

**PRACTICAL NO: 01**

**Aim:** To verify Career mate's functionality using structured manual testing techniques.

Sr. No	Test ID	Action	Description	Input	Expected Output	Actual Output	Test Result	Test Comment
1	register_01	User Registration	Test new user sign-up flow	Valid username, email, password	Account should be created and should redirect to dashboard/login	Success message is shown	Pass	Working as expected
2	register_02	Duplicate Registration	Check for duplicate email during registration	Already registered email	System should show "Email already exists" error	Error message is shown	Pass	Validation enforced
3	login_01	Login with valid credentials	Authenticate with correct credentials	Correct email and password	System should redirect to dashboard	User is redirected successfully	Pass	Login successful
4	login_02	Login with invalid credentials	Block incorrect login attempts	Wrong password	System should show "Incorrect credentials" error	Error message is shown	Pass	Login validation working

**SOFTWARE TESTING AND QUALITY ASSURANCE**

Dattaram Santosh Kolte

SYMCA-B (97)

5	login_03	Forgot Password	Test password recovery process	Registered email address	System should display reset instructions or should send reset email	Message is shown	Pass	Flow implemented correctly
6	login_04	Logout	Verify logout clears session	Click logout button	System should redirect to login page	Logout is successful	Pass	Session cleared
7	job_01	View Jobs	Display job listings after login	Click "Jobs" tab	System should show available job listings	Listings are visible	Pass	Display correct
8	job_02	Search Jobs	Verify job search/filter works	Search keyword (e.g., React Developer)	Filtered job list should be shown	Filter results are accurate	Pass	Search functional
9	job_03	Apply to Job	Apply to selected job	Click "Apply"	System should show confirmation message or should move to "Applied" section	Confirmation is shown	Pass	Apply logic verified
10	resume_01	Upload Resume with valid file	Upload a valid resume file	Upload PDF file	System should show "Resume uploaded"	Upload is successful	Pass	Functional

**SOFTWARE TESTING AND QUALITY ASSURANCE**

Dattaram Santosh Kolte

SYMCA-B (97)

					successfully"			
11	resume_02	Upload Resume with invalid file	Prevent upload of unsupported file types	Upload .exe, .js file	System should show "Invalid file type" error	Error message is shown	Pass	Security check passed
12	profile_01	Update Profile	Edit user details and save	Change name, skills, contact info	Updated profile should be displayed	Changes are saved	Pass	Update working
13	security_01	Unauthorized Dashboard Access	Prevent access without login	Visit dashboard URL without login	System should redirect to login page	Redirect is successful	Pass	Access control functional
14	uiux_01	Mobile View	Check layout responsiveness	Open in mobile browser/emulator	Layout should adjust without breaking	Responsive layout is visible	Pass	Mobile-friendly verified
15	admin_01	Admin Login	Admin authentication with correct credentials	Valid admin login	System should redirect to admin dashboard	Admin is logged in	Pass	Admin role verified
16	admin_02	Admin Adds Job	Test job posting via admin panel	Fill job form and submit	Job should appear in job listings	Job is listed	Pass	Create functionality works

**SOFTWARE TESTING AND QUALITY ASSURANCE**

Dattaram Santosh Kolte

SYMCA-B (97)

17	admin_03	Admin Deletes Job	Remove a job posting from system	Click delete on a job	Job should be deleted from DB and UI	Job is removed	Pass	Delete logic confirmed
----	----------	-------------------	----------------------------------	-----------------------	--------------------------------------	----------------	------	------------------------

## PRACTICAL NO: 02

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

### A) Chrome

package com.gh;

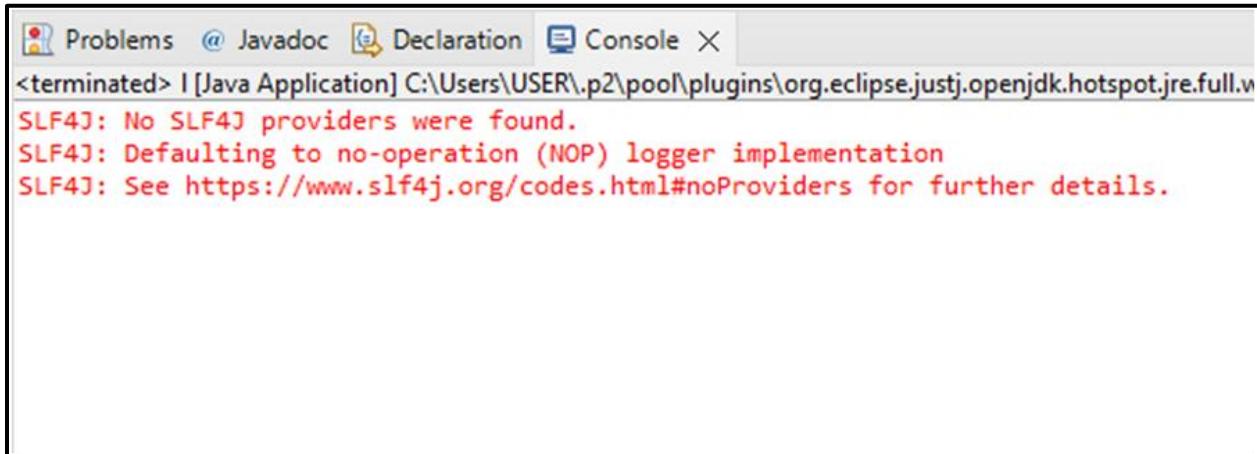
```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class l {

    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://google.com");

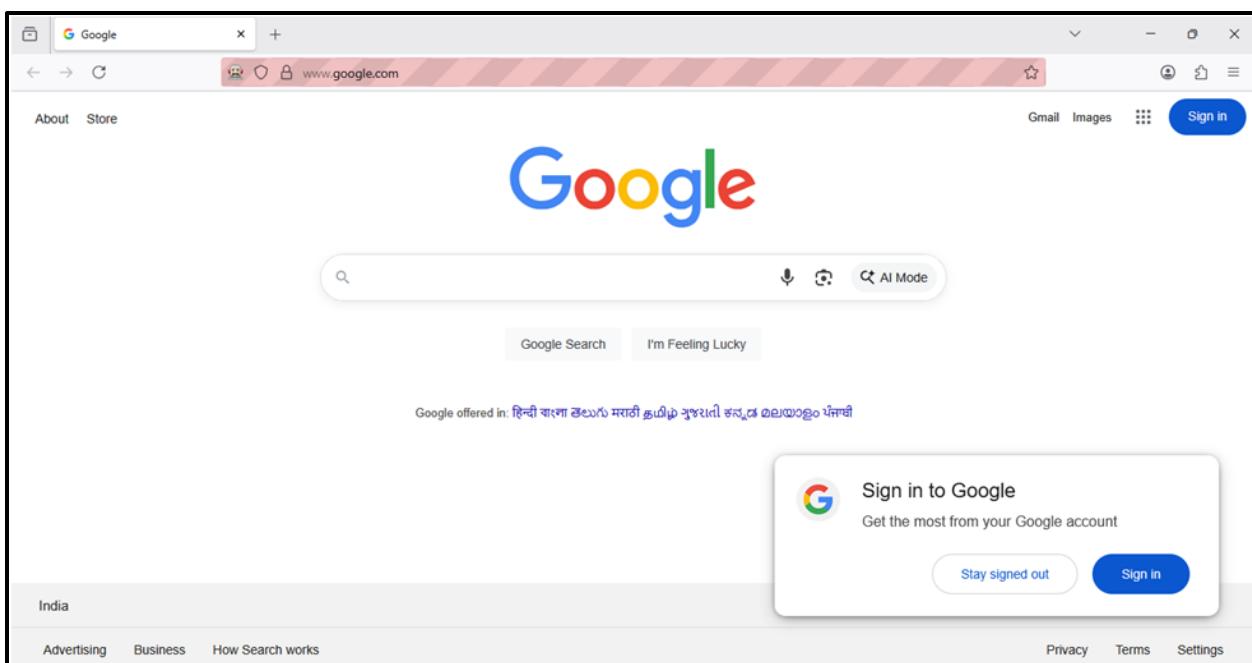
    }
}
```

### OUTPUT:



The screenshot shows the Eclipse IDE's Console view. The tabs at the top are Problems, Javadoc, Declaration, and Console. The console output is as follows:

```
<terminated> I [Java Application] C:\Users\USER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.v
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.
```

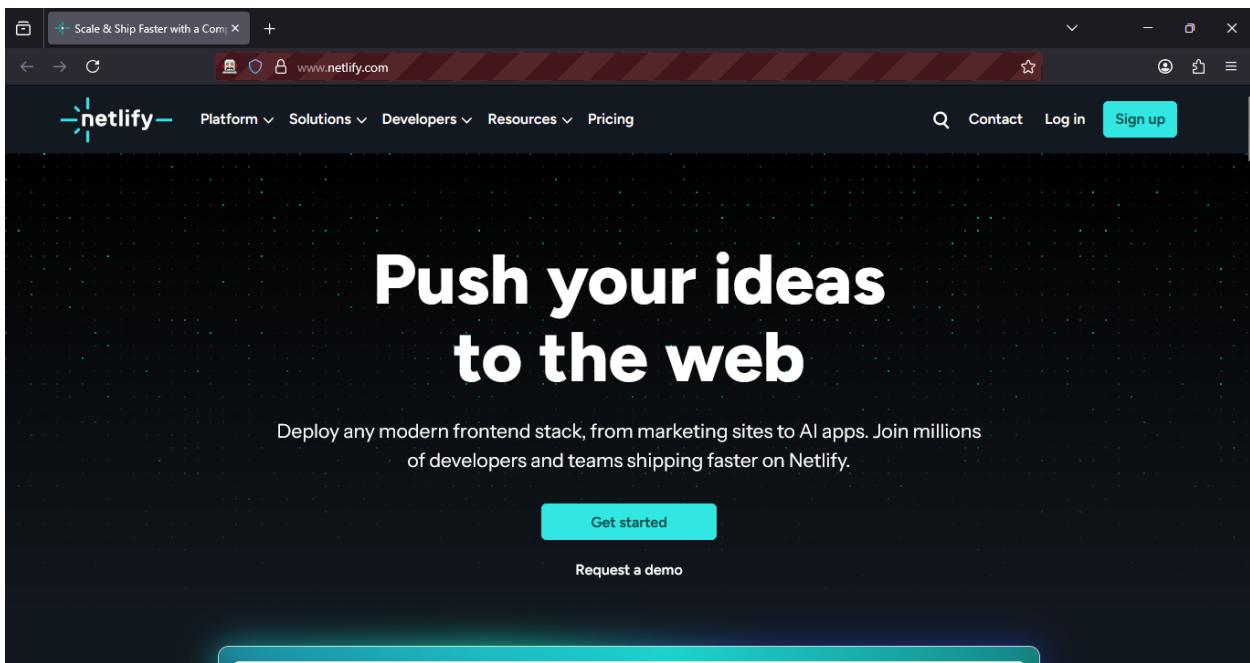


## B) FireFox

```
package com.prac2;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Practical2 {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\\\selenium
webdriver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("http://netlify.com");
    }
}
```

**OUTPUT:**



**C) Any Browser**

package prati;

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

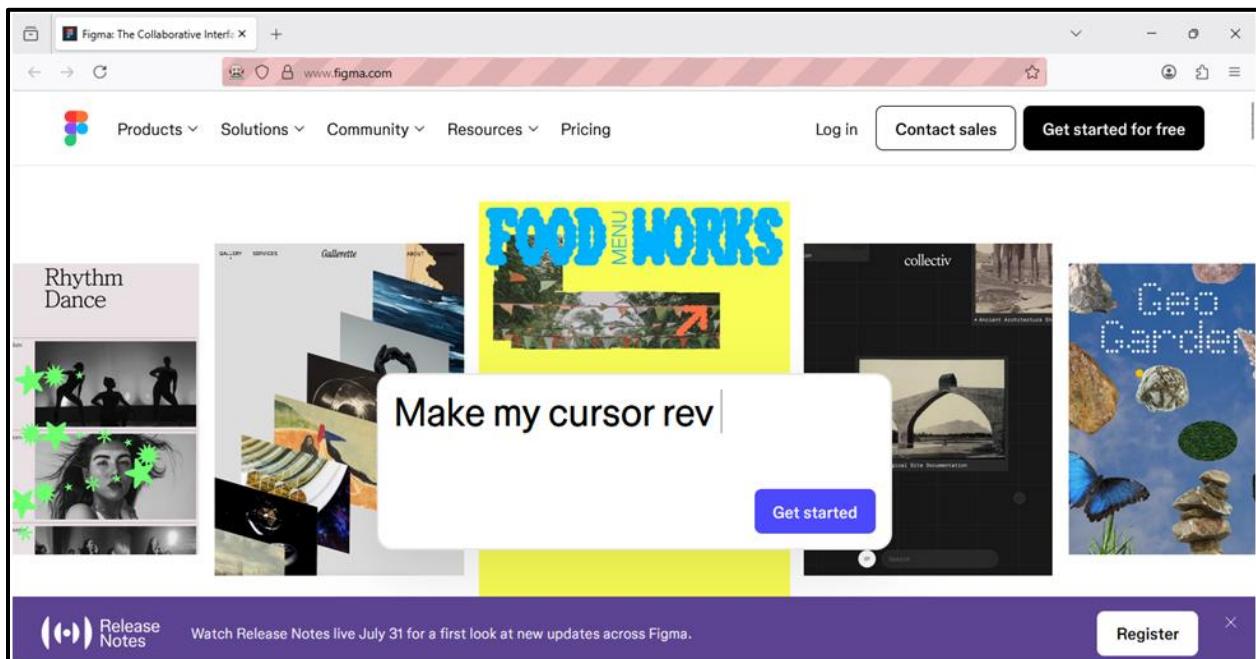
public class Browser {
    static WebDriver wd;
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Welcome");
        System.out.println("1.Chrome\n2.Foxfire\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. The top menu bar includes 'Problems', '@ Javadoc', 'Declaration', and 'Console'. The 'Console' tab is selected, displaying the following text:  
Browser [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_2  
Welcome  
1.Chrome  
2.Firefox  
  
