

G) TestNG using xml.

Code :

```
package testcase;

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();

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

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

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

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

- Title Bar:** Aj - SeleniumFramework/src/test/java/utilities/ReadPropertyFile.java - Eclipse IDE
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbars:** Standard toolbar with icons for New, Open, Save, Cut, Copy, Paste, Find, etc.
- Left Sidebar (Project Explorer):**
 - Java Project: SeleniumFramework
 - Source Folders: abc, Abcd, generics student, java, practicalsS, Sanskriti(93)
 - Test Folders: SeleniumFramework, src/test/java
 - testCase: MyFirstTest.java
 - utilities: ReadPropertyFile.java
 - Resources: src/test/resources
 - Libraries: JRE System Library [JavaSE-1.7], Maven Dependencies
 - Source Folders: src
 - main
 - test
 - testCase: MyFirstTest.java
 - utilities
 - resources
 - configfiles: config.properties
 - target, pom.xml, shruuti, sqta_20
- Central Area (Code Editor):** The code for `ReadPropertyFile.java` is displayed:

```
1 package utilities;
2
3 import java.io.FileInputStream;
4 import java.io.IOException;
5 import java.util.Properties;
6
7 public class ReadPropertyFile {
8     public static Properties getProperties() throws IOException {
9         Properties prop = new Properties();
10        try (FileInputStream fis = new FileInputStream("src/test/resources/configfiles/config.properties")) {
11            prop.load(fis);
12        }
13        return prop;
14    }
15
16
17 }
```
- Bottom Area (Console):** The console output shows the execution of `MyFirstTest.java`.

```
<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]
Nov 03, 2025 2:46:45 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 141
Nov 03, 2025 2:46:45 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: 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
Opened URL: https://www.zoho.com/
```

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

- Title Bar:** Aj - SeleniumFramework/src/test/resources/configfiles/config.properties - Eclipse IDE
- Menu Bar:** File Edit Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and run.
- Project Explorer (Left):** Shows the project structure:
 - abc
 - Abcd
 - generics student
 - java
 - practicalsS
 - Sanskriti(93)
 - SeleniumFramework
 - src/main/java
 - src/main/resources
 - src/test/java
 - testcase
 - MyFirstTest.java
 - utilities
 - ReadPropertyFile.java
 - src/test/resources
 - JRE System Library [JavaSE-1.7]
 - Maven Dependencies
 - src
 - main
 - test
 - java
 - MyFirstTest.java
 - utilities
 - resources
 - configfiles
 - config.properties
 - target
 - pom.xml
 - shruuti
 - stqa_20
- Code Editor (Center):** Displays the content of config.properties.

```
1 browser=chrome
2 testurl=https://www.zoho.com/
3 login_id=your.email@example.com
4 password=yourpassword
5
```
- Console (Bottom):** Shows the terminal output of the Java application.

```
<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]
Nov 03, 2025 2:46:45 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: 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]
Nov 03, 2025 2:46:45 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: 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
Opened URL: https://www.zoho.com/
```

The screenshot shows the Zoho Cloud Software Suite homepage. At the top, there's a navigation bar with links for Products, Enterprise, Customers, Partners, and Resources. On the right side of the header are search, language (EN), sign-in, and sign-up buttons. The main content area features a large headline: "Your life's work, powered by our life's work". Below this, a paragraph of text reads: "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 red "GET STARTED FOR FREE" button is located below the text. To the right, there's a sidebar titled "FEATURED APPS" with icons and names for CRM, Mail, Books, and People.

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

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.

GET STARTED FOR FREE >

FEATURED APPS

- CRM**
Comprehensive CRM platform >
for customer-facing teams.
- Mail**
Secure email service for teams >
of all sizes.
- Books**
Powerful accounting platform >
for growing businesses.
- People**

Practical No : 14

Aim: Demonstrate Library Architecture Framework

TestExecution.java

```
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();

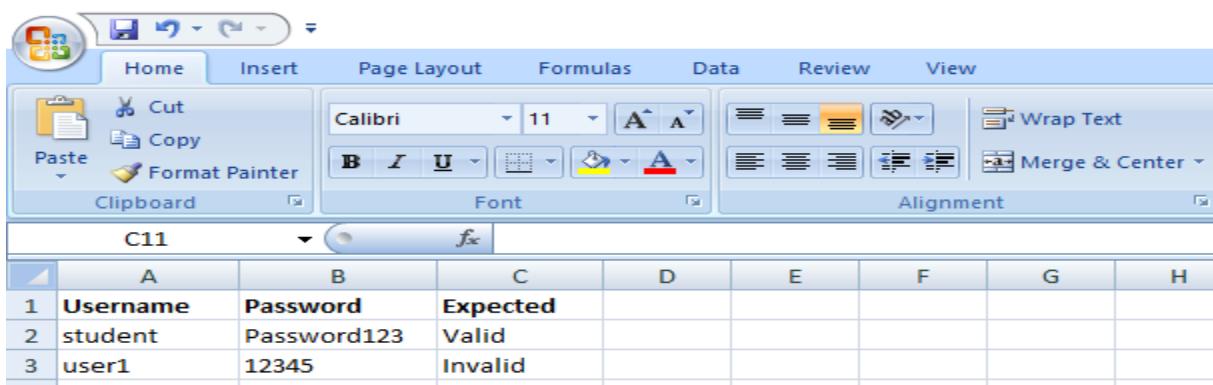
        System.out.println("--- Starting Library Architecture Framework ---");

        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 ---");
    }
}
```

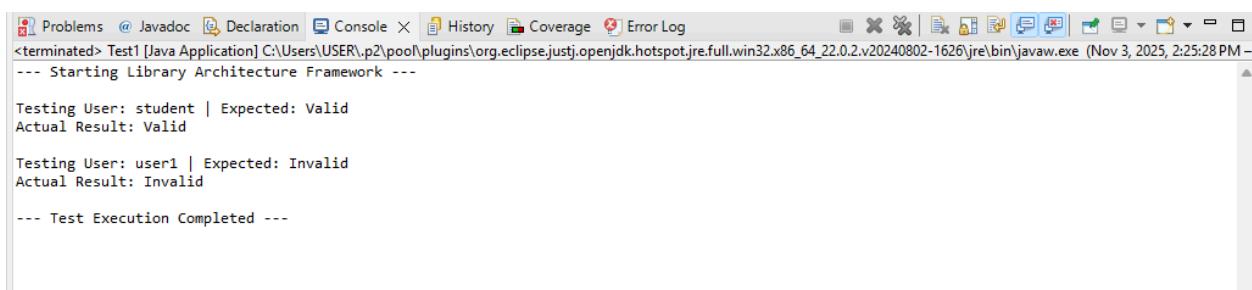
Output -

Name : Rupali Jadhav

Roll No. : 87



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



```
Problems @ Javadoc Declaration Console X History Coverage Error Log
<terminated> Test1 [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.jdt.core\org.eclipse.jdt.core_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

Program :

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 -



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the results of a TestNG test run. The output text is as follows:

```
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
```

Name : Rupali Jadhav

Roll No. : 87