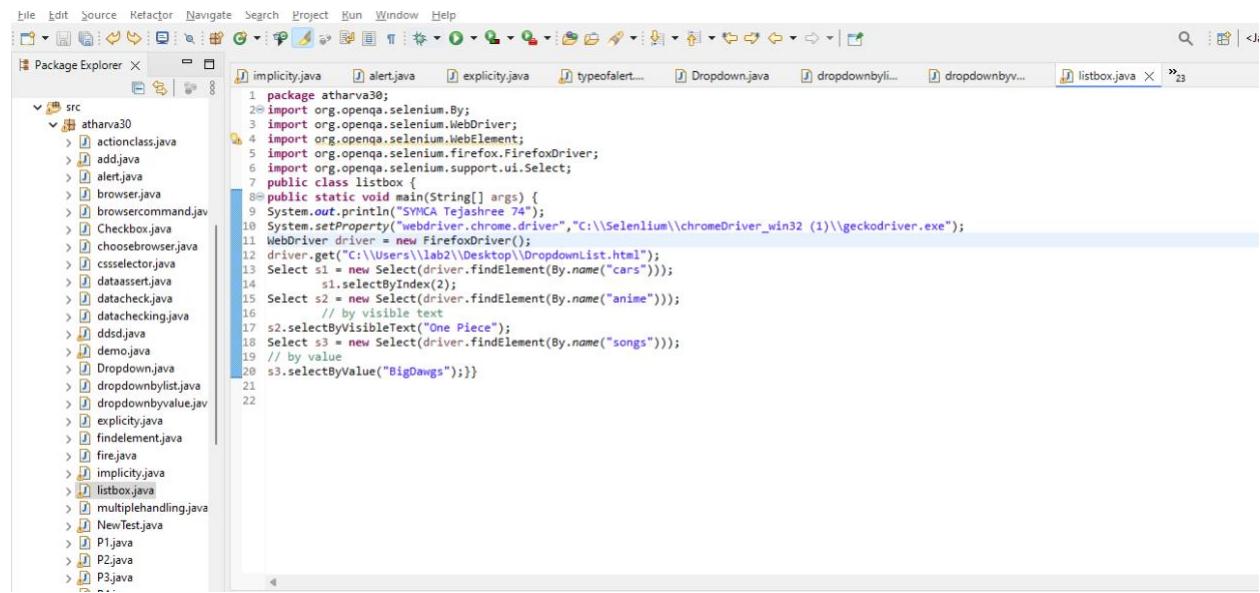
</html>
```

Listbox.java

## CODE:

```
package atharva30;
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.support.ui.Select;
public class listbox {
public static void main(String[] args) {
System.out.println("SYMCA Atharva30 74");
System.setProperty("webdriver.chrome.driver","C:\\\\Selenium\\\\chromeDriver_win32
(1)\\\\geckodriver.exe");
WebDriver driver = new FirefoxDriver();
driver.get("C:\\\\Users\\\\lab2\\\\Desktop\\\\DropdownList.html");
Select s1 = new Select(driver.findElement(By.name("cars")));
s1.selectByIndex(2);
Select s2 = new Select(driver.findElement(By.name("anime")));
// by visible text
s2.selectByVisibleText("One Piece");
Select s3 = new Select(driver.findElement(By.name("songs")));
// by value
s3.selectByValue("BigDawgs");}}
```

## OUTPUT:



The screenshot shows the Eclipse IDE interface with the 'Package Explorer' view open. The project structure under 'src' includes packages like 'atharva30' containing various Java files such as 'actionclass.java', 'add.java', 'alert.java', 'browser.java', 'browsercommand.java', 'Checkbox.java', 'choosebrowser.java', 'cssselector.java', 'datasheet.java', 'datacheck.java', 'datachecking.java', 'ddd.java', 'demo.java', 'Dropdown.java', 'dropdownbylist.java', 'dropdownbyvalue.java', 'explicity.java', 'findelement.java', 'fire.java', 'implicit.java', 'listbox.java', 'multiplehandling.java', 'NewTest.java', 'P1.java', 'P2.java', and 'P3.java'. The 'listbox.java' file is currently selected and its code is displayed in the main editor area. The code uses Selenium WebDriver to interact with dropdown menus.

```
1 package atharva30;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.WebElement;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6 import org.openqa.selenium.support.ui.Select;
7 public class listbox {
8     public static void main(String[] args) {
9         System.out.println("SYMCA Tejasree 74");
10        System.setProperty("webdriver.chrome.driver", "C:\\Selenium\\chromedriver.exe");
11        WebDriver driver = new FirefoxDriver();
12        driver.get("C:\\Users\\lab2\\Desktop\\DropdownList.html");
13        Select s1 = new Select(driver.findElement(By.name("cars")));
14        s1.selectByIndex(2);
15        Select s2 = new Select(driver.findElement(By.name("anime")));
16        // by visible text
17        s2.selectByVisibleText("One Piece");
18        Select s3 = new Select(driver.findElement(By.name("songs")));
19        // by value
20        s3.selectByValue("BigDawgs");}}
```

Choose a car:



A dropdown menu with the following options: Volvo, Saab, Mercedes, and Audi. The option 'Mercedes' is highlighted.

Choose an anime:



A dropdown menu with the following options: Naruto, One Piece, Bleach, and Attack on Titan. The option 'One Piece' is highlighted.

Choose a song:



A dropdown menu with the following options: Senorita, Shape of You, Big Dawgs, and Dandelious. The option 'Big Dawgs' is highlighted.

## PRACTICAL 10

---

### **AIM:- Demonstrate**

**A)Radio Button**

**B)Checkbox**

**A)Radio Buttton**

**CODE:**

**Radio.html**

```
<html>
<head>
<body>
<form>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname">
    <br>
    <label for="fav_language">Favourite Language</label><br>
    <input type="radio" id="html" name="fav_language" value="HTML">
    <label for="html">HTML</label><br>
    <input type="radio" id="css" name="fav_language" value="CSS">
    <label for="css">CSS</label><br>
    <input type="radio" id="javascript" name="fav_language" value="JavaScript">
    <label for="javascript">JavaScript</label>
</form>
</body>
</html>
```

**Radio.java**

```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class Radio
{public static void main(String[] args)
{
```

```
System.setProperty("webdriver.chrome.driver", "C:\\ Selenium
webdriver\\geckodriver\\geckodriver.exe");
WebDriver wd = new FirefoxDriver();
wd.get("file:///C:/Users/lab2/Desktop/Radio.html");
wd.findElement(By.xpath("//input[@value='HTML']")).click();
}
}
```

## OUTPUT:

The screenshot shows the Eclipse IDE interface. The top part displays the Java code for a 'Radio' class. The bottom part shows a browser window displaying a form with a radio button group.

**Eclipse IDE Screenshot:**

- Java Code (Radio.java):**

```
1 package atharva30;
2 import org.openqa.selenium.*;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.WebElement;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6 public class Radio
7 {
8     public static void main(String[] args)
9     {
10         System.setProperty("webdriver.chrome.driver", "C:\\ Selenium webdriver\\geckodriver\\geckodriver.exe");
11         WebDriver wd = new FirefoxDriver();
12         wd.get("file:///C:/Users/lab2/Desktop/Radio.html");
13         wd.findElement(By.xpath("//input[@value='HTML']")).click();
14     }
15 }
```
- Browser Output:**

A screenshot of a web browser window showing a form. The URL bar indicates the page is loaded from 'file:///C:/Users/lab2/Desktop/Radio.html'. The form contains fields for 'First Name' and 'Last Name', and a section for 'Favourite Langauge' with three radio buttons: 'HTML' (selected), 'CSS', and 'JAVASCRIPT'.

First Name:

Last Name:

Favourite Langauge

HTML

CSS

JAVASCRIPT

## B) Checkbox

### Checkbox.html

```
<html>
<head>
    <title>Checkbox Example</title>
</head>
<body>
    <h2>Checkbox Example</h2>
    <form id="checkboxForm">
        <label><input type="checkbox" id="checkbox1" value="option1">C</label>
        <br>
        <label><input type="checkbox" id="checkbox2" value="option2">C++</label><br>
        <label><input type="checkbox" id="checkbox3" value="option3">Java</label><br>
        <button type="button" onclick="submitForm()">Submit</button>
    </form>
    <script>
        function submitForm() {
            document.getElementById("checkboxForm").submit();
        }
    </script>
</body>
</html>
```

### Checkbox.java

#### CODE:

```
package atharva30;

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

public class Checkbox {
    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub
        System.setProperty("webdriver.chrome.driver", "D:\\Selenium\\chromedriver-win64\\chromedriver.exe");
        WebDriver wd = new ChromeDriver();

        wd.get("C:\\Users\\USER\\Desktop\\Checkbox\\index.html");
        WebElement vehicle1 = wd.findElement(By.id("vehicle1"));
        vehicle1.click();

        WebElement vehicle2 = wd.findElement(By.xpath("//input[@value='option2']"));
    }
}
```

```
vehicle2.click();

WebElement vehicle3 =
wd.findElement(By.xpath("//label[contains(text(),'Option3')]/input"));
vehicle3.click();

try {
    Thread.sleep(2000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
wd.quit();
}
```

## }OUTPUT:

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is the Package Explorer view, which displays a package named 'src' containing several Java files: actionclass.java, add.java, alert.java, browser.java, browsercommand.java, Checkbox.java, choosebrowser.java, cssselector.java, dataassert.java, datacheckbox.java, datachecking.java, ddd.java, demo.java, Dropdown.java, dropdownbylist.java, dropdownbyvalue.java, explicity.java, findelement.java, fire.java, implicity.java, listbox.java, multiplehandling.java, NewTest.java, P1.java, P2.java, P3.java, and P4.java. The 'Checkbox.java' file is currently selected and open in the editor. The code in the editor is as follows:

