Capstone Project

Project Name: **Human Resource Management**

Submitted By: Ritik Prajapat [StarAgile Student Trainee]

Submitted To: [StarAgile Consulting Pvt Ltd]

GitHub link:
https://github.com/ritikprajapat8085/AutomationProjects/tree/OHRM

Table of Contents

1. Project Overview

OrangeHRM is an open-source Human Resource Management tool used globally to manage employee records, leave, attendance, and recruitment. In this capstone project, I automated critical HR functionalities using Selenium, Java, and TestNG to reduce manual effort and enhance test reliability.

Human Resource Management is a product that offers services like below:

- Adding new employees,
- Modifying existing employees
- Deleting existing employees
- Searching employees
- Raising a request for hiring a new employee
- Approving a new request
- Submitting timesheet
- Approving timesheet, etc.

2. Objective

- To validate the core functionalities of the OrangeHRM application, like user login/logout, adding employees, managing leaves, etc., and ensure they are working as expected.
- Implement Excel-based data-driven testing to verify login and other functionality against multiple data sets, including valid and invalid combinations.
- Design and implement a modular, scalable, and maintainable test framework using the Page Object Model design pattern.
- Automate the execution of test cases through TestNG and generate detailed execution reports using ExtentReports to track the status of each test case.

3. Scope of the Project:

The project covers UI-based functional testing of the OrangeHRM application modules like Login, Add Employee, Search Employee, and Logout. It includes positive and negative test cases, Excel data-driven testing, and result reporting.

1 . Login/logout

2. Admin

2.1. User Management

4. Tools &Technologies Used:

Tool / Tech	Purpose
● Java	Programming Language
● Selenium WebDriver	Automation Testing Tool
● TestNG	Test Management and Assertions
● Apache POI	Reading Excel files
● ExtentReports	Report Generation
● Eclipse / IntelliJ	IDE
● Maven	Project Build Tool
● GitHub	Version Control
● OrangeHRM	Application Under Test

5. Test Case Scenarios:

Scenario 1:

Automate Orange HRM login and logout with 5 different datasets, including valid and invalid datasets.

1. Save the dataset in an Excel file
2. Write a script to read data from Excel
3. Prepare a script for Login and logout
4. Perform assertion for a valid data set (use Username: Admin and Password:admin123) test case should pass, and an invalid data set (other than the given data) test case should fail.
5. Capture a Screenshot for every login functionality
6. Generate an Extent Report for the same.

Scenario 2:

Create a Page Object Model for two pages Login Page and the Admin Pageand write an automation script for the Login functionality and Admin search feature.

Create a Page and Class for the Login Page and automate the functionality of Orange HRM login for Valid Test Data: (username: Admin and Password: admin123).

Create a Page and a class for Admin where you can prepare 4 important test cases:

1. Create a test case to get all 12 options from the left side menu and print the count, which should be 12 from that list. Click on Admin, and then the Admin page will open.

2. Create a test case for the search For Existing Employee search by username (): Here, send the username Admin to the username text box and click on the search button and display the total number of records found and refresh the page.

3. Create a test case for search For Existing Employee search ByUserRole(): here, automate the dropdown and select Role Admin and click on the search button and display the total record found and refresh the page

4. Create a test case for the search For Existing Employee search ByUserStatus(): here, automate the dropdown and select status Enabled or Disabled, then click on the search button and display the total record found.

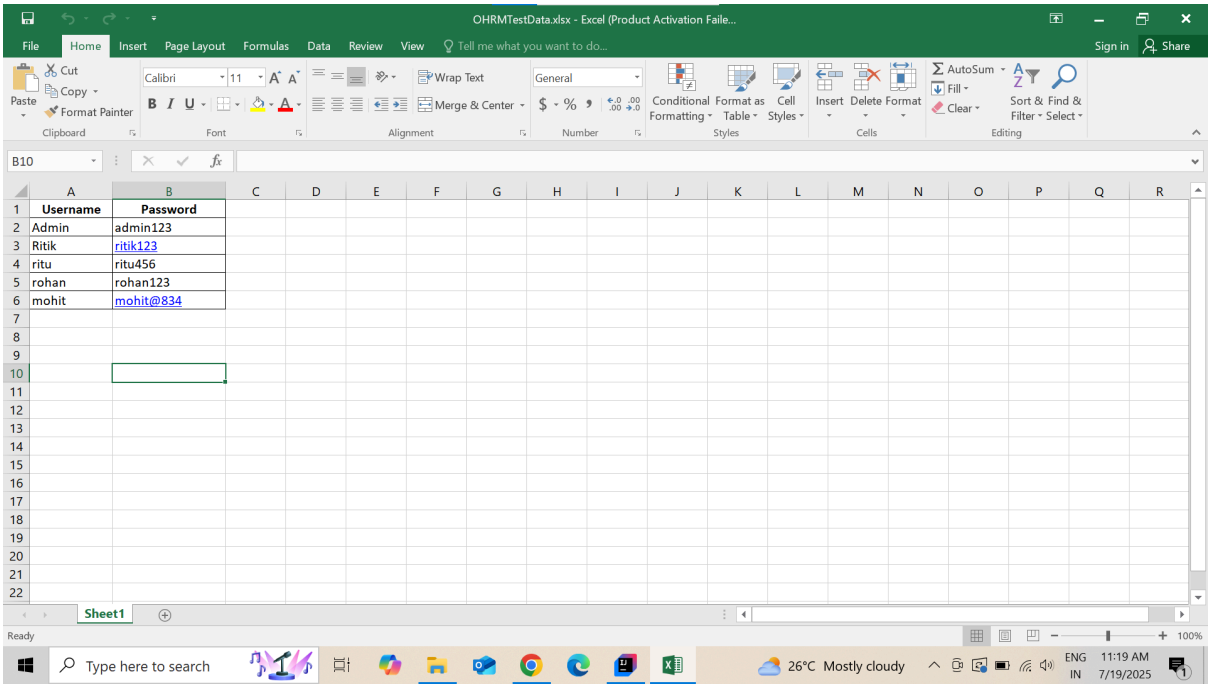
Note: In the Second scenario, use Selenium TestNG and design a Page Object Model where you can implement other Object Oriented Principles.

6. Testcase Design & Execution:

Scenario 1:

1. Save the dataset in an Excel file

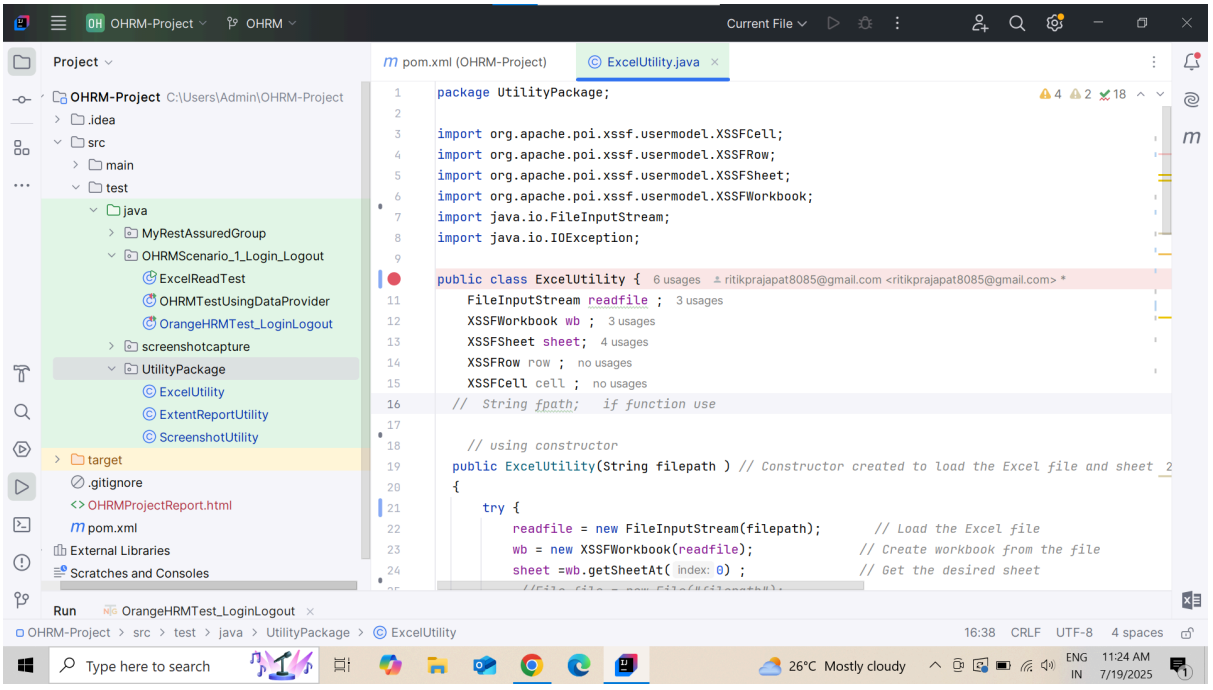
Test data is maintained in an Excel file. The Apache POI library is used to read multiple sets of data, including usernames and passwords.



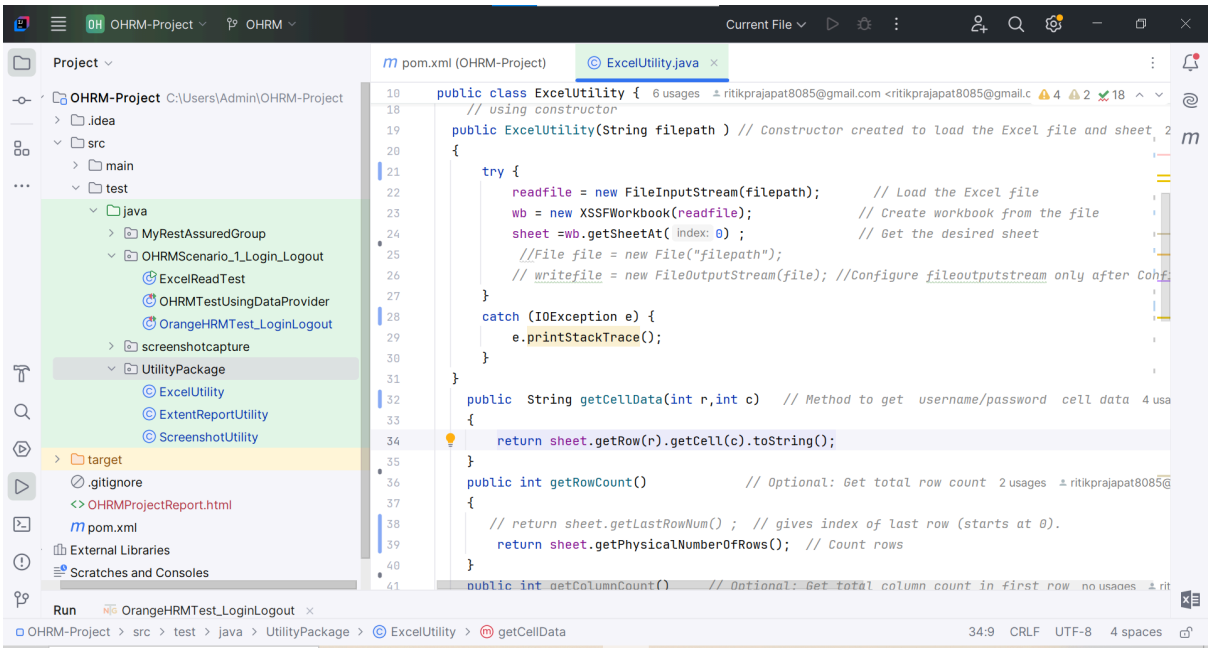
Username	Password
Admin	admin123
Ritik	ritik123
ritu	ritu456
rohan	rohan123
mohit	mohit@834

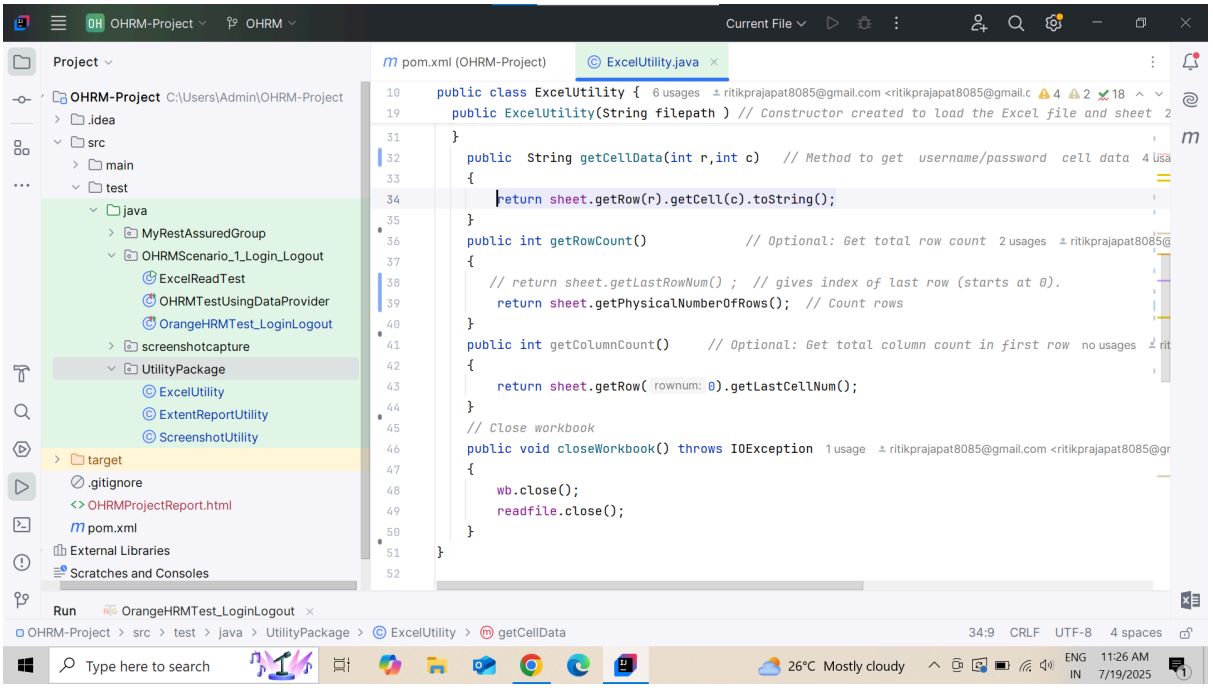
2. Write a script to read data from Excel

I have created a separate Excel utility class.