Choice  
2



## PRACTICAL NO: 03

**Aim:** Demonstrate wait command in selenium.

```
package JavaPracticals;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.support.ui.ExpectedConditions;
import java.time.Duration;

public class OrderPlace {
    public static void main(String[] args) {
        System.setProperty("webdriver.gecko.driver",
            "C:\\\\selenium web driver\\\\ChromeDriver\\\\chromedriver-
win64\\\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://demo.frontaccounting.eu/");
        WebDriverWait wait = new WebDriverWait(wd, Duration.ofSeconds(10));
        wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.name("user_name_entry_field"))).sendKeys("your_username");
        wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.name("password"))).sendKeys("your_password");
        Select companySelect = new
        Select(wd.findElement(By.name("company_login_name")));
        wd.findElement(By.name("SubmitUser")).click();
        wait.until(ExpectedConditions.elementToBeClickable(By.xpath(
            "/html/body/table[1]/tbody/tr/td/table[1]/tbody/tr/td/div[2]/table/tbody/tr[1]/t
d/table/tbody/tr[2]/td[1]/a[1]"
        ))).click();
        Select customerSelect = new Select(wd.findElement(By.name("customer_id")));
        customerSelect.selectByVisibleText("Donald Easter");
        wd.findElement(By.name("AddItem")).click();
        wd.findElement(By.name("ProcessOrder")).click();
        wd.quit();
    }
}
```

**OUTPUT:**

The screenshot shows the FrontAccounting 2.4.12 Login page. The URL is https://demo.frontaccounting.eu. The login form has the following fields:

- User name: DemoUser
- Password: \*\*\*\*\*
- Company: Training Co.

A red error message at the bottom of the form states: "invalid password or username. Please, try again." Below the form is a "Login -->" button. At the bottom of the page, there is a timestamp: 09/21/2024 | 07:46 am, and a footer with the text: "FrontAccounting 2.4.12 - Theme: default" and "FrontAccounting".

The screenshot shows the "New Sales Quotation Entry" page. The URL is https://demo.frontaccounting.eu/sales/sales\_order\_entry.php?NewQuotation=Yes. The top navigation bar includes links for Sales, Purchases, Items and Inventory, Manufacturing, Fixed Assets, Dimensions, Banking and General Ledger, and Setup. The main form contains the following data:

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

The "Sales Quotation Items" section shows one item entry:

Item Code	Item Description	Quantity	Unit	Price after Tax	Discount %	Total	Add Item
	Samsung	1	each	0.00	0.0	0.00	<input type="button" value="Add Item"/>

Below the items, there is a "Cash payment" section with fields for Deliver from Location (Default), Cash account (Petty Cash account), and Comments. At the bottom of the form are "Place Quotation" and "Cancel Quotation" buttons.

The screenshot shows the "New Sales Quotation Entry" page after a successful operation. The URL is https://demo.frontaccounting.eu/sales/sales\_order\_entry.php?AddedQU=1241. A green success message at the top of the page reads: "Quotation # 1241 has been entered." Below this message are several buttons and links:

- New This Quotation
- Print This Quotation
- Email This Quotation
- Make Sales Order Against This Quotation
- Enter a New Quotation

At the bottom of the page, there is a "Back" link and a footer with the text: "FrontAccounting 2.4.12 - Theme: default" and "FrontAccounting".

## **PRACTICAL NO: 04**

**Aim:** Implement Browser command and navigation commands

```
package com.prac4;

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

public class Prac4 {

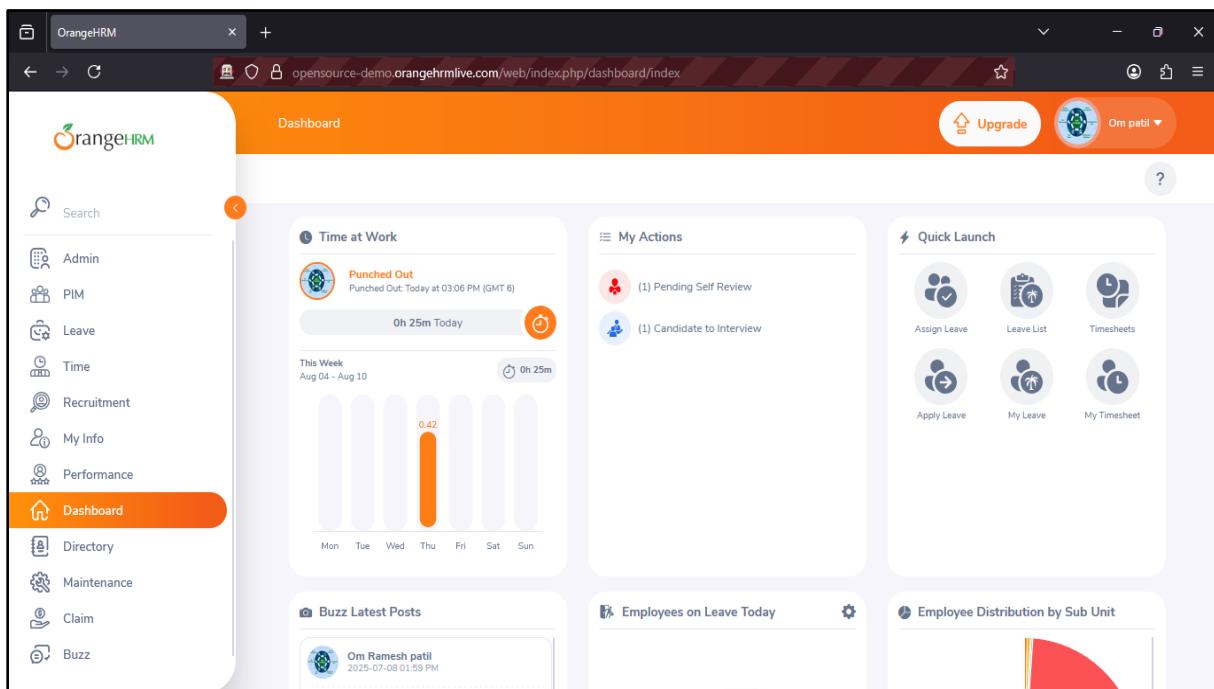
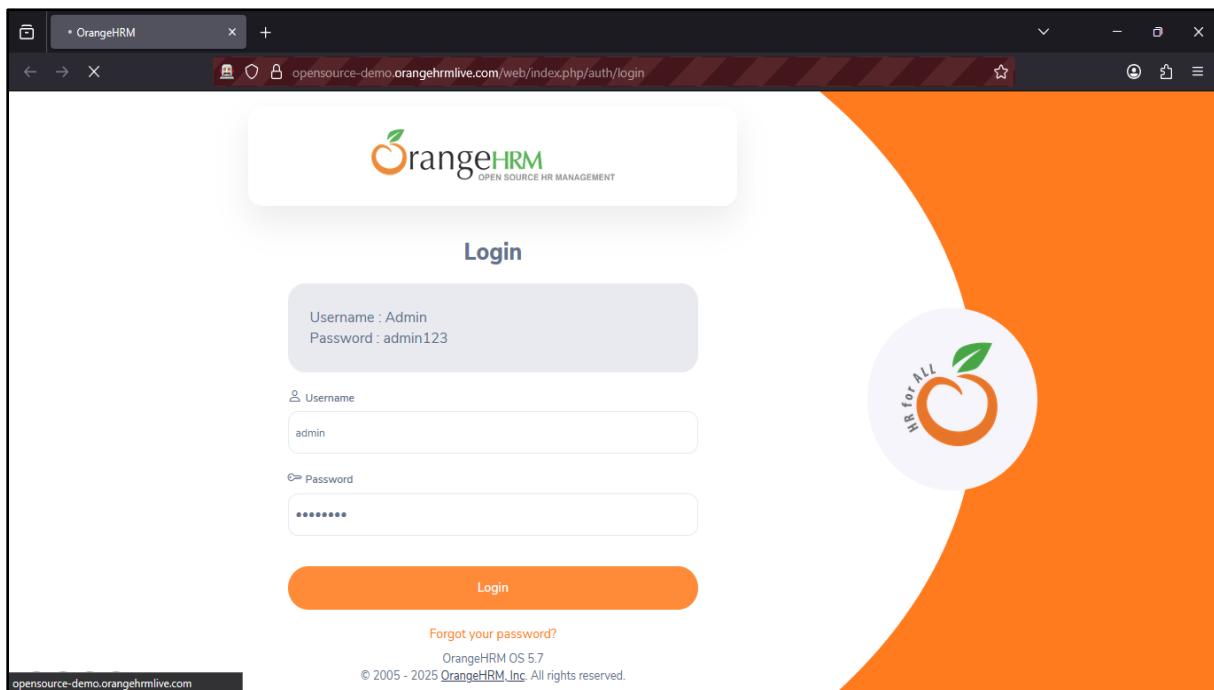
    public static void main(String[] args) throws Exception
    {
        System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://opensource-demo.orangehrmlive.com/");
        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[1]/div/div[2]/form/div[3]/b
utton")).click();
        Thread.sleep(2000);
        wd.navigate().back();
        Thread.sleep(2000);
        wd.navigate().forward();
        wd.navigate().to("https://opensource-demo.orangehrmlive.com/");
        wd.navigate().refresh();
    }
}
```