```
1 package atharva30;
2
3 import org.openqa.selenium.By;
4
5 public class Checkbox {
6     public static void main(String[] args) throws Exception {
7         // TODO Auto-generated method stub
8         System.setProperty("webdriver.chrome.driver", "D:\\Selenium\\chromedriver-win64\\chromedriver.exe");
9         WebDriver wd = new ChromeDriver();
10
11         wd.get("file:///C:/Users/USER/Desktop/Checkbox/index.html");
12         WebElement vehicle1 = wd.findElement(By.id("vehicle1"));
13         vehicle1.click();
14
15         WebElement vehicle2 = wd.findElement(By.xpath("//input[@value='option2']"));
16         vehicle2.click();
17
18         WebElement vehicle3 = wd.findElement(By.xpath("//label[contains(text(),'Option3')]/input"));
19         vehicle3.click();
20
21         try {
22             Thread.sleep(2000);
23         } catch (InterruptedException e) {
24             e.printStackTrace();
25         }
26         wd.quit();
27     }
28 }
29
30 }
```

Below the editor, the Eclipse status bar shows various toolbars and icons. At the bottom, there is a browser window titled 'Checkbox Example'. The address bar indicates the URL is 'file:///C:/Users/lab2/Desktop/Checkbox.html'. The browser displays a simple form with three checkboxes labeled 'C', 'C++', and 'Java'. The first two checkboxes are checked, while the third is unchecked.

## Checkbox Example

C  
 C++  
 Java

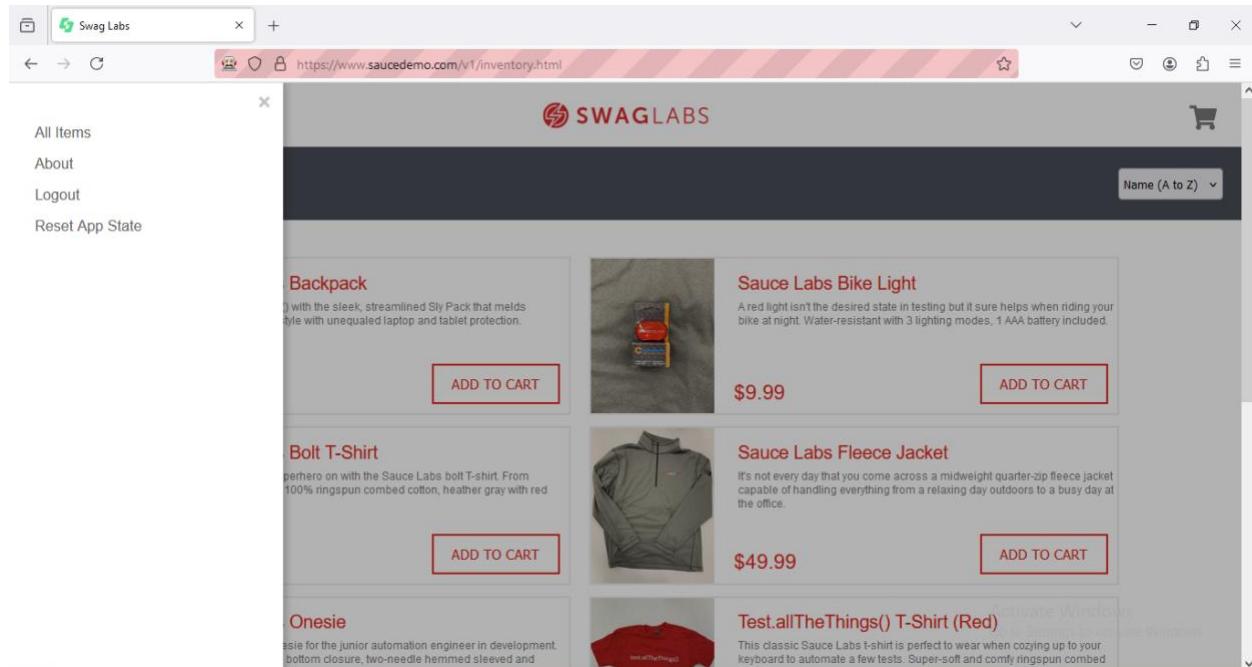
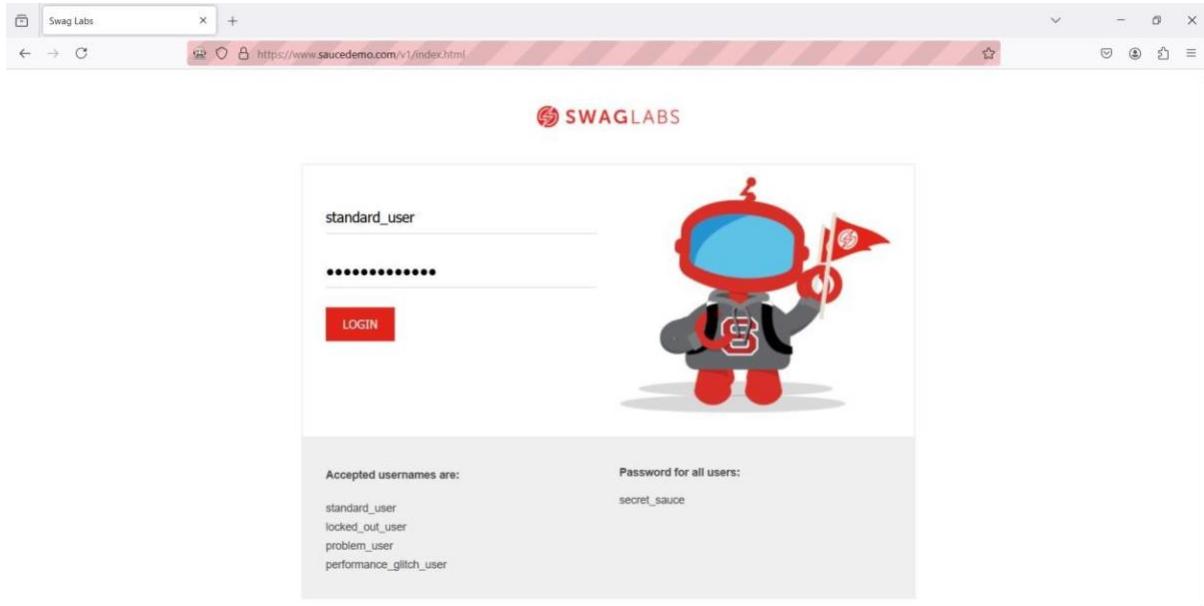
## PRACTICAL 11

### Aim:- Demonstrate action classes in Selenium