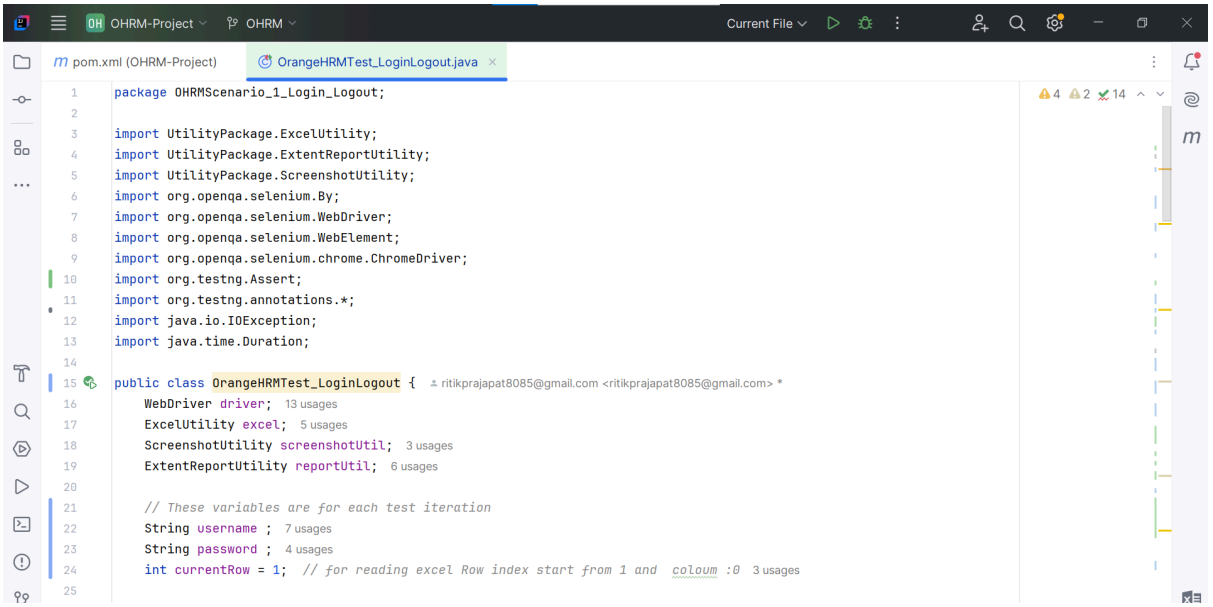



```
package UtilityPackage;

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 java.io.FileInputStream;
import java.io.IOException;

public class ExcelUtility {
    FileInputStream readfile ;
    XSSFWorkbook wb ;
    XSSFSheet sheet;
    XSSFRow row ;
    XSSFCell cell ;
  //  String fpath;    if function use

    // using constructor
  public ExcelUtility(String filepath ) // Constructor created to load the Excel file and sheet
  {
      try {
          readfile = new FileInputStream(filepath);       // Load the Excel file
          wb = new XSSFWorkbook(readfile);                // Create workbook from the file
          sheet =wb.getSheetAt( index: 0) ;               // Get the desired sheet
```

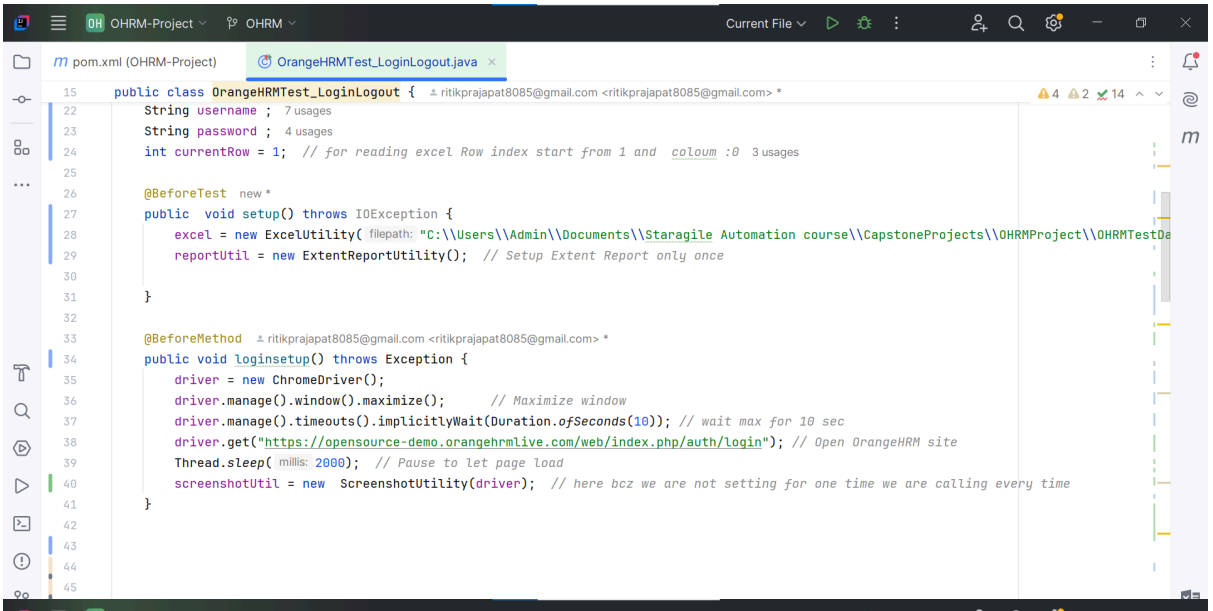

OHRM-Project
OHRM
Current File
Project
pom.xml (OHRM-Project)
ExcelUtility.java
OHRM-Project C:\Users\Admin\OHRM-Project
.idea
src
main
test
java
MyRestAssuredGroup
OHRMScenario_1_Login_Logout
ExcelReadTest
OHRMTestUsingDataProvider
OrangeHRMTest_LoginLogout
screenshotcapture
UtilityPackage
ExcelUtility
ExtentReportUtility
ScreenshotUtility
target
.gitignore
OHRMProjectReport.html
pom.xml
External Libraries
Scratches and Consoles
public class ExcelUtility { 6 usages ritikprajapat8085@gmail.com <ritikprajapat8085@gmail.c 4 2 18
// using constructor
public ExcelUtility(String filepath) // Constructor created to load the Excel file and sheet
{
try {
readfile = new FileInputStream(filepath); // Load the Excel file
wb = new XSSFWorkbook(readfile); // Create workbook from the file
sheet =wb.getSheetAt(index: 0) ; // Get the desired sheet
//File file = new File("filepath");
// writefile = new FileOutputStream(file); //Configure fileoutputstream only after Conf
}
catch (IOException e) {
e.printStackTrace();
}
}
public String getCellData(int r,int c) // Method to get username/password cell data 4 usa
{
return sheet.getRow(r).getCell(c).toString();
}
public int getRowCount() // Optional: Get total row count 2 usages ritikprajapat8085@
{
// return sheet.getLastRowNum() ; // gives index of last row (starts at 0).
return sheet.getPhysicalNumberOfRows(); // Count rows
}
public int getColumnCount() // Optional: Get total column count in first row no usages rit
Run
OrangeHRMTest_LoginLogout
OHRM-Project > src > test > java > UtilityPackage > ExcelUtility > getCellData
34:9 CRLF UTF-8 4 spaces


OHRM-Project
OHRM
Current File
Project
pom.xml (OHRM-Project)
ExcelUtility.java
OHRM-Project C:\Users\Admin\OHRM-Project
.idea
src
main
test
java
MyRestAssuredGroup
OHRMScenario_1_Login_Logout
ExcelReadTest
OHRMTestUsingDataProvider
OrangeHRMTest_LoginLogout
screenshotcapture
UtilityPackage
ExcelUtility
ExtentReportUtility
ScreenshotUtility
target
.gitignore
OHRMProjectReport.html
pom.xml
External Libraries
Scratches and Consoles
public class ExcelUtility { 6 usages ritikprajapat8085@gmail.com <ritikprajapat8085@gmail.c 4 2 18
public ExcelUtility(String filepath) // Constructor created to load the Excel file and sheet
}
public String getCellData(int r,int c) // Method to get username/password cell data 4 usa
{
return sheet.getRow(r).getCell(c).toString();
}
public int getRowCount() // Optional: Get total row count 2 usages ritikprajapat8085@
{
// return sheet.getLastRowNum() ; // gives index of last row (starts at 0).
return sheet.getPhysicalNumberOfRows(); // Count rows
}
public int getColumnCount() // Optional: Get total column count in first row no usages rit
{
return sheet.getRow(rownum: 0).getLastCellNum();
}
// Close workbook
public void closeWorkbook() throws IOException 1 usage ritikprajapat8085@gmail.com <ritikprajapat8085@gr
{
wb.close();
readfile.close();
}
}
Run
OrangeHRMTest_LoginLogout
OHRM-Project > src > test > java > UtilityPackage > ExcelUtility > getCellData
34:9 CRLF UTF-8 4 spaces
Type here to search
26°C Mostly cloudy
ENG IN
11:26 AM 7/19/2025

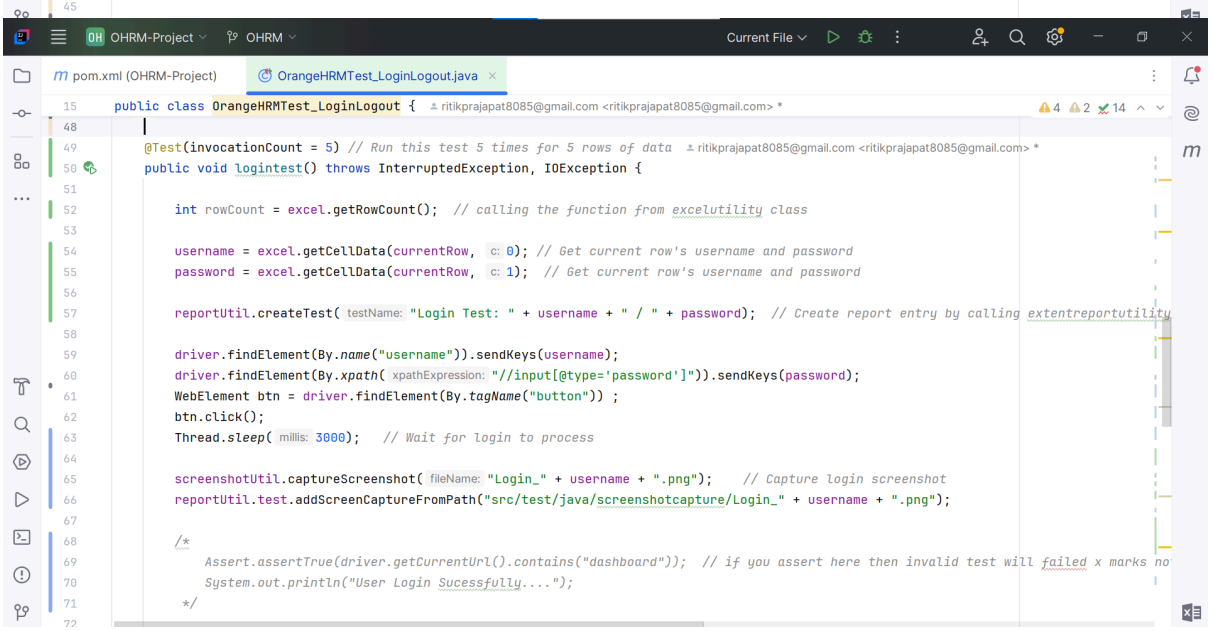
3. Prepare a script for Login and logout



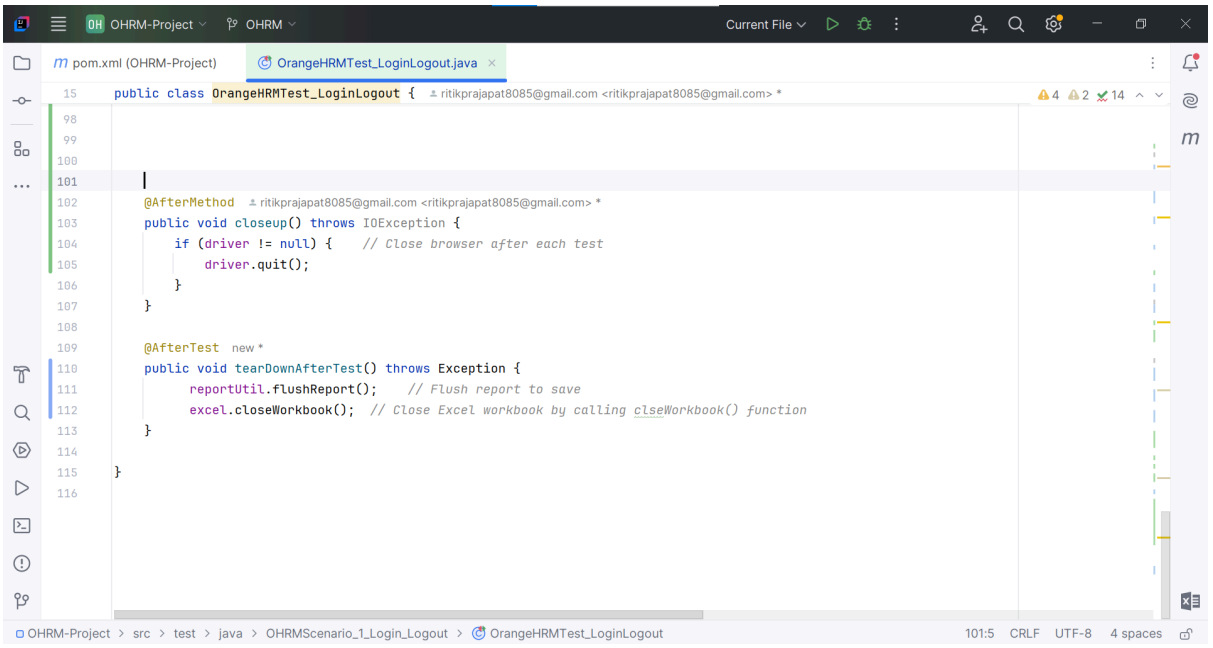
```java
package OHRMScenario_1_Login_Logout;

import UtilityPackage.ExcelUtility;
import UtilityPackage.ExtentReportUtility;
import UtilityPackage.ScreenshotUtility;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.*;
import java.io.IOException;
import java.time.Duration;

public class OrangeHRMTest_LoginLogout {
    WebDriver driver;
    ExcelUtility excel;
    ScreenshotUtility screenshotUtil;
    ExtentReportUtility reportUtil;

    // These variables are for each test iteration
    String username ;
    String password ;
    int currentRow = 1;  // for reading excel Row index start from 1 and  coloum :0
```



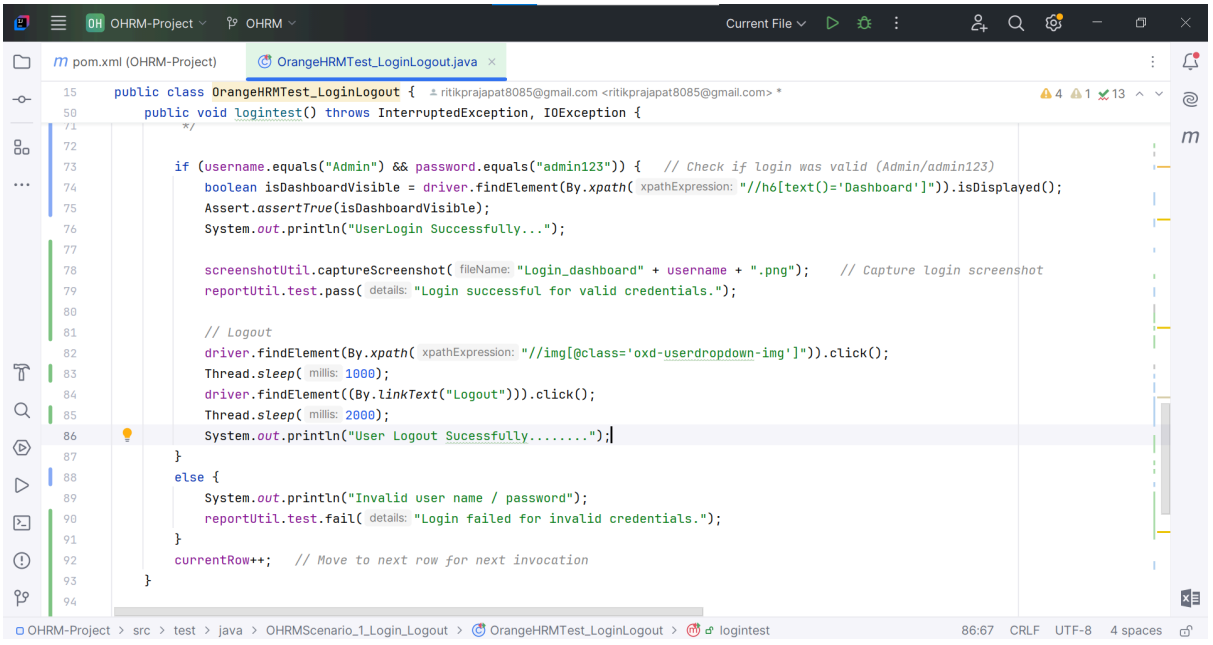
```java
public class OrangeHRMTest_LoginLogout {
    String username ;
    String password ;
    int currentRow = 1;  // for reading excel Row index start from 1 and  coloum :0

    @BeforeTest
    public  void setup() throws IOException {
        excel = new ExcelUtility( filepath: "C:\\Users\\Admin\\Documents\\Staragile Automation course\\CapstoneProjects\\OHRMProject\\OHRMTestDa
        reportUtil = new ExtentReportUtility();  // Setup Extent Report only once

    }

    @BeforeMethod
    public void loginsetup() throws Exception {
        driver = new ChromeDriver();
        driver.manage().window().maximize();      // Maximize window
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10)); // wait max for 10 sec
        driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login"); // Open OrangeHRM site
        Thread.sleep( millis: 2000);  // Pause to let page load
        screenshotUtil = new  ScreenshotUtility(driver);  // here bcz we are not setting for one time we are calling every time
    }
```



```java
public class OrangeHRMTest_LoginLogout {

    @Test(invocationCount = 5) // Run this test 5 times for 5 rows of data
    public void logintest() throws InterruptedException, IOException {

        int rowCount = excel.getRowCount();  // calling the function from excelutility class

        username = excel.getCellData(currentRow,  c: 0); // Get current row's username and password
        password = excel.getCellData(currentRow,  c: 1);  // Get current row's username and password

        reportUtil.createTest( testName: "Login Test: " + username + " / " + password);  // Create report entry by calling extentreportutility

        driver.findElement(By.name("username")).sendKeys(username);
        driver.findElement(By.xpath( xpathExpression: "//input[@type='password']")).sendKeys(password);
        WebElement btn = driver.findElement(By.tagName("button")) ;
        btn.click();
        Thread.sleep( millis: 3000);   // Wait for login to process

        screenshotUtil.captureScreenshot( fileName: "Login_" + username + ".png");    // Capture login screenshot
        reportUtil.test.addScreenCaptureFromPath("src/test/java/screenshotcapture/Login_" + username + ".png");

        /*
            Assert.assertTrue(driver.getCurrentUrl().contains("dashboard"));  // if you assert here then invalid test will failed x marks no
            System.out.println("User Login Sucessfully....");
         */
```