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

### **OUTPUT:**



## **PRACTICAL NO: 05**

**Aim:** Implement the find element command.

```
package com.prac5;

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

public class Practical5 {

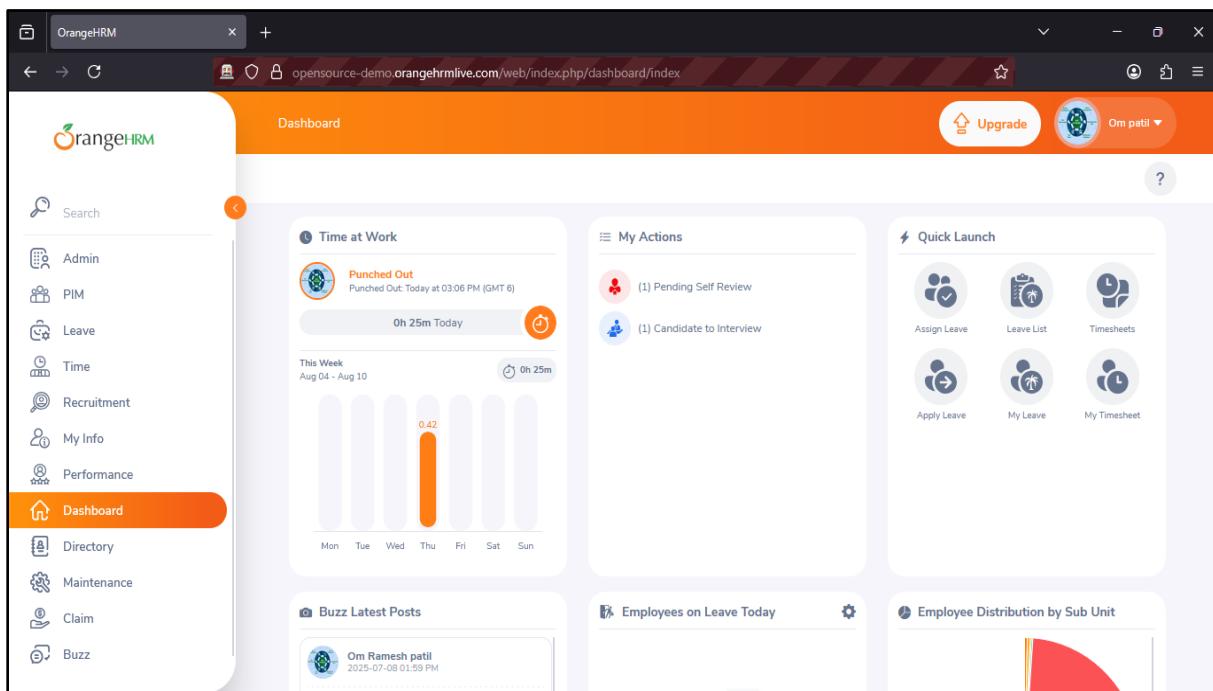
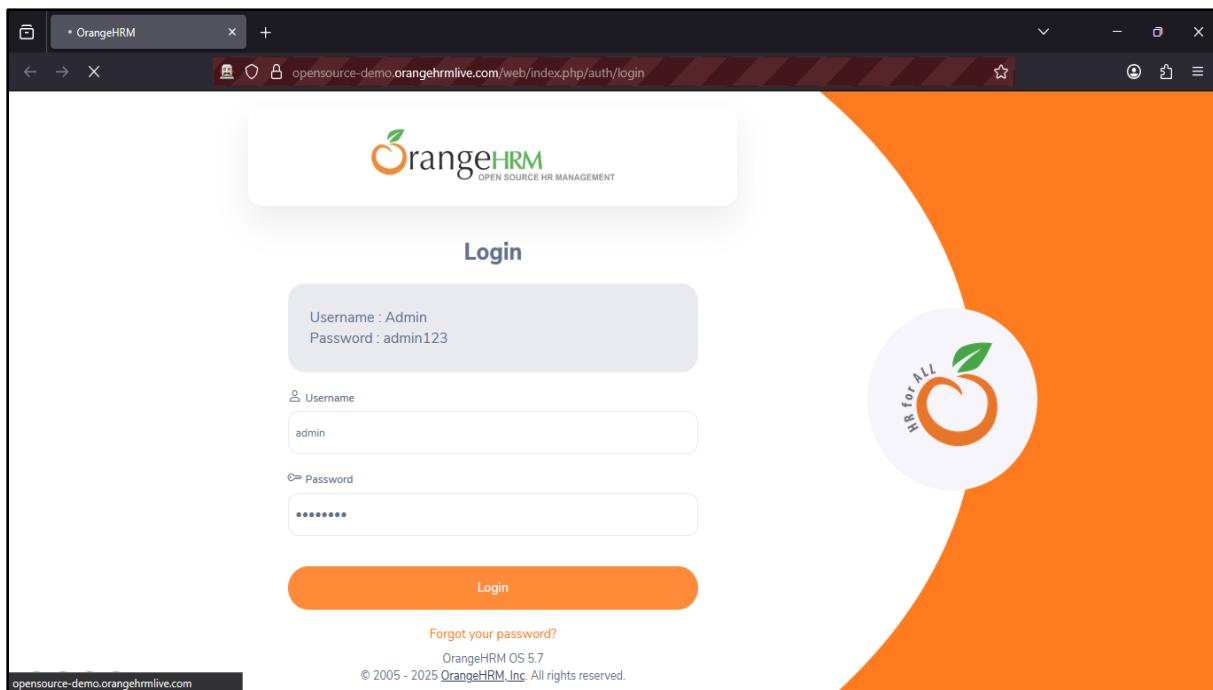
    public static void main(String[] args) throws Exception
    {
        System.setProperty("webdriver.chrome.driver","C:\\\\selenium web
driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://opensource-demo.orangehrmlive.com/");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

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

### **OUTPUT:**



## **PRACTICAL NO: 06**

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

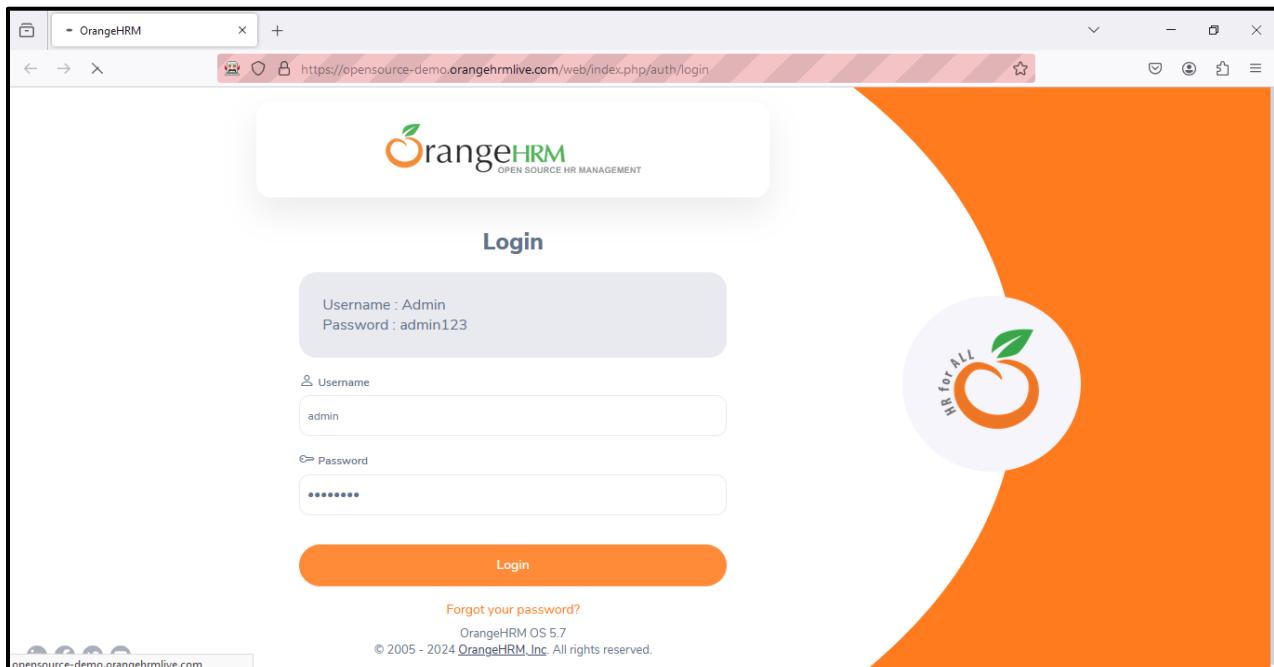
### **1) cssSelector**

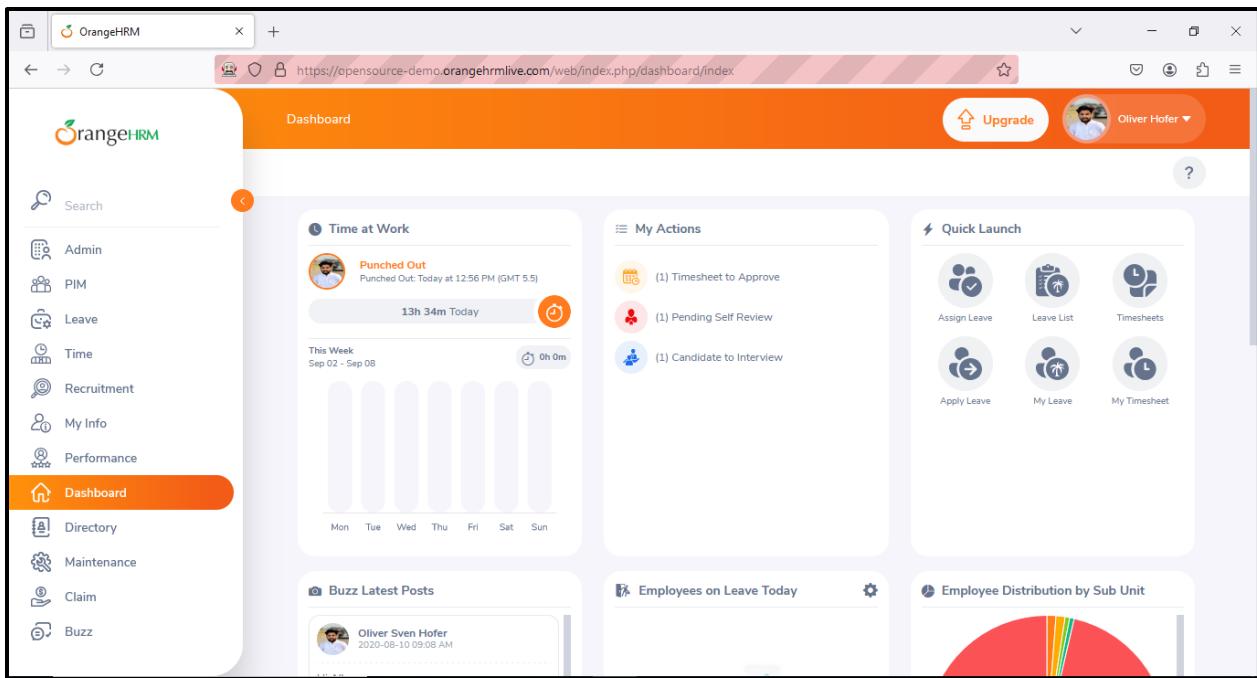
```
package JavaPracticals;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class CssSelector {
    public static void main(String[] args) throws Exception {
        System.setProperty("webdriver.gecko.driver", "C:\\selenium 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:**





## 2) XPath

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

public class Practical4 {
    public static void main(String[] args) throws Exception {
        System.setProperty(
            "webdriver.gecko.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]/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");

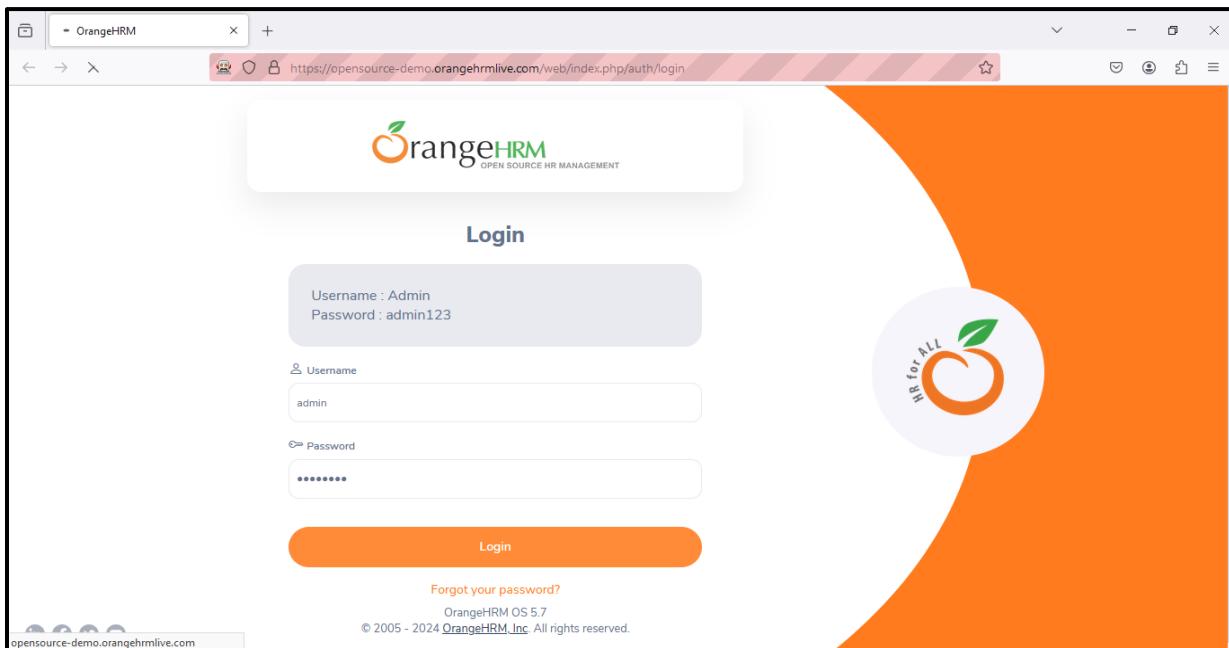
        wd.findElement(By.xpath("//html/body/div/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();
    }
}
```

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

### **OUTPUT:**



A screenshot of the OrangeHRM dashboard. The URL in the browser is https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index. The dashboard has a header with the OrangeHRM logo, a user profile for "Oliver Hofer", and an "Upgrade" button. On the left is a sidebar with links: Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and a highlighted "Dashboard". The main area is divided into several cards: "Time at Work" showing "Punched Out" at 12:56 PM (GMT 5.5), "My Actions" with 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" (Oliver Sven Hofer, 2020-08-10 09:08 AM) and "Employees on Leave Today".

### 3) PathSelector

```
package JavaPracticals;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver; // Using Firefox driver

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

        // Set the system property for GeckoDriver (used by Firefox)
        System.setProperty(
            "webdriver.gecko.driver",
            "C:\\selenium web driver\\ChromeDriver\\chromedriver-win64\\chromedriver-
win64\\geckodriver.exe"
        );

        // Initialize the Firefox WebDriver
        WebDriver wd = new FirefoxDriver();

        // Open the OrangeHRM login page
        wd.get("https://opensource-demo.orangehrmlive.com/");

        // Wait for 2 seconds to ensure page loads
        Thread.sleep(2000);

        // Enter username in the username input field

        wd.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[1]/div/div[2]/input"))
            .sendKeys("admin");

        // Enter password in the password input field

        wd.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[2]/div/div[2]/input"))
            .sendKeys("admin123");

        // Click the login button

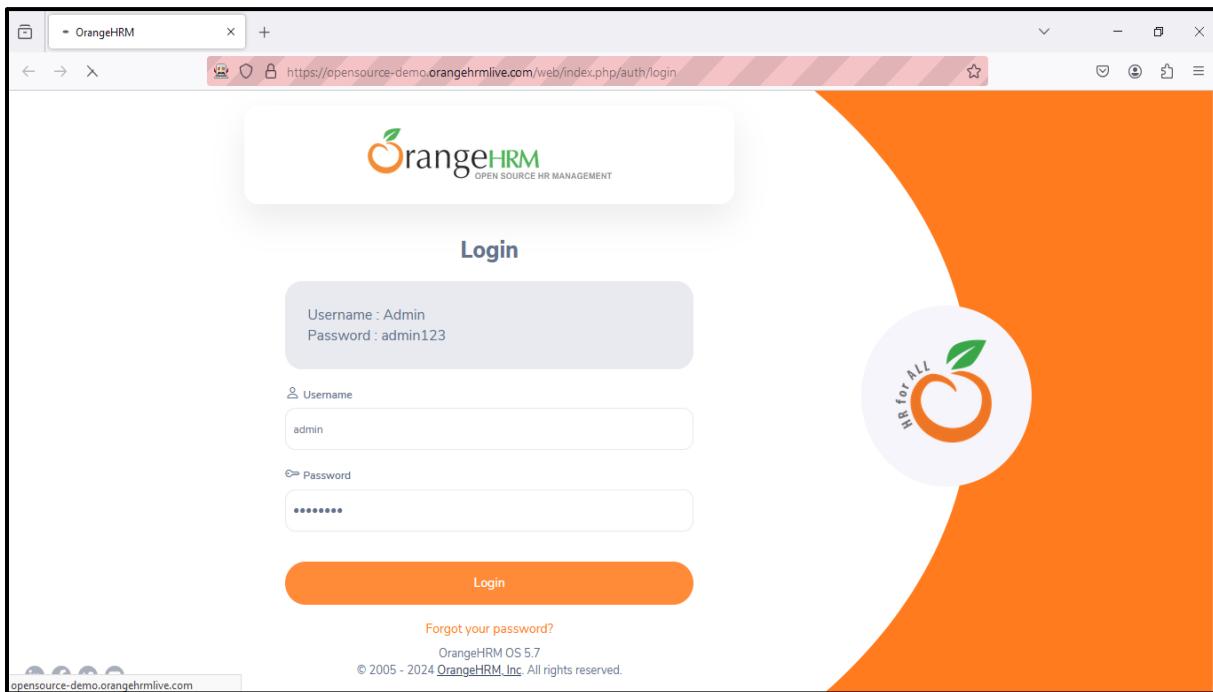
        wd.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button"))
            .click();
    }
}
```

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

### **OUTPUT:**



The screenshot shows the OrangeHRM dashboard. On the left is a sidebar with navigation links: Search, Admin, PIM, Leave, Time, Recruitment, My Info, Performance, Dashboard (which is selected and highlighted in orange), Directory, Maintenance, Claim, and Buzz. The main dashboard area has several modules: "Time at Work" showing a punch-in record for "Nimra Shehoni" at 10:15 PM (GMT+4) on Aug 11, "My Actions" listing pending self-reviews and candidates for interview, and "Quick Launch" buttons for Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, and My Timesheet. Other visible modules include "Buzz Latest Posts" and "Employees on Leave Today".

## PRACTICAL NO: 07

**Aim:** Demonstrate Synchronisation in Selenium (Implicit wait,Explicitly wait)

### **1) Implicit wait**