```
package atharva30;
import java.util.List;
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 actionclasses {
    public static void main(String[] args) throws InterruptedException
    {System.setProperty("webdriver.chrome.driver", "C:\\ Selenium
webdriver\\geckodriver\\geckodriver.exe");
    WebDriver wd=new FirefoxDriver();
    wd.get("https://www.saucedemo.com/v1/");
    Thread.sleep(3000);
    wd.findElement(By.name("user-name")).sendKeys("standard_user");
    wd.findElement(By.name("password")).sendKeys("secret_sauce");
    wd.findElement(By.id("login-button")).click();
    Thread.sleep(3000);
    wd.findElement(By.className("bm-burger-button")).click();
    Actions act = new Actions(wd);
    Thread.sleep(3000);
    List<WebElement> menu = wd.findElements(By.className("bm-item-list"));
    for(int i=0; i<=menu.size()-1; i++)
    System.out.println(menu.get(i).getText());
    act.moveToElement(menu.get(i)).perform(); } }
```

The screenshot shows a Java code editor with the code for 'actionclasses' class. The code uses Selenium WebDriver and Actions to interact with the Sauce Demo website. The code includes imports for WebDriver, WebElement, Actions, and various By locators. It sets up the driver, navigates to the site, logs in, and then iterates through the items in the sidebar, printing their text and moving the mouse over them. The code is annotated with line numbers from 5 to 33. Below the code editor is a standard Eclipse-style toolbar with icons for file operations, and at the bottom, there's a status bar showing the terminal output and some navigation links.

}

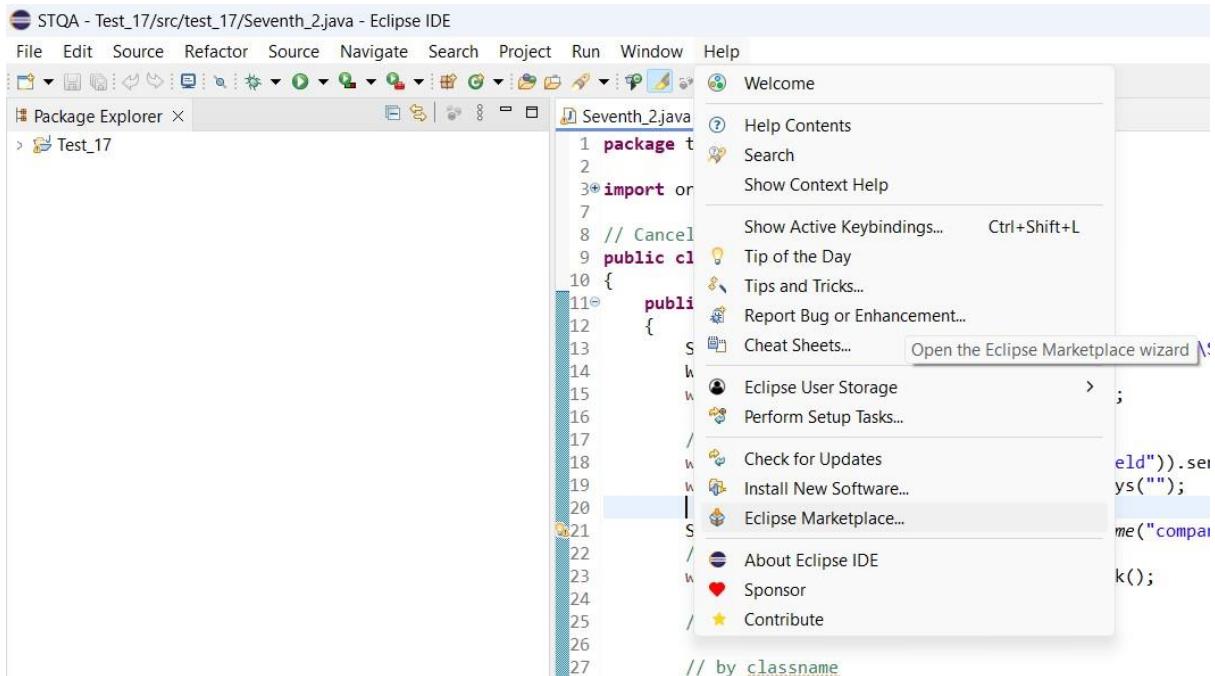


## PRACTICAL 12

**Aim: Installation of TestNg, running TestNg & TestNg annotations.**

### Installation:

Click on Help > Eclipse Marketplace



Then Type testNg and press Go

Eclipse Marketplace

## Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.  
Press the "more info" link to learn more about a solution.

Search Recent Popular Favorites Installed Research at the Eclipse

Find:  All Markets All Categories Go

**TestNG for Eclipse**

This plug-in lets you run your TestNG tests from Eclipse. You can run suites, groups or individual methods. Errors are reported in a separate tab that lets you... [more info](#)  
by [Cédric Beust](#), Apache 2.0  
[testing](#) [junit](#) [testing](#) [unit](#) [integration](#) [functional](#) [selenium](#)

★ 775 Installs: 2.12M (20,988 last month) [Install](#)

Eclipse Marketplace

### Confirm Selected Features

Press Confirm to continue with the installation. Or go back to choose more solutions to install.

-  TestNG for Eclipse <https://testing.org/testng-eclipse-update-site>
-  TestNG (required)
-  TestNG M2E (Maven) Integration (Optional)

Eclipse Marketplace

### Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

Licenses:	License text:
<a href="#">Apache License</a>	Apache License Version 2.0, January 2004 <a href="http://www.apache.org/licenses/">http://www.apache.org/licenses/</a> TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions: "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity controlling it. Apache License Version 2.0, January 2004 <a href="http://www.apache.org/licenses/">http://www.apache.org/licenses/</a> TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions: "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity controlling it.

I accept the terms of the license agreement   
I do not accept the terms of the license agreement

?

< Install More Confirm > Finish Cancel

?

< Back Next > Finish Cancel

**Trust Authorities**

Do you trust content originating from the following authorities? Installing content involves performing actions that alter the installation's configuration and may potentially be used for malicious purposes.

Authority / Update Site	Units Secured
> https://testing.org	19 ✓

Remember selected authorities    Always trust all authorities    Select All    Deselect All    Expand All    Collapse All

Details... Export...  Details...  ?  Trust Selected  Cancel

**Trust Artifacts**

⚠ Do you trust unsigned content of unknown origin?

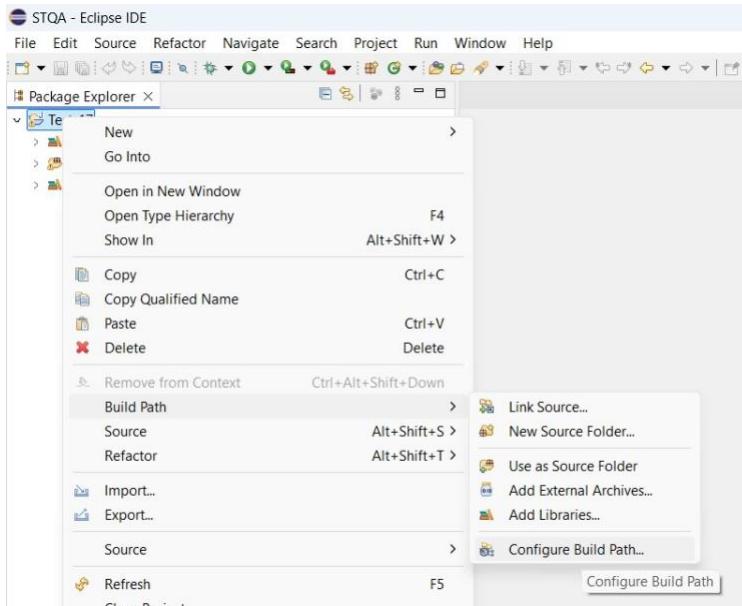
Type	Id/Fingerprint	Name	Validity Dates
Unsigned	n/a	Unknown	n/a

Always trust all content    Select All    Deselect All

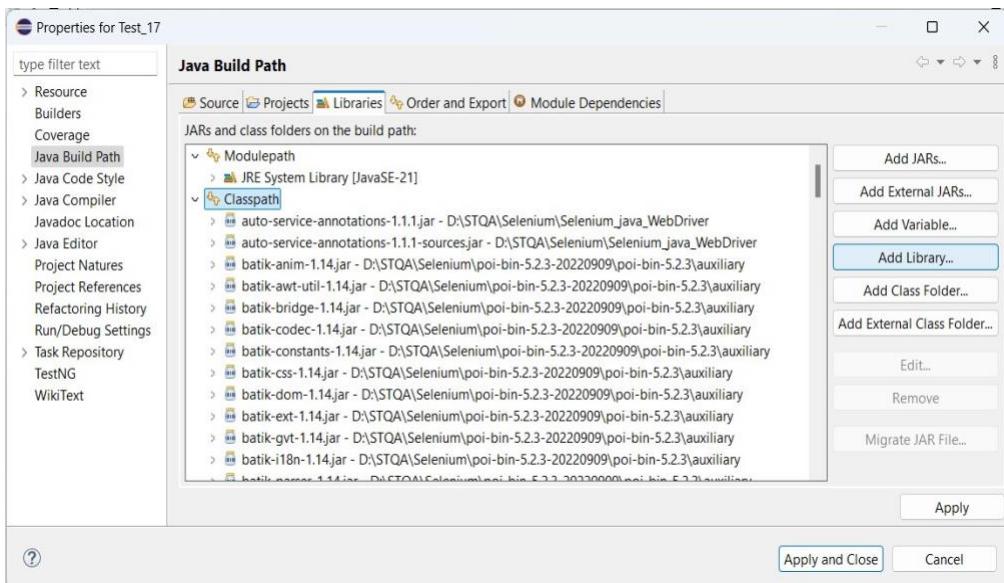
Classifier	Id	Version
osgi.bundle	com.beust.jcommander	1.78.0
osgi.bundle	com.beust.jcommander.source	1.78.0
osgi.bundle	org.apache-extras.beanshell.bsh	2.0.0.b6
osgi.bundle	org.apache-extras.beanshell.bsh.source	2.0.0.b6
osgi.bundle	org.testing	7.4.0.r202105021533
osgi.bundle	org.testing.eclipse	7.10.0.r202404131040
osgi.bundle	org.testing.eclipse.feature.group	7.10.0.r202404131040
osgi.bundle	org.testing.eclipse.feature.jar	7.10.0.r202404131040
osgi.bundle	org.testing.eclipse.maven	7.10.0.r202404131040
osgi.bundle	org.testing.eclipse.maven.feature.feature....	7.10.0.r202404131040
osgi.bundle	org.testing.source	7.4.0.r202105021533

?  Trust Selected  Cancel

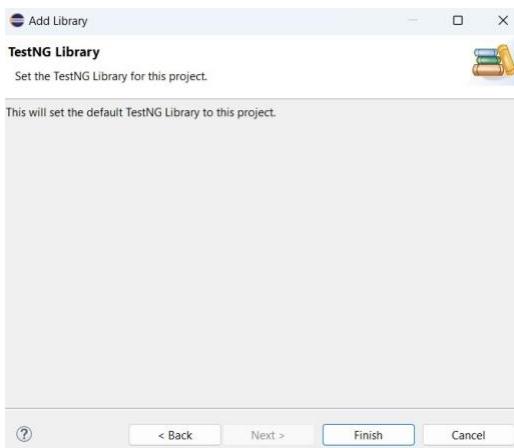
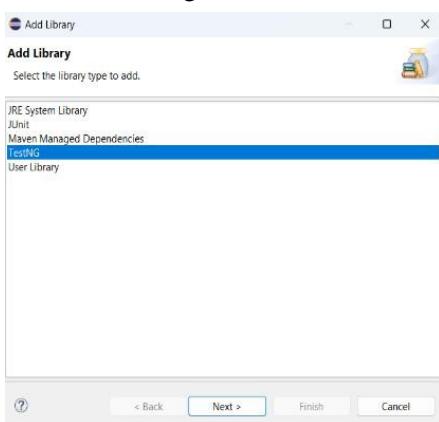
Right-Click on Project an click on Build Path then click on “Configure Build Path”

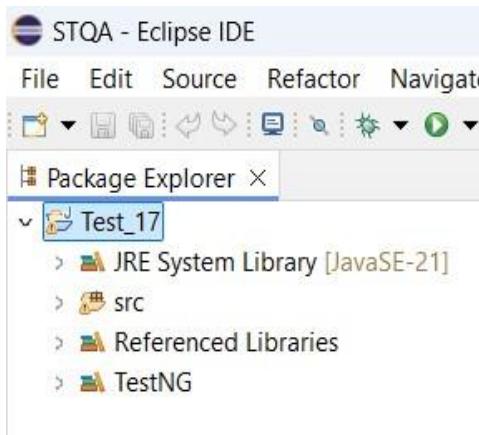
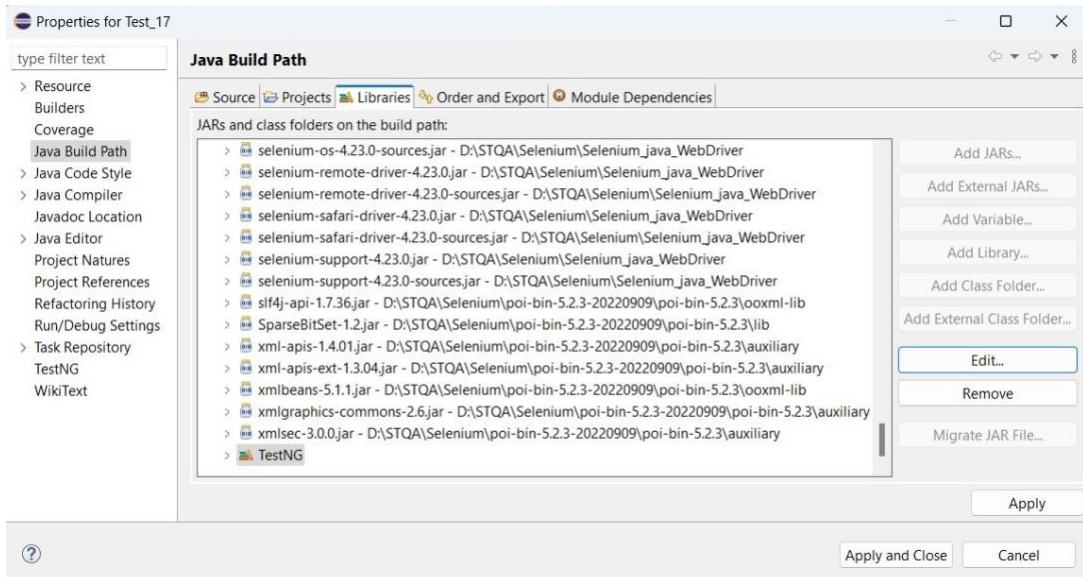


### Click on Add Library



### Click on TestNg then Click on Next





## A) DemoDataProvider

CODE:

```
package testNG;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
public class prcat12
{
    @Test(dataProvider = "dp")
    public void f(Integer n, String s)
    {
        System.out.println(n + " " + s);
    }
    @DataProvider
    public Object[][] dp()
    {
        return new Object[][]
        {
            new Object[] { 1, "A" },
            new Object[] { 2, "B" },
            new Object[] { 3, "C" },
            new Object[] { 4, "Atharva" }
        };
    }
}
```

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Package Explorer:** Shows a tree view of Java files in the project, including: Checkbox.java, demo.java, NewTest.java, dropdownbylist.java, NewTest2.java, dropdownbyvalue.java, explicity.java, findelement.java, fire.java, implicitity.java, listbox.java, multiplehandling.java, NewTest.java, P1.java, P2.java, P3.java, P4.java, P5.java, P6\_XPATH.java, pathselector.java, prompt.java, Radio.java, TestSelenium.java, typeofalert.java, and ..
- Editor:** Displays the code for `NewTest2.java`. The code defines a class `NewTest2` with a single method `f` annotated with `@Test` and a parameter `n` annotated with `dataProvider = "dp"`. It also contains a `@DataProvider` annotation returning an `Object[][]` array containing four rows of `{ Integer, String }` pairs: `{ 1, "A" }, { 2, "B" }, { 3, "C" }, { 4, "Atharva" }`.
- Console:** Shows the output of the test execution:  
<terminated> NewTest2 [TestNG] C:\Users\USER\p2\pool\plugins\org.eclipse.jdt.core\hotspot\jre\full\win32\x86\_64\_22.0.1.v20240426-1149\jre\bin\javaw.exe (Oct 25, 2024, 4:30:16 PM)  
[RemoteTestNG] detected TestNG version 7.4.0  
1 a

## Output:-

```
Console X P3.java P4.java P5.java P6_XPATH.java prompt.java choosebrowser.java

<terminated> NewTest2 [TestNG] C:\Users\USER\p2\pool\plugins\org.eclipse.jst.junit.framework\org.eclipse.jst.junit.framework_1.1.0.v20240426-1149\jre\bin\javaw.exe (Oct 25, 2024, 4:30:16 PM - 4:30:16 PM)
[RemoteTestNG] detected TestNG version 7.4.0
1 a
2 b
3 c
4 Atharva
PASSED: f(2, "b")
PASSED: f(1, "a")
PASSED: f(3, "c")
PASSED: f(4, "Atharva")

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

=====
Default suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
=====
```

## B) DemoAnnotation

### CODE:

```
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 pract12B {
    @Test
    public void f() {System.out.println("Roll no: 30");
    }
    @BeforeMethod
    public void beforeMethod() {
        System.out.println("Before Method");
    }

    @AfterMethod
    public void afterMethod() {
        System.out.println("After Method");
    }

    @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");
    }

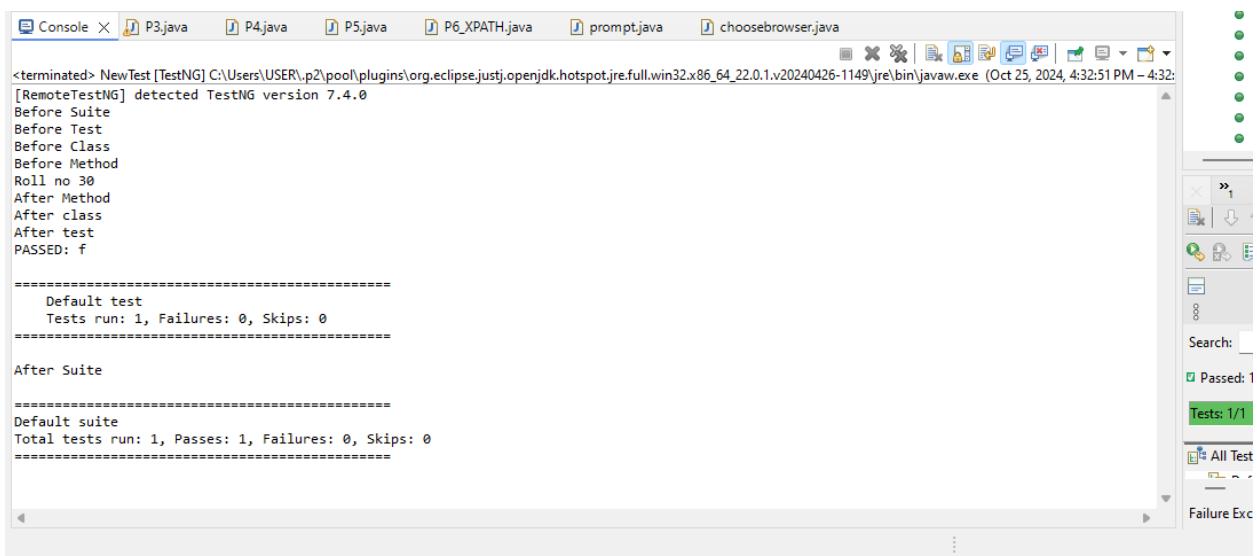
    @BeforeSuite
    public void beforeSuite() {
        System.out.println("Before Suite");
    }