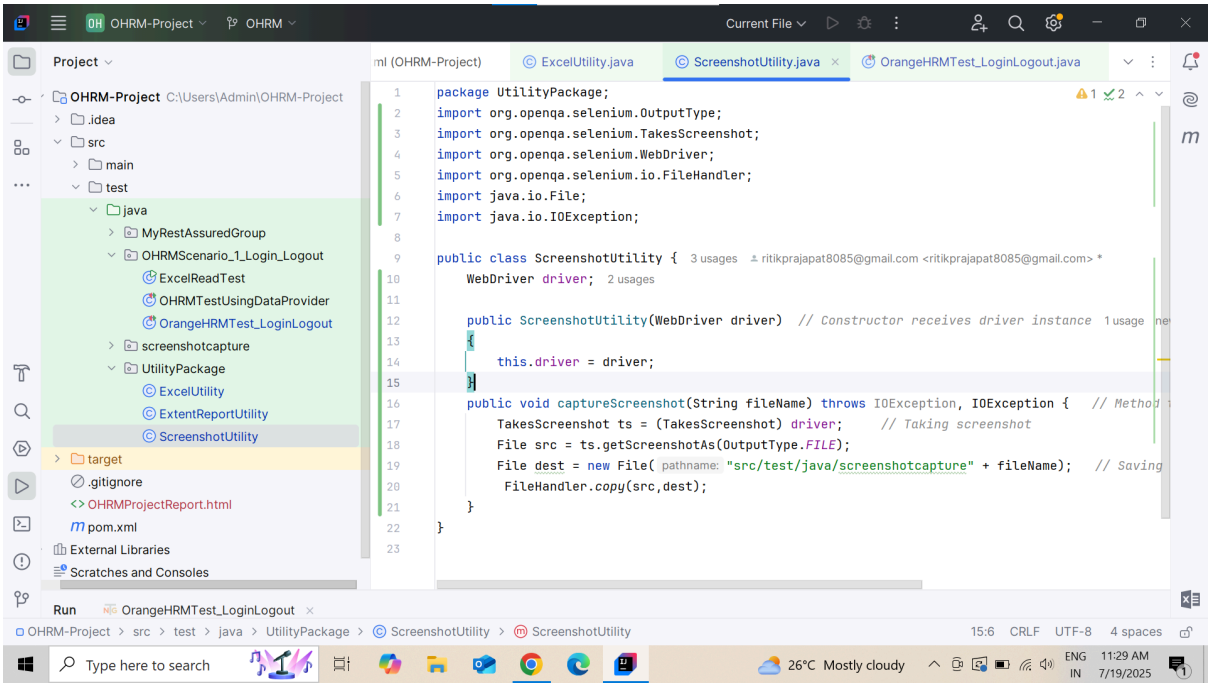


4. Perform assertion for a valid data set (use Username: Admin and Password:admin123) test case should pass, and an invalid data set (other than the given data) test case should fail.



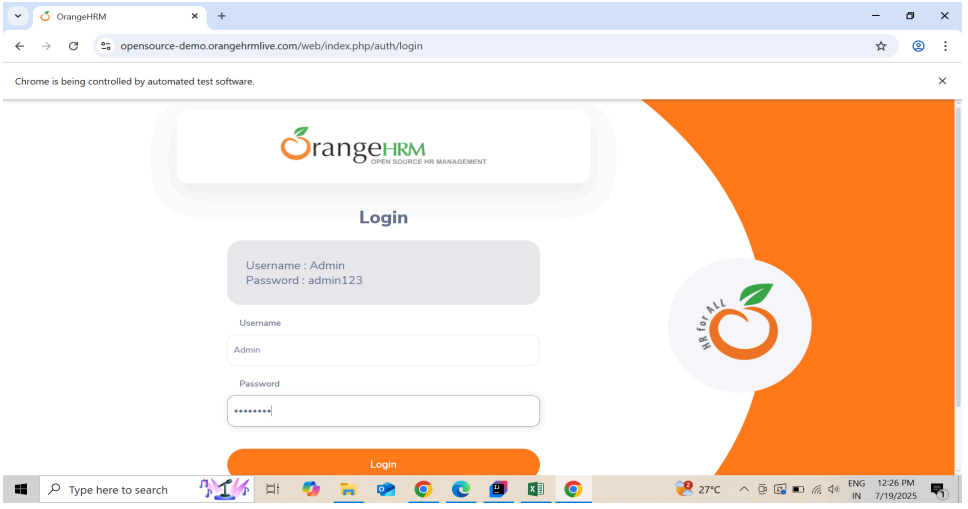
5. Capture a Screenshot for every login functionality

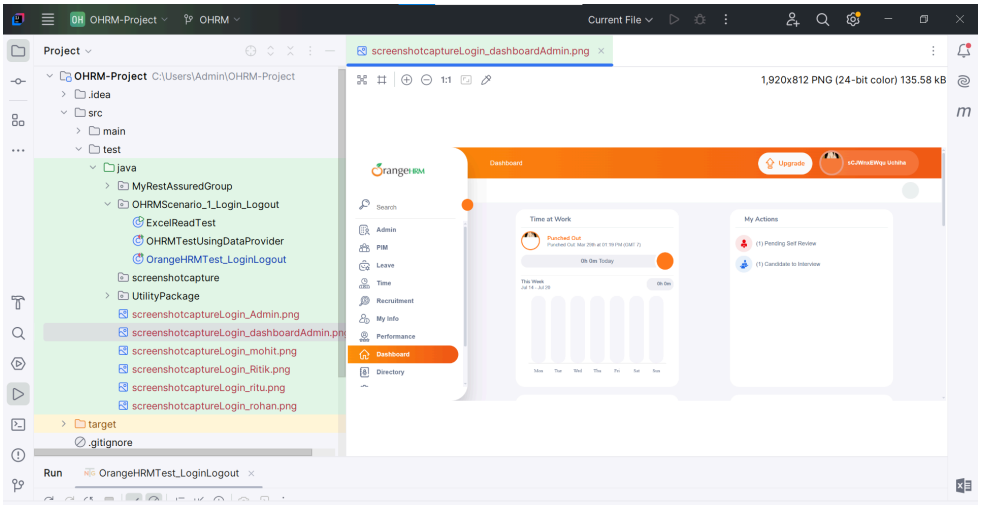
I have created a separate **Screenshot** utility class.



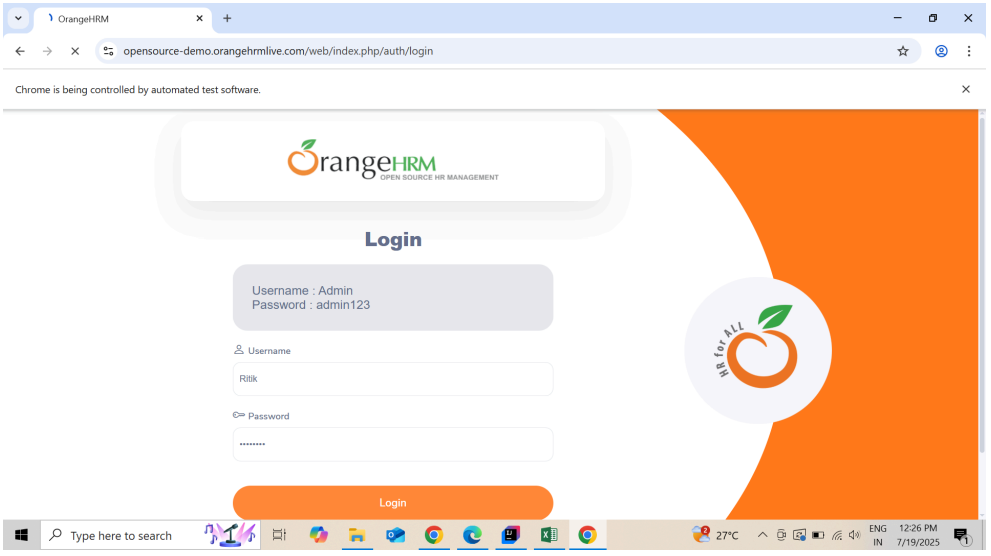
```java
package UtilityPackage;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.io.FileHandler;
import java.io.File;
import java.io.IOException;

public class ScreenshotUtility {
    WebDriver driver;

    public ScreenshotUtility(WebDriver driver) {
        this.driver = driver;
    }
    public void captureScreenshot(String fileName) throws IOException, IOException {
        TakesScreenshot ts = (TakesScreenshot) driver;
        File src = ts.getScreenshotAs(OutputType.FILE);
        File dest = new File("src/test/java/screenshotcapture" + fileName);
         FileHandler.copy(src,dest);
    }
}
```

