Capstone Project - Human Resource Management Domain Project

Introduction: Orange HRM is a comprehensive Human Resource Management (HRM) System that captures all the essential functionalities required for any enterprise.

Domain: Human Resource Management

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

Adding new employees

Modifying existing employees

Deleting existing employees

Searching employees

Raising request for hiring new employee

Approving new request

Submitting time sheet

Approving timesheet etc.

Modules Login, Admin, PIM, Time, Leave, Recruitment, My Info, Performance, Dashboard, Directory, Maintenance, Claim, Buzz

Modules In Scope for Automation Testing

1. Login

2. Admin

2.1. User Management

Objective: As a tester, you need to understand the below requirements and prepare an Automation script for the same using Selenium and TestNG

Module 1. Login:

Scenario 1: Automate Orange HRM login and logout with 5 different data sets including valid and invalid data sets.

1. Save the data set in an Excel file

2. Write a script to read data from Excel

3. Prepare a script for Login and logout

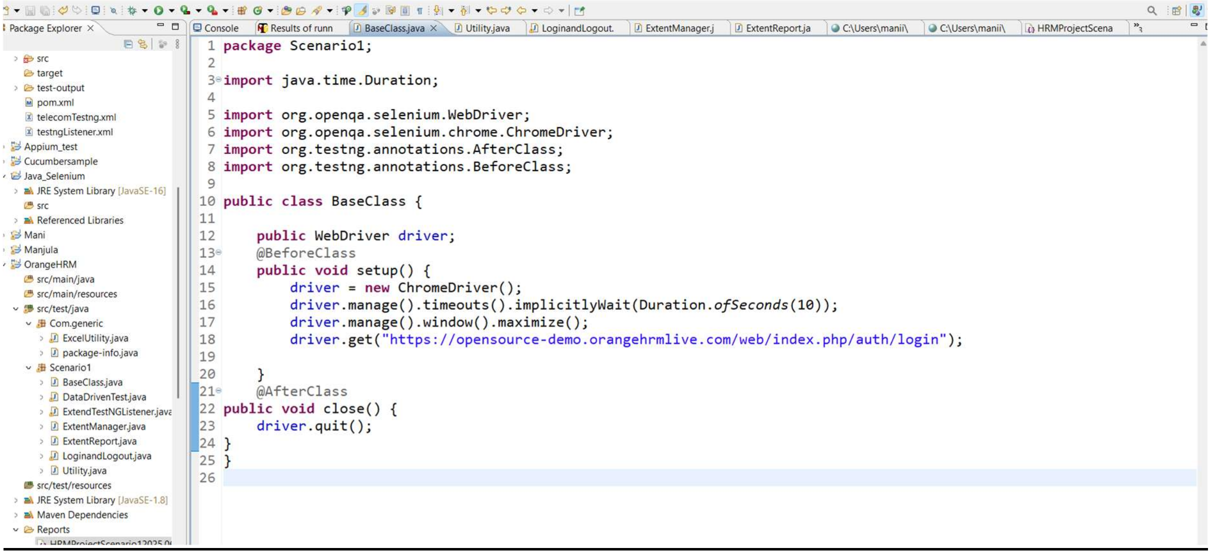
4. Perform assertion for valid data set (use Username: Admin and Password: admin123) test case should be passed and invalid data set (other than given data) test case should fail.

5. Capture Screenshot for every login functionality

6. Generate Extent Report for the same.

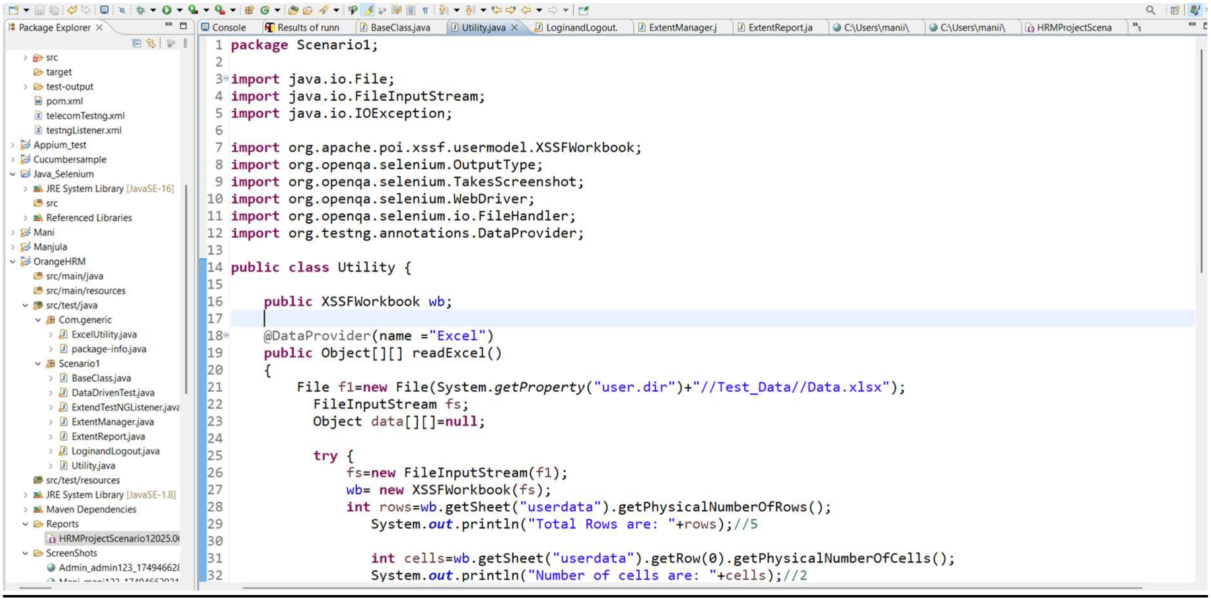
<u>Git Hub URL</u>:

<u>Base Class:</u>

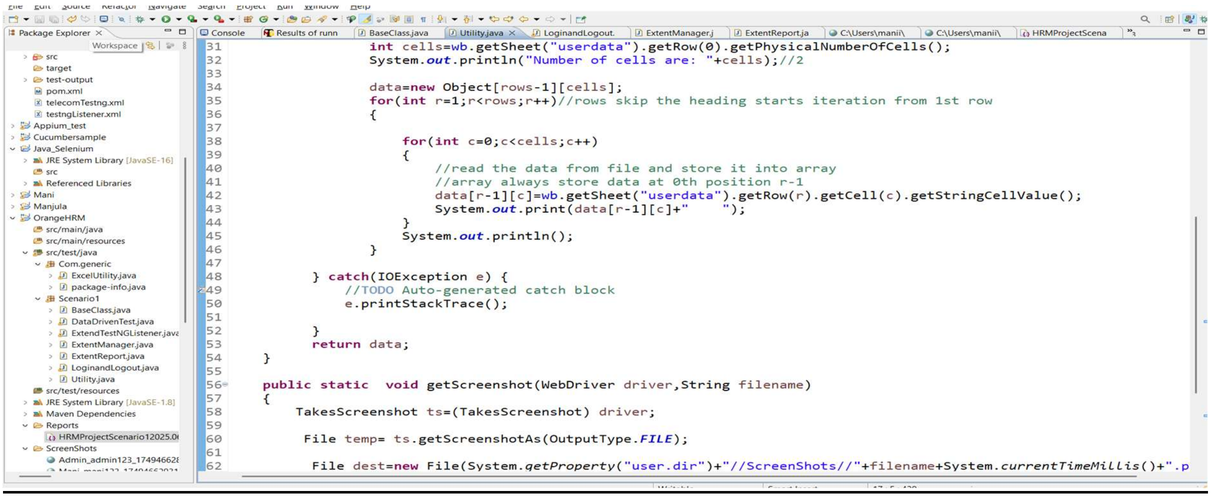


```java
package Scenario1;

import java.time.Duration;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;

public class BaseClass {

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

    }
    @AfterClass
public void close() {
    driver.quit();
}
}
```