    @AfterSuite
```

```
public void afterSuite() {
    System.out.println("After Suite");
}

}
```

## Output:



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The console window displays the output of a TestNG test run. The output includes the following text:

```
<terminated> NewTest [TestNG] C:\Users\USER\p2\pool\plugins\org.eclipse.jdt.core\org.eclipse.jdt.core_22.0.1.v20240426-1149\jre\bin\javaw.exe (Oct 25, 2024, 4:32:51 PM – 4:32:51 PM)
[RemoteTestNG] detected TestNG version 7.4.0
Before Suite
Before Test
Before Class
Before Method
Roll no 30
After Method
After class
After test
PASSED: f

=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====
After Suite
=====
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
=====
```

The right side of the interface shows the TestNG results summary, indicating 1 passed test.

### C) DataOHM

#### CODE:

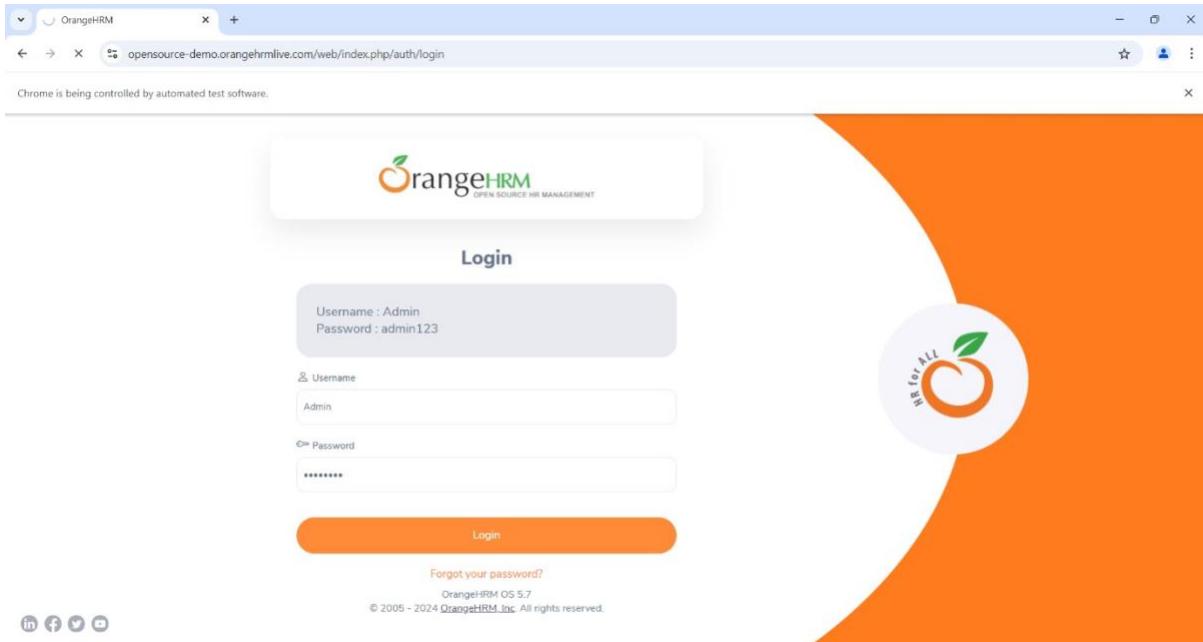
```
package testNG;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Test;
import org.testng.annotations.DataProvider;

@Test(dataProvider = "dp")
public class prcat12C
{
    public void f(String u, String p)
    {
        System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.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(u);//locator id
        wd.findElement(By.name("password")).sendKeys(p);//locater name
        wd.findElement(By.className("oxd-button")).click();

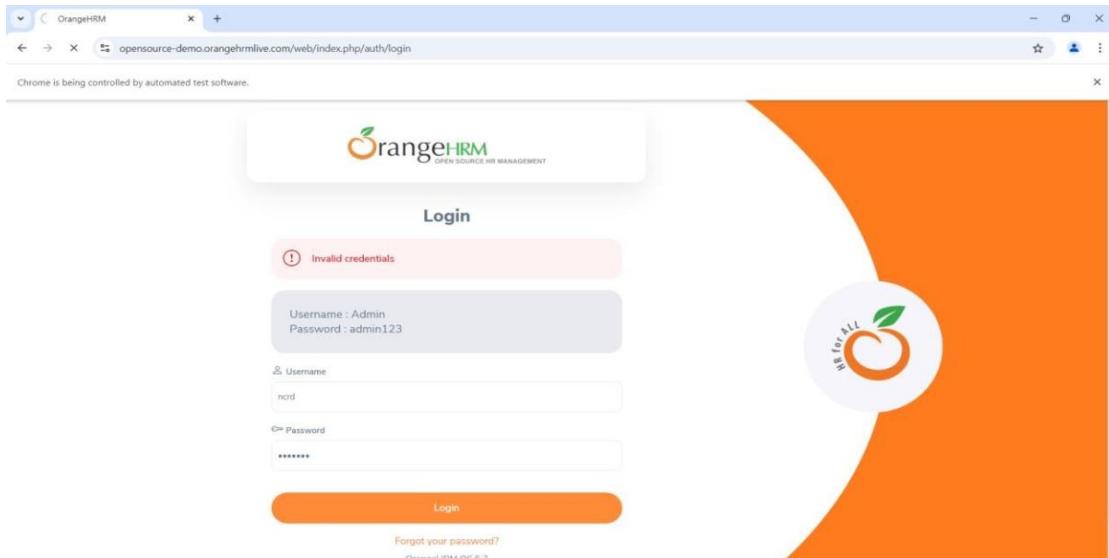
        try
        {
            wd.findElement(By.className("oxd-userdropdown-name")).click();
            wd.findElement(By.className("oxd-userdropdown-link")).click();
            System.out.println("Pass");
        }
        catch (Exception e)
        {
            System.out.println("Fail");
        }
    }

    @DataProvider
    public Object[][] dp()
    {
        return new Object[][]
        {
            new Object[] { "Admin", "admin123" },
            new Object[] { "add", "b" },
        };
    }
}
```

## Output:-



The screenshot shows the OrangeHRM dashboard. The left sidebar includes links for Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and Dashboard (which is currently selected). The main dashboard area has several modules: "Time at Work" showing a punch record for today, "My Actions" with pending self-review and candidate-to-interview tasks, "Quick Launch" with icons for Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, and My Timesheet, "Buzz Latest Posts" showing a recent post from Nimna Shehani, "Employees on Leave Today" (empty), and "Employee Distribution by Sub Unit" (empty). The top right corner shows a user profile for Nimna Shehani with options for Upgrade, About, Support, Change Password, and Logout.



```
Console X Servers Results of running class prcat12C
<terminated> prcat12C [TestNG] C:\Program Files\Java\jdk-22\bin\javaw.exe (23 Oct 2024, 12:08:41 pm - 12:09:21 pm) [pid: 6848]
Pass
Fail
PASSED: f("Admin", "admin123")
PASSED: f("add", "b")

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

## D) DataCheck

### CODE:

```
package testNG;

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

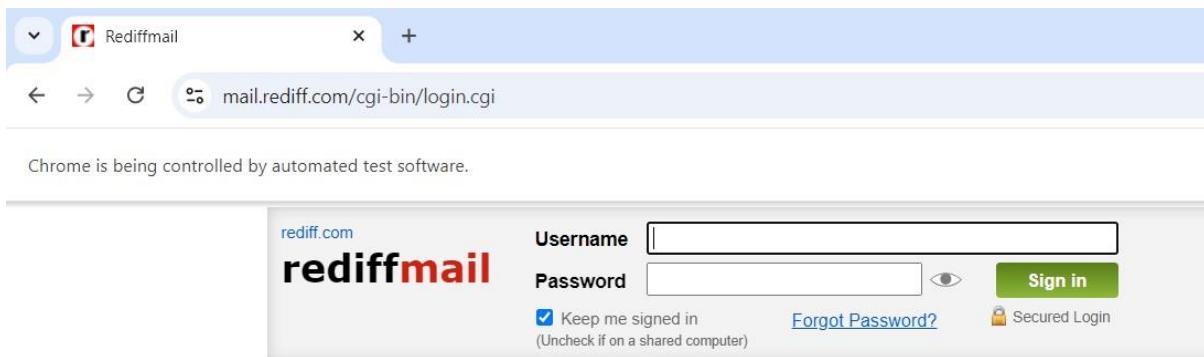
public class pract12D
{
    @Test(description="Checking status of checkbox")
    public void ChekBox()
    {

        System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        WebDriver wd=new FirefoxDriver();

        wd.get("https://mail.rediff.com/cgi-bin/login.cgi");

        boolean sts = wd.findElement(By.name("remember")).isSelected();
        Assert.assertTrue(sts);
    }
}
```

### Output:-



```
Console X Servers Results of running class pract120
<terminated> pract120 [TestNG] C:\Program Files\Java\jdk-22\bin\javaw.exe (23 Oct 2024, 1:09:08 pm - 1:09:18 pm) [pid: 11848]
[RemoteTestNG] detected TestNG version 7.4.0
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
ERROR StatusLogger Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger to log to the console...
PASSED: CheckBox
    Checking status of checkbox

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

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

## E) DemoAssertFA

### CODE:

```
package TestNG;
import org.testng.Assert;
import org.testng.annotations.Test;
public class WebDriver
{
    @Test
    public void frontAccounting()
    {
        System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-
win64\\\\geckodriver.exe");
        WebDriver wd=new WebDriver();
        wd.get("https://demo.frontaccounting.eu/index.php");
        String expectedTitle = "FrontAccounting 2.4.12 - Login";//if title is not matching
        execution will be fail
        String actualTitle = wd.getTitle();
        Assert.assertEquals(actualTitle, expectedTitle);
        System.out.println("Title of Webpage =====> " + actualTitle);
    }
    private String getTitle() {
        // TODO Auto-generated method stub
        return null;
    }
    private void get(String string) {
        // TODO Auto-generated method stub
    }
}
```

### Output:

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The Package Explorer view on the left lists various Java files such as actionclass.java, add.java, alert.java, browser.java, browsercommand.java, Checkbox.java, choosebrowser.java, cssselector.java, dataassert.java, datacheck.java, datachecking.java, ddd.java, demo.java, Dropdown.java, dropdownbylist.java, dropdownbyvalue.java, explicity.java, findelement.java, fire.java, implicit.java, listbox.java, multiplehandling.java, NewTest.java, P1.java, P2.java, P3.java, P4.java, P5.java, P6\_XPATH.java, prompt.java, and choosebrowser.java. The code editor window displays the Java code for the WebDriver class, specifically the frontAccounting() method. The code sets up a WebDriver instance, navigates to a URL, and performs an assertEquals assertion on the page title. The bottom status bar indicates the terminal output: <terminated> NewTest [TestNG] C:\Users\USER\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86\_64\_22.0.1.v20240426-1149\jre\bin\javaw.exe (Oct 25, 2024, 4:32).

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X
actionclass.java *NewTest3.java demo.java NewTest.java NewTest2.java *listbox.java *Radio.java WebDriver.java
1 package TestNG;
2
3
4 import org.testng.Assert;
5 import org.testng.annotations.Test;
6
7 public class WebDriver
8 {
9     @Test
10    public void frontAccounting()
11    {
12        System.setProperty("webdriver.chrome.driver","C:\\\\selenium web driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-
13        wd=new WebDriver();
14        wd.get("https://demo.frontaccounting.eu/index.php");
15
16        String expectedTitle = "FrontAccounting 2.4.12 - Login";//if title is not matching
17        execution will be fail
18
19        String actualTitle = wd.getTitle();
20        Assert.assertEquals(actualTitle, expectedTitle);
21
22        System.out.println("Title of Webpage =====> " + actualTitle);
23    }
24
25    private String getTitle() {
26        // TODO Auto-generated method stub
27        return null;
28    }
29}
```

FrontAccounting 2.4.12 - Login x +

demo.frontaccounting.eu/index.php

Chrome is being controlled by automated test software.