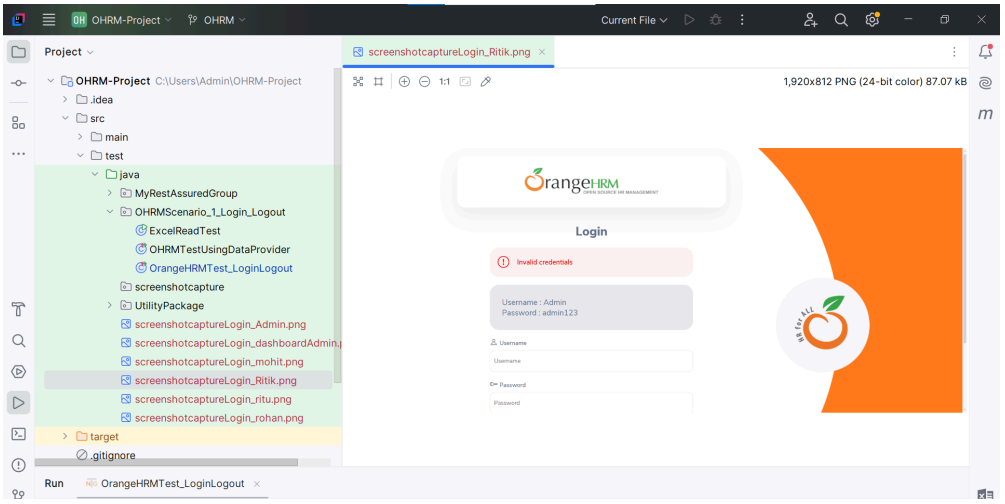
1. Username:Admin & Password: admin123



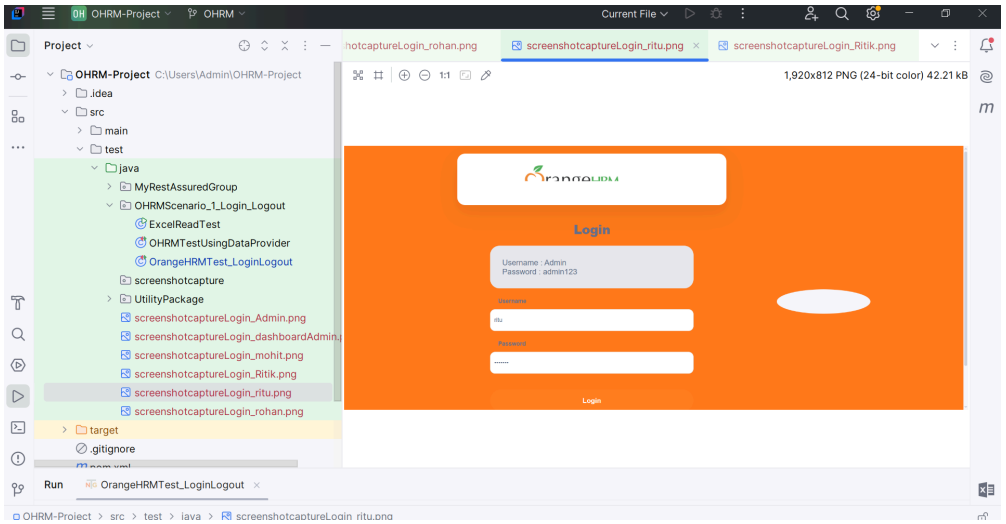


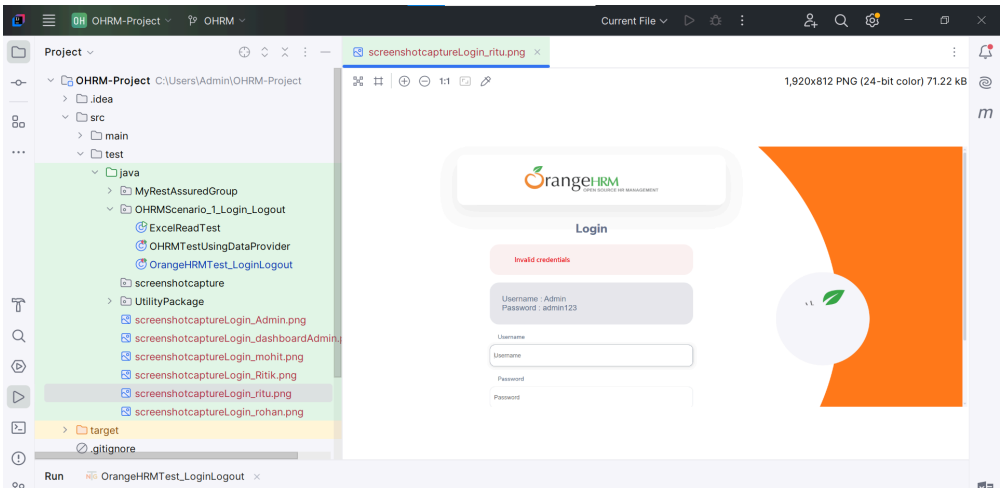
2. Username:ritik & Password: ritik123



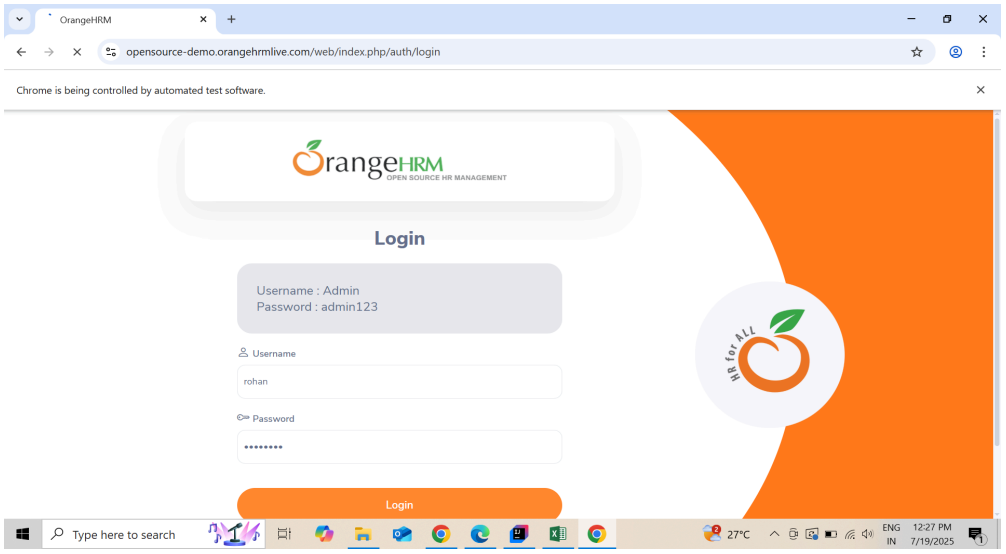


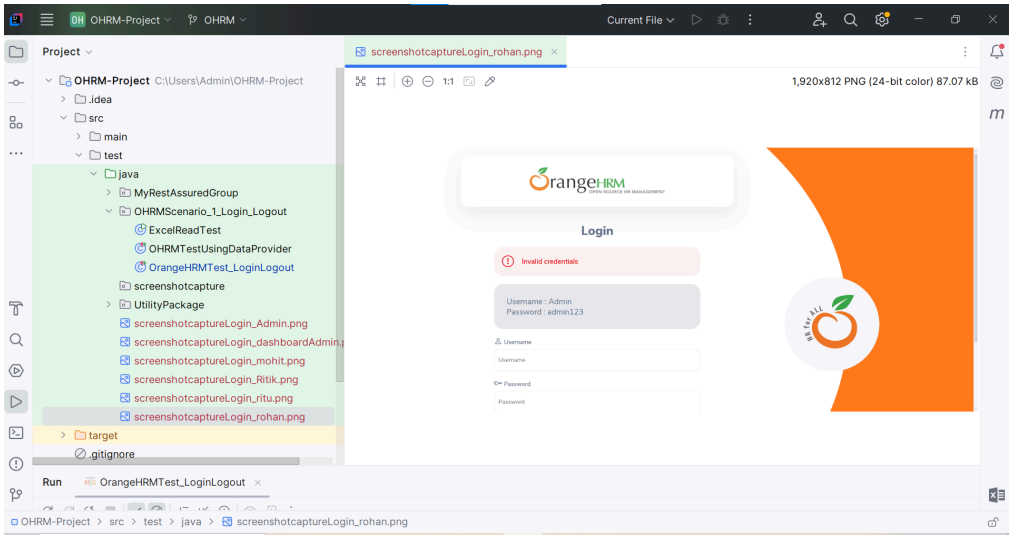
3. Username:ritu & Password:ritu456



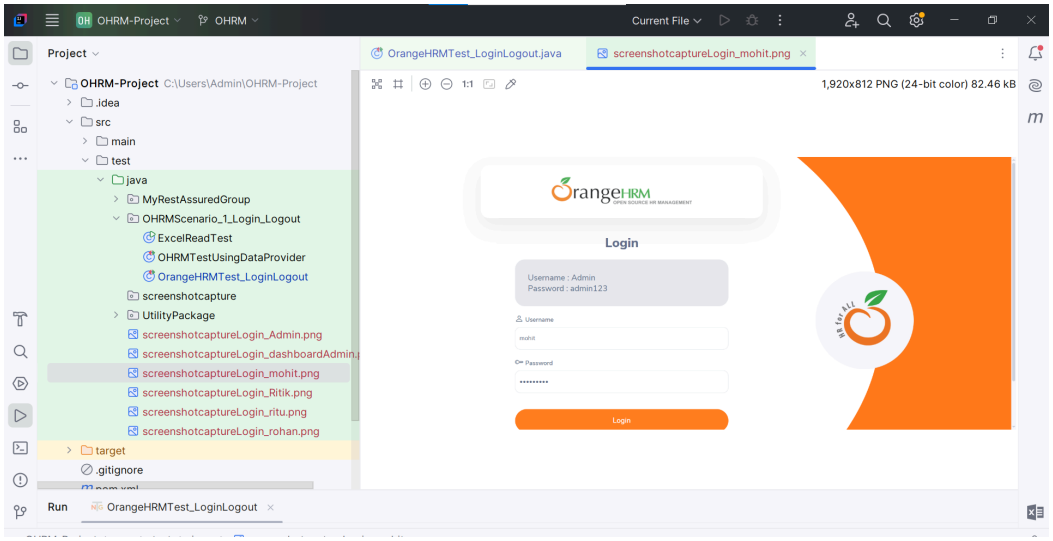


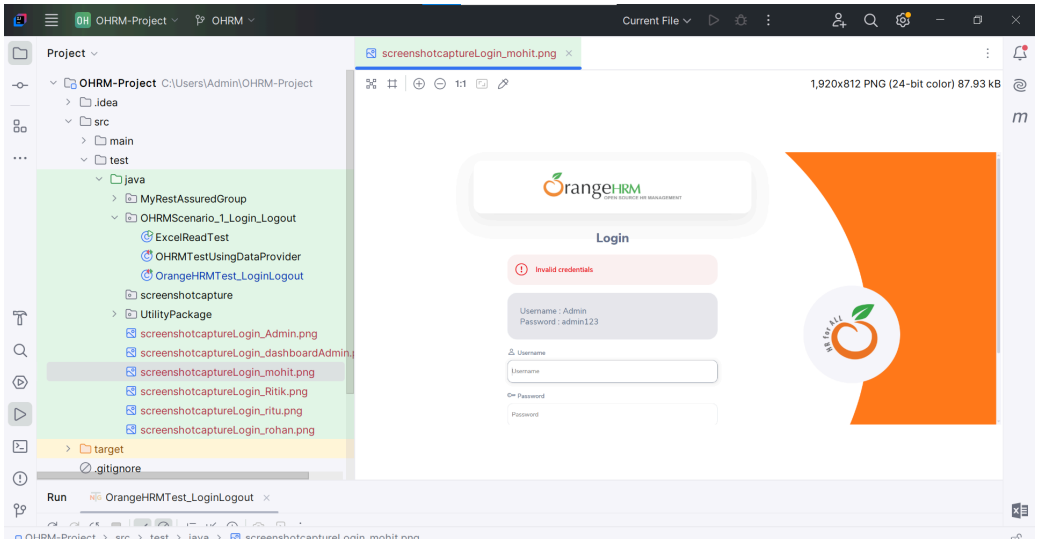
4. Username:rohan & Password: rohan123





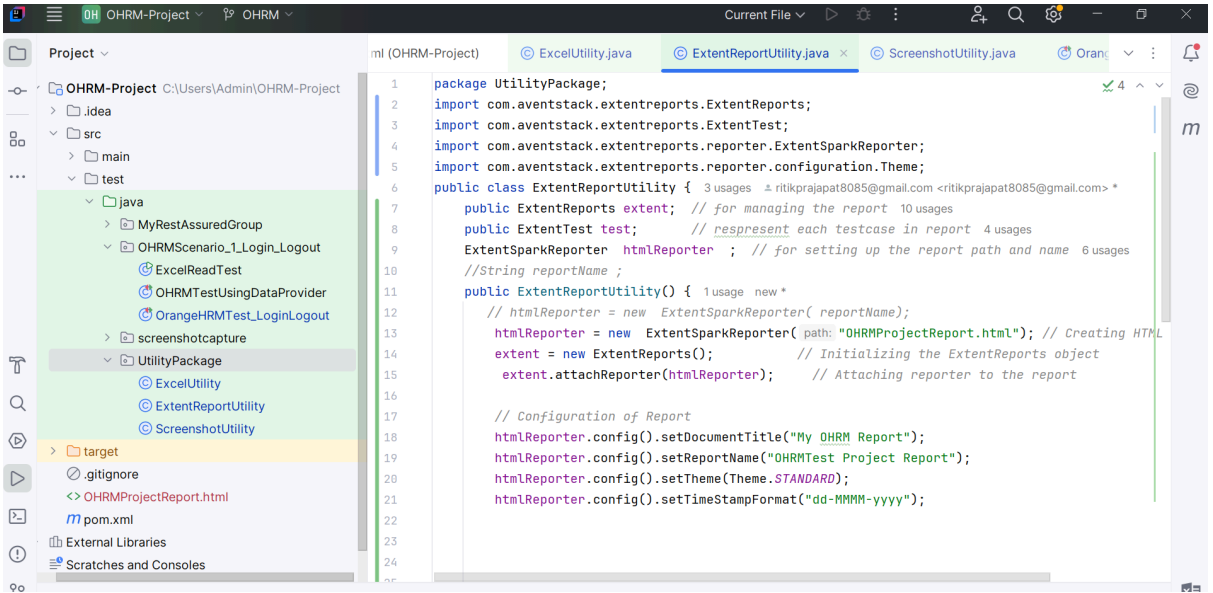
5. Username:mohit & Password: mohit@834

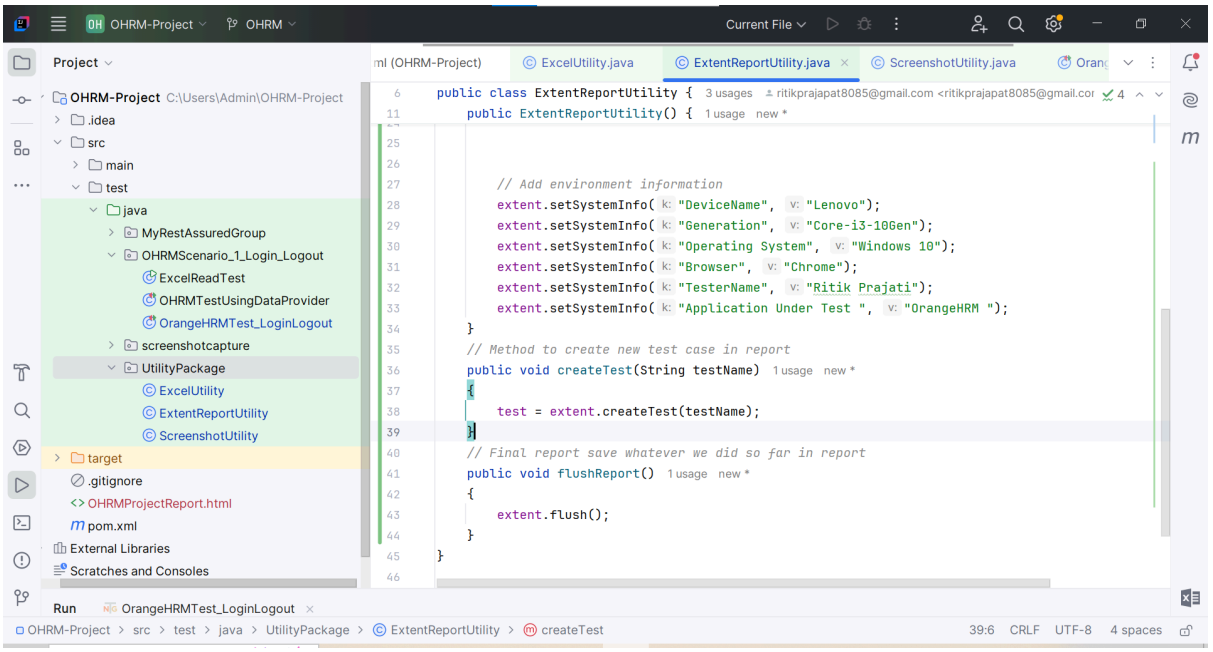




6. Generate an Extent Report for the same.

I have created a separate **Extent Report** utility class



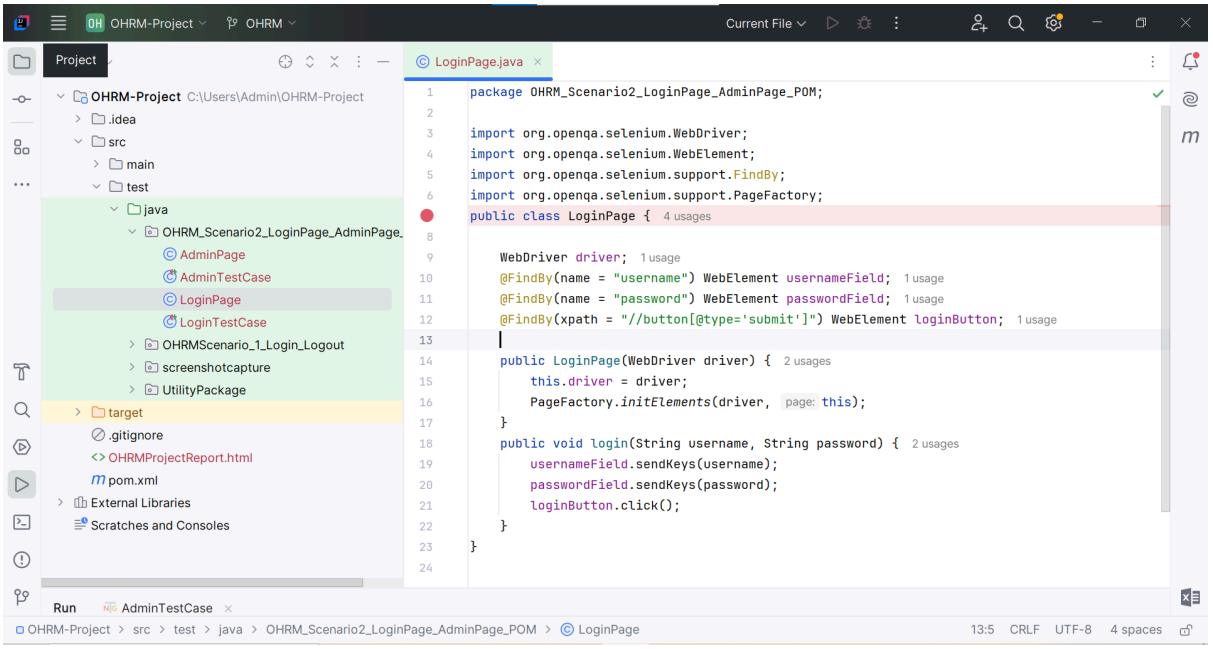


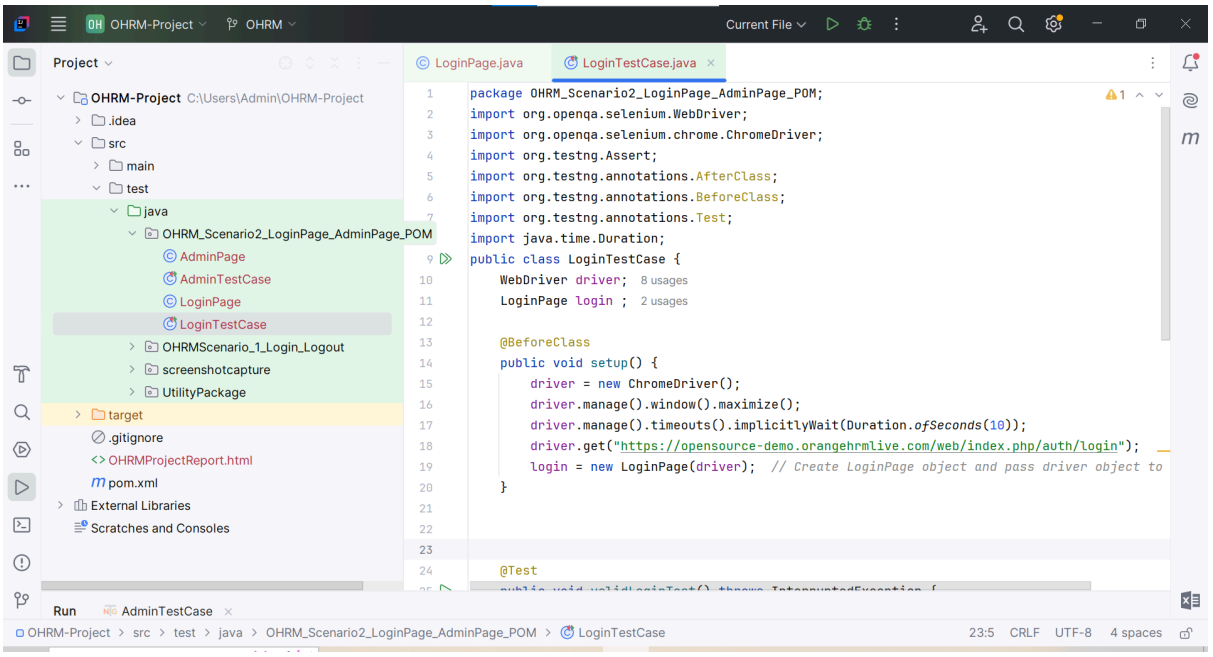
Scenario 2:

Create a Page Object Model for two pages **Login Page and the Admin Page,** and write an automation script for the Login functionality and Admin search feature.

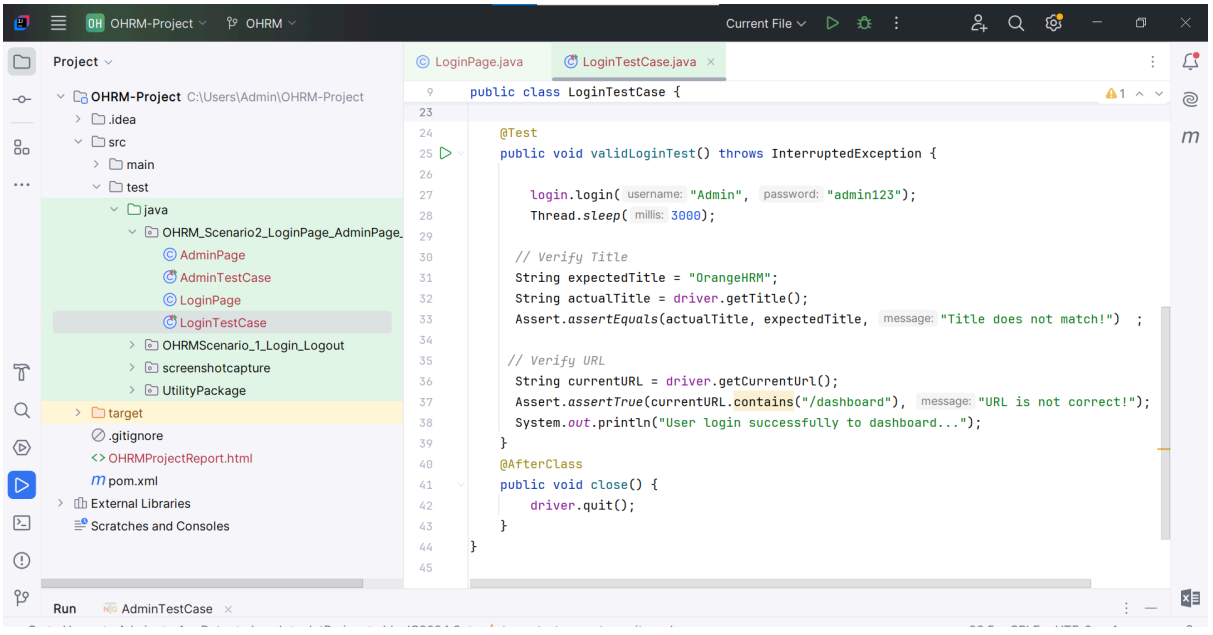
1. Create a Page and Class for the Login Page and automate the functionality of Orange HRM login for Valid Test Data: (username: Admin and Password: admin123).