```
Package JavaPracticals;
```

```
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Practical2 {

    @SuppressWarnings("deprecation")
    public static void main(String[] args) throws Exception {

        System.setProperty(
            "webdriver.gecko.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);

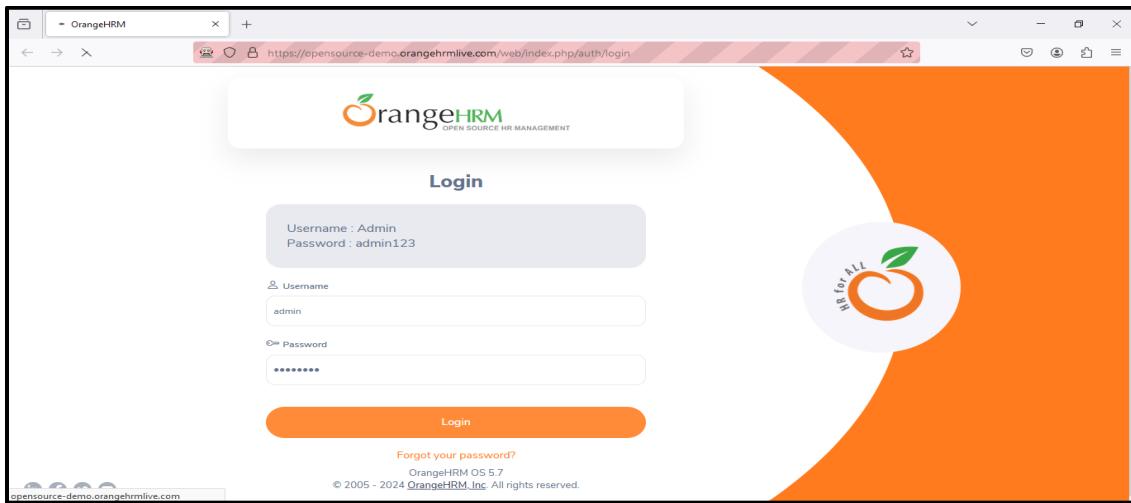
        wd.findElement(By.name("username")).sendKeys("admin");
        wd.findElement(By.name("password")).sendKeys("admin123");
        wd.findElement(By.className("oxd-button")).click();
        wd.close();
    }
}
```

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

### **OUTPUT:**



The screenshot shows the OrangeHRM dashboard. The left sidebar has a "Search" field and a navigation menu with items like Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and Dashboard (which is highlighted). The main dashboard area has several cards: "Time at Work" showing a punch out at 11:24 PM (GMT 4), "My Actions" with pending self reviews and interview candidates, "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 Nimma Shehani; "Employees on Leave Today"; and "Employee Distribution by Sub Unit". The top right corner shows an "Upgrade" button and a user profile for Nimma Shehani.

## 2) Explicit wait

```
package JavaPracticals;

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 Practical3 {
    public static void main(String[] args) throws Exception {

        System.setProperty(
            "webdriver.gecko.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");

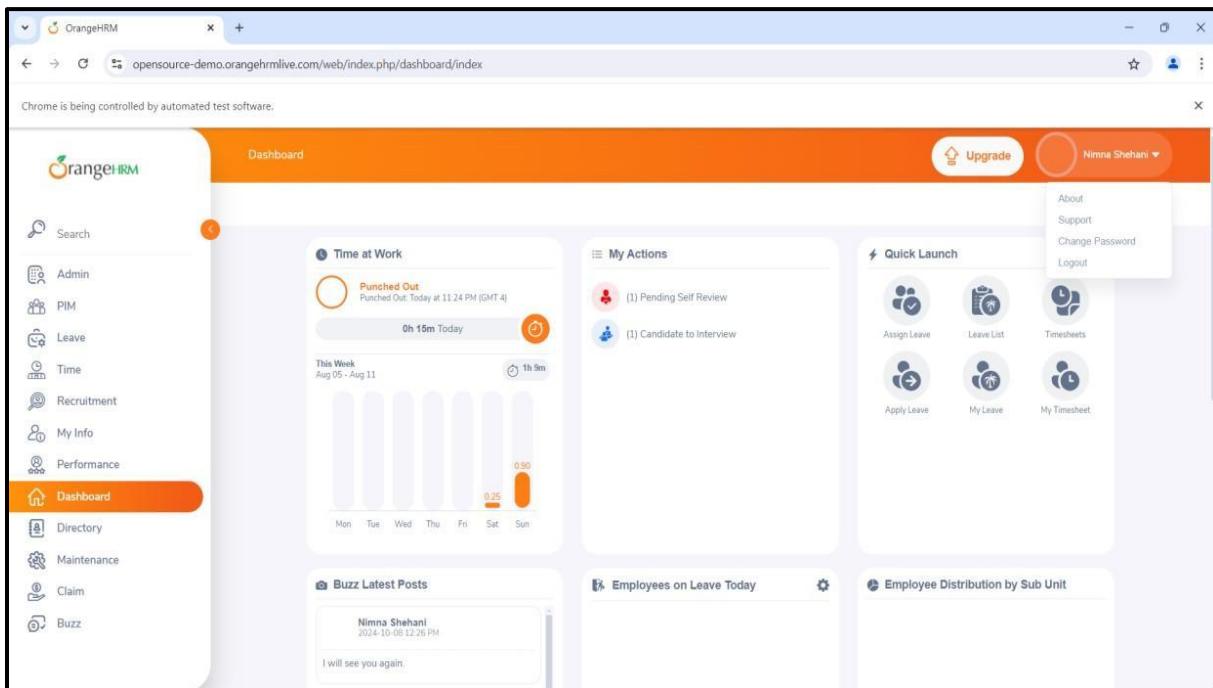
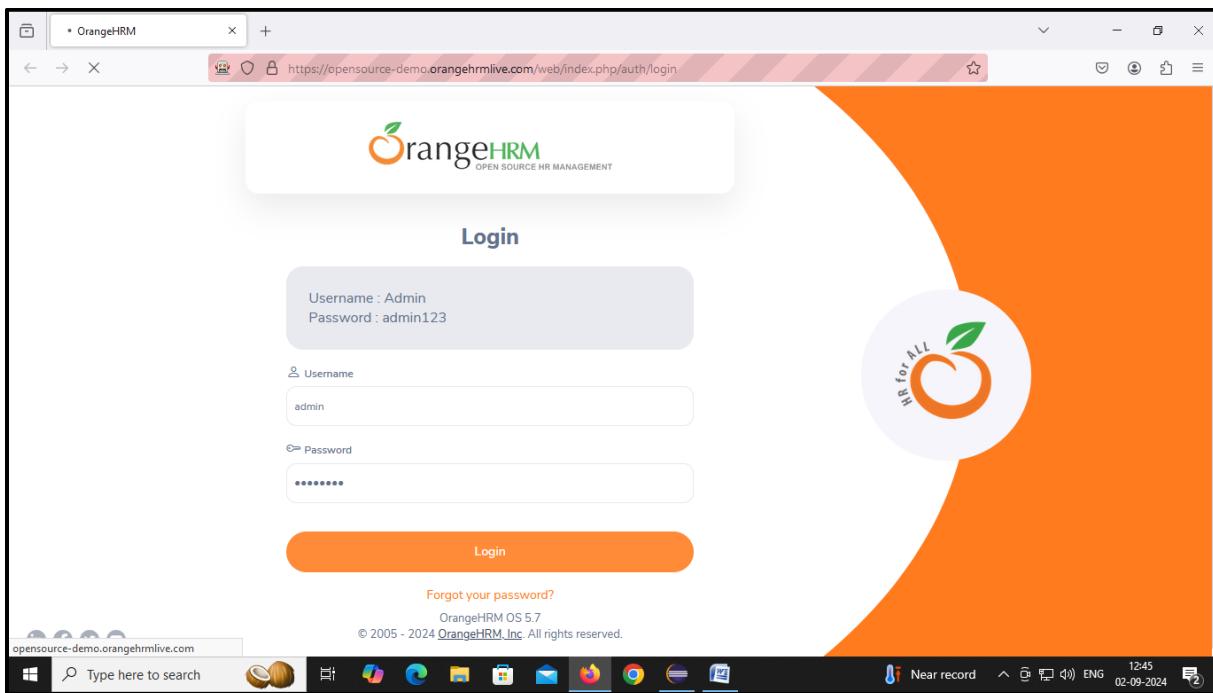
        wt.until(ExpectedConditions.visibilityOfElementLocated(By.name("password"))).sendKeys("admin12
3");
        wt.until(ExpectedConditions.visibilityOfElementLocated(By.className("oxd-button"))).click();
        wt.until(ExpectedConditions.elementToBeClickable(By.className("oxd-userdropdown-
tab"))).click();
        Thread.sleep(5000);
        wt.until(ExpectedConditions.elementToBeClickable(By.partialLinkText("Logout"))).click();
    }
}
```

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

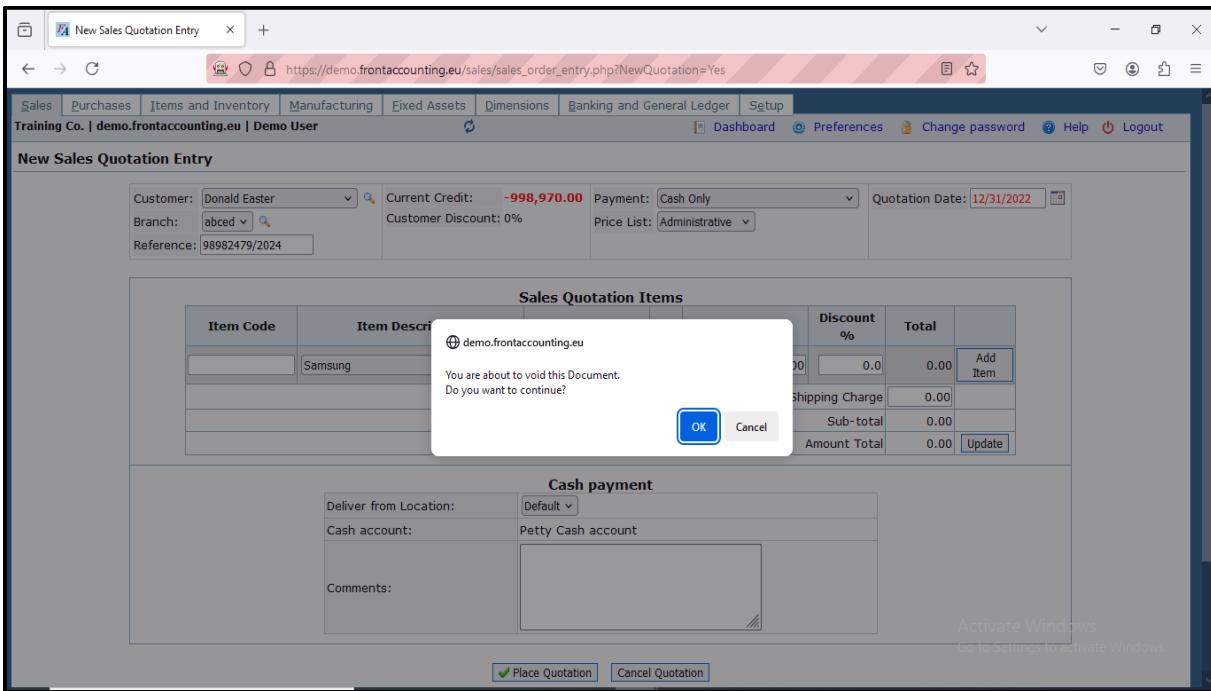
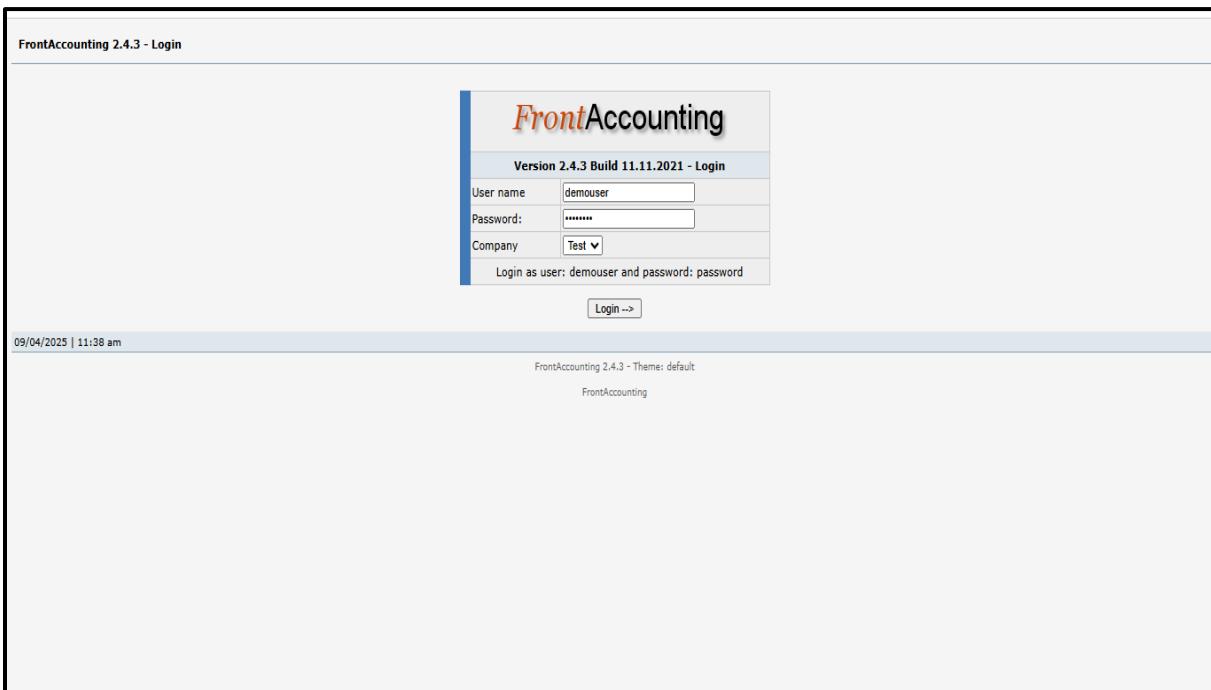
### **OUTPUT:**



## PRACTICAL NO: 08

**Aim:** Demonstrate Different types of alerts.