FrontAccounting 2.4.12 - Login

**FrontAccounting**  
Version 2.4.12 Build 13.07.2022 - Login

User name: demouser  
Password: .....  
Company: Training Co.  
Login as user: demouser and password: password

Login ->

09/20/2024 | 07:08 pm

FrontAccounting 2.4.12 - Theme: default

FrontAccounting

Console X Servers Results of running class pract12E

```
<terminated> pract12E [testNG] C:\Program Files\Java\jdk-22\bin\java.exe [23 Oct 2024, 1:17:22 pm – 1:17:32 pm] [pid: 11992]
[RemoteTestNG] detected TestNG version 7.4.0
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
ERROR StatusLogger Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger to log to the console...
Title of Webpage =====> FrontAccounting 2.4.12 - Login
PASSED: frontAccounting

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

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

**F) DemoAlert :-**

**1) Alertbox**

**CODE**

```
package atharva30;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

public class alert
{@Test
    public void f() throws Exception
{
    System.setProperty("C:\\\\Selenium\\\\chromedriver-
win64\\\\ChromeDriver\\\\chromedriver-win64","C:\\\\Selenium\\\\geckodriver-v0.34.0-
win64\\\\geckodriver.exe");
    WebDriver wd = new ChromeDriver();
    wd.get("https://vinothqaacademy.com/alert-and-popup/");

    wd.findElement(By.name("alertbox")).click();
    Thread.sleep(2000);
    Alert alt = wd.switchTo().alert();
    System.out.println("Text of alert" + alt.getText());
    alt.accept();
}

}
```

## Output:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a Java project named "atharva30" containing various test classes like Checkbox.java, NewTest3.java, demo.java, alert.java, etc.
- Code Editor:** Displays the content of Checkbox.java, which includes code for interacting with an alert box using Selenium WebDriver.
- Console:** Shows the output of the TestNG execution, indicating the detection of TestNG version 7.4.0 and the start of the test suite.
- Browser Preview:** A screenshot of a web browser window titled "Demo Site - Alert and Popup" showing a VNOH Q.A ACADEMY logo. An alert box is displayed with the message "vinothqaacademy.com says I am an alert box!". The browser status bar indicates "Chrome is being controlled by automated test software."
- Bottom Buttons:** Three buttons labeled "Alert Box", "Confirm Alert Box", and "Prompt Alert Box" are visible at the bottom of the browser preview area.

Demo Site - Alert and Popup - +

vinothaacademy.com/alert-and-popup/

Chrome is being controlled by automated test software.

+91-6383544892 (WhatsApp Only) | Email Id : vinothrwrs@gmail.com

Home Self Paced Video Course Tutorials Demo Sites About Me & Feedback Free YouTube Courses

VNOH Q.A ACADEMY "LEARN WITH CLARITY"

# Alert and PopUp

Alert Box

Confirm Alert Box

Prompt Alert Box

You clicked on OK!

```
Console X JRE Servers Results of running class pract12E
terminated> pract12F [Java Application] C:\Program Files\Java\jdk-22-bin\javaw.exe (24 Oct 2024, 12:41:28 pm – 12:41:53 pm) [pid: 5272]
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
ERROR StatusLogger Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger to log to the console...
Text of Alert I am an alert box!
```

## 2) Confirm Alertbox:-

### CODE:

```
package atharva30;

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

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

        System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-
win64\\\\geckodriver.exe");
        WebDriver wd=new FirefoxDriver();

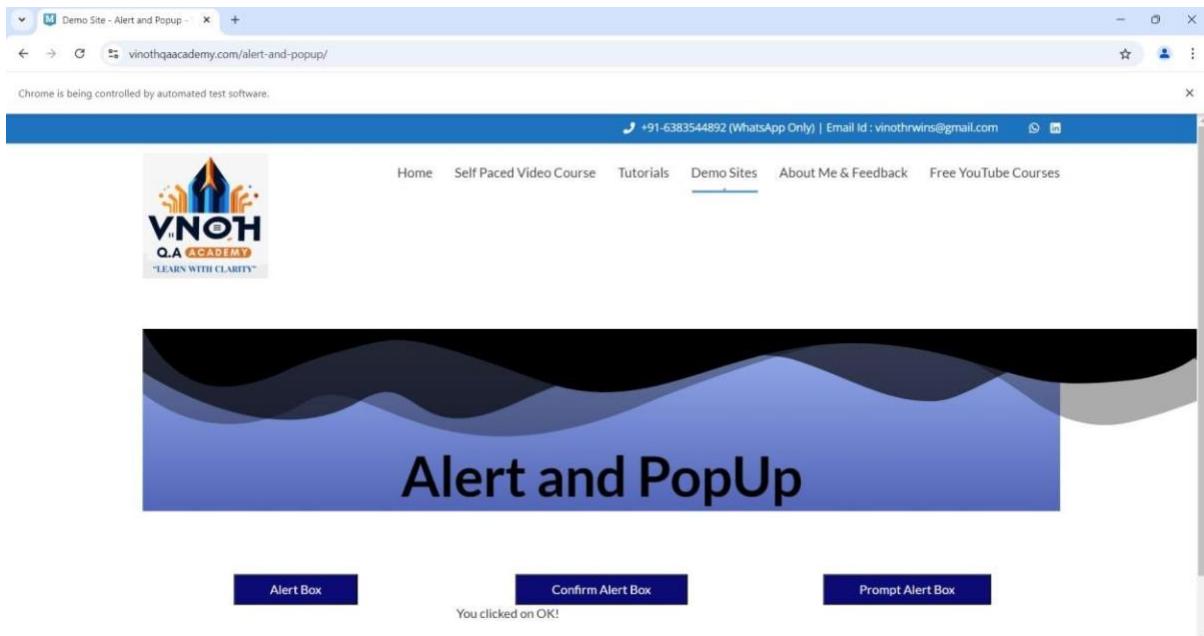
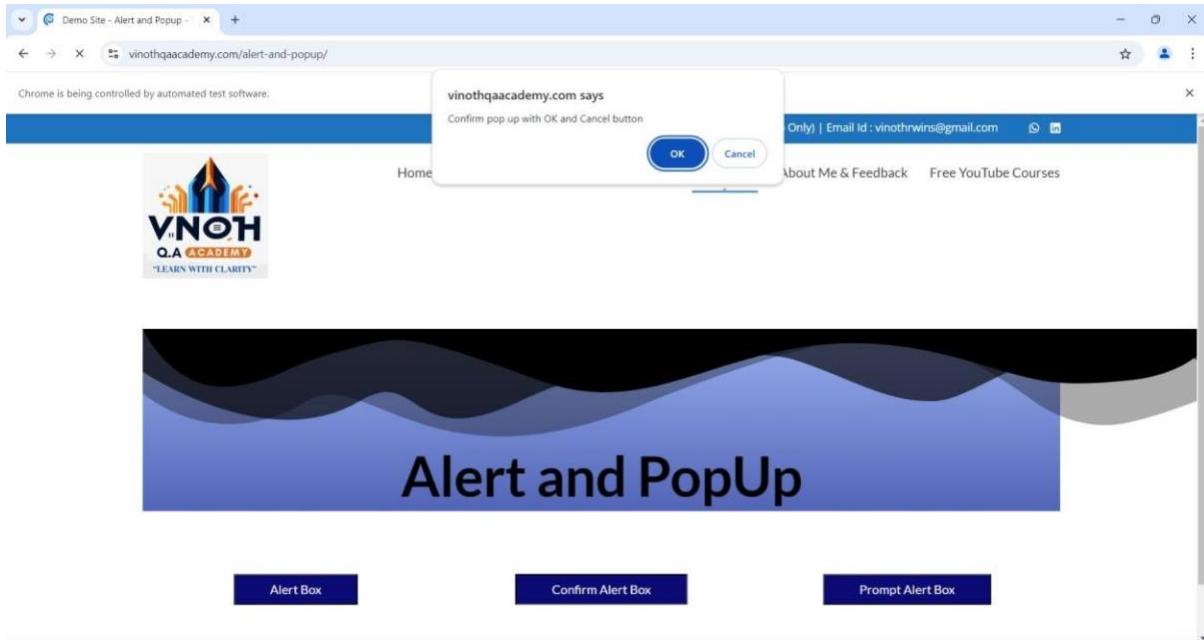
        wd.get("https://vinothqaacademy.com/alert-and-popup/");
        wd.findElement(By.name("confirmalertbox")).click();
        Thread.sleep(2000);

        Alert alt = wd.switchTo().alert();
        System.out.println("Text of Alert " + alt.getText());//capture text
        alt.accept();//click on OK button
    }
}
```

### Output:-

The screenshot shows the Eclipse IDE interface with the Java code for 'confomalert' in the center. The code is identical to the one provided above. To the right of the code, the 'Console' tab is active, showing the output of the program's execution. The output text is:  
Text of Alert **Confirm Alert**  
This indicates that the alert box was successfully displayed and accepted.

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer X
JRE System Library [JavaSE-1]
src
  atharva30
    actionclass.java
    add.java
    alert.java
    browser.java
    browsercommand.java
    Checkbox.java
    choosebrowser.java
    confomalert.java
    cssselector.java
    dataassert.java
    datacheck.java
    datachecking.java
    ddd.java
    demo.java
    Dropdown.java
    dropdownbylist.java
    dropdownbyvalue.java
    explicity.java
    findelement.java
    fire.java
    implicity.java
    listbox.java
    multiplehandling.java
Checkbox.java *NewTest3.java demo.java alert.java NewTest.java NewTest2.java WebDriver.java *confomal... 28
1 package atharva30;
2
3 import org.openqa.selenium.Alert;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.firefox.FirefoxDriver;
7
8 public class confomalert
9 {
10     public static void main(String[] args) throws Exception
11     {
12
13         System.setProperty("webdriver.chrome.driver","C:\\\\selenium web driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
14         WebDriver wd=new FirefoxDriver();
15
16         wd.get("https://vinothqaacademy.com/alert-and-popup/");
17         wd.findElement(By.name("confirmalertbox")).click();
18         Thread.sleep(2000);
19
20         Alert alt = wd.switchTo().alert();
21         System.out.println("Text of Alert " + alt.getText());//capture text
22         alt.accept();//click on OK button
23     }
24
25
Console X P3.java P4.java P5.java P6 XPATH.java prompt.java choosebrowser.java
```