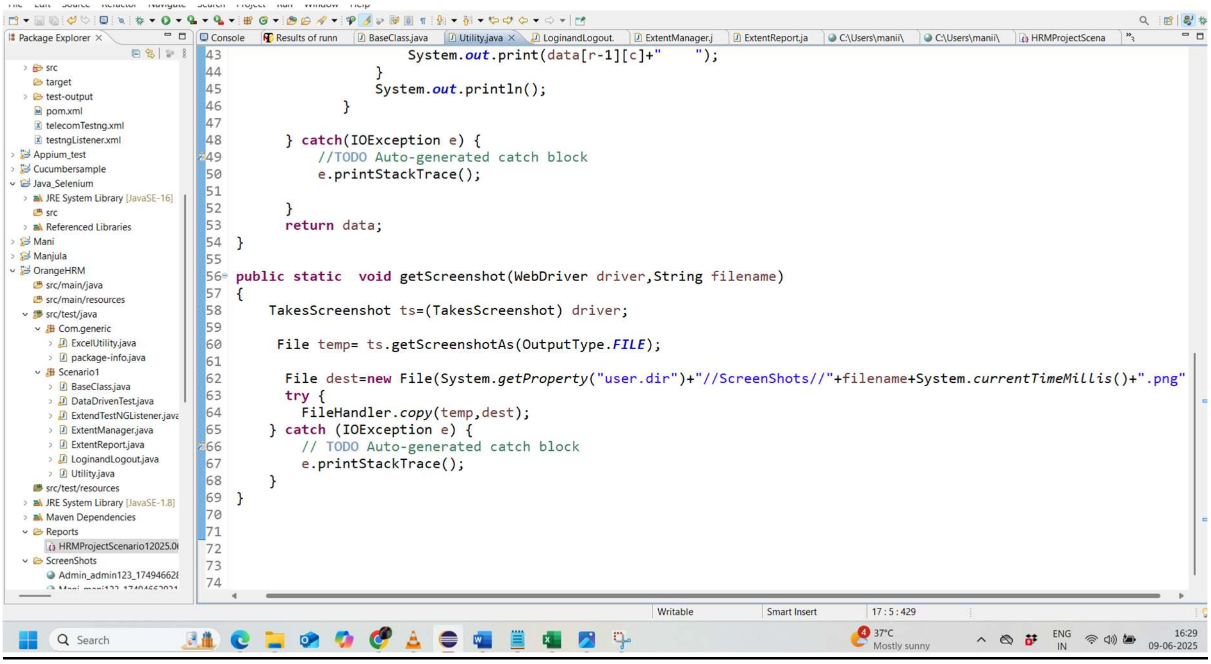
<u>Utility:</u>



```java
package Scenario1;

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

import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.io.FileHandler;
import org.testng.annotations.DataProvider;

public class Utility {

    public XSSFWorkbook wb;

    @DataProvider(name ="Excel")
    public Object[][] readExcel()
    {
        File f1=new File(System.getProperty("user.dir")+"//Test_Data//Data.xlsx");
         FileInputStream fs;
         Object data[][]=null;

         try {
             fs=new FileInputStream(f1);
             wb= new XSSFWorkbook(fs);
             int rows=wb.getSheet("userdata").getPhysicalNumberOfRows();
                System.out.println("Total Rows are:  "+rows);//5

                int cells=wb.getSheet("userdata").getRow(0).getPhysicalNumberOfCells();
                System.out.println("Number of cells are:  "+cells);//2
```

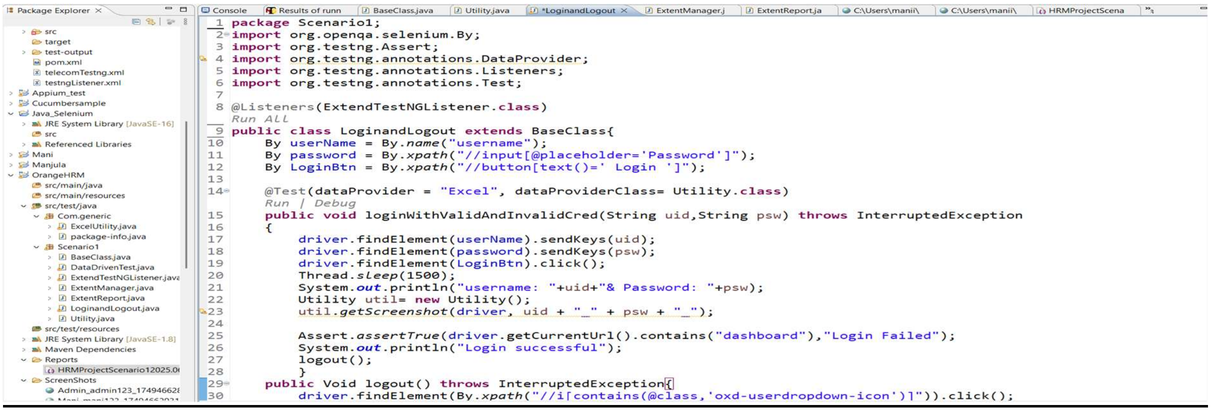


```java
                int cells=wb.getSheet("userdata").getRow(0).getPhysicalNumberOfCells();
                System.out.println("Number of cells are: "+cells);//2

                data=new Object[rows-1][cells];
                for(int r=1;r<rows;r++)//rows skip the heading starts iteration from 1st row
                {

                    for(int c=0;c<cells;c++)
                    {
                        //read the data from file and store it into array
                        //array always store data at 0th position r-1
                        data[r-1][c]=wb.getSheet("userdata").getRow(r).getCell(c).getStringCellValue();
                        System.out.print(data[r-1][c]+"     ");
                    }
                    System.out.println();
                }

            } catch(IOException e) {
                //TODO Auto-generated catch block
                e.printStackTrace();

            }
            return data;
        }

    public static  void getScreenshot(WebDriver driver,String filename)
    {
        TakesScreenshot ts=(TakesScreenshot) driver;

         File temp= ts.getScreenshotAs(OutputType.FILE);

          File dest=new File(System.getProperty("user.dir")+"//ScreenShots//"+filename+System.currentTimeMillis()+".p
```

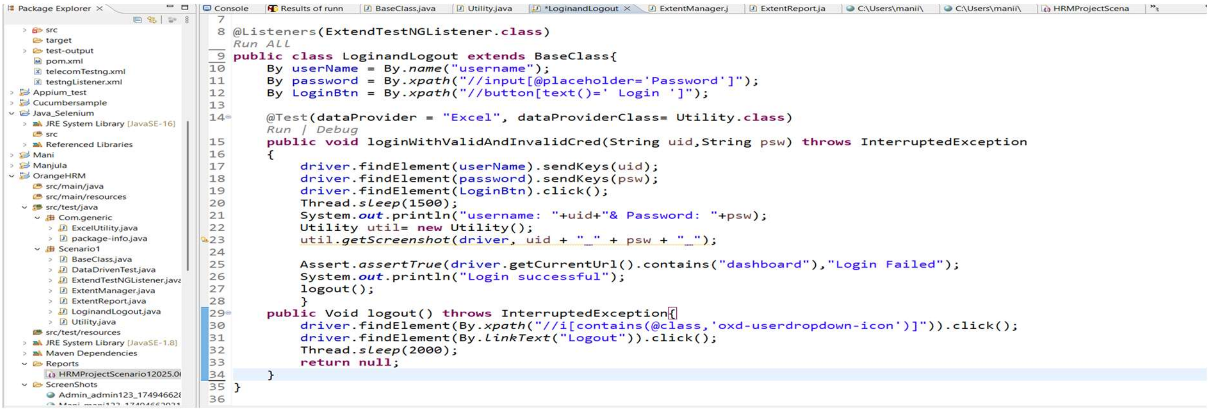


```java
                        System.out.print(data[r-1][c]+"     ");
                    }
                    System.out.println();
                }

            } catch(IOException e) {
                //TODO Auto-generated catch block
                e.printStackTrace();

            }
            return data;
        }

    public static  void getScreenshot(WebDriver driver,String filename)
    {
        TakesScreenshot ts=(TakesScreenshot) driver;

         File temp= ts.getScreenshotAs(OutputType.FILE);

          File dest=new File(System.getProperty("user.dir")+"//ScreenShots//"+filename+System.currentTimeMillis()+".png"
          try {
            FileHandler.copy(temp,dest);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
```

Login Logout:

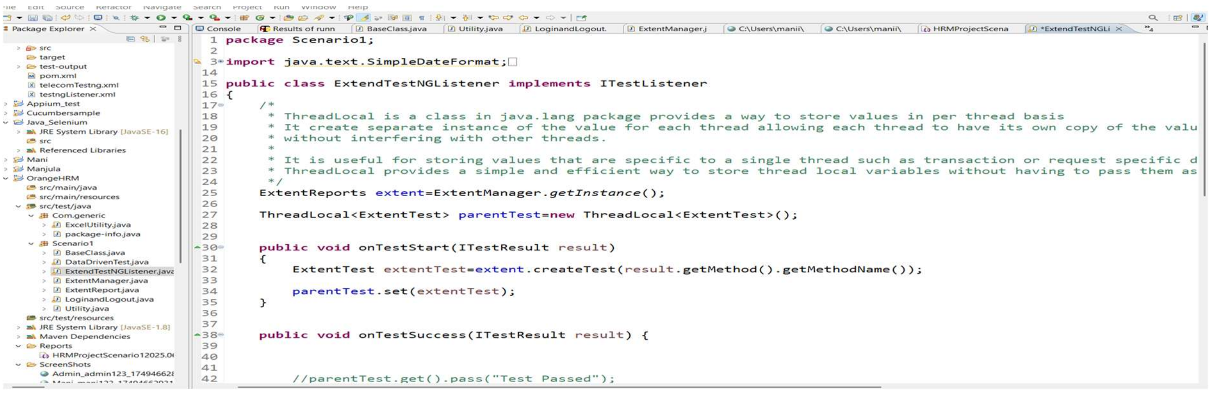


```java
package Scenario1;
import org.openqa.selenium.By;
import org.testng.Assert;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Listeners;
import org.testng.annotations.Test;

@Listeners(ExtendTestNGListener.class)
public class LoginandLogout extends BaseClass{
    By userName = By.name("username");
    By password = By.xpath("//input[@placeholder='Password']");
    By LoginBtn = By.xpath("//button[text()=' Login ']");

    @Test(dataProvider = "Excel", dataProviderClass= Utility.class)
    public void loginWithValidAndInvalidCred(String uid,String psw) throws InterruptedException
    {
        driver.findElement(userName).sendKeys(uid);
        driver.findElement(password).sendKeys(psw);
        driver.findElement(LoginBtn).click();
        Thread.sleep(1500);
        System.out.println("username: "+uid+"& Password: "+psw);
        Utility util= new Utility();
        util.getScreenshot(driver, uid + "_" + psw + "_");

        Assert.assertTrue(driver.getCurrentUrl().contains("dashboard"),"Login Failed");
        System.out.println("Login successful");
        logout();
        }
    public Void logout() throws InterruptedException{
        driver.findElement(By.xpath("//i[contains(@class,'oxd-userdropdown-icon')]")).click();
```

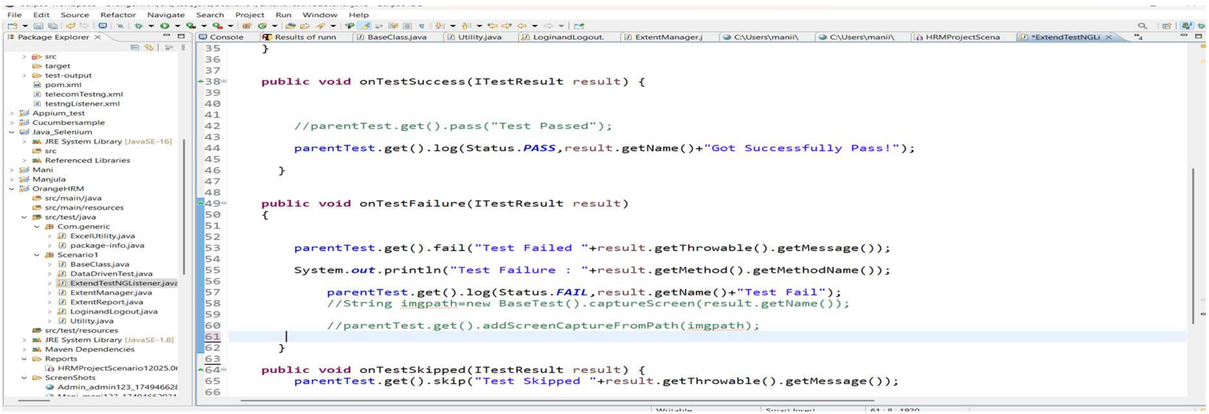


```java
@Listeners(ExtendTestNGListener.class)
Run ALL
public class LoginandLogout extends BaseClass{
    By userName = By.name("username");
    By password = By.xpath("//input[@placeholder='Password']");
    By LoginBtn = By.xpath("//button[text()=' Login ']");

    @Test(dataProvider = "Excel", dataProviderClass= Utility.class)
    Run | Debug
    public void loginWithValidAndInvalidCred(String uid,String psw) throws InterruptedException
    {
        driver.findElement(userName).sendKeys(uid);
        driver.findElement(password).sendKeys(psw);
        driver.findElement(LoginBtn).click();
        Thread.sleep(1500);
        System.out.println("username:  "+uid+"& Password:  "+psw);
        Utility util= new Utility();
        util.getScreenshot(driver, uid + "_" + psw + "_");

        Assert.assertTrue(driver.getCurrentUrl().contains("dashboard"),"Login Failed");
        System.out.println("Login successful");
        logout();
        }
    public Void logout() throws InterruptedException{
        driver.findElement(By.xpath("//i[contains(@class,'oxd-userdropdown-icon')]")).click();
        driver.findElement(By.linkText("Logout")).click();
        Thread.sleep(2000);
        return null;
    }
}
```

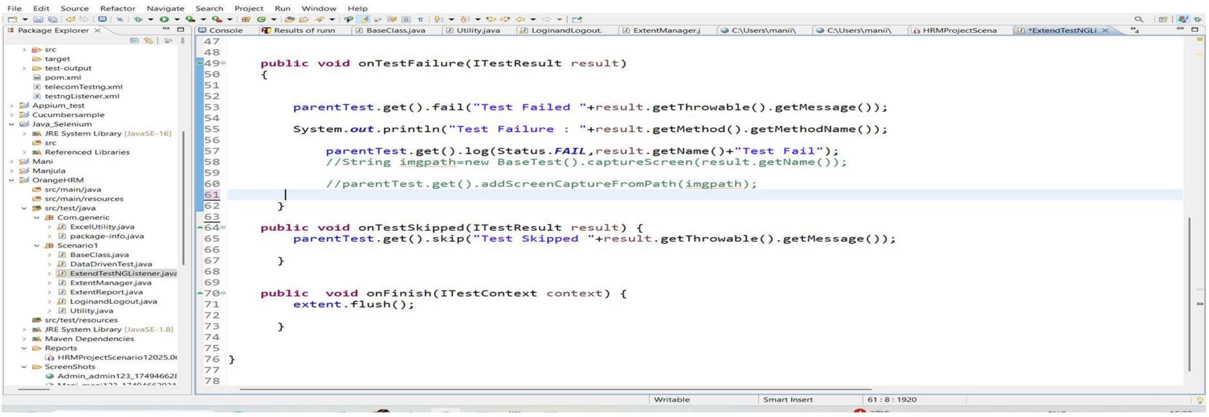
TestNG Listener:



```java
package Scenario1;

import java.text.SimpleDateFormat;

public class ExtendTestNGListener implements ITestListener
{
    /*
     * ThreadLocal is a class in java.lang package provides a way to store values in per thread basis
     * It create separate instance of the value for each thread allowing each thread to have its own copy of the valu
     * without interfering with other threads.
     *
     * It is useful for storing values that are specific to a single thread such as transaction or request specific d
     * ThreadLocal provides a simple and efficient way to store thread local variables without having to pass them as
     */
    ExtentReports extent=ExtentManager.getInstance();

    ThreadLocal<ExtentTest> parentTest=new ThreadLocal<ExtentTest>();


    public void onTestStart(ITestResult result)
    {
        ExtentTest extentTest=extent.createTest(result.getMethod().getMethodName());

        parentTest.set(extentTest);
    }


    public void onTestSuccess(ITestResult result) {



        //parentTest.get().pass("Test Passed");
```