```
package JavaPracticals;
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 alert {
    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();
    }
}
```

**OUTPUT:**

## **PRACTICAL NO: 09**

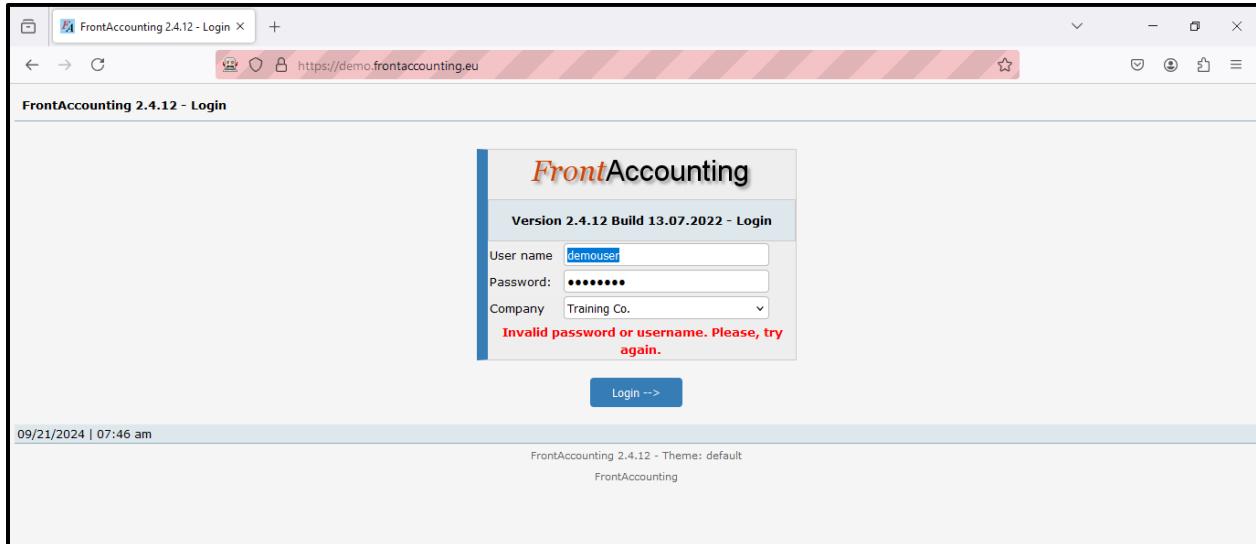
**Aim:** Demonstrate Multi Window Handling.

```
package JavaPracticals;

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 MultipleWindowHandling {
    public static void main(String[] args) {
        //Set the path for GeckoDriver (since you're using Firefox)
        System.setProperty("webdriver.gecko.driver",
            "C:\\selenium web driver\\ChromeDriver\\chromedriver-
win64\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        // Open FrontAccounting demo site
        wd.get("https://demo.frontaccounting.eu/");

        // Login
        wd.findElement(By.name("user_name_entry_field")).sendKeys("your_username"); // Replace with valid username
        wd.findElement(By.name("password")).sendKeys("your_password"); // Replace with valid password
        // Select company (if required)
        Select s = new Select(wd.findElement(By.name("company_login_name")));
        // s.selectByVisibleText("Default Company"); // Uncomment if needed
        wd.findElement(By.name("SubmitUser")).click();
        // Click link to open a new window
        wd.findElement(By.xpath(
            "/html/body/table[1]/tbody/tr/td/table[1]/tbody/tr/td/div[2]/table/tbody/tr[1]/td/t
able/tbody/tr[2]/td[1]/a"
        )).click();
        // Get parent window ID
        String parentId = wd.getWindowHandle();
        System.out.println("Parent Window: " + parentId);
        // Click a link that opens another window
        wd.findElement(By.xpath(
            "/html/body/table[1]/tbody/tr/td/table[1]/tbody/tr/td/div[2]/form/center[1]/table/t
body/tr/td[2]/table/tbody/tr[1]/td[2]/a"
        
```

```
]).click();
// Handle all window IDs
Set<String> allWindows = wd.getWindowHandles();
for (String windowId : allWindows) {
    wd.switchTo().window(windowId);
    System.out.println("Window Title: " + wd.getTitle());
    if (!windowId.equals(parentId)) {
        // Perform actions in the new window
        Select ss = new Select(wd.findElement(By.id("customer_id"))); // Ensure ID is
correct
        ss.selectByVisibleText("Donald Easter");
        wd.manage().window().maximize();
        wd.close(); // Close the child window after work
    }
}
// Switch back to parent window
wd.switchTo().window(parentId);
// Logout
wd.findElement(By.linkText("Logout")).click();
// Close browser
wd.quit();
}
```

**OUTPUT:**

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

The screenshot shows the 'Customer Transactions' page of the FrontAccounting software. At the top, there is a search bar with dropdowns for 'Select a customer' (set to 'Jash'), 'All Types' (set to 'All'), 'From' (set to '09/23/2024'), 'To' (set to '10/23/2024'), and a checkbox for 'Zero values'. Below the search bar is a table header with columns: Currency, Terms, Current, 1-30 Days, 31-60 Days, Over 60 Days, and Total Balance. A single row is present in the table, showing 'USD' in the Currency column and 'Cash Only' in the Terms column. The 'Current' column shows '0.00' for all three time periods and '0.00' in the Total Balance column. Below the table, a message says 'No records' and 'Marked items are overdue.' There is a 'Back' button at the bottom left. The status bar at the bottom shows the date and time: '10/23/2024 | 08:50 am' and the window title 'F2 - Customers'. The footer displays the software version 'FrontAccounting 2.4.12 - Theme: default - FrontAccounting'.

The screenshot shows the 'Logout' page of the FrontAccounting software. The main heading is 'FrontAccounting'. Below it, a message says 'Thank you for using FrontAccounting 2.4.12' and 'Click here to Login Again.' The status bar at the bottom shows the date and time: '10/23/2024 | 08:52 am' and the window title 'Logout'. The footer displays the software version 'FrontAccounting 2.4.12 - Theme: default - FrontAccounting'.

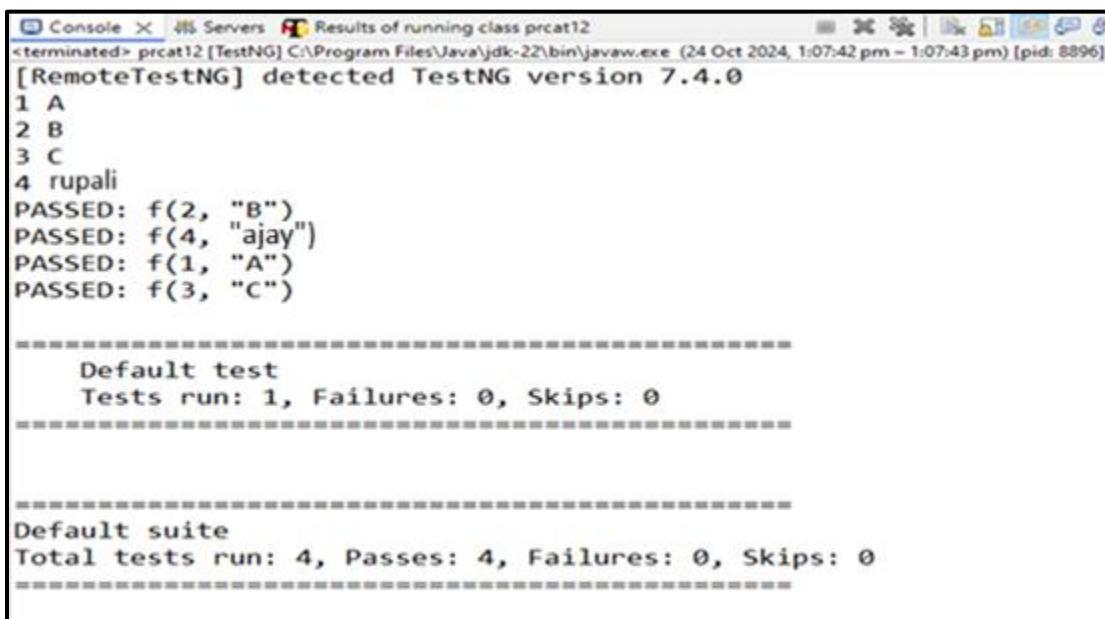
## PRACTICAL NO: 10

**Aim:** Demonstrate TestNG

### **A) Demo DataProvider**

```
package testNG;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
public class Pract12 {
    @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, "ajay"},
        };
    }
}
```

### OUTPUT:



The screenshot shows a Java console window with the following output:

```
Console X Servers Results of running class pcat12
<terminated> pcat12 [TestNG] C:\Program Files\Java\jdk-22\bin\javaw.exe (24 Oct 2024, 1:07:42 pm – 1:07:43 pm) [pid: 8896]
[RemoteTestNG] detected TestNG version 7.4.0
1 A
2 B
3 C
4 rupali
PASSED: f(2, "B")
PASSED: f(4, "ajay")
PASSED: f(1, "A")
PASSED: f(3, "C")

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

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

## B) Demo Annotation

```
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: 111");
    }
    @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:**

```

terminated> pract12B [TestNG] C:\Users\Ajay\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe (17 Oct 2025, 8:5
[RemoteTestNG] detected TestNG version 7.11.0
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
Before Suite
Before Test
Before Class
Before Method
Roll no: 111
After Method
After Class
After Test
PASSED: testNG.pract12B.f

=====
Default test
Tests run: 1, PASEDG.pract1111. f
=====

After Suite

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

**C) Data OHM**

```

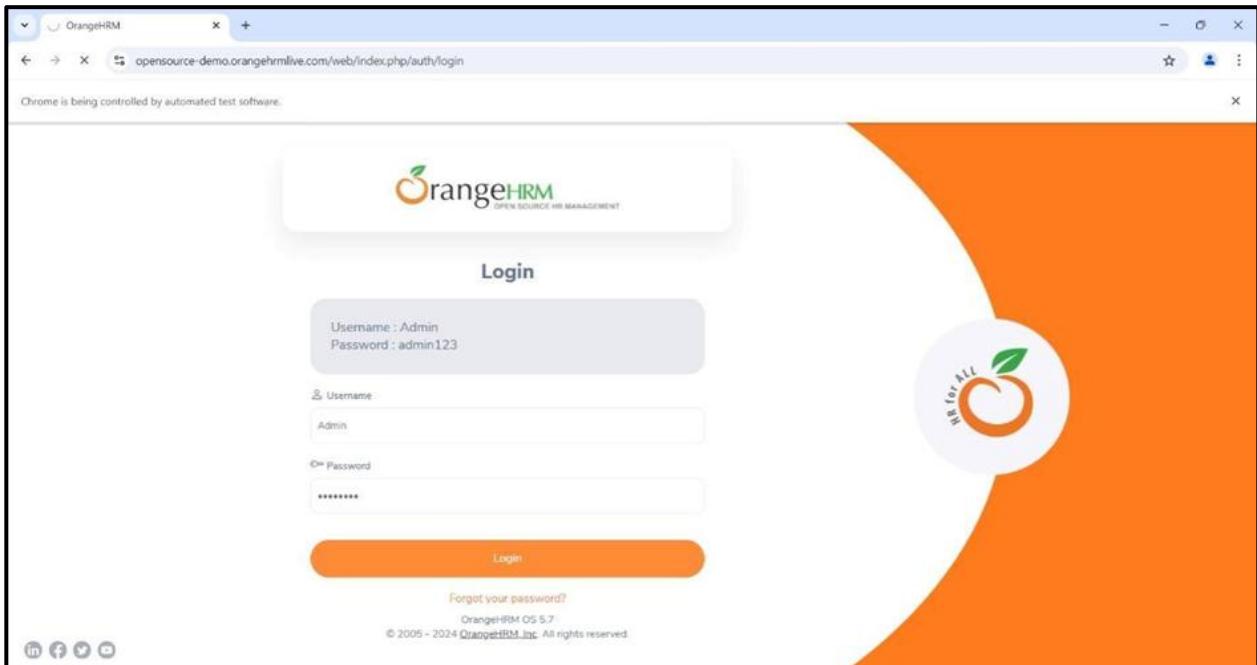
package testNG;

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;
public class pracat12C {
    @Test(dataProvider = "dp")
    public void f(String u, String p) {
        System.setProperty("webdriver.gecko.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 name
        wd.findElement(By.name("password")).sendKeys(p); // locator 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");
    } finally {
        wd.quit();
    }
}
@DataProvider
public Object[][] dp() {
    return new Object[][] {
        { "Admin", "admin123" },
        { "add", "b" }
    };
}
```

**OUTPUT:**



## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

The screenshot shows the OrangeHRM dashboard. On the left is a sidebar with icons for Search, Admin, PIM, Leave, Time, Recruitment, My Info, Performance, and a highlighted Dashboard. The main area has a header "Dashboard" and a sub-header "Time at Work". It displays a circular icon indicating "Punched Out" and "0h 16m Today". Below this is a bar chart for the week of Aug 05 - Aug 11, with a legend showing 0.2% and 0.9%. To the right are sections for "My Actions" (Pending Self Review, Candidate to Interview), "Quick Launch" (Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, My Timesheet), and "Buzz Latest Posts" (Nimra Shahani's post: "I will see you again"). A top navigation bar includes Upgrade, Nimra Shahani, and links for About, Support, Change Password, and Logout.