### 3) Prompt Alertbox :-

#### CODE:

```
package atharva30;

import org.testng.annotations.Test;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.BeforeClass;

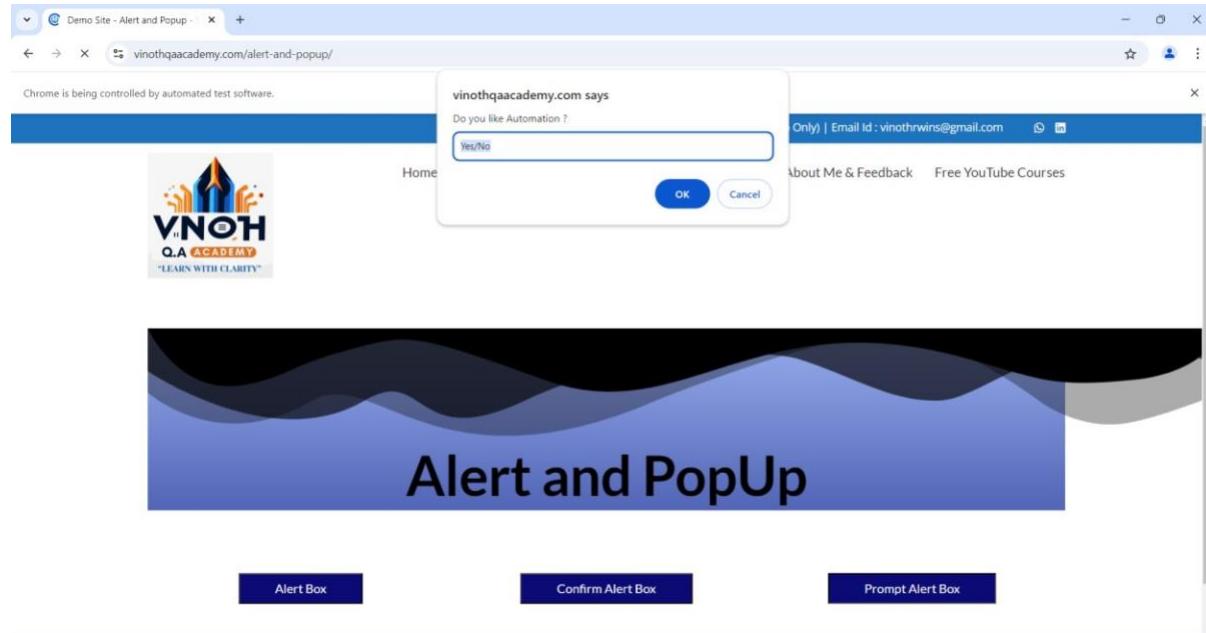
public class prompt {
    WebDriver wd;
    @BeforeClass
    public void beforeClass() {
        System.setProperty("C:\\\\Selenium\\\\chromedriver-
win64\\\\ChromeDriver\\\\chromedriver-win64","C:\\\\Selenium\\\\geckodriver-v0.34.0-
win64\\\\geckodriver.exe");
        wd = new ChromeDriver();
        wd.get("https://vinothqaacademy.com/alert-and-popup/");
    }
    @Test
    public void f() throws InterruptedException {
        wd.findElement(By.name("promptalertbox1234")).click();
        Thread.sleep(20000);

        String Text ="Yes";
        wd.switchTo().alert().sendKeys(Text);
        Alert alt =wd.switchTo().alert();
        System.out.println("Text of alert" + alt.getText());
        alt.accept();
    }
}
```

## Output:-

The screenshot shows the Eclipse IDE interface. The left pane displays the Package Explorer with a project named 'atharva30' containing numerous Java files. The right pane shows a code editor with a Java file named 'prompt.java'. The code implements a Selenium WebDriver test to interact with an alert box. The console tab at the bottom shows the execution results:

```
prompt [Testing] C:\Users\USER\p2\pool\plugins\org.eclipse.jdt.core\1.22.0.20240426-1149\jre\bin\javaw.exe (Oct 25, 2024, 4:43:42 PM - 4:43:44 PM)
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====
```



Demo Site - Alert and Popup - [vinothqaacademy.com/alert-and-popup/](http://vinothqaacademy.com/alert-and-popup/)

Chrome is being controlled by automated test software.

+91-6383544892 (WhatsApp Only) | Email Id : vinothrwrs@gmail.com

Home Self Paced Video Course Tutorials Demo Sites About Me & Feedback Free YouTube Courses

**VNOH**  
Q.A ACADEMY  
"LEARN WITH CLARITY"

# Alert and PopUp

[Alert Box](#) [Confirm Alert Box](#) [Prompt Alert Box](#)

Thanks for Liking Automation

```
[RemoteTestNG] detected TestNG version 7.4.0
Sept 20, 2024 11:13:25 PM org.openqa.selenium.devtools.CdpVersionFinder fin
WARNING: Unable to find an exact match for CDP version 128, returning the c
Text of Alert Do you like Automation ?
PASSED: testAlertInput

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

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

## G) DemoOHM

### CODE:

```
package testNG;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
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 pract12G
{
    WebDriver wd;

    @BeforeSuite

    public void openBrowser()
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        wd = new FirefoxDriver();
    }

    @BeforeClass
    public void loginOHM()
    {
        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");//locater name
        wd.findElement(By.className("oxd-button")).click();//locatorclassName
    }

    @Test(priority = 2)
    public void myInfo()
    {
        wd.findElement(By.className("oxd-main-menu-item")).click();
    }

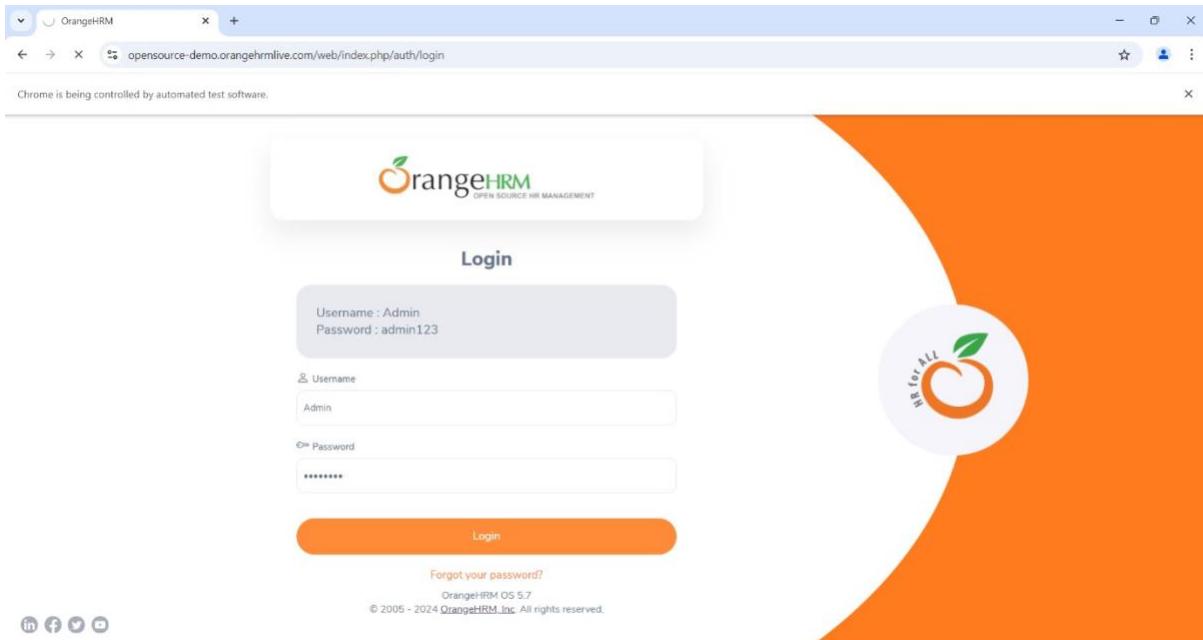
    @Test(priority = 1)
    public void pim()
    {
        wd.findElement(By.className("oxd-main-menu-item")).click();
    }
}
```

```
@AfterClass
public void logoutOHM() throws Exception
{
    wd.findElement(By.className("oxd-userdropdown-name")).click();//locator partiallinkText
    Thread.sleep(2000);
    wd.findElement(By.partialLinkText("Logout")).click();//locatorlinkText
}

@AfterSuite
public void closeBrowser()
{
    wd.quit();
}

}
```

## Output



OrangeHRM

opensource-demo.orangehrmlive.com/web/index.php/dashboard/index

Chrome is being controlled by automated test software.

Dashboard

Search

Admin

PIM

Leave

Time

Recruitment

My Info

Performance

Dashboard

Directory

Maintenance

Claim

Buzz

Upgrades

manda user

?

Time at Work

Punched Out: Mar 29th at 01:19 PM (GMT 7)

0h 0m Today

0h 0m

This Week Sep 16 - Sep 22

My Actions

(1) Pending Self Review

(1) Candidate to Interview

Quick Launch

Assign Leave

Leave List

Timeheets

Apply Leave

My Leave

My Timesheet

Buzz Latest Posts

manda akhil user 2020-08-10 09:09 AM

Hi AR,

Employees on Leave Today

manda user CAN - FMLA [09:00 - 17:00]

muser

Employee Distribution by Sub Unit

The screenshot shows the OrangeHRM dashboard. On the left is a sidebar with links for Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and several other modules. The main area has sections for 'Time at Work' (showing a punch-in message), 'My Actions' (listing pending self reviews and candidates for interview), 'Quick Launch' (with icons for leave assignment, leave list, timeheets, apply leave, my leave, and my timesheet), 'Buzz Latest Posts' (a social feed), 'Employees on Leave Today' (listing an employee named manda user), and 'Employee Distribution by Sub Unit' (a pie chart).

OrangeHRM

opensource-demo.orangehrmlive.com/web/index.php/admin/viewSystemUsers

Chrome is being controlled by automated test software.

Admin / User Management

User Management Job Organization Qualifications Nationalities Corporate Branding Configuration

About Support Change Password Logout

Admin

PIM

Leave

Time

Recruitment

My Info

Performance

Dashboard

Directory

Maintenance

Claim

Buzz

Upgrades

manda user

?

System Users

Username	User Role	Employee Name	Status	Actions
Admin	Admin	manda user	Enabled	
alicia.heller	ESS	Lissey Schaefer	Enabled	

+ Add

(8) Records Found

Username	User Role	Employee Name	Status	Actions
Admin	Admin	manda user	Enabled	
alicia.heller	ESS	Lissey Schaefer	Enabled	

The screenshot shows the 'Admin / User Management' section of the OrangeHRM application. It includes tabs for User Management, Job, Organization, Qualifications, Nationalities, Corporate Branding, and Configuration. On the left is a sidebar with Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and other modules. The main area displays a table of 'System Users' with columns for Username, User Role, Employee Name, and Status. It also includes a search bar and buttons for Reset and Search. Below the table, there's a section for adding new users and a summary of 8 records found.

```
[RemoteTestNG] detected TestNG version 7.4.0
Sept 20, 2024 10:51:56 PM org.openqa.selenium.devtools.CdpVersi
WARNING: Unable to find an exact match for CDP version 128, ret
PASSED: myInfo
PASSED: pim

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

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

## PRACTICAL 13

---

### Aim:- Demonstrate data driven Framework

- 1) File read data from Excel and perform testing using selenium.