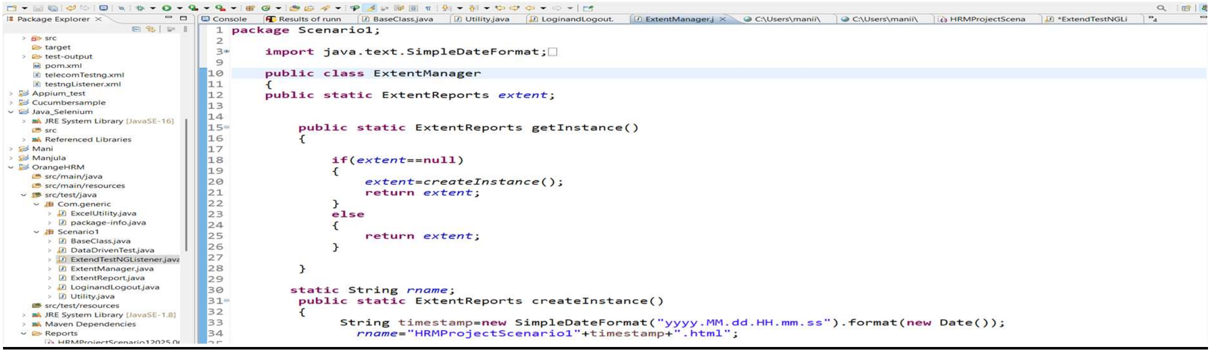


```java
    }


    public void onTestSuccess(ITestResult result) {



        //parentTest.get().pass("Test Passed");

        parentTest.get().log(Status.PASS,result.getName()+"Got Successfully Pass!");

    }


    public void onTestFailure(ITestResult result)
    {


        parentTest.get().fail("Test Failed "+result.getThrowable().getMessage());

        System.out.println("Test Failure :  "+result.getMethod().getMethodName());

            parentTest.get().log(Status.FAIL,result.getName()+"Test Fail");
            //String imgpath=new BaseTest().captureScreen(result.getName());

            //parentTest.get().addScreenCaptureFromPath(imgpath);

    }

    public void onTestSkipped(ITestResult result) {
        parentTest.get().skip("Test Skipped "+result.getThrowable().getMessage());

```

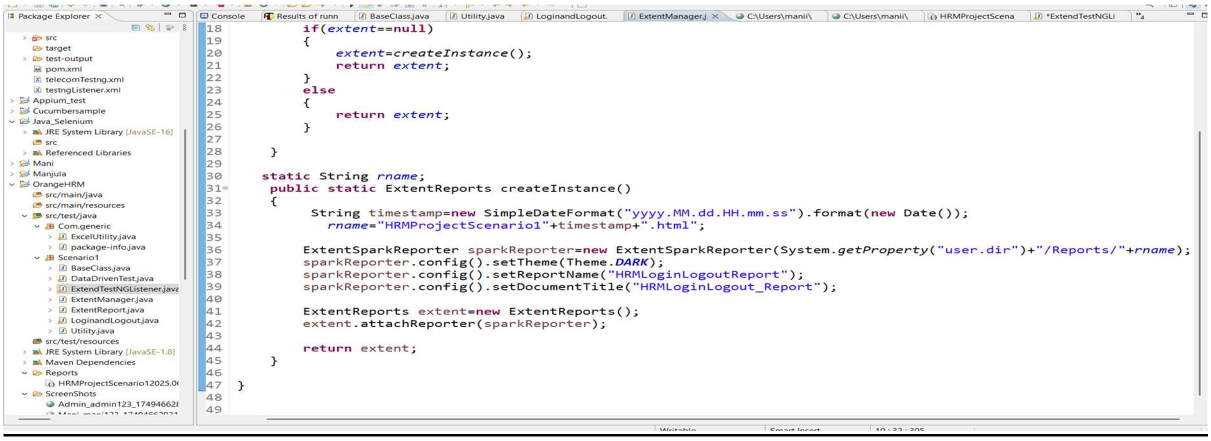


```java

    public void onTestFailure(ITestResult result)
    {


        parentTest.get().fail("Test Failed "+result.getThrowable().getMessage());

        System.out.println("Test Failure :  "+result.getMethod().getMethodName());

            parentTest.get().log(Status.FAIL,result.getName()+"Test Fail");
            //String imgpath=new BaseTest().captureScreen(result.getName());

            //parentTest.get().addScreenCaptureFromPath(imgpath);

    }

    public void onTestSkipped(ITestResult result) {
        parentTest.get().skip("Test Skipped "+result.getThrowable().getMessage());

    }


    public  void onFinish(ITestContext context) {
        extent.flush();

    }


}


```

Extend Manager Listener:



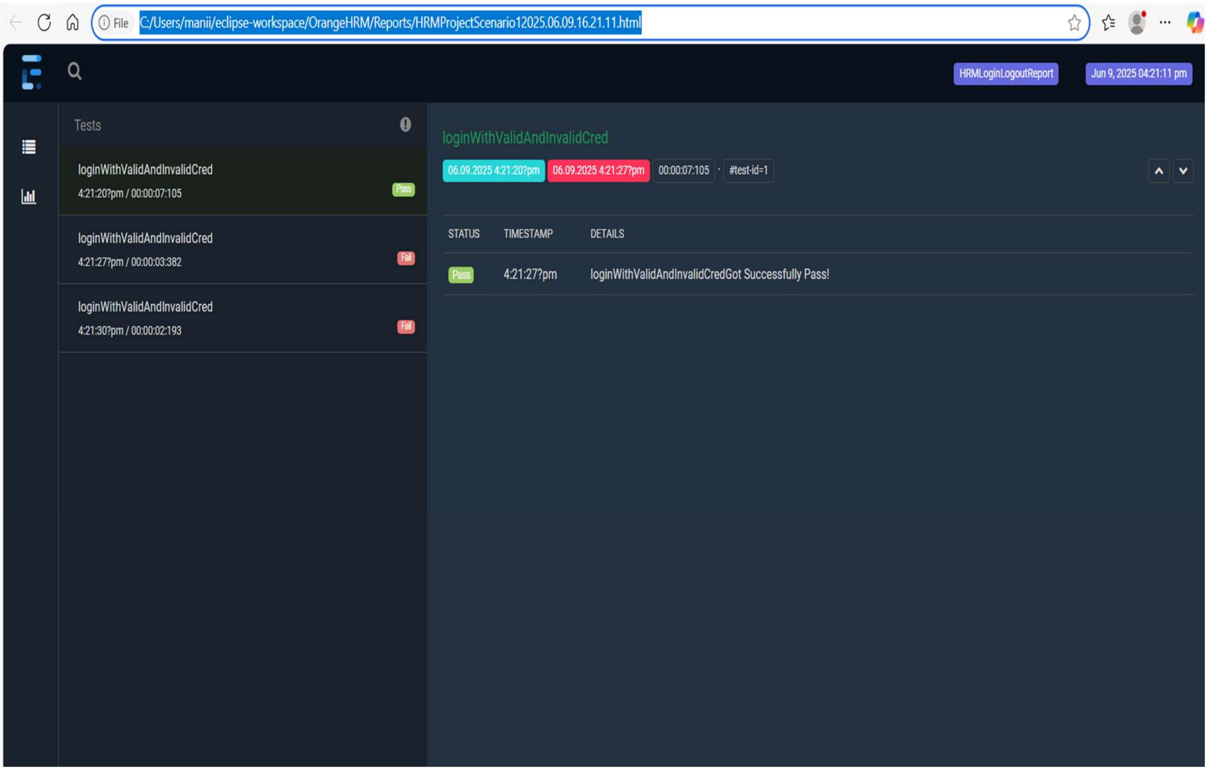
```java
package Scenario1;

    import java.text.SimpleDateFormat;

    public class ExtentManager
    {
    public static ExtentReports extent;


        public static ExtentReports getInstance()
        {

            if(extent==null)
            {
                extent=createInstance();
                return extent;
            }
            else
            {
                return extent;
            }

        }

       static String rname;
        public static ExtentReports createInstance()
        {
             String timestamp=new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new Date());
               rname="HRMProjectScenario1"+timestamp+".html";
```



```java
            if(extent==null)
            {
                extent=createInstance();
                return extent;
            }
            else
            {
                return extent;
            }

        }

       static String rname;
        public static ExtentReports createInstance()
        {
             String timestamp=new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new Date());
               rname="HRMProjectScenario1"+timestamp+".html";

            ExtentSparkReporter sparkReporter=new ExtentSparkReporter(System.getProperty("user.dir")+"/Reports/"+rname);
            sparkReporter.config().setTheme(Theme.DARK);
            sparkReporter.config().setReportName("HRMLoginLogoutReport");
            sparkReporter.config().setDocumentTitle("HRMLoginLogout_Report");

            ExtentReports extent=new ExtentReports();
            extent.attachReporter(sparkReporter);

            return extent;
        }

    }
```