The screenshot shows the OrangeHRM login page. The URL in the address bar is "opensource-demo.orangehrmlive.com/web/index.php/auth/login". The page features the OrangeHRM logo and a large orange background graphic with a stylized orange icon. The login form includes fields for "Username" (ncrd) and "Password" (\*\*\*\*\*). A red error message box says "(!) Invalid credentials". Below the form are "Login" and "Forgot your password?" buttons. At the bottom, it says "OrangeHRM © 2014-2024".

```
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")

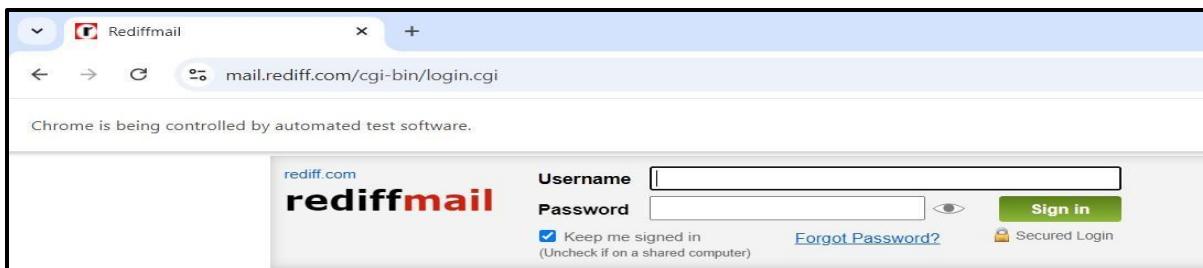
=====
Default test
Tests run: 1, Failures: 0, Skips: 0
=====
```

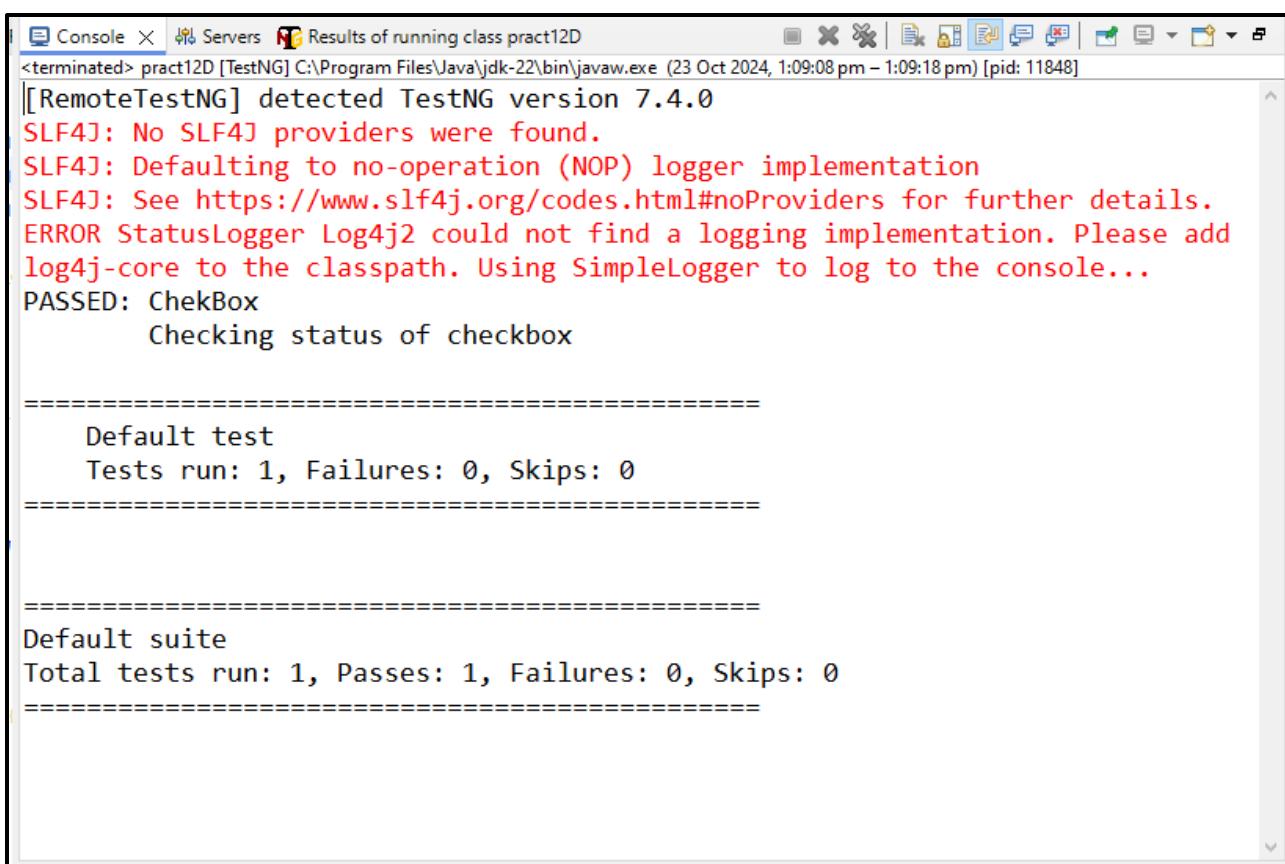
## D) DataCheck

```
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 CheckBox() {
        System.setProperty("webdriver.gecko.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, "Checkbox is not selected!");
        wd.quit();
    }
}
```

## OUTPUT:



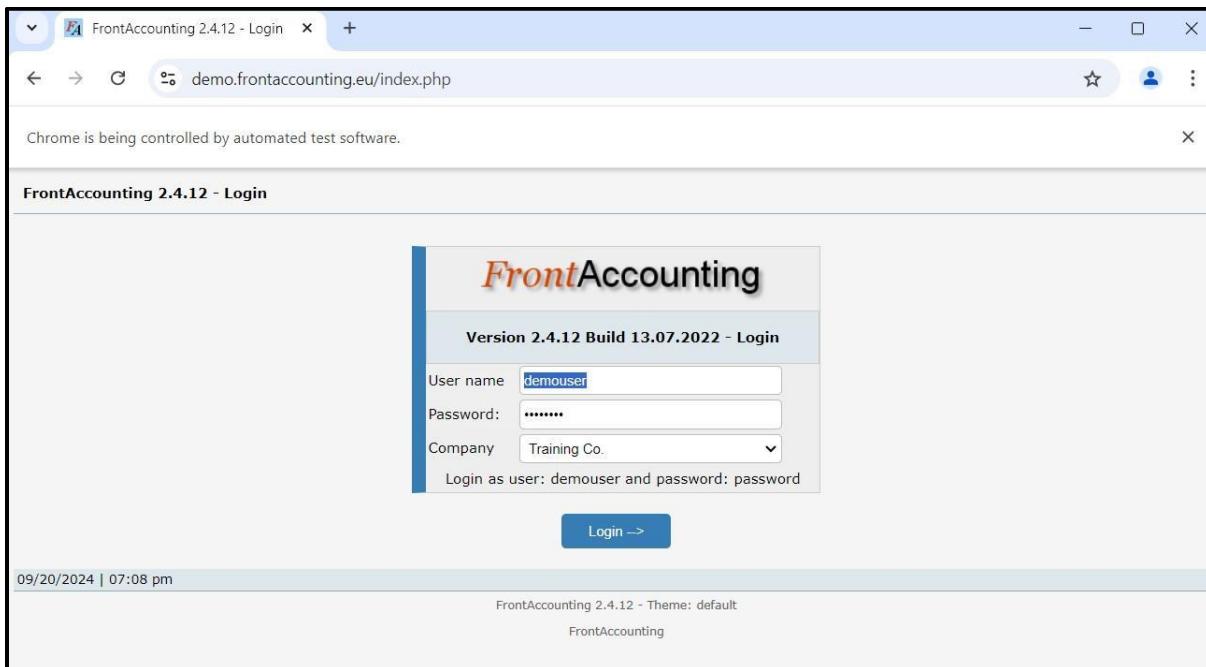


The screenshot shows a Java console window with the following output:

```
<terminated> pract12D [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

```
package testNG;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.Assert;
import org.testng.annotations.Test;
public class pract12E {
    @Test
    public void frontAccounting() {
        System.setProperty("webdriver.gecko.driver",
            "C:\\\\selenium web driver\\\\ChromeDriver\\\\chromedriver-win64\\\\chromedriver-win64\\\\geckodriver.exe");
        WebDriver wd = new FirefoxDriver();
        wd.get("https://demo.frontaccounting.eu/index.php");
        String expectedTitle = "FrontAccounting 2.4.12 - Login"; // if title is not matching,
        execution will fail
        String actualTitle = wd.getTitle();
        Assert.assertEquals(actualTitle, expectedTitle, "Title does not match!");
        System.out.println("Title of Webpage =====> " + actualTitle);
        wd.quit();
    }
}
```

**OUTPUT:**