LoginPage

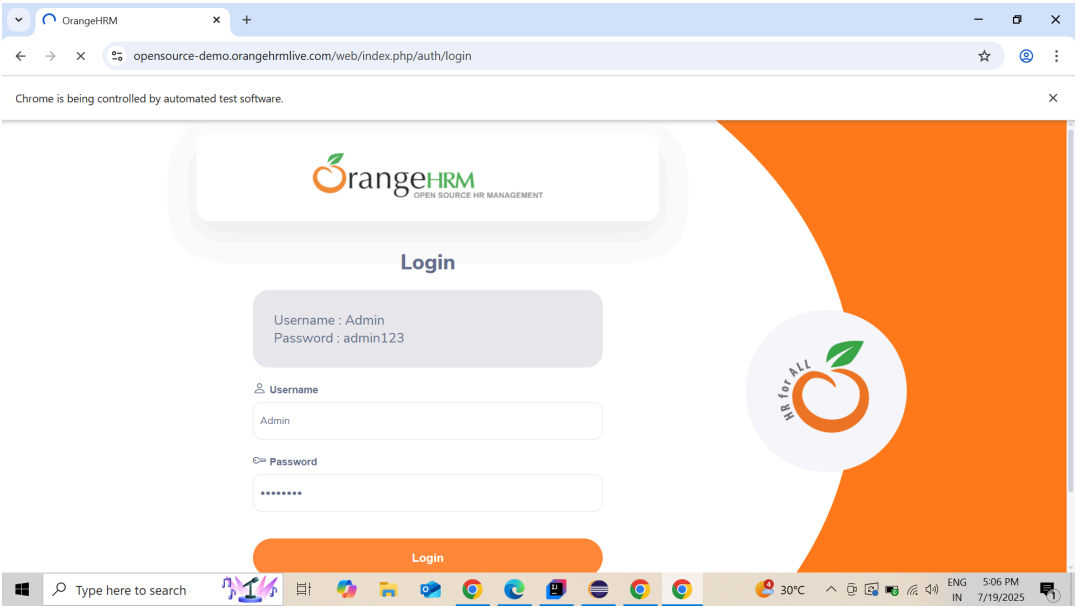


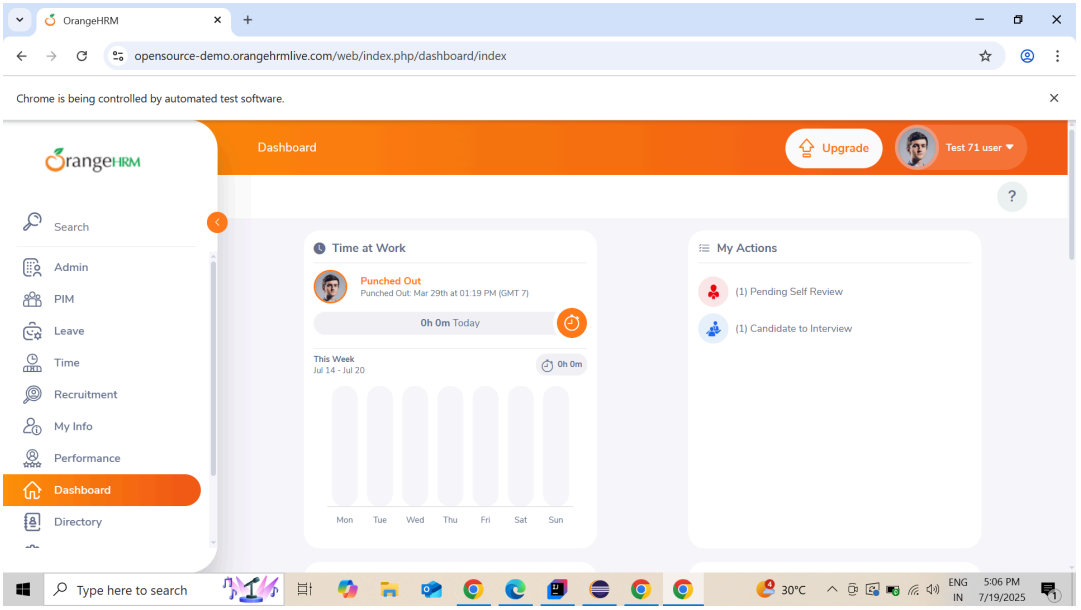


LoginTestCase



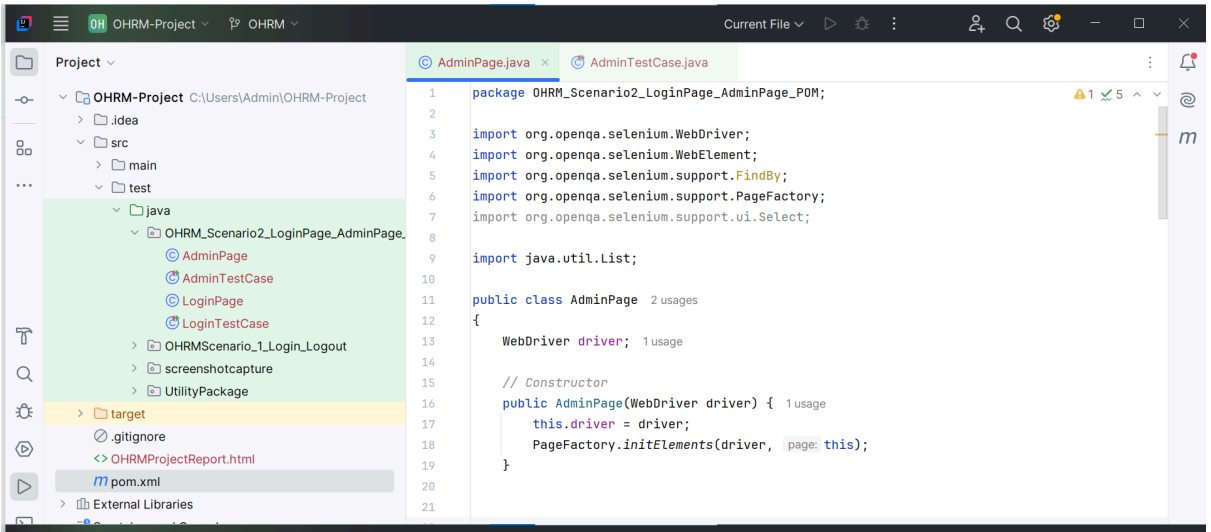
Screenshot

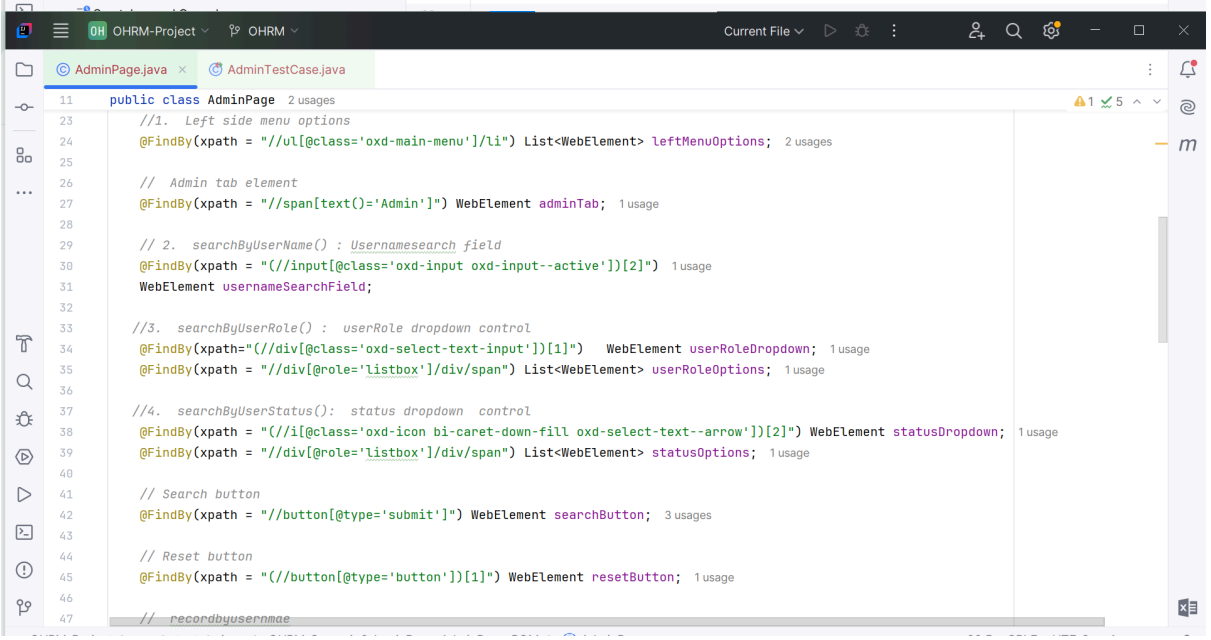

OrangeHRM
opensource-demo.orangehrmlive.com/web/index.php/auth/login
Chrome is being controlled by automated test software.
OrangeHRM
OPEN SOURCE HR MANAGEMENT
Login
Username : Admin
Password : admin123
Username
Admin
Password
Login
HR for ALL
Type here to search
30°C
ENG
IN
5:06 PM
7/19/2025

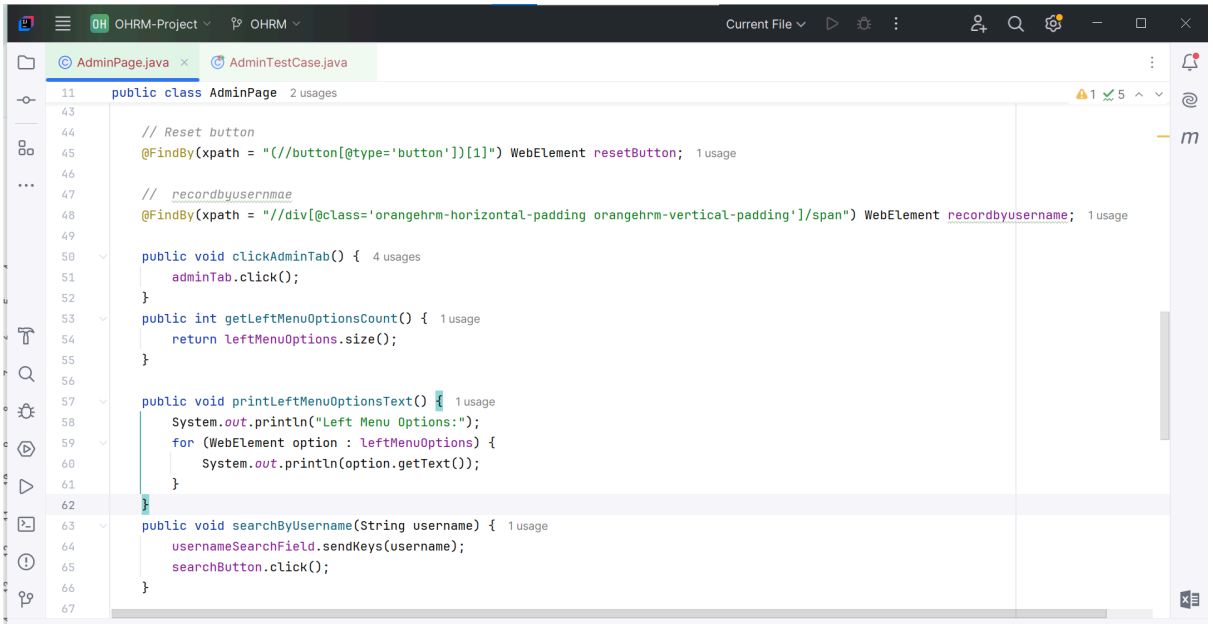

OrangeHRM
opensource-demo.orangehrmlive.com/web/index.php/dashboard/index
Chrome is being controlled by automated test software.
OrangeHRM
Dashboard
Upgrade
Test 71 user
Search
Admin
PIM
Leave
Time
Recruitment
My Info
Performance
Dashboard
Directory
Time at Work
Punched Out
Punched Out: Mar 29th at 01:19 PM (GMT 7)
0h 0m Today
This Week
Jul 14 - Jul 20
0h 0m
Mon
Tue
Wed
Thu
Fri
Sat
Sun
My Actions
(1) Pending Self Review
(1) Candidate to Interview
Type here to search
30°C
ENG
IN
5:06 PM
7/19/2025

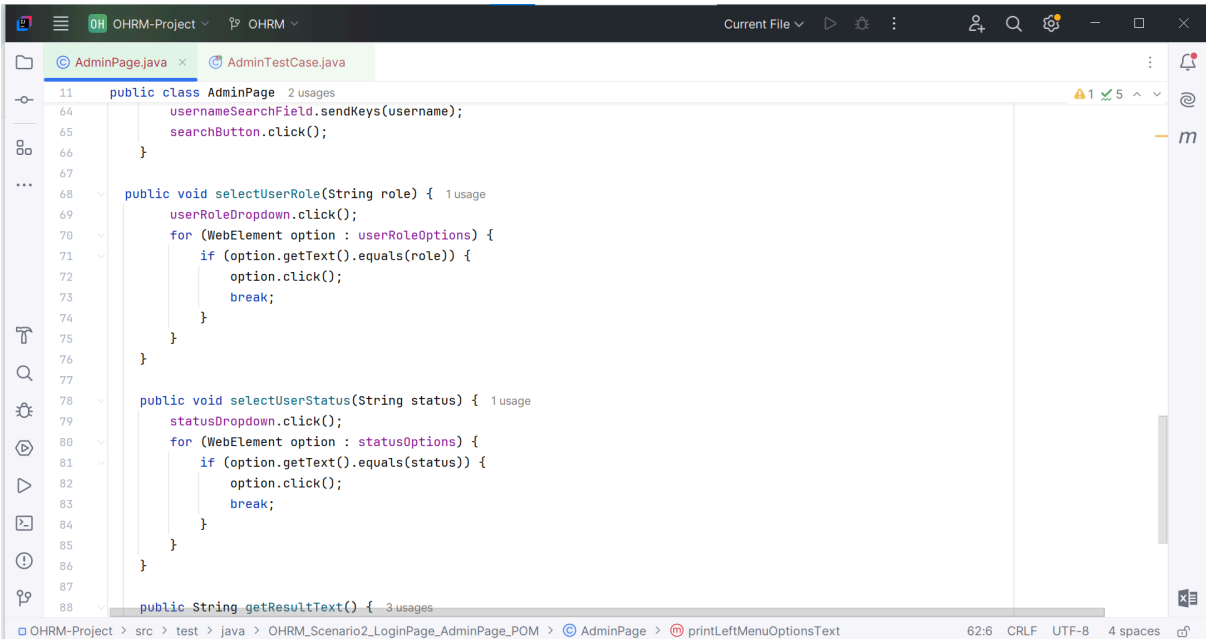
2. Create a Page and a class for Admin where you can prepare 4 important test cases:

AdminPage:

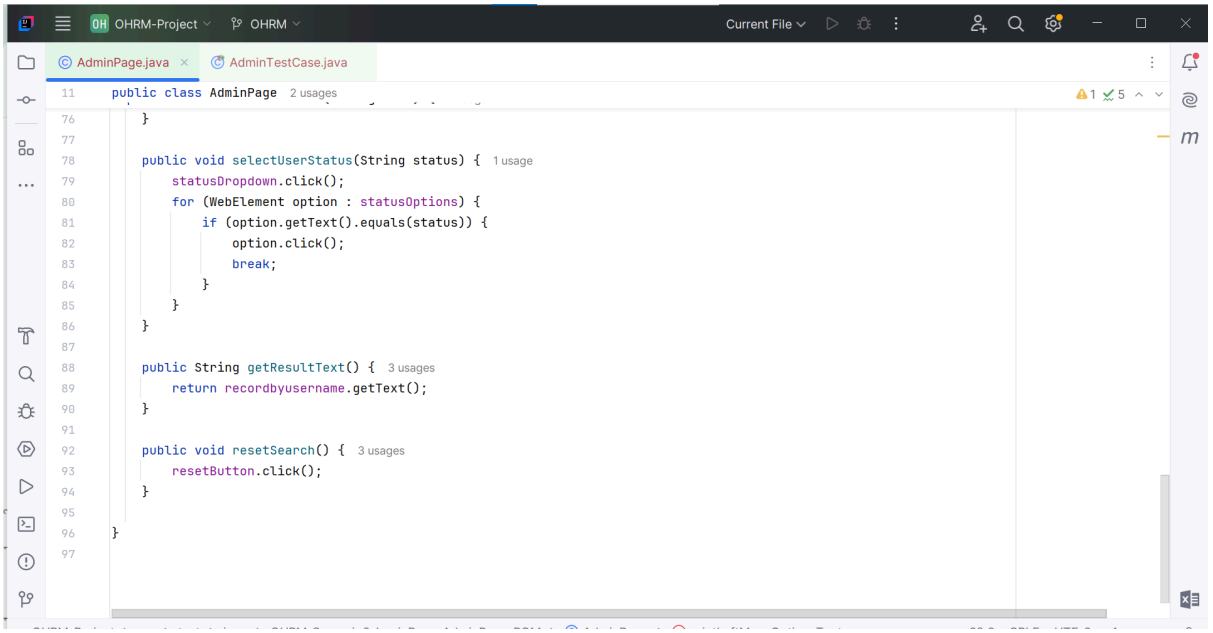






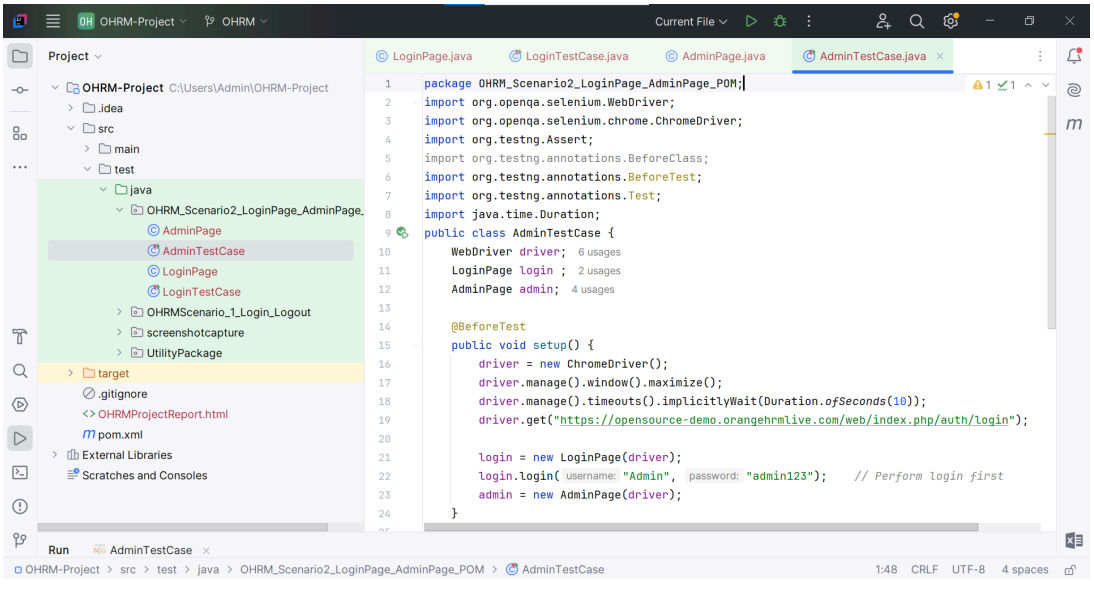


```
public class AdminPage  2 usages
        usernameSearchField.sendKeys(username);
        searchButton.click();
    }

public void selectUserRole(String role) {  1 usage
        userRoleDropdown.click();
        for (WebElement option : userRoleOptions) {
            if (option.getText().equals(role)) {
                option.click();
                break;
            }
        }
    }

    public void selectUserStatus(String status) {  1 usage
        statusDropdown.click();
        for (WebElement option : statusOptions) {
            if (option.getText().equals(status)) {
                option.click();
                break;
            }
        }
    }

    public String getResultText() {  3 usages
```



```
public class AdminPage  2 usages
    }

    public void selectUserStatus(String status) {  1 usage
        statusDropdown.click();
        for (WebElement option : statusOptions) {
            if (option.getText().equals(status)) {
                option.click();
                break;
            }
        }
    }

    public String getResultText() {  3 usages
        return recordbyusername.getText();
    }

    public void resetSearch() {  3 usages
        resetButton.click();
    }

}
```

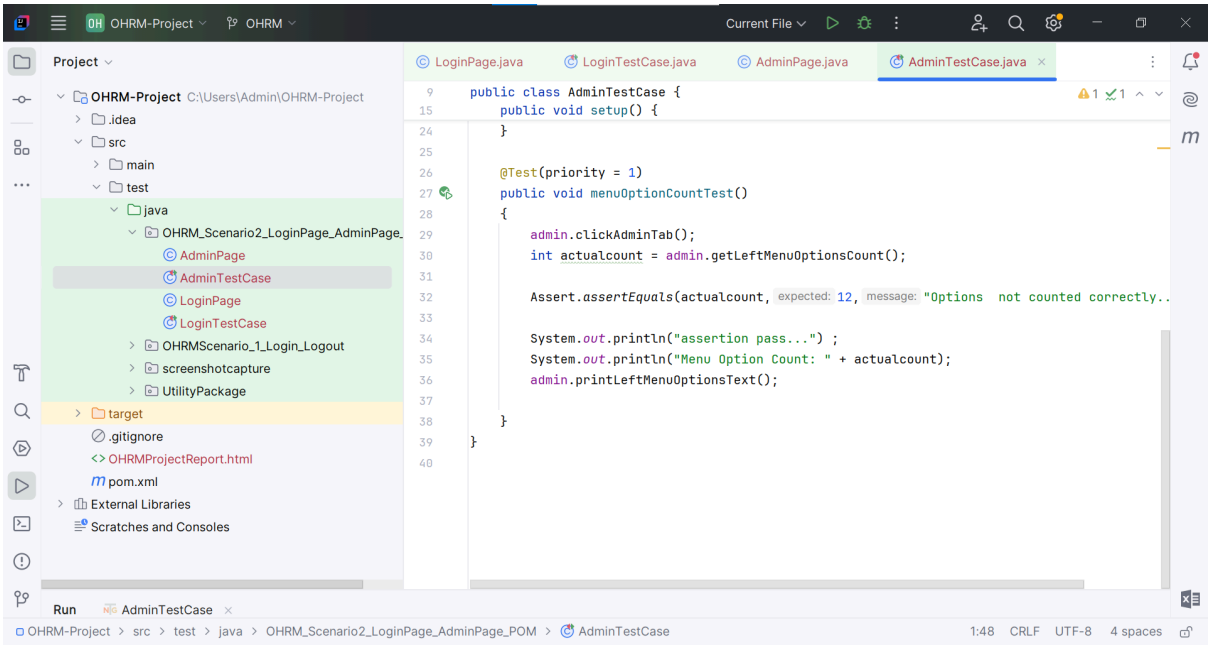
BeforeTest Method:



```
package OHRM_Scenario2_LoginPage_AdminPage_POM;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;
import java.time.Duration;
public class AdminTestCase {
    WebDriver driver;  6 usages
    LoginPage login ;  2 usages
    AdminPage admin;  4 usages

    @BeforeTest
    public void setup() {
        driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
        driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

        login = new LoginPage(driver);
        login.login( username: "Admin",  password: "admin123");    // Perform login first
        admin = new AdminPage(driver);
    }
```

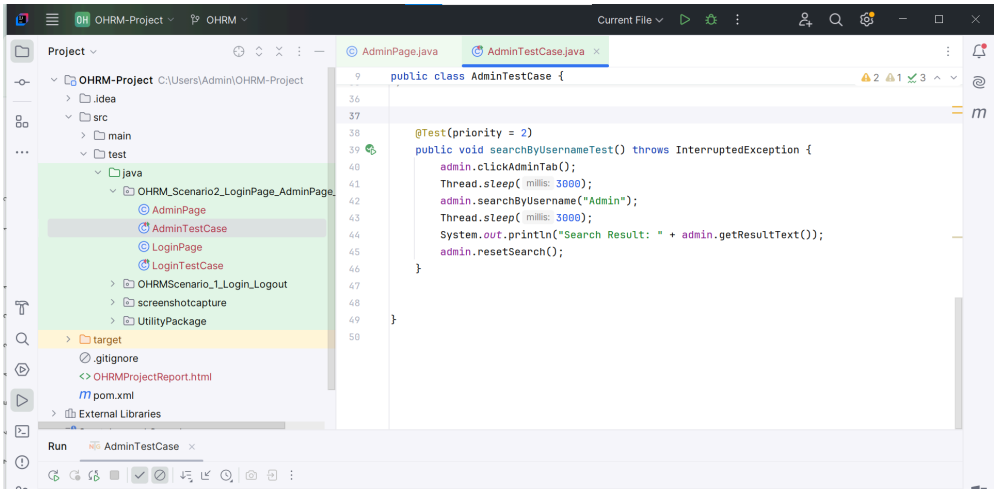
2.1. Create a test case to get all 12 options from the left side menu and print the count, which should be 12 from that list. Click on Admin, and then the Admin page will open.

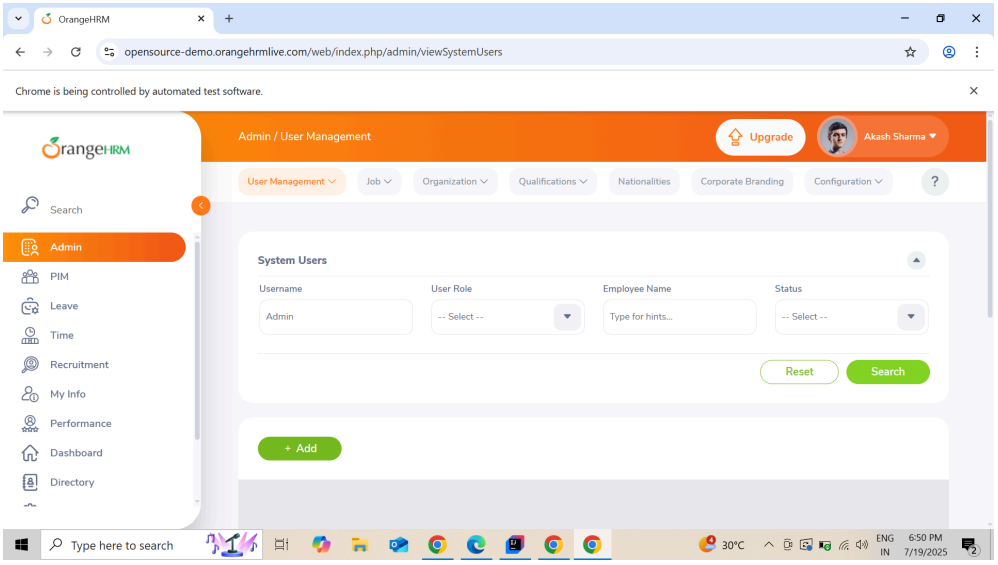
LeftMenuOptions Method:

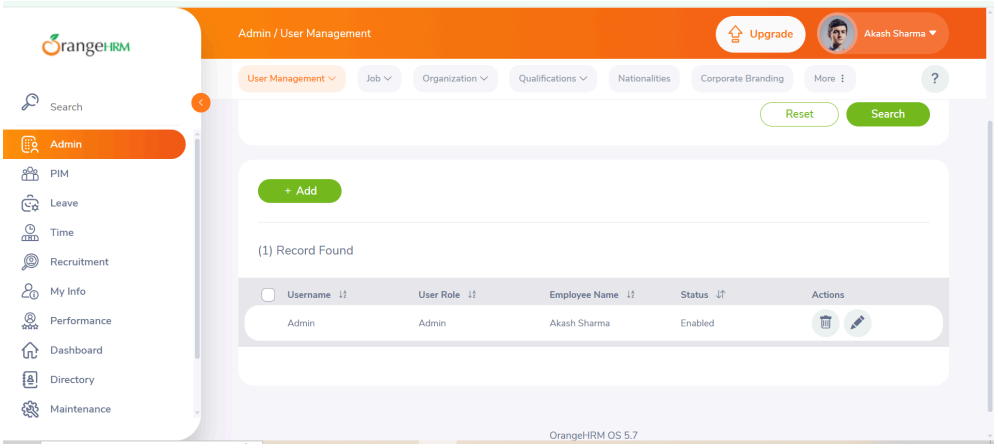


2.2. Create a test case for the search For Existing Employee search by username (): Here, send the username Admin to the username text box and click on the search button and display the total number of records found and refresh the page.

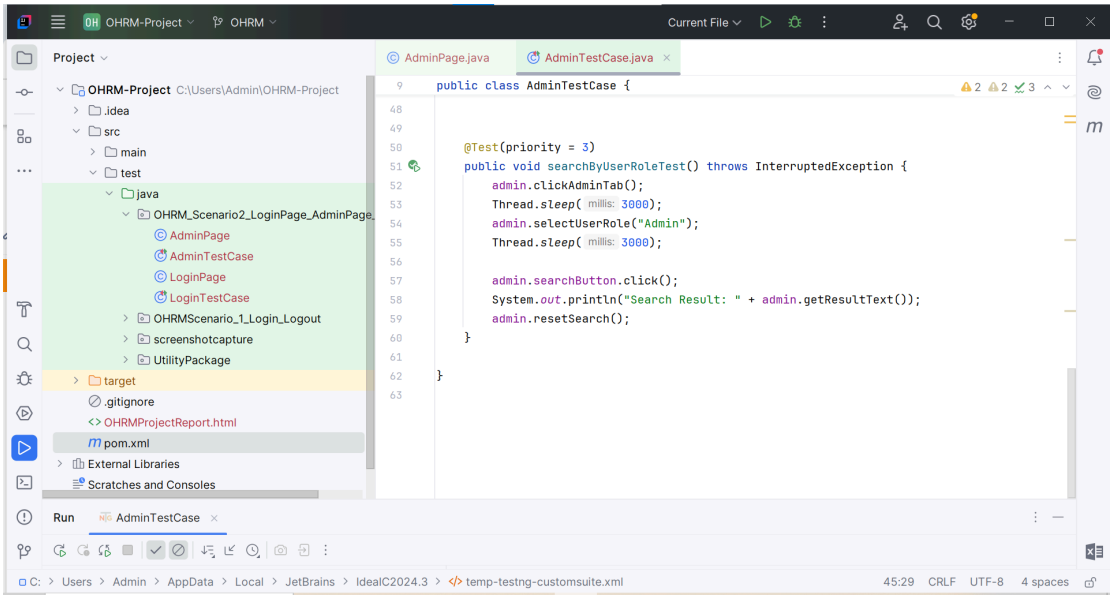
SearchbyUserName()