Report:

HRMLoginLogout_Report


File
C:/Users/manii/eclipse-workspace/OrangeHRM/Reports/HRMProjectScenario12025.06.09.16.21.11.html
HRMLoginLogoutReport
Jun 9, 2025 04:21:11 pm
Tests

loginWithValidAndInvalidCred
4:21:20?pm / 00:00:07:105 Pass

loginWithValidAndInvalidCred
4:21:27?pm / 00:00:03:382 Fail

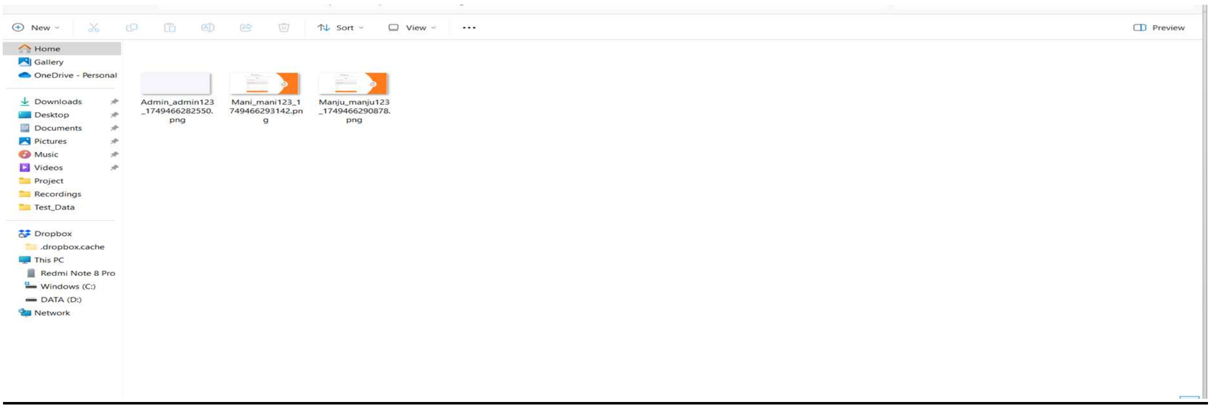
loginWithValidAndInvalidCred
4:21:30?pm / 00:00:02:193 Fail

loginWithValidAndInvalidCred

06.09.2025 4:21:20?pm 06.09.2025 4:21:27?pm 00:00:07:105 · #test-id=1

STATUS	TIMESTAMP	DETAILS
Pass	4:21:27?pm	loginWithValidAndInvalidCredGot Successfully Pass!

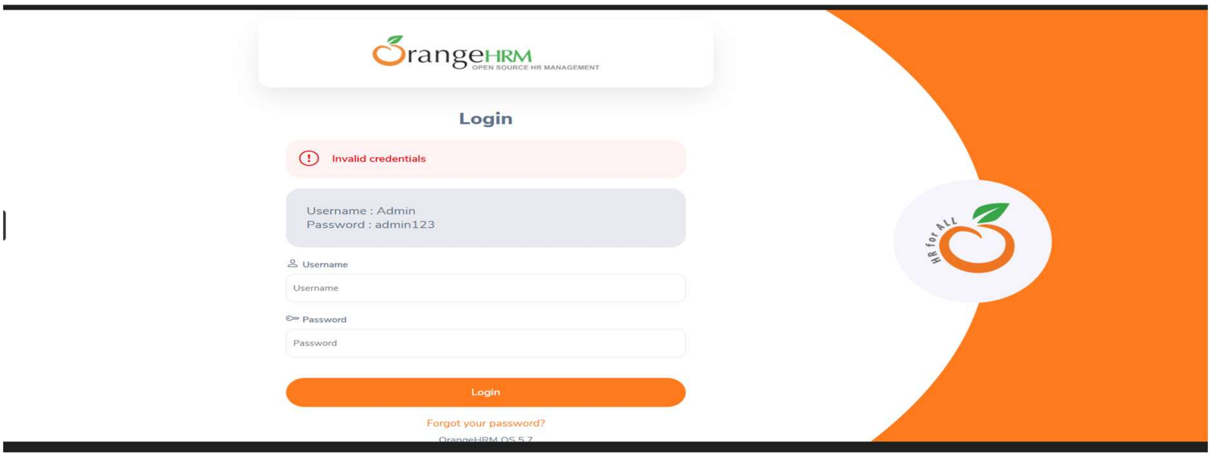
Screenshot:



Admin_admin123_1749466282550.png

Mani_mani123_1749466293142.png

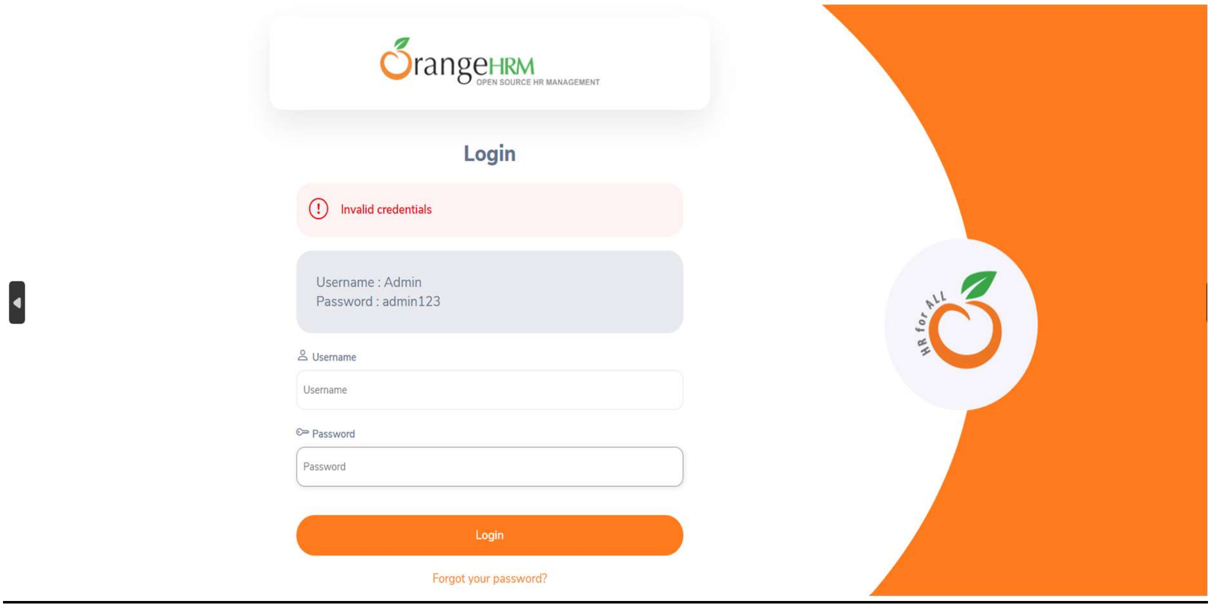
Manju_manju123_1749466290878.png



Invalid credentials

Username : Admin
Password : admin123

OrangeHRM OS 5.7



Invalid credentials

Username : Admin
Password : admin123

Scenario 2: **Create a Page Object Model for two pages Login Page and an Admin Page and write an automation script for 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 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 then the Admin page will open.