```
Console × Servers Results of running class pract12E
<terminated> pract12E [TestNG] C:\Program Files\Java\jdk-22\bin\javaw.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
=====
```

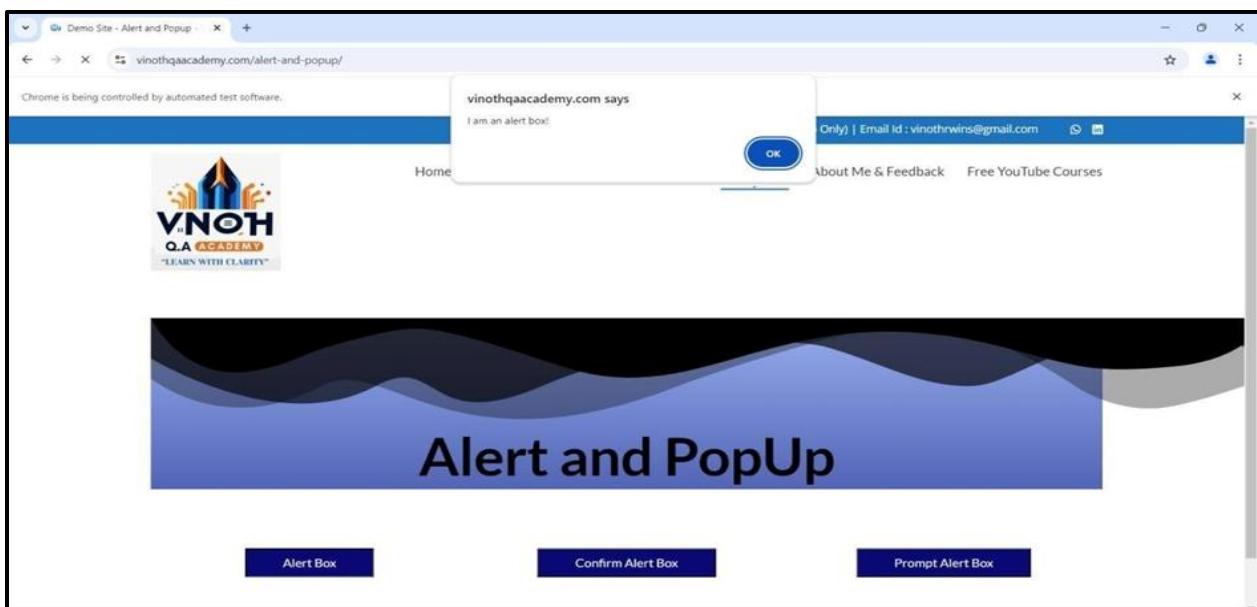
## F) DemoAlert

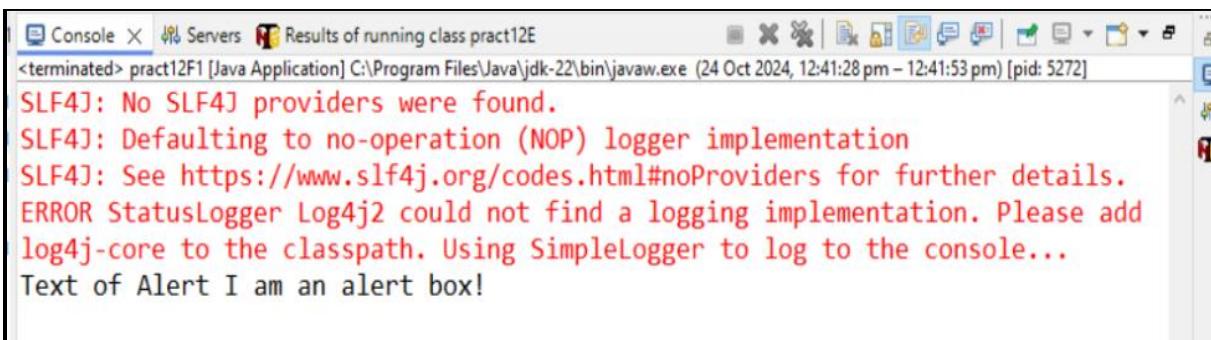
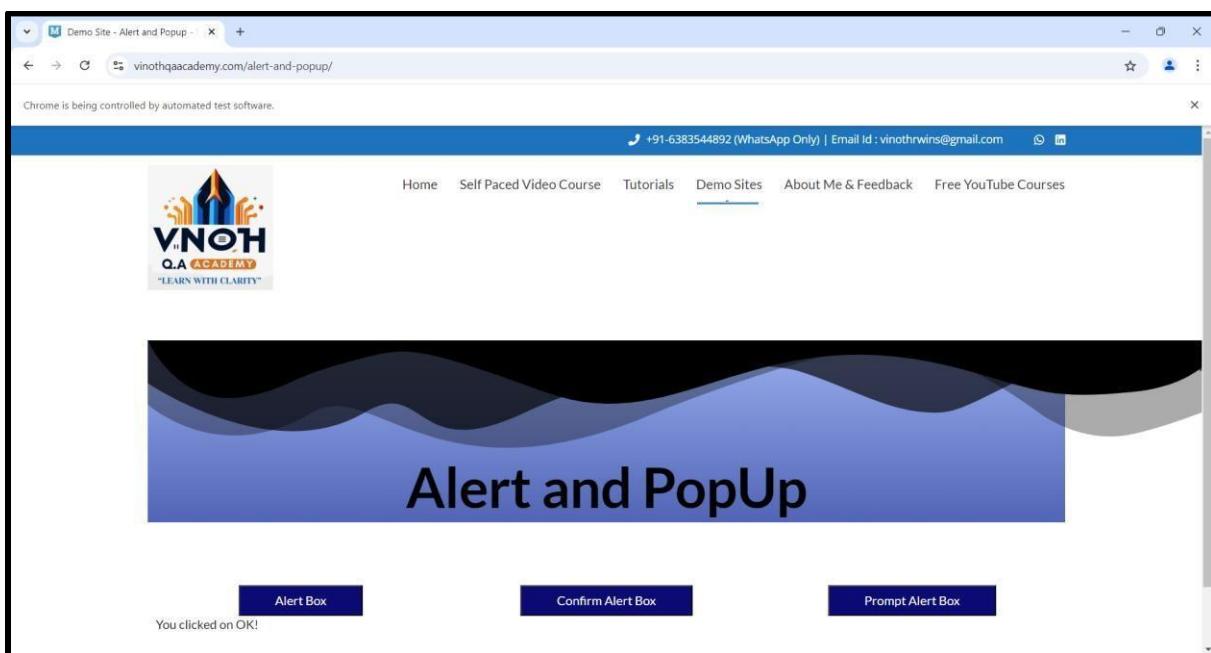
### 1) Alertbox

```
package testNG;
import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class pract12F1 {
    public static void main(String[] args) throws Exception {
        System.setProperty("webdriver.gecko.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("alertbox")).click();
        Thread.sleep(2000);
        Alert alt = wd.switchTo().alert();
        System.out.println("Text of Alert: " + alt.getText()); // capture alert text
        alt.accept(); // click on OK button
        wd.quit();
    }
}
```

### OUTPUT:





### G) TestNG using xml

```
package testNG;

import utilities.ReadPropertyFile; import java.io.IOException; import java.util.Properties;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver; import
io.github.bonigarcia.wdm.WebDriverManager;

public class MyFirstTest {
public static void main(String[] args) throws IOException, InterruptedException {
Properties prop = ReadPropertyFile.getProperties();
String browser = prop.getProperty("browser"); String url = prop.getProperty("testurl");
if (browser.equalsIgnoreCase("chrome")) { WebDriverManager.chromedriver().setup();
WebDriver driver = new ChromeDriver();
```

```
to ( driver.manage().window().maximize(); driver.get(url);
System.out.println("Opened URL: " + url);
```

```
I/ Example locators (update according to the real page)
I/ driver.findElement(By.linkText("Sign in")).click();
I/ driver.findElement(By.id("login id")).sendKeys(prop.getProperty("login id"));
I/ driver.findElement(By.id("password")).sendKeys(prop.getProperty("password"));
I/ driver.findElement(By.xpath("//button[@id='nextbtn']")).click();

Thread.sleep(3000);
} finally { driver.quit();
} else {
System.out.println("Browser not supported: " + browser);
```

**OUTPUT:**

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

- Project Structure:** The left sidebar shows the project tree for "Aj\_SeleniumFramework". It includes packages like abc, Ahrd, generics student, java, practicals5, Sankriti (33), and SeleniumFramework. Under SeleniumFramework, there are src/main/java, src/main/resources, src/test/java (containing testcase and utilities), and src/test/resources.
- Code Editor:** The main editor window displays Java code for a utility class named ReadPropertyFile. The code reads a properties file from the resources directory and returns its contents as a Properties object.
- Console Output:** The bottom right panel shows the terminal output of the Java application. It includes the command run, the Java version (JavaSE-1.7), and the execution path (C:\Program Files\Java\jdk-17\bin\javaw.exe). It also shows several warning messages related to CDP implementation and version matching, along with the message "Opened URL: https://www.zomato.com".

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

A screenshot of the Eclipse IDE interface. The left pane shows a project structure for 'Aj - SeleniumFramework'. The 'src/test/resources/configfiles/config.properties' file is open, displaying the following content:

```
1 browser=chrome\n2 testurl=https://www.zoho.com/\n3 login_id=your.email@example.com\n4 password=yourpassword\n5
```

The right pane shows the 'Console' tab with the output of a Java application running. The output includes:

```
<terminated> MyFirstTest [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 3, 2025, 2:46:43 PM – 2:46:52 PM) [pid: 21440]\nNov 03, 2025 2:46:45 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch\nWARNING: Unable to find CDP imple <terminated> MyFirstTest [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 3, 2025, 2:46:43 PM – 2:46:52 PM) [pid: 21440]\nNov 03, 2025 2:46:45 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5\nWARNING: Unable to find version of CDP to use for 141.0.7390.123. You may need to include a dependency on a specific version of the C\nOpened URL: https://www.zoho.com/
```

A screenshot of the Zoho Cloud Software Suite homepage. The top navigation bar includes links for Products, Enterprise, Customers, Partners, and Resources. The main headline reads:

# Your life's work, powered by our life's work

A subtext below the headline states:

A unique and powerful software suite to transform the way you work. Designed for businesses of all sizes, built by a company that values your privacy.

A 'GET STARTED FOR FREE' button is located at the bottom left. On the right side, there is a 'FEATURED APPS' section with icons and descriptions for CRM, Mail, Books, and People.

The URL in the address bar is: <https://accounts.zoho.in/signin?servicename=ZohoHome&signupurl=https://www.zoho.com/>

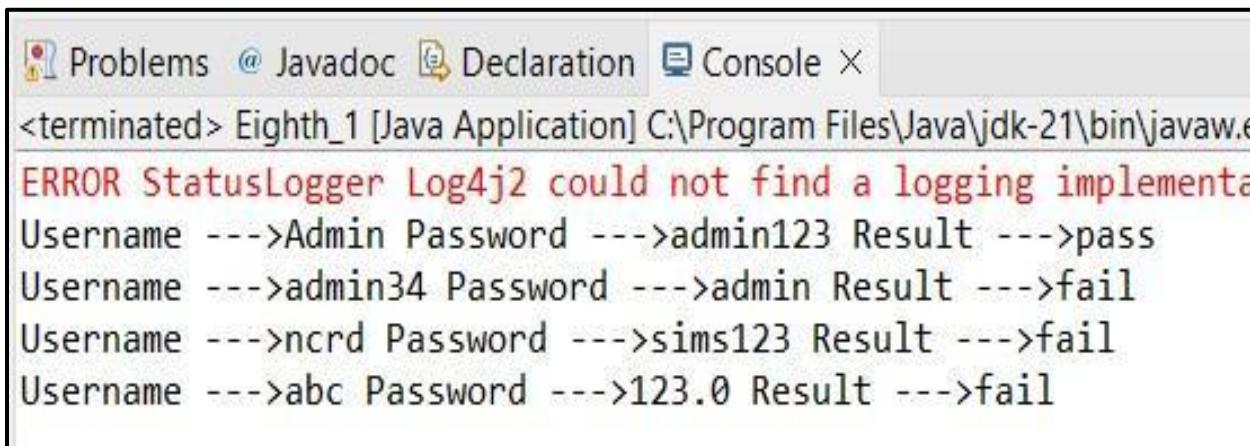
## PRACTICAL NO: 11

**Aim:** Demonstrate data driven Framework

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

```
package JavaPracticals;
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\\Excel1.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);
        }
        wb.close();
        fis.close();
    }
}
```

**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
```

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

A screenshot of Microsoft Excel with a table titled 'B12'. The table has three columns: 'Username' (A), 'Password' (B), and 'Result' (C). The data rows are:

	Username	Password	Result
1	Admin	admin123	pass
2	admin34	admin	fail
3	ncrd	sims123	fail
4	abc		1234 fail

A screenshot of a web browser displaying the OrangeHRM login page. The URL in the address bar is [opensource-demo.orangehrmlive.com/web/index.php/auth/login](https://opensource-demo.orangehrmlive.com/web/index.php/auth/login). The page features the OrangeHRM logo and a large orange circular graphic on the right. The login form includes fields for 'Username' (Admin) and 'Password' (admin123), both pre-filled with placeholder text. A 'Login' button is at the bottom. Below the form are links for 'Forgot your password?' and 'OrangeHRM OS 5.7'. The footer contains the copyright notice '© 2005 - 2024 OrangeHRM Inc. All rights reserved.'

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

The screenshot shows the OrangeHRM dashboard. On the left is a sidebar with icons for Search, Admin, PIM, Leave, Time, Recruitment, My Info, Performance, Dashboard (which is selected), Directory, Maintenance, Claim, and Buzz. The main area has a header with the OrangeHRM logo, an Upgrade button, and a user profile for Nimma Shehani. Below the header are several cards: 'Time at Work' (Punched Out, 0h 15m Today, 1h 9m, Mon-Sun), 'My Actions' (Pending Self Review, Candidate to Interview), 'Quick Launch' (Assign Leave, Leave List, Timesheets, Apply Leave, My Leave, My Timesheet), 'Buzz Latest Posts' (Nimma Shehani, 2024-10-08 12:26 PM, I will see you again.), 'Employees on Leave Today' (empty), and 'Employee Distribution by Sub Unit' (empty). A context menu is open over the user profile.

The screenshot shows the OrangeHRM login page. The URL is [opensource-demo.orangehrmlive.com/web/index.php/auth/login](https://opensource-demo.orangehrmlive.com/web/index.php/auth/login). The page features the OrangeHRM logo and a large orange circular graphic on the right. The login form includes fields for Username (nrcd) and Password (\*\*\*\*\*), both of which are highlighted in red. A message at the top says "Invalid credentials". Below the form is a "Login" button and links for "Forgot your password?" and "OrangeHRM HOME".