**Code:-**

```
package Atharva3074;

import java.io.FileInputStream; import
java.io.IOException; import
org.apache.poi.xssf.usermodel.XSSFCell; import
org.apache.poi.xssf.usermodel.XSSFRow; import
org.apache.poi.xssf.usermodel.XSSFSheet;
import
org.apache.poi.xssf.usermodel.XSSFWorkbook;

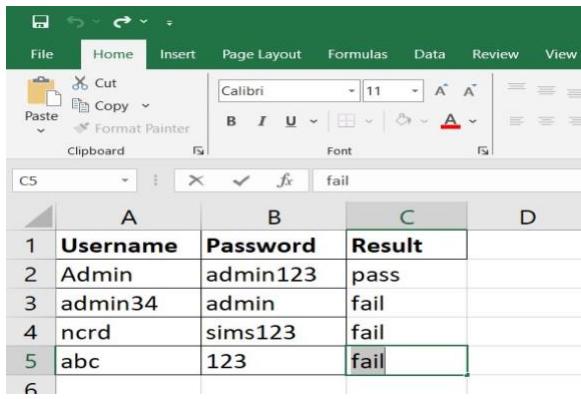
// Get Data From Excel
public class Eighth_1
{
    public static void main(String[] args) throws IOException
    {
        FileInputStream fis = new FileInputStream("D:\\STQA\\ExcelData.xlsx");

        XSSFWorbook wb = new XSSFWorbook(fis);
        XSSFSheet sh = wb.getSheet("Sheet1");

        for (int i = 1; i <= sh.getLastRowNum(); i++)
        {
            XSSFRow rw = sh.getRow(i);
            XSSFCell un = rw.getCell(0);
            XSSFCell pw = rw.getCell(1);
            XSSFCell rs = rw.getCell(2);

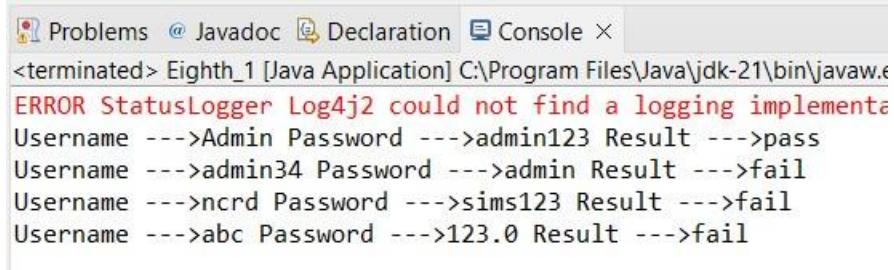
            System.out.println("Username --->" + un + " " + "Password --->" + pw +
" " +"Result --->" + rs );
        }
    }
}
```

}



	A	B	C	D
1	Username	Password	Result	
2	Admin	admin123	pass	
3	admin34	admin	fail	
4	ncrd	sims123	fail	
5	abc	123	fail	
6				

Output:-



```
Problems @ Javadoc Declaration Console ×
<terminated> Eighth_1 [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe
ERROR StatusLogger Log4j2 could not find a logging implementation
Username --->Admin Password --->admin123 Result --->pass
Username --->admin34 Password --->admin Result --->fail
Username --->ncrd Password --->sims123 Result --->fail
Username --->abc Password --->123.0 Result --->fail
```

**2) Write a Program to Read data from an Excel file and login in to site.**

Code:-

```
package Atharva3074;

import java.io.FileInputStream;
import
java.io.FileOutputStream;
import java.io.IOException;
import
java.util.concurrent.TimeUnit;

import org.apache.poi.xssf.usermodel.XSSFCell;
import org.apache.poi.xssf.usermodel.XSSFRow;
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 Eighth_3
{
    public static void main(String[] args) throws IOException
    {
        System.setProperty("webdriver.chrome.driver","D:\\STQA\\Selenium\\chromedriverwin6
4\\chromedriver.exe");
        WebDriver wd = new ChromeDriver();
        wd.get("https://opensource-demo.orangehrmlive.com/");

        FileInputStream fis = new FileInputStream("D:\\STQA\\ExcelData_1.xlsx");

        XSSFWorbook wb = new XSSFWorbook(fis);
        XSSFSheet sh = wb.getSheet("WriteData");

        for(int i=1;i<=sh.getLastRowNum();i++)
        {
            XSSFRow rw = sh.getRow(i);
            XSSFCel un = rw.getCell(0);
            XSSFCel pw = rw.getCell(1);
            XSSFCel res = rw.createCell(2);
```

```
System.out.println("username --> "+ un + " " + "Password --> "+ pw);

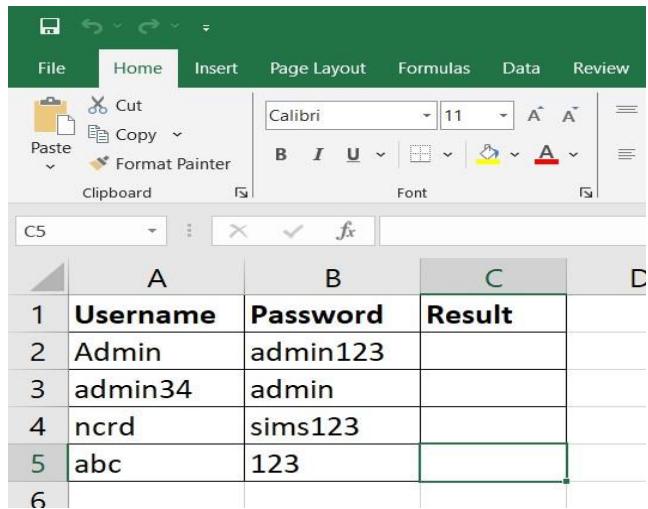
wd.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
wd.findElement(By.name("username")).sendKeys(un.toString());
wd.findElement(By.name("password")).sendKeys(pw.toString());
wd.findElement(By.className("oxd-button")).click();

try
{
    wd.findElement(By.className("oxd-userdropdownname")).click();
    wd.findElement(By.partialLinkText("Logout")).click();

    System.out.println("Login Pass");
    res.setCellValue("Valid");
}
catch
(Exception e)
{
    System.out.println("Login Fail");
    res.setCellValue("Invalid");
}

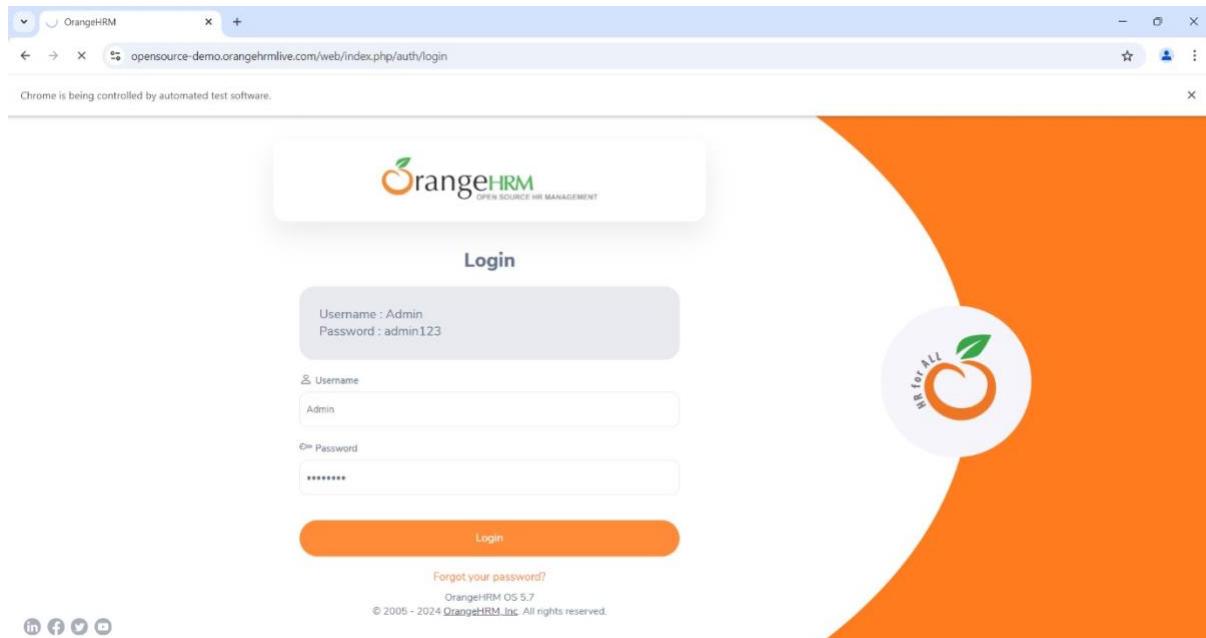
FileOutputStream fos = new
FileOutputStream("D:\\STQA\\ExcelData_1.xlsx");
wb.write(fos);

fis.close();
}
```



	A	B	C	D
1	Username	Password	Result	
2	Admin	admin123		
3	admin34	admin		
4	ncrd	sims123		
5	abc	123		
6				

Output:-



The screenshot shows the OrangeHRM dashboard. At the top right, there is a user profile for 'Nimna Shehani' with options for Upgrade, About, Support, Change Password, and Logout. On the left, a sidebar menu includes Search, Admin, PIM, Leave, Time, Recruitment, My Info, Performance, Dashboard (which is selected), Directory, Maintenance, Claim, and Buzz. The main content area features several cards: 'Time at Work' showing a punch-out at 11:24 PM on Aug 11; 'My Actions' with pending self-review and candidate-to-interview tasks; 'Quick Launch' with links for Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, and My Timesheet; 'Buzz Latest Posts' from Nimna Shehani; 'Employees on Leave Today'; and 'Employee Distribution by Sub Unit'. A message at the bottom states 'Chrome is being controlled by automated test software.'

The screenshot shows the OrangeHRM login page. It features a large orange background graphic with the text 'HR for ALL' and an orange fruit logo. The login form has fields for 'Username' (containing 'nord') and 'Password' (containing '\*\*\*\*\*'). Below the form is a 'Login' button. A red error message 'Invalid credentials' is displayed above the form. A note at the bottom says 'Forgot your password?' and 'Copyright © 2014-2024 OrangeHRM Inc. All Rights Reserved.' The URL in the address bar is 'opensource-demo.orangehrmlive.com/web/index.php/auth/login'. A message at the bottom states 'Chrome is being controlled by automated test software.'

```
Problems @ Javadoc Declaration Console X
<terminated> Eighth_3 [Java Application] C:\Program Files\Java\j
Sept 19, 2024 8:25:51 PM org.openqa.selenium.dev
WARNING: Unable to find an exact match for CDP v
ERROR StatusLogger Log4j2 could not find a loggi
username --> Admin Password --> admin123
Login Pass
username --> admin34 Password --> admin
Login Fail
username --> ncrd Password --> sims123
Login Fail
username --> abc Password --> 123.0
Login Fail
```

## PRACTICAL 14

---

### AIM:-Validation Testing using chrome and bing

#### A) Chrome

#### CODE:

```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class validation
{
    public static void main(String[] args) throws Exception
    {System.setProperty("webdriver.chrome.driver", "C:\\\\ Selenium
webdriver\\\\geckodriver\\\\geckodriver.exe");
    WebDriver wd=new FirefoxDriver();
    wd.get("https://google.com/");
    //wd.get("https://bing.com/");
    Thread.sleep(2000);

    wd.findElement(By.className("gLFyf")).sendKeys("Fruits");
    wd.findElement(By.className("gLFyf")).sendKeys(Keys.ENTER);}}
```

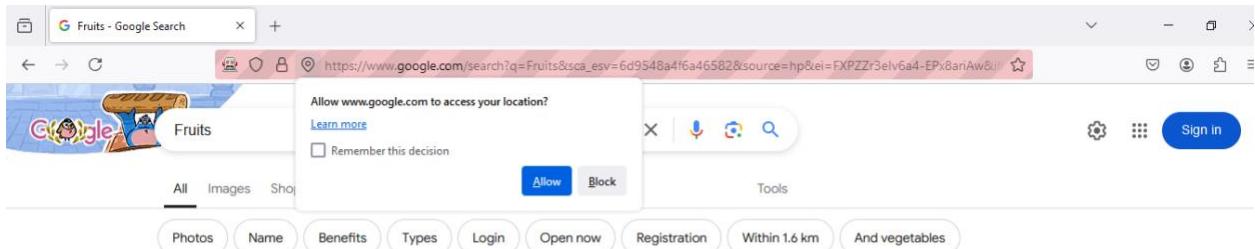
## **OUTPUT:**

The screenshot shows the Eclipse IDE interface with the following details:

- File menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Includes icons for file operations like Open, Save, Print, and search.
- Package Explorer:** Shows a tree view of Java files in the project, including:
  - choosebrowser.java
  - confomalet.java
  - cssselector.java
  - datasert.java
  - datacheck.java
  - datachecking.java
  - ddt.java
  - demo.java
  - Dropdown.java
  - dropdownbylist.java
  - dropdownbyvalue.java
  - explicity.java
  - findelement.java
  - fire.java
  - implicitly.java
  - listbox.java
  - multiplehandling.java
  - NewTest.java
  - P1.java
  - P2.java
  - P3.java
  - P4.java
  - P5.java
  - P6\_XPATH.java
  - pathselector.java
  - prompt.java
  - Radio.java
  - TestSelenium.java
  - typeofalert.java
  - validation.java
- Code Editor:** Displays the `validation.java` file content:

```
1 package atharava30;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.Keys;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6 public class validation
7 {
8 public static void main(String[] args) throws Exception
9 {
10     System.setProperty("webdriver.chrome.driver", "C:\\ Selenium webdriver\\geckodriver\\geckodriver.exe");
11     WebDriver wd=new FirefoxDriver();
12     wd.get("https://google.com/");
13     //wd.get("https://bing.com/");
14     Thread.sleep(2000);
15     wd.findElement(By.className("gLFyf")).sendKeys("Fruits");
16     wd.findElement(By.className("gLFyf")).sendKeys(Keys.ENTER);}}
```
- Console:** Shows the output of the test run:

```
<terminated> prompt [Testing] C:\Users\USER\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v20240426-1149\jre\bin\javaw.exe (Oct 25, 2024, 4:43:18 PM - 4d)
Total tests run: 1, Passes: 0, Failures: 1, Skips: 0
=====
```



Images ...