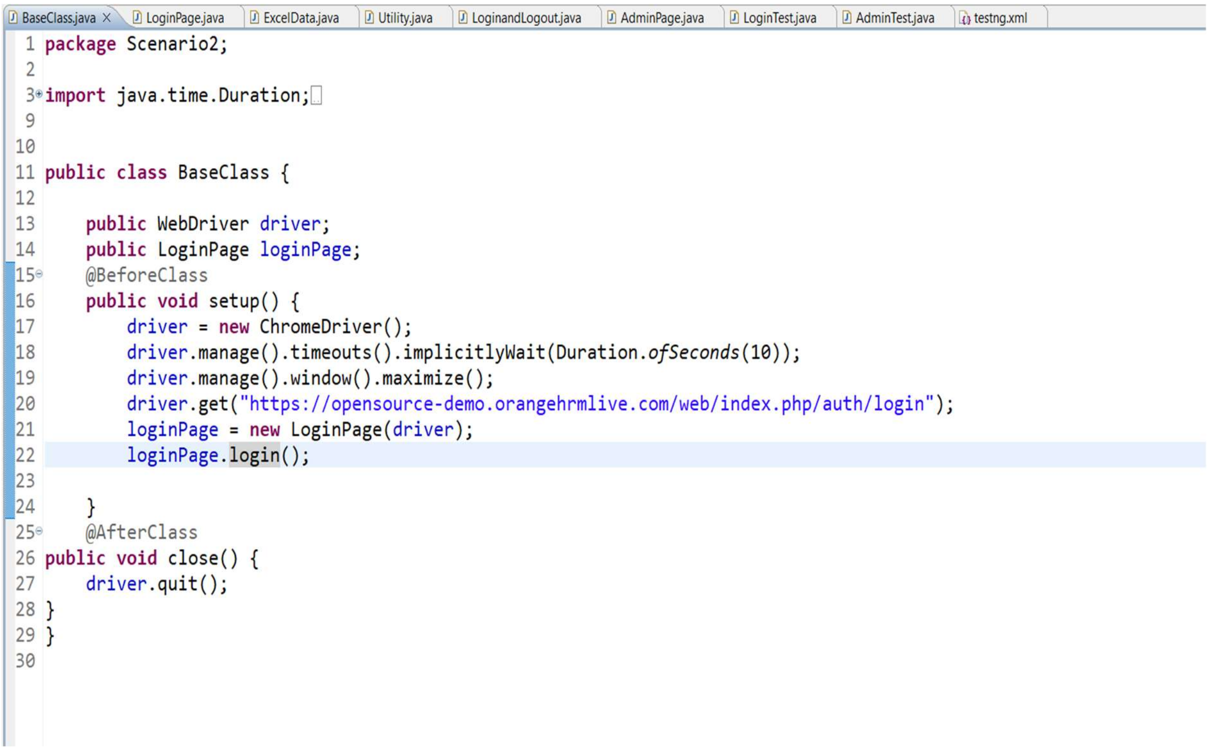
2. Create test case for search For Existing Employee search ByUserName(): here send username Admin to username text box and click on the search button and display total record found and refresh page.

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

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

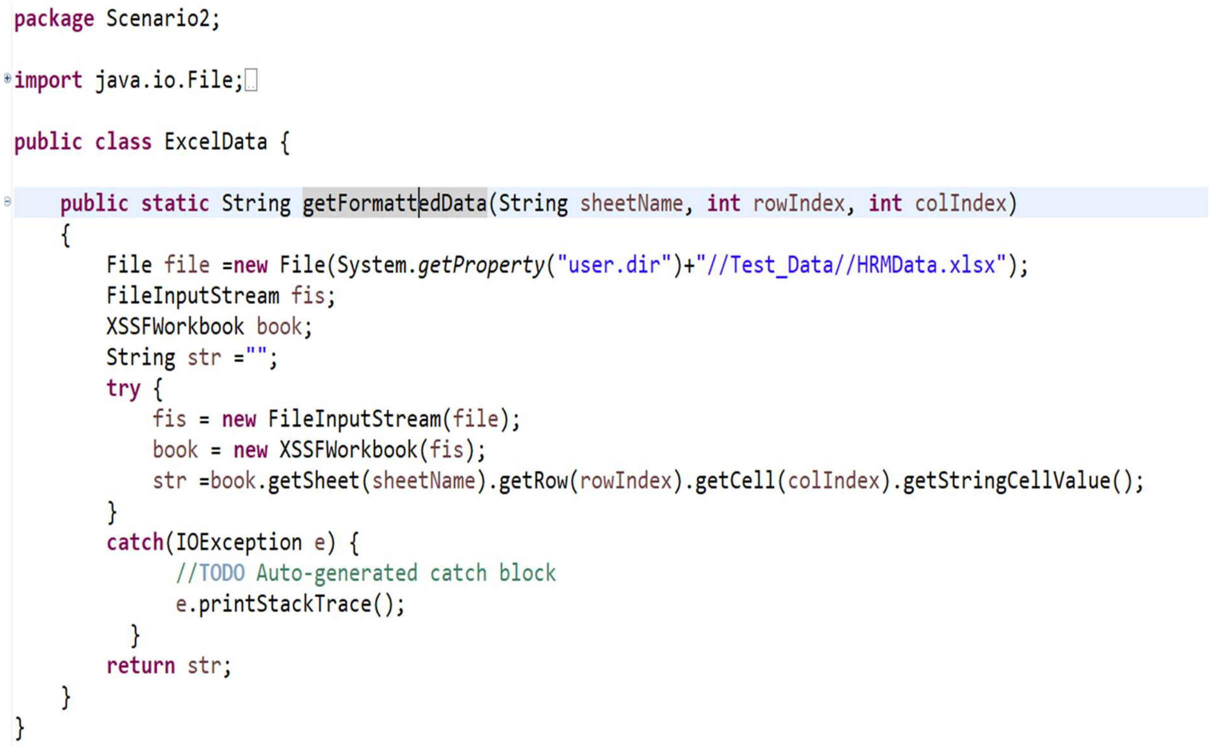
Note: In the Second scenario use Selenium +TestNG and Design Page Object Model where you can implement Object Oriented Principle

BaseClass:



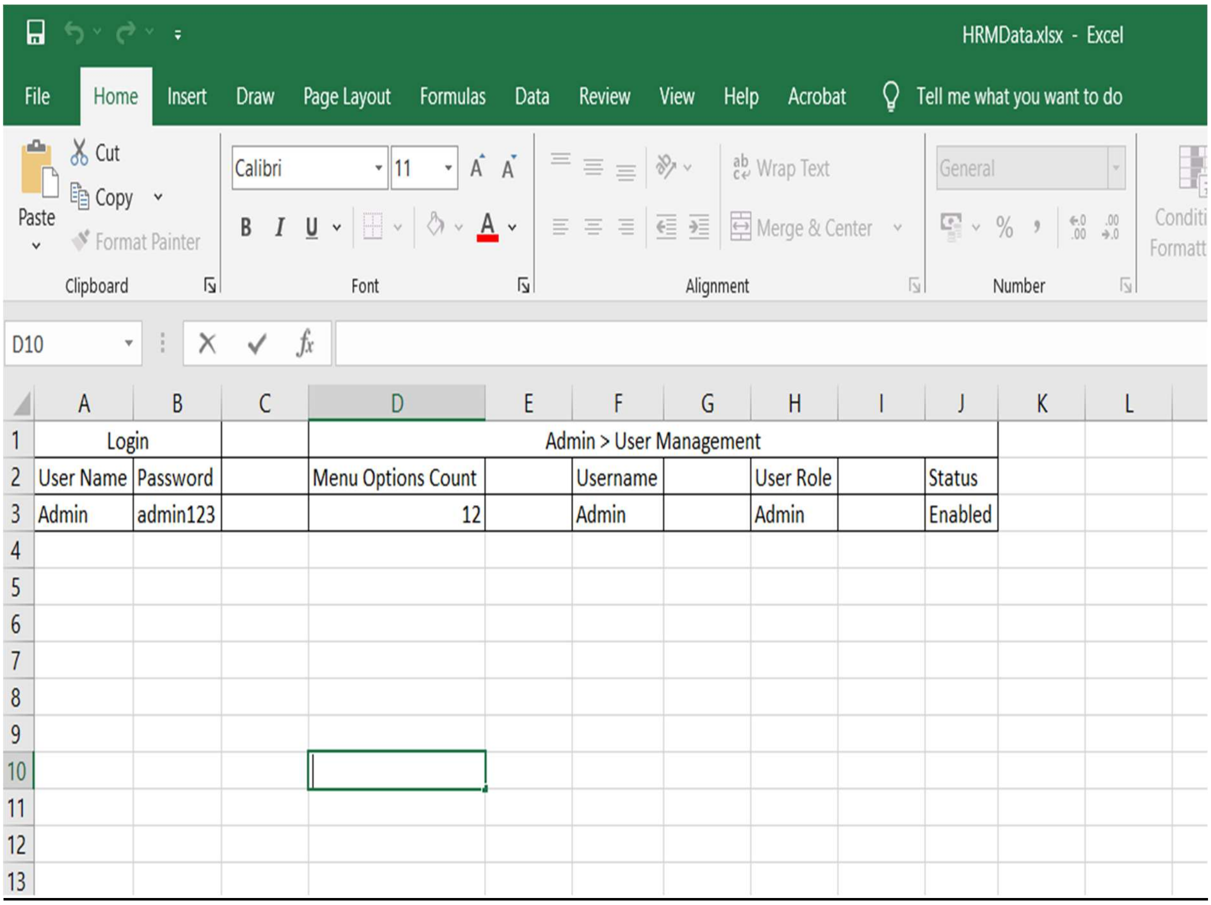
```java
package Scenario2;

import java.time.Duration;


public class BaseClass {

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

    }
    @AfterClass
public void close() {
    driver.quit();
}
}