## **PRACTICAL NO: 12**

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

```
package JavaPracticals;
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\\chromedriverwin64\\chromedriver.exe");
        WebDriver wd = new ChromeDriver();
        wd.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        wd.get("https://opensource-demo.orangehrmlive.com/");
        FileInputStream fis = new FileInputStream("D:\\STQA\\Excel1.xlsx");
        XSSSSFWorkbook wb = new XSSSSFWorkbook(fis);
        XSSFSheet sh = wb.getSheet("WriteData");
        for (int i = 1; i <= sh.getLastRowNum(); i++) {
            XSSFRow rw = sh.getRow(i);
            XSSFCell un = rw.getCell(0);
            XSSFCell pw = rw.getCell(1);
            XSSFCell res = rw.createCell(2);
            System.out.println("Username --> " + un + " | Password --> " + pw);
            wd.findElement(By.name("username")).clear();
            wd.findElement(By.name("username")).sendKeys(un.toString());
            wd.findElement(By.name("password")).clear();
            wd.findElement(By.name("password")).sendKeys(pw.toString());
            wd.findElement(By.className("oxd-button")).click();
            try {
                wd.findElement(By.className("oxd-userdropdown-name")).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);
fos.close();
fis.close();
wb.close();
wd.quit();
System.out.println("Test completed and results written to Excel file.");
}
}
```

**OUTPUT:**

	A	B	C	D	E	F	G	H	I	J
1	Username	Password	Result							
2	Admin	admin123	pass							
3	admin34	admin	fail							
4	ncrd	sims123	fail							
5	abc		1234 fail							

Problems @ Javadoc Declaration Console ×

<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

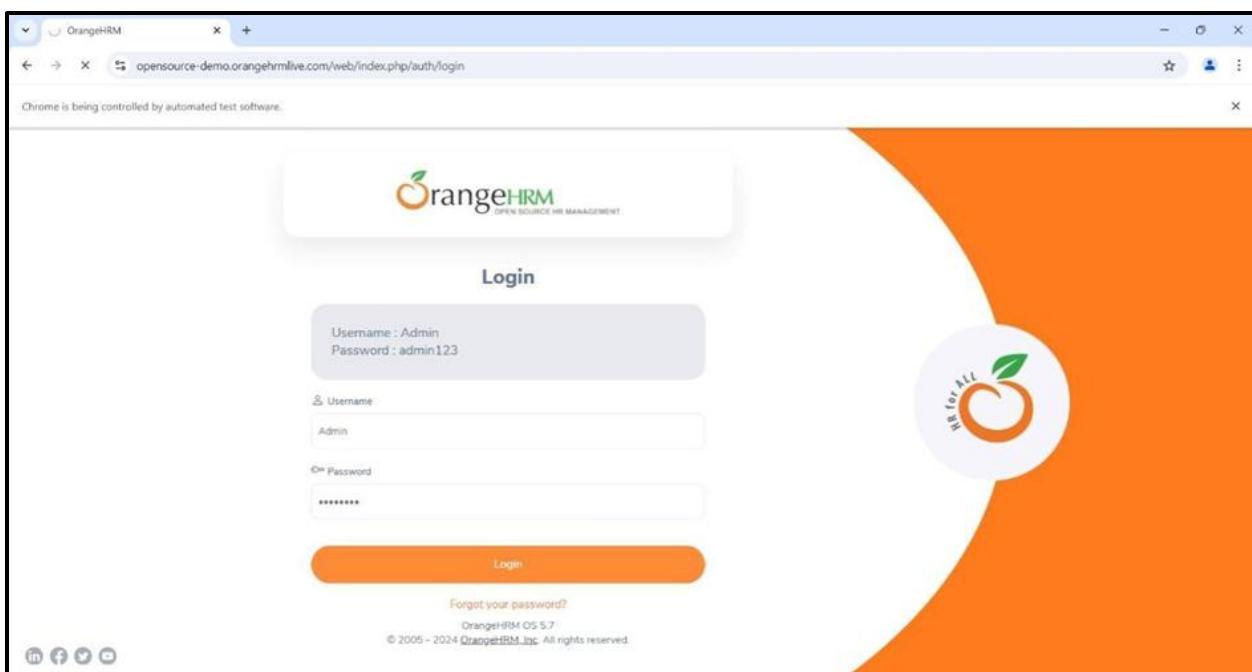
username --> abc Password --> 123.0

Login Fail

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)



The screenshot shows the OrangeHRM dashboard for the user "Nimra Shahani". The left sidebar contains a navigation menu with items like "Search", "Admin", "PIM", "Leave", "Time", "Recruitment", "My Info", "Performance", and "Dashboard" (which is highlighted). Other items in the sidebar include "Directory", "Maintenance", "Claim", and "Buzz". The main dashboard area has several sections: "Time at Work" showing a punch record ("Punched Out, Today at 11:24 PM (GMT 4)"), a weekly time card for Aug 05 - Aug 11, and a summary for the day (0h 18m Today); "My Actions" listing "Pending Self Review" and "Candidate to Interview"; "Quick Launch" buttons for "Assign Leave", "Leave List", "Timesheets", "Apply Leave", "My Leave", and "My Timesheet"; "Buzz Latest Posts" showing a recent post from "Nimra Shahani"; "Employees on Leave Today" showing one employee; and "Employee Distribution by Sub Unit" which is currently empty. The top right corner shows the user profile "Nimra Shahani" and a dropdown menu with options: "About", "Support", "Change Password", and "Logout".

## SOFTWARE TESTING AND QUALITY ASSURANCE

Dattaram Santosh Kolte

SYMCA-B (97)

The screenshot shows a Chrome browser window with the URL [opensource-demo.orangehrmlive.com/web/index.php/auth/login](https://opensource-demo.orangehrmlive.com/web/index.php/auth/login). The page is titled "Login" and features the OrangeHRM logo. A prominent orange error message box displays the text "Invalid credentials". Below this, a light gray box shows the entered credentials: "Username : Admin" and "Password : admin123". The login form includes fields for "Username" (containing "nord") and "Password" (containing "\*\*\*\*\*"). An orange "Login" button is at the bottom. To the right of the form is a large orange graphic element with a white circular logo containing an orange fruit and the text "Nord". At the top left of the browser window, it says "Chrome is being controlled by automated test software."

## **PRACTICAL NO: 13**

**Aim:** Validation Testing using Chrome and Bing.

A) Chrome

Package JavaPracticals;

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

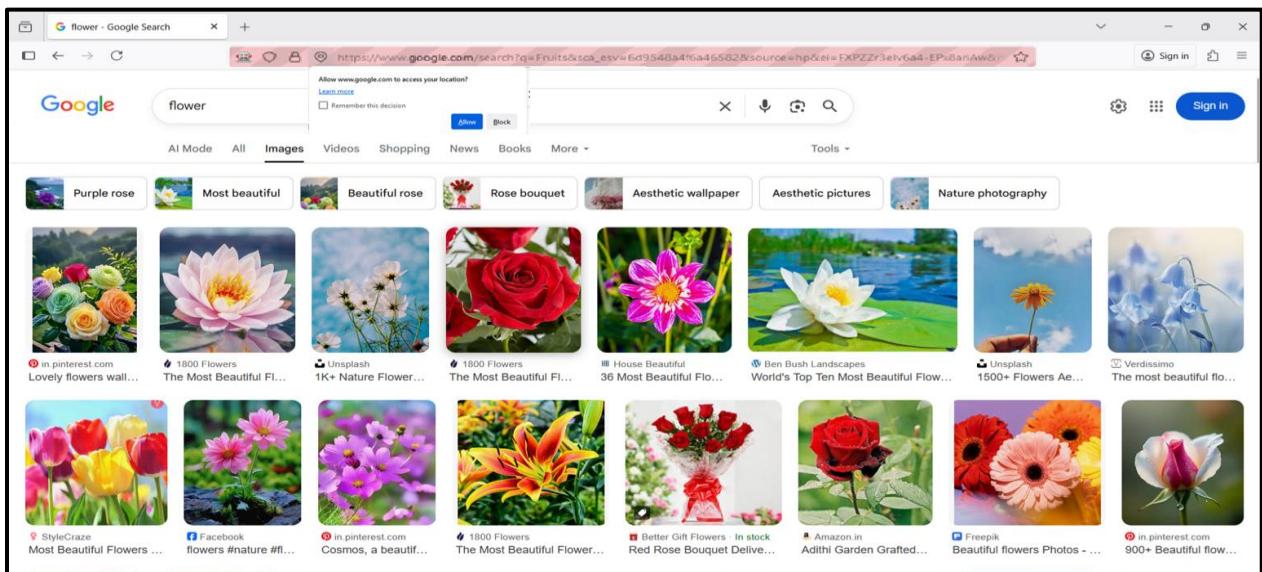
public class Practical8 {
    public static void main(String[] args) throws Exception {
        System.setProperty("webdriver.gecko.driver", "C:\\\\Selenium
webdriver\\\\geckodriver\\\\geckodriver.exe");

        WebDriver wd = new FirefoxDriver();
        wd.get("https://google.com/");
        // wd.get("https://bing.com/");

        Thread.sleep(2000);

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

## **OUTPUT:**



**B)Bing**

```
package JavaPracticals;

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

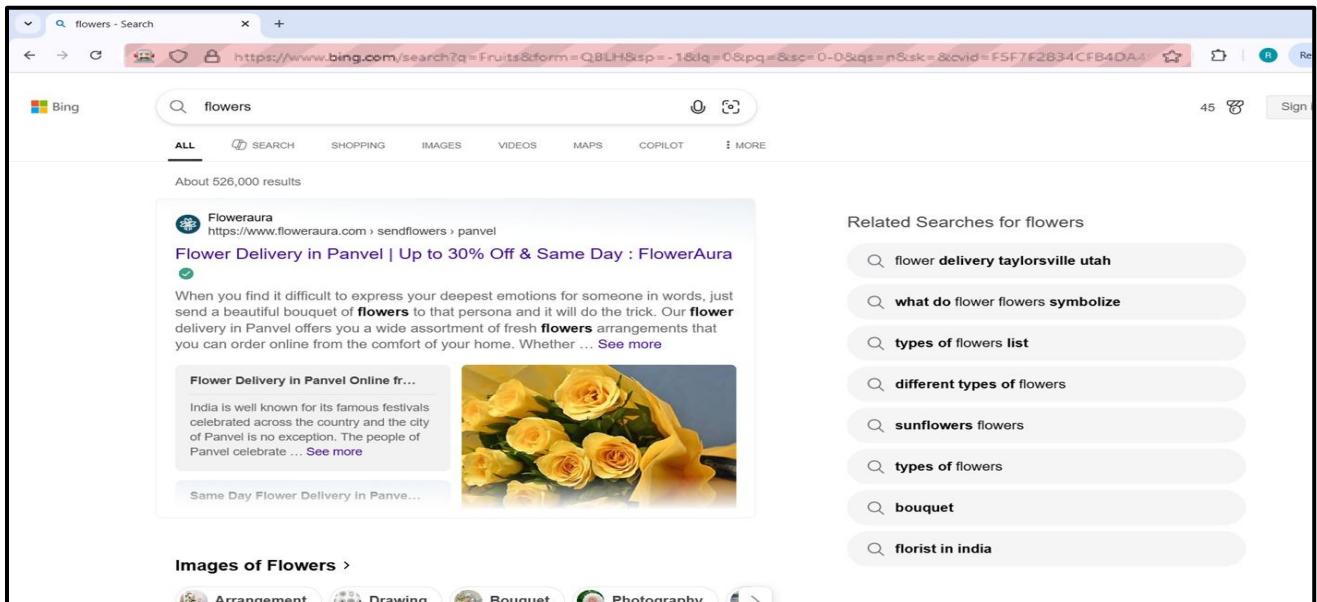
public class Practical8 {
    public static void main(String[] args) throws Exception {
        System.setProperty("webdriver.gecko.driver", "C:\\\\Selenium
webdriver\\\\geckodriver\\\\geckodriver.exe");

        WebDriver wd = new FirefoxDriver();
        wd.get("https://google.com/");
        // wd.get("https://bing.com/");

        Thread.sleep(2000);

        // For Firefox (Google)
        // 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:**

## PRACTICAL NO: 14

**Aim:** Demonstrate Library Architecture Framework.

```
package testcases;

import library.AppLibrary;
import utils.ExcelLibrary;

public class TestExecution {
    public static void main(String[] args) throws Exception {
        String excelPath = "C:\\\\Users\\\\Asus\\\\Downloads\\\\ExcelData.xlsx";
        ExcelLibrary excel = new ExcelLibrary(excelPath, "Sheet1");
        AppLibrary app = new AppLibrary();

        System.out.println("--- Starting Library Architecture Framework ---");

        for (int i = 1; i <= excel.getRowCount(); i++) {
            String username = excel.getCellData(i, 0);
            String password = excel.getCellData(i, 1);
            String expected = excel.getCellData(i, 2);

            System.out.println("\nTesting User: " + username + " | Expected: " + expected);

            app.launchBrowser();
            app.openApplication();

            String actualResult = app.performLogin(username, password);
            System.out.println("Actual Result: " + actualResult);

            excel.setCellData(i, 3, actualResult); // write result to Excel

            app.closeBrowser();
        }

        excel.closeWorkbook();
        System.out.println("--- Test Execution Completed ---");
    }
}
```

**OUTPUT:**

A screenshot of Microsoft Excel showing a spreadsheet titled 'C11'. The spreadsheet contains three rows of data:

	A	B	C	D	E	F	G	H
1	Username	Password	Expected					
2	student	Password123	Valid					
3	user1	12345	Invalid					

A screenshot of the Eclipse IDE Console tab. The output shows the execution of a Java application named 'Test1'.

```
<terminated> Test1 [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (Nov 3, 2025, 2:25:28 PM -)
--- Starting Library Architecture Framework ---

Testing User: student | Expected: Valid
Actual Result: Valid

Testing User: user1 | Expected: Invalid
Actual Result: Invalid

--- Test Execution Completed ---
```

## **PRACTICAL NO: 15**

**Aim:** Perform Regression Testing.

### **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

**Calculator.java**

```
package regression;
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
    public int subtract(int a, int b) {
        return a - b;
    }
    public int multiply(int a, int b) {
        return a * b;
    }
}
```

**RegressionTest.java**

```
package regression;
import org.testng.Assert;
import org.testng.annotations.Test;
public class RegressionTest {

    Calculator calc = new Calculator();
    @Test(priority = 1)
    public void testAddition() {
        Assert.assertEquals(calc.add(10, 5), 15);
        System.out.println("Addition test passed!");
    }

    @Test(priority = 2)
    public void testSubtraction() {
        Assert.assertEquals(calc.subtract(10, 5), 5);
        System.out.println("Subtraction test passed!");
    }

    // New test case added after modification
    @Test(priority = 3)
    public void testMultiplication() {
        Assert.assertEquals(calc.multiply(10, 5), 50);
        System.out.println("Multiplication test passed!");
    }
}
```

**OUTPUT:**

```
Console > Servers Results of running class prcat12
[RemoteTestNG] detected TestNG version 7.4.0
Addition test passed!
Subtraction test passed!
Multiplication test passed!
PASSED: testAddition
PASSED: testSubtraction
PASSED: testMultiplication
=====
Default test
Tests run: 3, Failures: 0, Failures 0, Skips: 0
```