**Activate Windows**  
Go to Settings to activate Windows.

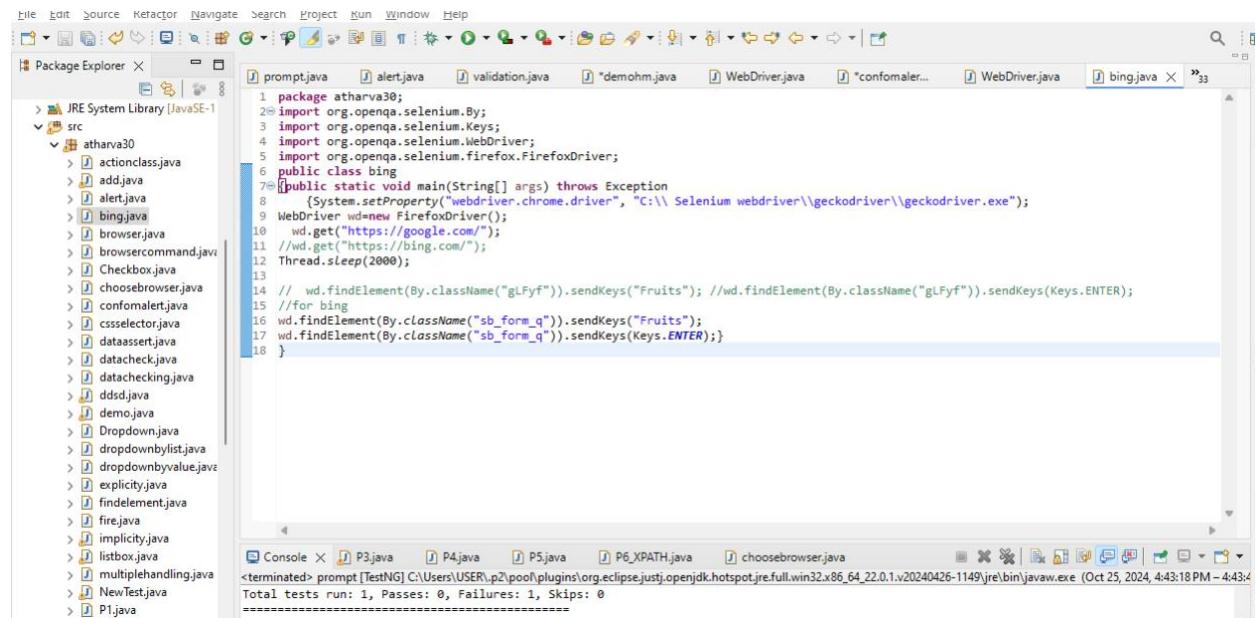
## B)Bing

### CODE:

```
package atharva30;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class bing
{public static void main(String[] args) throws Exception
    {System.setProperty("webdriver.chrome.driver", "C:\\ Selenium
webdriver\\geckodriver\\geckodriver.exe");
WebDriver wd=new FirefoxDriver();
    wd.get("https://google.com/");
//wd.get("https://bing.com/");
Thread.sleep(2000);

// wd.findElement(By.className("gLFyf")).sendKeys("Fruits");
//wd.findElement(By.className("gLFyf")).sendKeys(Keys.ENTER);
//for bing
wd.findElement(By.className("sb_form_q")).sendKeys("Fruits");
wd.findElement(By.className("sb_form_q")).sendKeys(Keys.ENTER);}
}
```

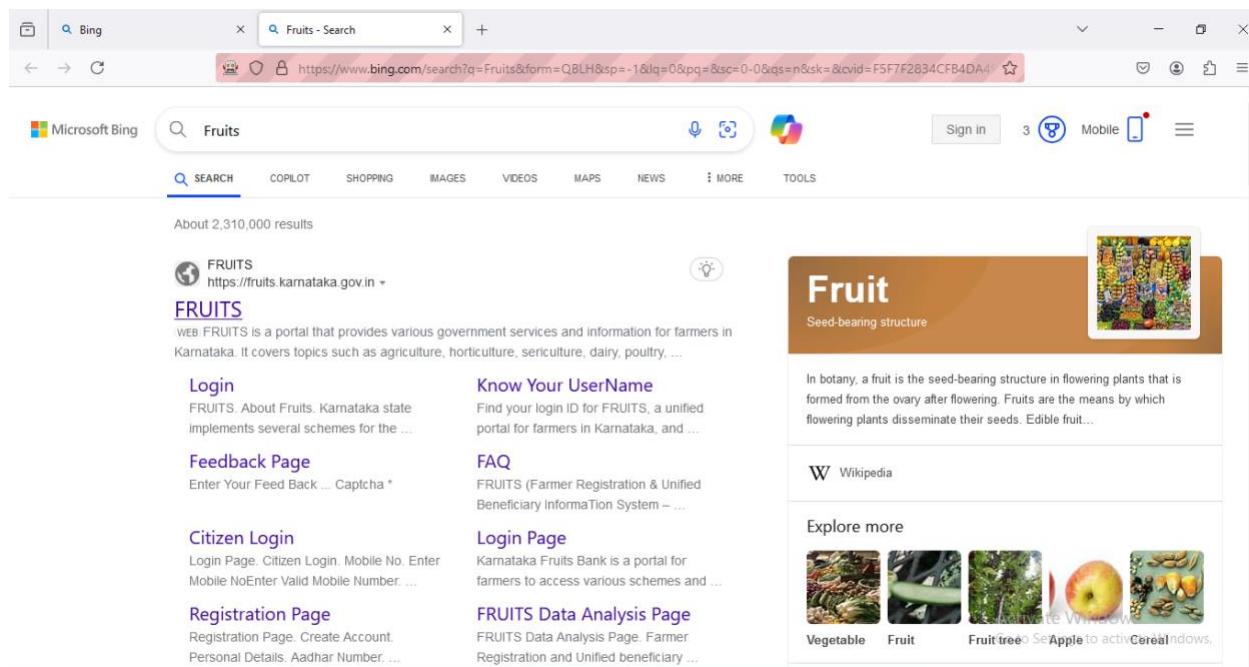
## Output:



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The left sidebar displays the Package Explorer with a JRE System Library (JavaSE-1) and a src folder containing numerous Java files such as actionclass.java, add.java, alert.java, bing.java, browser.java, browsercommand.java, Checkbox.java, choosebrowser.java, conformal.java, csselement.java, dataassert.java, datacheck.java, datachecking.java, ddd.java, demo.java, Dropdown.java, dropdownbylist.java, dropdownbyvalue.java, explicity.java, findelement.java, fire.java, implicity.java, listbox.java, multiplehandling.java, NewTest.java, and P1.java. The main editor window shows a Java file named promptjava with the following code:

```
1 package atharva30;
2 import org.openqa.selenium.By;
3 import org.openqa.selenium.Keys;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.firefox.FirefoxDriver;
6 public class bing {
7     public static void main(String[] args) throws Exception {
8         System.setProperty("webdriver.chrome.driver", "C:\\\\ Selenium webdriver\\\\geckodriver\\\\geckodriver.exe");
9         WebDriver wd=new FirefoxDriver();
10        wd.get("https://google.com/");
11        wd.get("https://bing.com/");
12        Thread.sleep(2000);
13
14        // wd.findElement(By.className("gLFyf")).sendKeys("Fruits"); //wd.findElement(By.className("gLFyf")).sendKeys(Keys.ENTER);
15        //for bing
16        wd.findElement(By.className("sb_form_q")).sendKeys("Fruits");
17        wd.findElement(By.className("sb_form_q")).sendKeys(Keys.ENTER);
18    }
}
```

The bottom status bar indicates the terminal output: <terminated> prompt [TestNG] C:\Users\USER\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86\_64\_22.0.1.v20240426-1149\jre\bin\javaw.exe (Oct 25, 2024, 4:43:18 PM - 4:43:42 Total tests run: 1, Passes: 0, Failures: 1, Skips: 0)



The screenshot shows a Microsoft Bing search results page. The search query is "Fruits". The results include:

- FRUITS** <https://fruits.karnataka.gov.in/> - A portal for farmers in Karnataka.
- Login** [FRUITS. About Fruits. Karnataka state implements several schemes for the ...](#)
- Feedback Page** [Enter Your Feed Back ... Captcha \\*](#)
- Citizen Login** [Login Page. Citizen Login. Mobile No. Enter Mobile NoEnter Valid Mobile Number. ....](#)
- Registration Page** [Registration Page. Create Account. Personal Details. Aadhar Number. ....](#)
- Know Your UserName** [Find your login ID for FRUITS, a unified portal for farmers in Karnataka, and ...](#)
- FAQ** [FRUITS \(Farmer Registration & Unified Beneficiary Information System – ...](#)
- Login Page** [Karnataka Fruits Bank is a portal for farmers to access various schemes and ...](#)
- FRUITS Data Analysis Page** [FRUITS Data Analysis Page. Farmer Registration and Unified beneficiary ...](#)

On the right side, there is a sidebar titled "Fruit" with a sub-section "Seed-bearing structure". It includes a snippet of text about fruits being seed-bearing structures and a small image of various fruits. Below this, there is a "Explore more" section with links to "Vegetable", "Fruit", "Private Windows", "Fruit tree", "Apple", "Seeds", "Cereal", and "Windows".

## PRACTICAL 15

---

### **Aim:- Perform regression testing**

#### **Theory:-**

##### **What is Regression Testing?**

REGRESSION TESTING is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features. Regression Testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine. This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done.

##### **Need of Regression Testing**

The Need of Regression Testing mainly arises whenever there is requirement to change the code and we need to test whether the modified code affects the other part of software application or not. Moreover, regression testing is needed, when a new feature is added to the software application and for defect fixing as well as performance issue fixing.

##### **How to do Regression Testing**

In order to do Regression Testing process, we need to first debug the code to identify the bugs. Once the bugs are identified, required changes are made to fix it, then the regression testing is done by selecting relevant test cases from the test suite that covers both modified and affected parts of the code.

Software maintenance is an activity which includes enhancements, error corrections, optimization and deletion of existing features. These modifications may cause the system to work incorrectly. Therefore, Regression Testing becomes necessary.

##### **Regression Test Selection**

Regression Test Selection is a technique in which some selected test cases from test suite are executed to test whether the modified code affects the software application or not. Test cases are categorized into two parts, reusable test cases which can be used in further regression cycles and obsolete test cases which can not be used in succeeding cycles