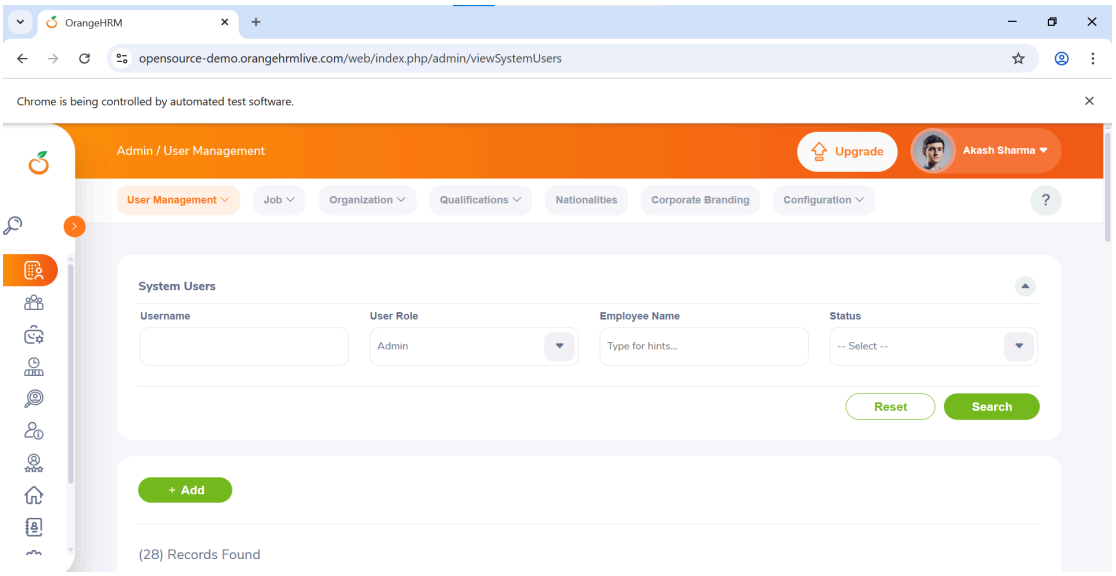


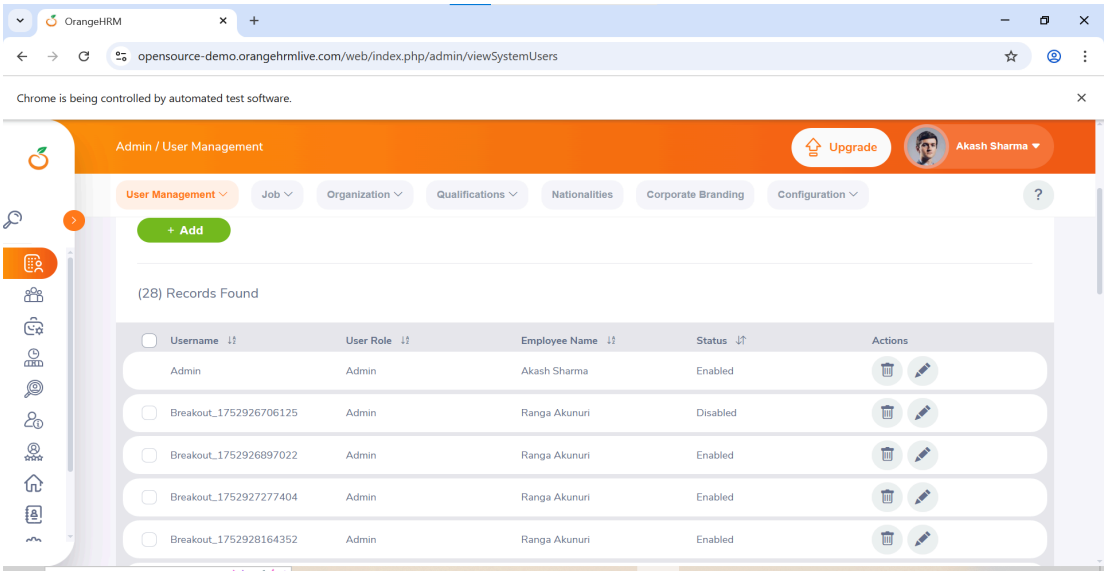




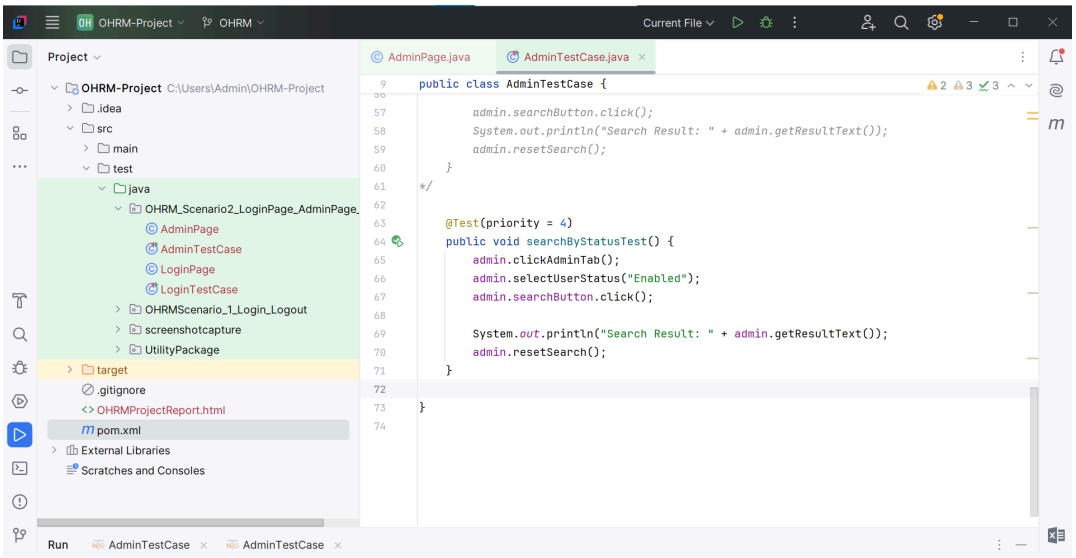
2.3. Create a test case for search For Existing Employee search ByUserRole(): Here, automate the dropdown and select Role Admin and click on the search button and display the total record found and refresh the page.

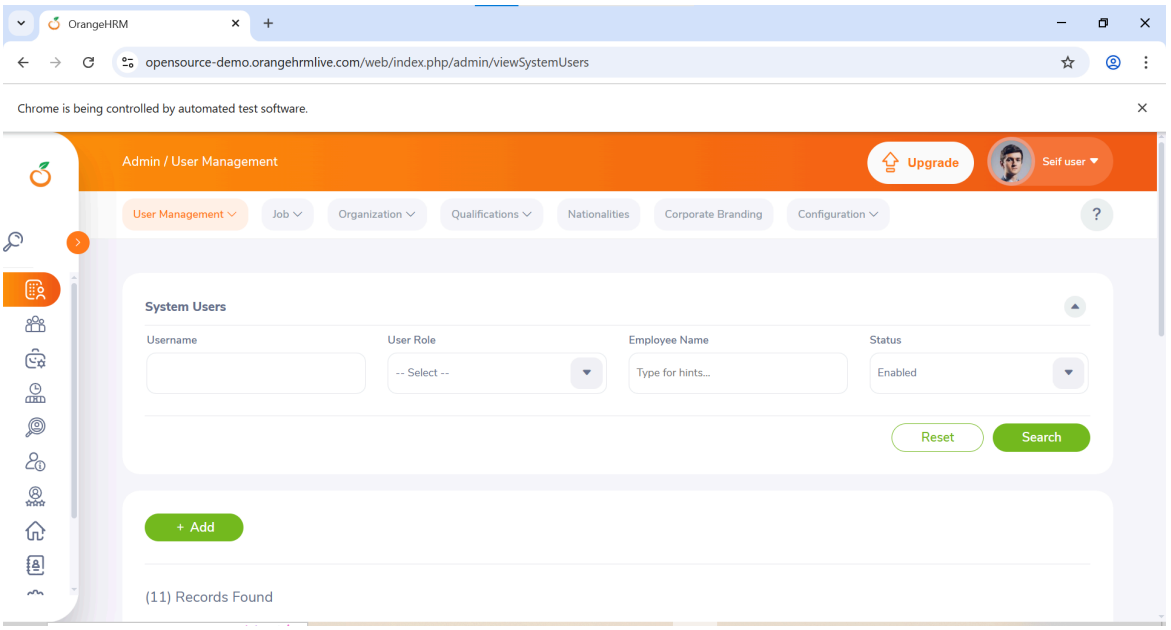


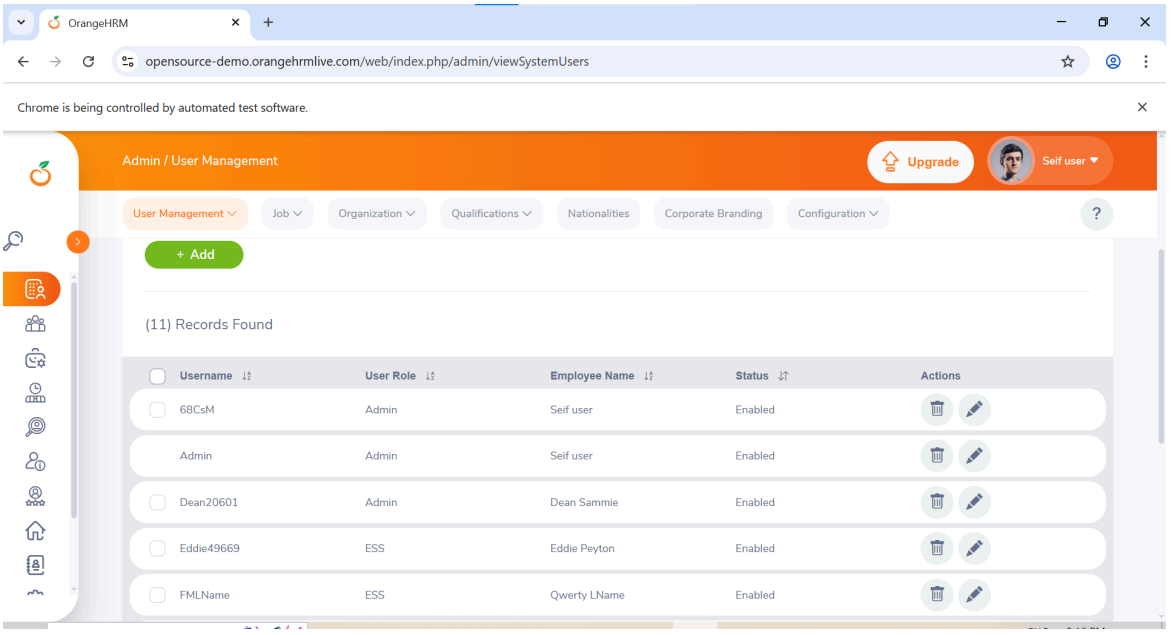




2.4. Create a test case for the search For Existing Employee search ByUserStatus(): here, automate the dropdown and select status Enabled or Disabled, then click on the search button and display the total record found.



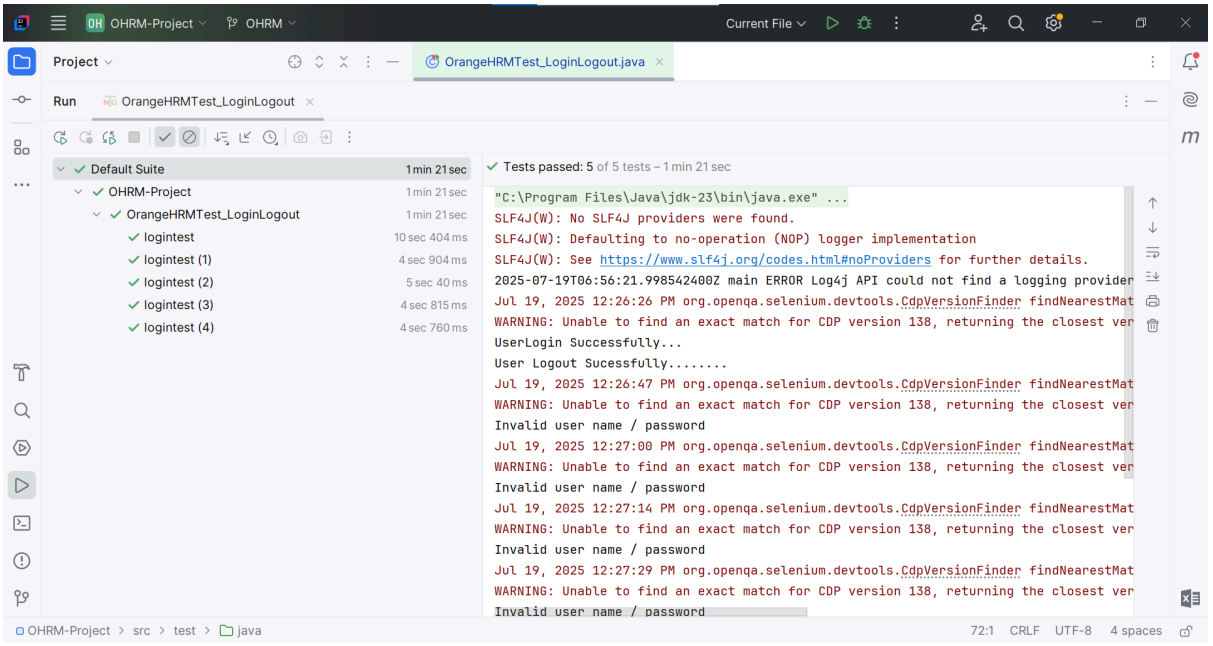

OrangeHRM
opensource-demo.orangehrmlive.com/web/index.php/admin/viewSystemUsers
Chrome is being controlled by automated test software.
Admin / User Management
Upgrade
Seif user
User Management
Job
Organization
Qualifications
Nationalities
Corporate Branding
Configuration
System Users
Username
User Role
-- Select --
Employee Name
Type for hints...
Status
Enabled
Reset
Search
+ Add
(11) Records Found

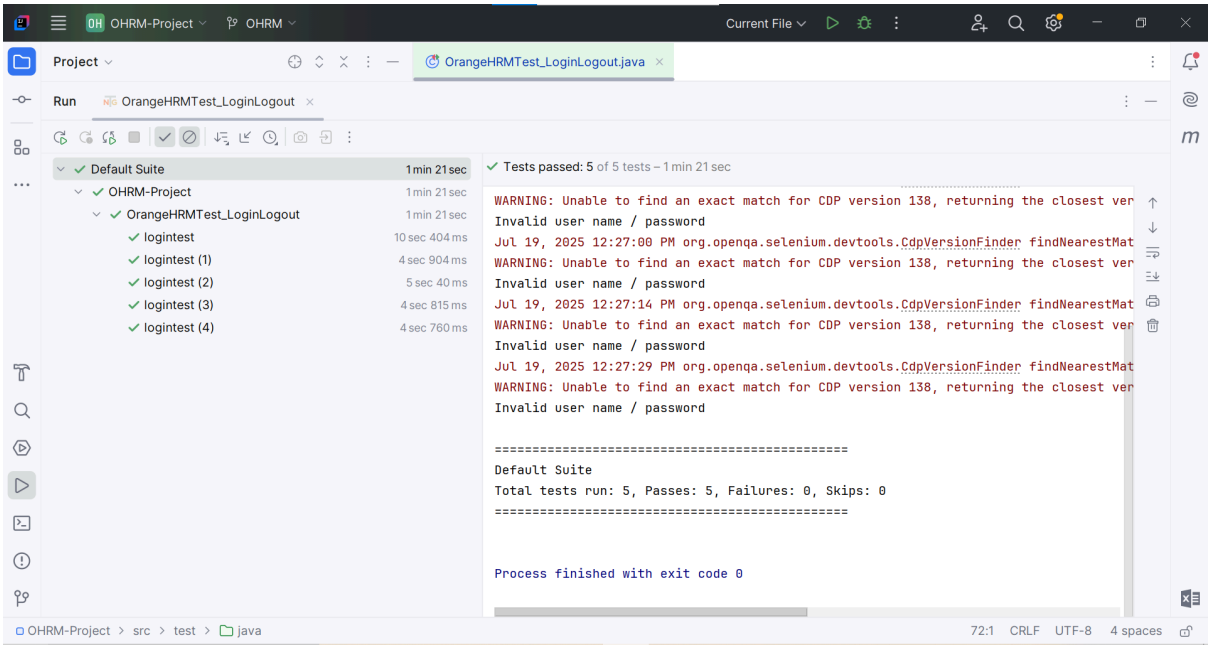

OrangeHRM
opensource-demo.orangehrmlive.com/web/index.php/admin/viewSystemUsers
Chrome is being controlled by automated test software.
Admin / User Management
Upgrade
Seif user
User Management
Job
Organization
Qualifications
Nationalities
Corporate Branding
Configuration
+ Add
(11) Records Found
Username
User Role
Employee Name
Status
Actions
68CsM
Admin
Seif user
Enabled
Admin
Admin
Seif user
Enabled
Dean20601
Admin
Dean Sammie
Enabled
Eddie49669
ESS
Eddie Peyton
Enabled
FMLName
ESS
Qwerty LName
Enabled

7. Test Result Analysis & Reporting

Scenario 1:

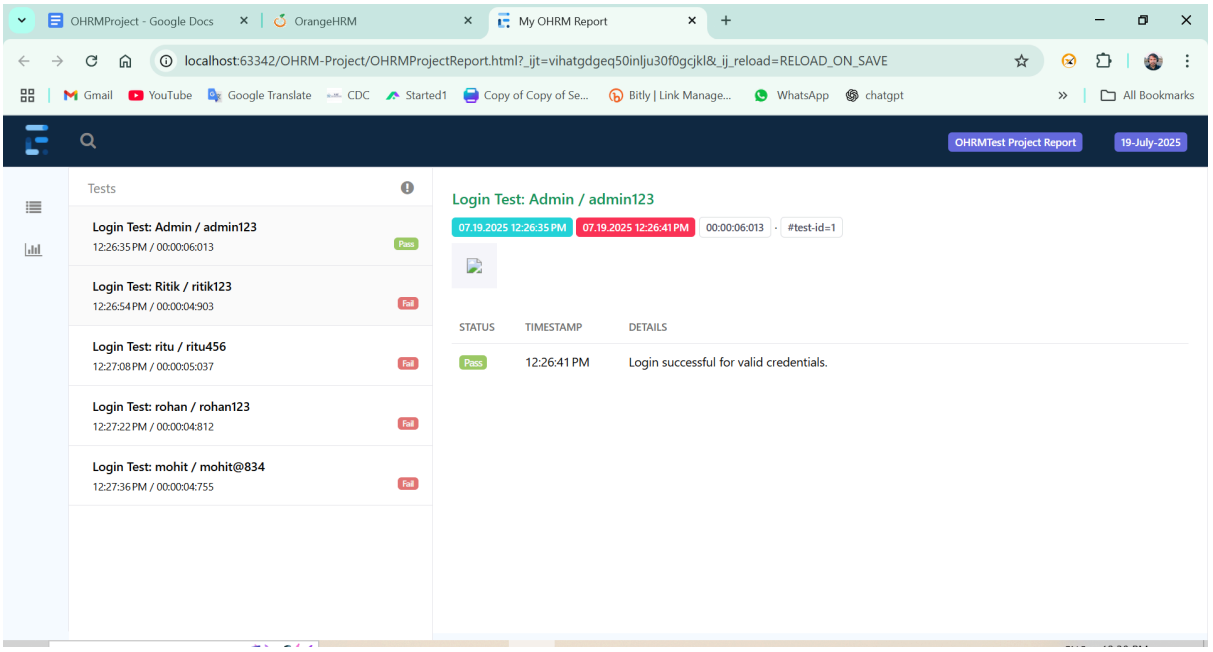
TestNG Report:

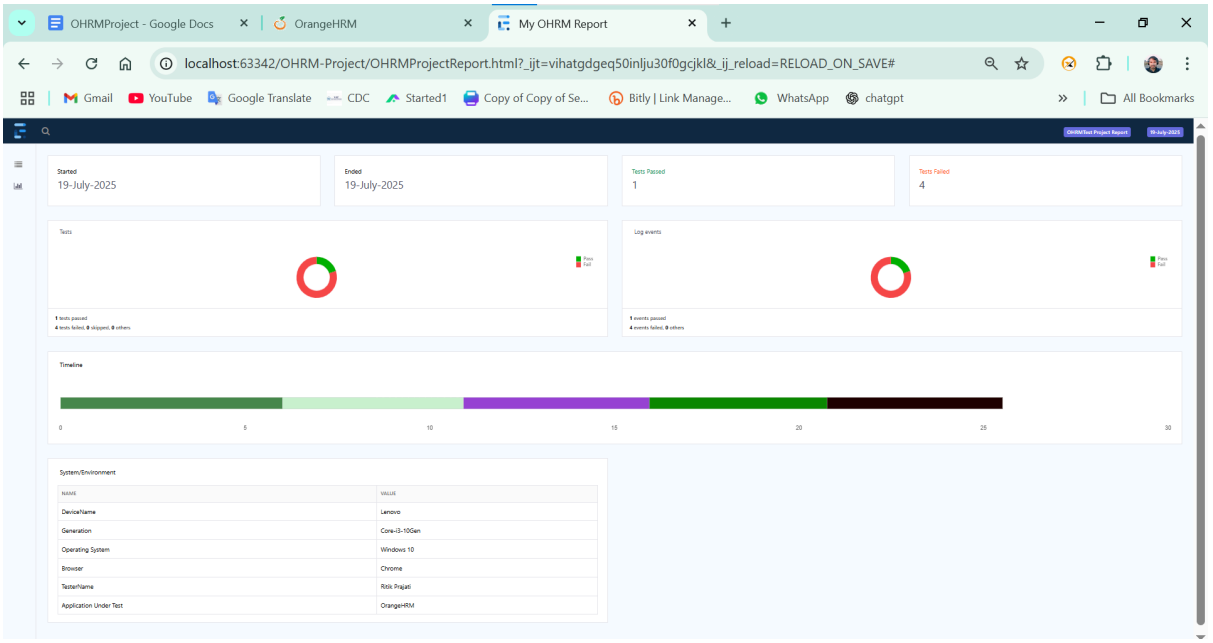




ExtentReports:

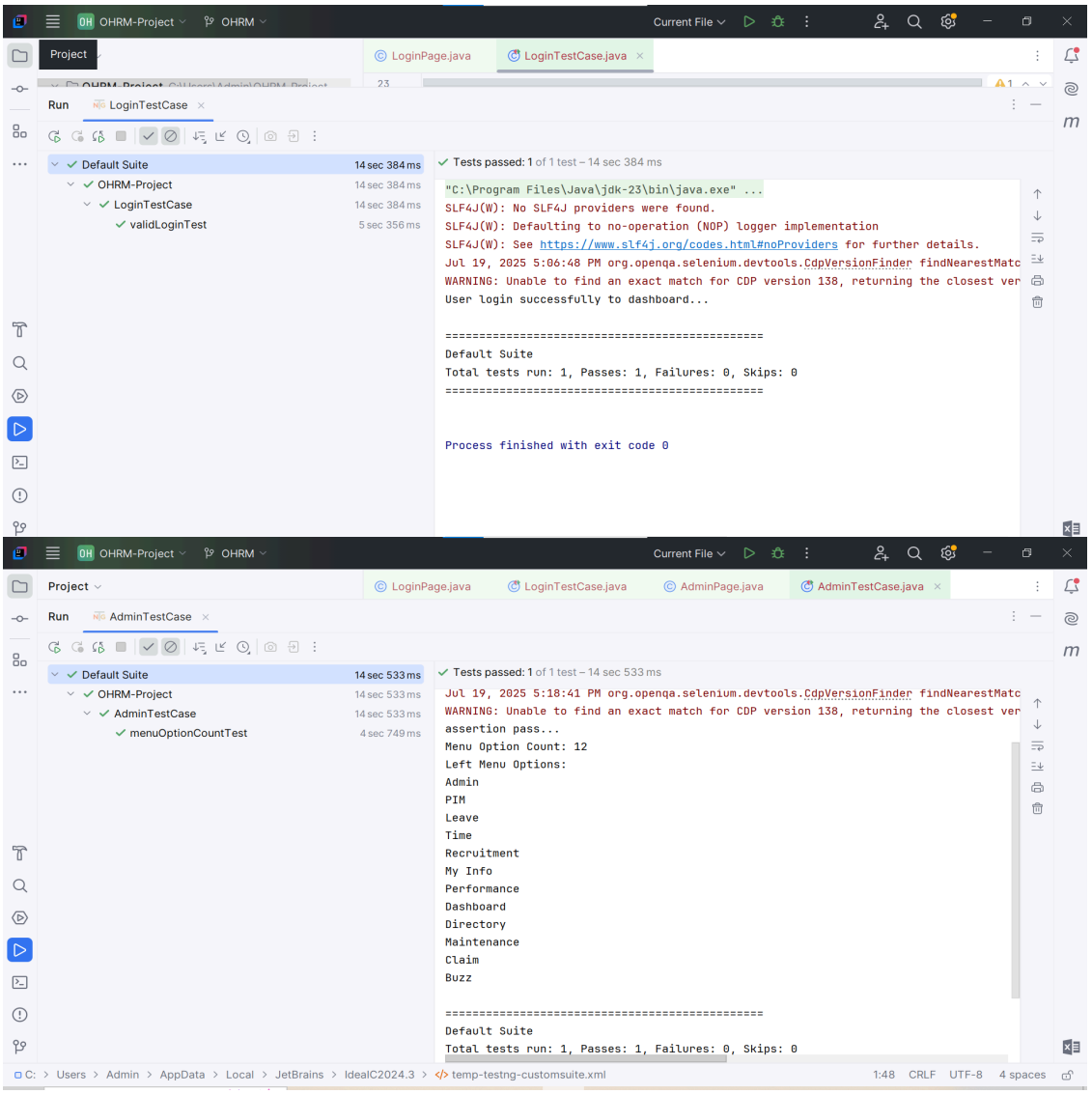
http://localhost:63342/OHRM-Project/OHRMProjectReport.html?_ijt=vihatgdgeq50inlju30f0gcjkl&_ij_reload=RELOAD_ON_SAVE#



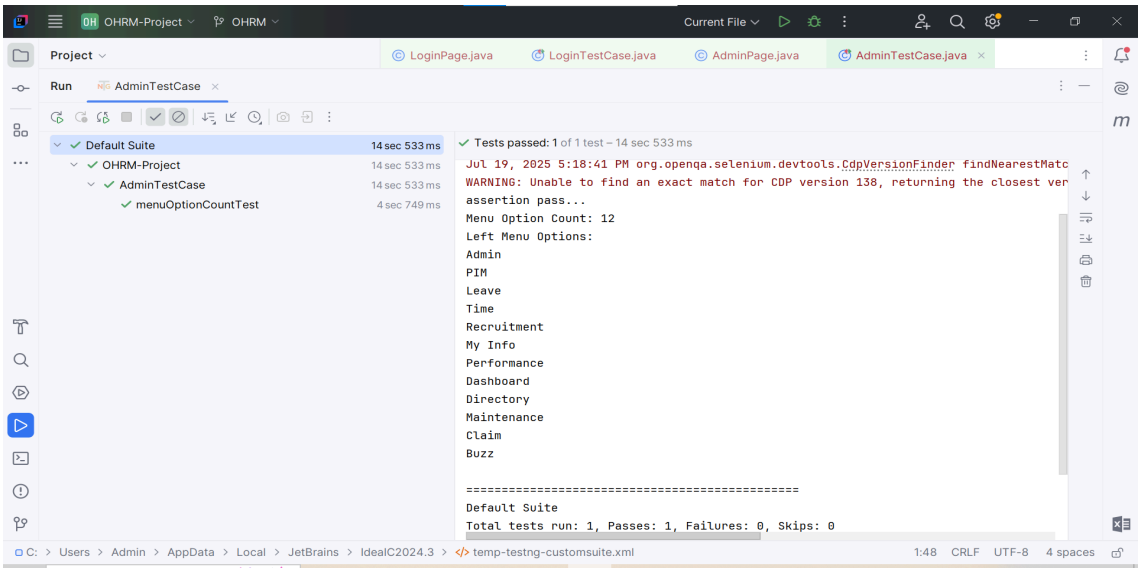


Scenario 2:

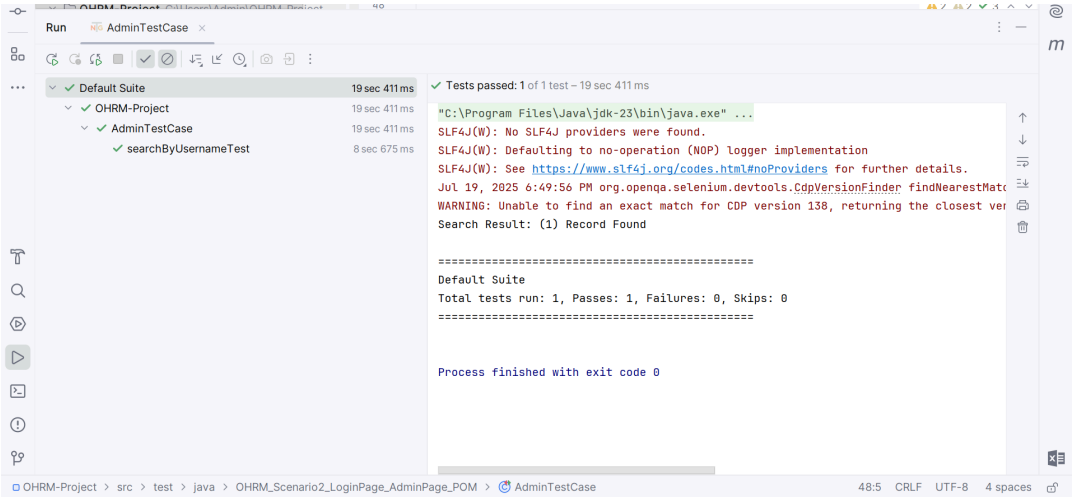
TestNG report: 1.LoginPageTest



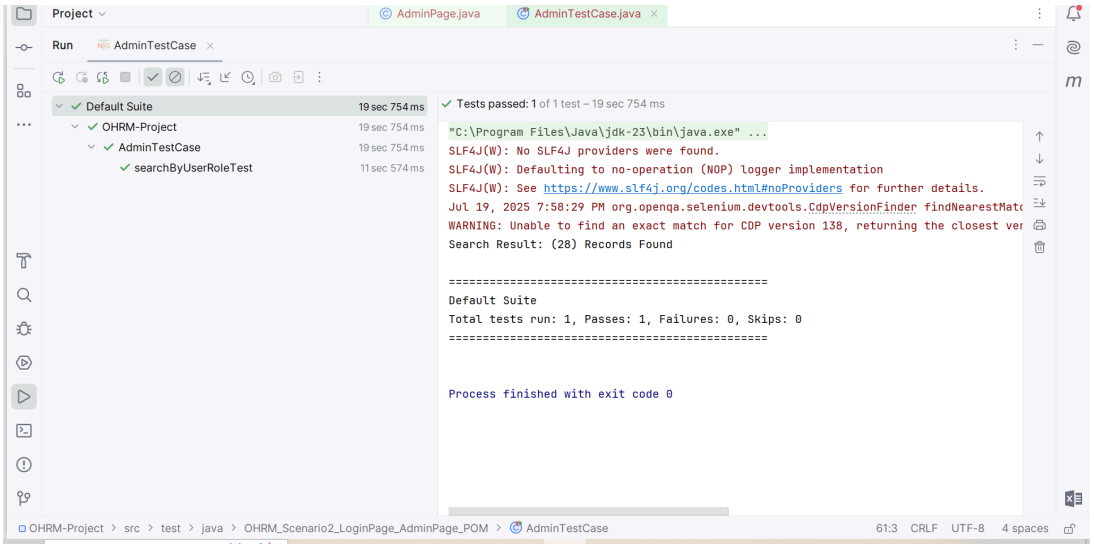
TestNG report: 2.1 LeftMenuOptions



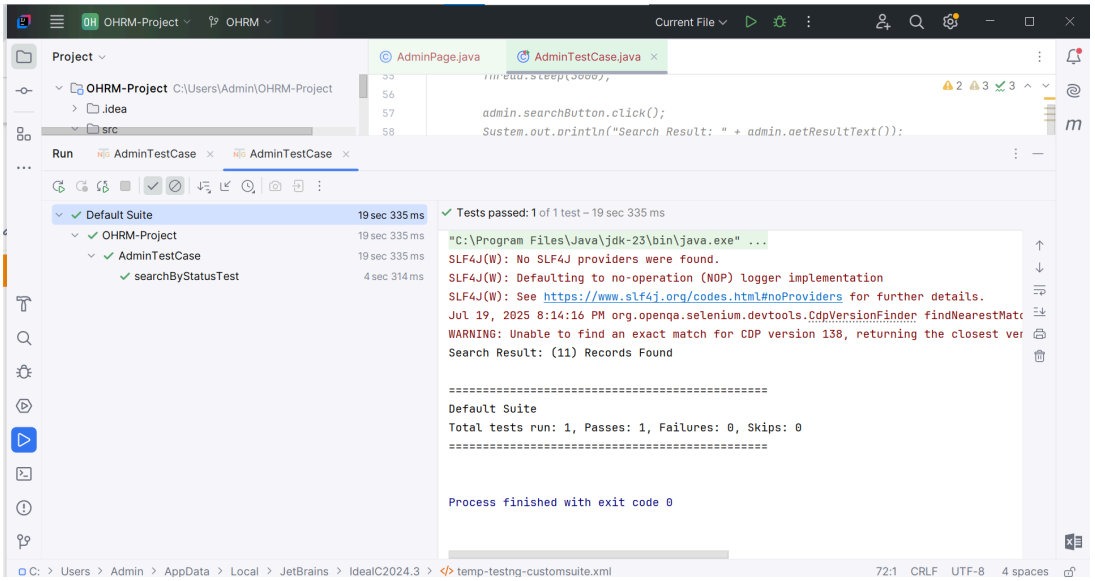
TestNG report: 2.2 searchByUserName()



TestNG report: 2.3 searchByUserRole()



TestNG report: 2.4 searchByUserRole()



8. Conclusion

The automation of OrangeHRM using the **Page Object Model (POM)** framework and **TestNG** has successfully validated critical functional components of the application. The testing effort focused on two primary modules—**Login** and **Admin**, which are fundamental for any HRMS user flow.

In **first scenario** the login functionality was automated with valid and invalid credentials (Admin/admin123). The test script successfully located and interacted with the username, password, and login button fields.

In the **second scenario,** I implemented POM for LoginPage and AdminPage. I tested below four key functionalities:

1. All 12 options from the left-side navigation menu were identified and counted.

2. The username field accepted input dynamically. Search was executed for an existing user **(Admin)**, and results were displayed correctly.

3. The User Role dropdown was handled programmatically. The script selected the **Admin** role and successfully retrieved all associated users.

4. The Status dropdown was accessed, and both values (Enabled, Disabled) were tested . Results displayed the correct status-filtered data.

—---------- END—----------