```

Excel Data:



```
package Scenario2;

import java.io.File;

public class ExcelData {

    public static String getFormattedData(String sheetName, int rowIndex, int colIndex)
    {
        File file =new File(System.getProperty("user.dir")+"//Test_Data//HRMData.xlsx");
        FileInputStream fis;
        XSSFWorkbook book;
        String str ="";
        try {
            fis = new FileInputStream(file);
            book = new XSSFWorkbook(fis);
            str =book.getSheet(sheetName).getRow(rowIndex).getCell(colIndex).getStringCellValue();
        }
        catch(IOException e) {
             //TODO Auto-generated catch block
             e.printStackTrace();
          }
        return str;
    }
}
```

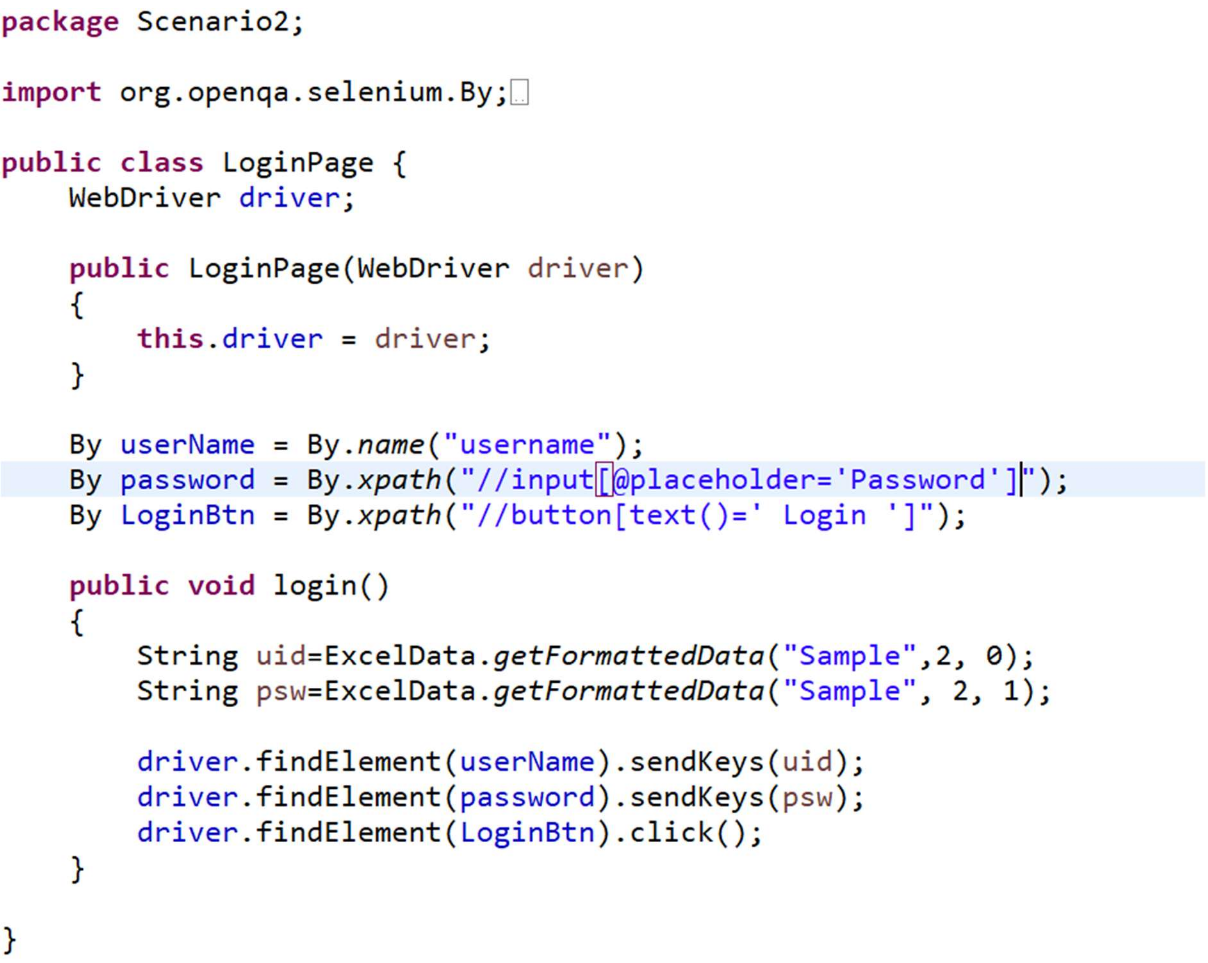
Excel File:



D10

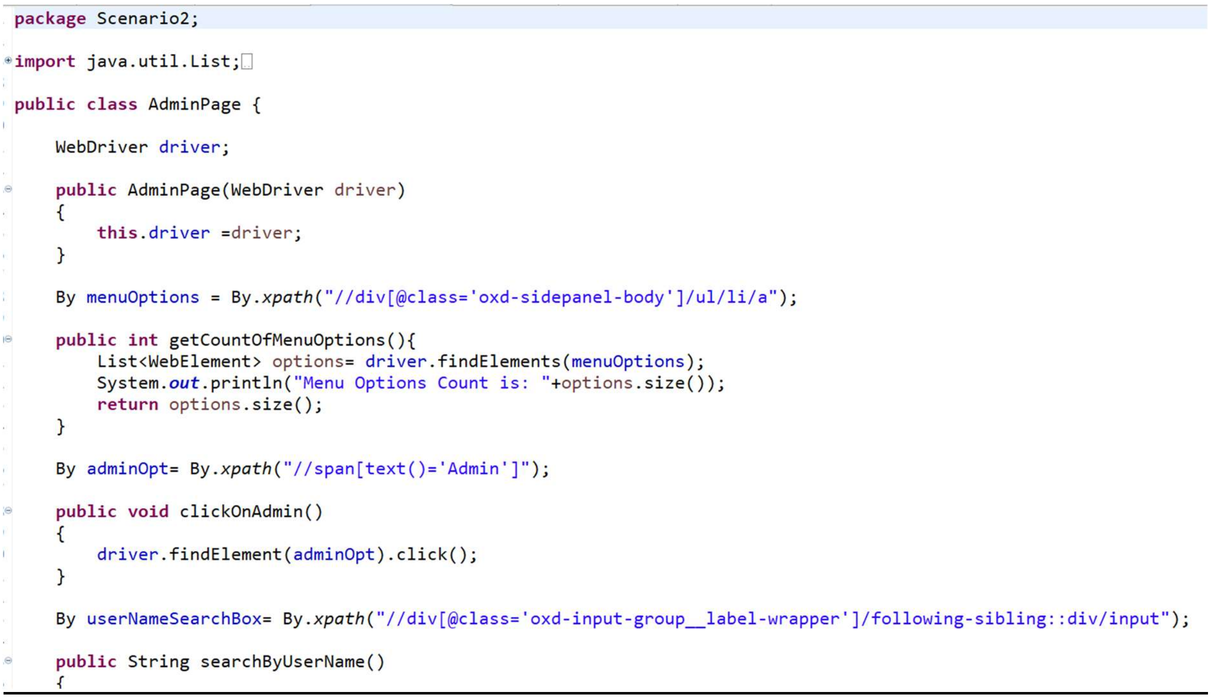
	A	B	C	D	E	F	G	H	I	J	K	L
1	Login			Admin > User Management								
2	User Name	Password		Menu Options Count		Username		User Role		Status		
3	Admin	admin123		12		Admin		Admin		Enabled		
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												

LoginPage:

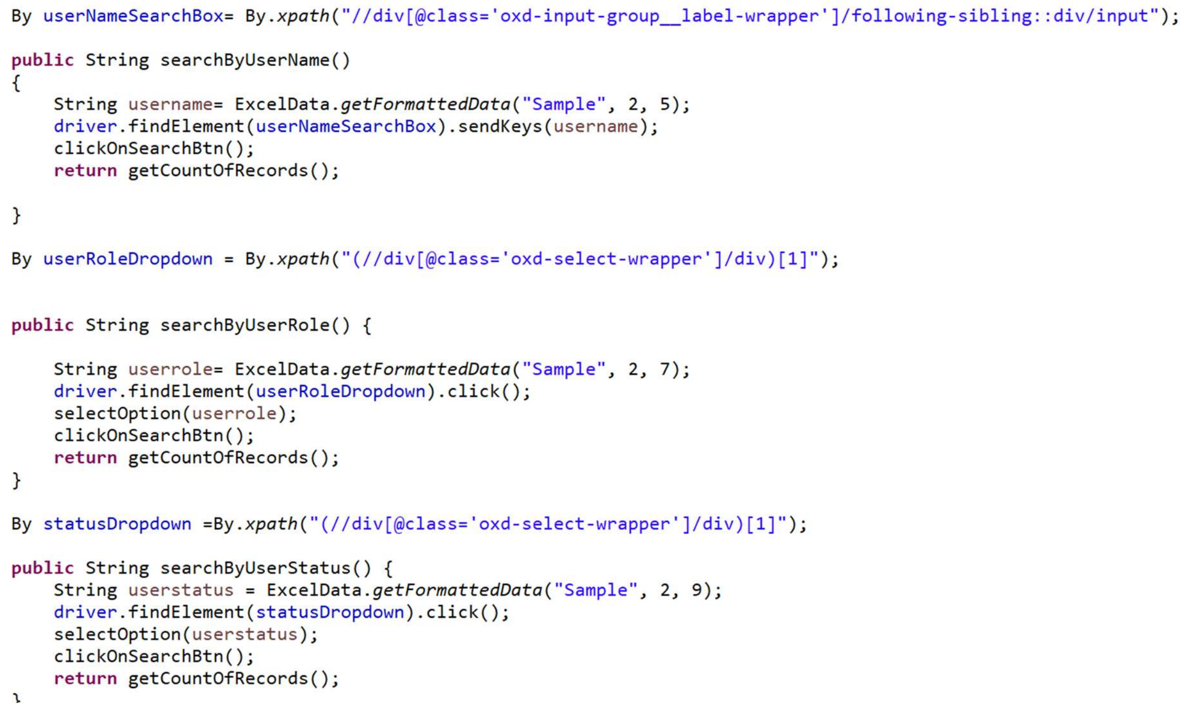


```
package Scenario2;

import org.openqa.selenium.By;

public class LoginPage {
    WebDriver driver;

    public LoginPage(WebDriver driver)
    {
        this.driver = driver;
    }

    By userName = By.name("username");
    By password = By.xpath("//input[@placeholder='Password']");
    By LoginBtn = By.xpath("//button[text()=' Login ']");

    public void login()
    {
        String uid=ExcelData.getFormattedData("Sample",2, 0);
        String psw=ExcelData.getFormattedData("Sample", 2, 1);

        driver.findElement(userName).sendKeys(uid);
        driver.findElement(password).sendKeys(psw);
        driver.findElement(LoginBtn).click();
    }

}
```

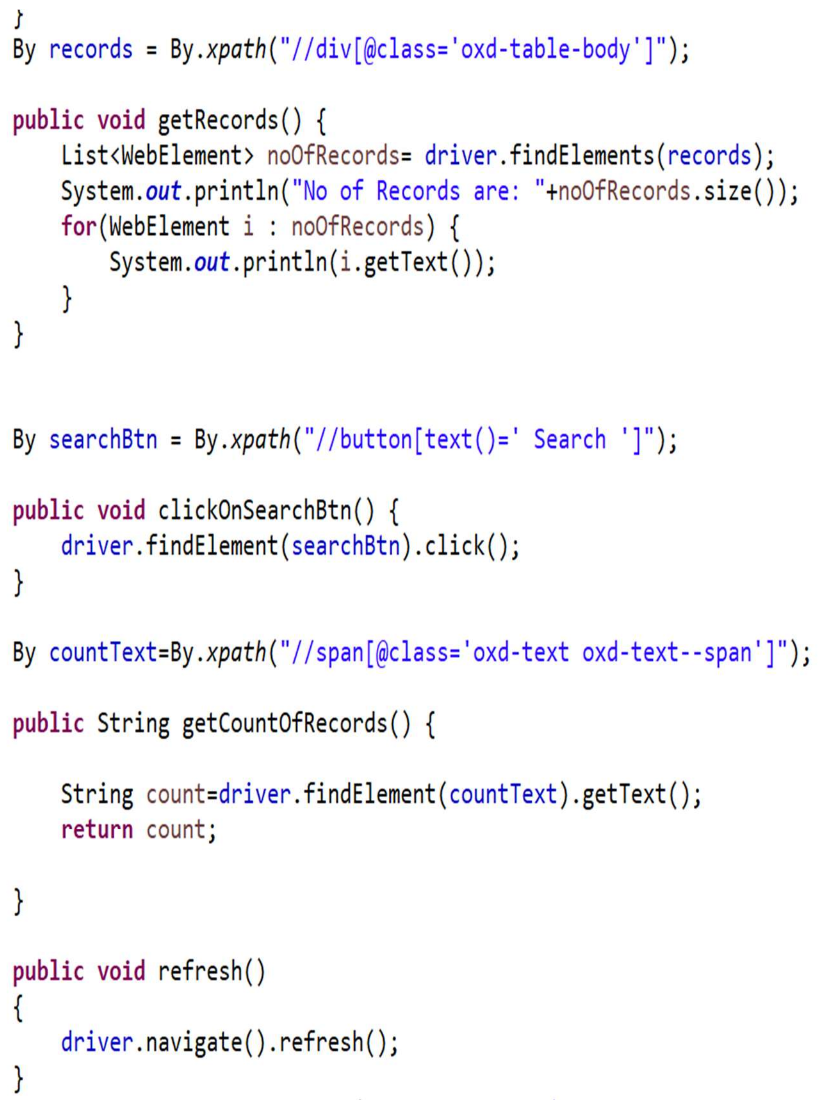
AdminPage:



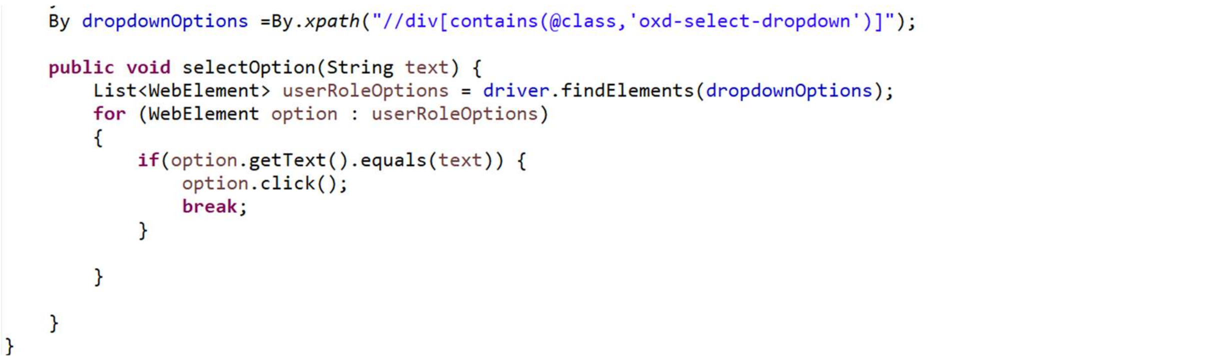
```
package Scenario2;

import java.util.List;

public class AdminPage {

    WebDriver driver;

    public AdminPage(WebDriver driver)
    {
        this.driver =driver;
    }

    By menuOptions = By.xpath("//div[@class='oxd-sidepanel-body']/ul/li/a");

    public int getCountOfMenuOptions(){
        List<WebElement> options= driver.findElements(menuOptions);
        System.out.println("Menu Options Count is: "+options.size());
        return options.size();
    }

    By adminOpt= By.xpath("//span[text()='Admin']");

    public void clickOnAdmin()
    {
        driver.findElement(adminOpt).click();
    }

    By userNameSearchBox= By.xpath("//div[@class='oxd-input-group__label-wrapper']/following-sibling::div/input");

    public String searchByUserName()
    {
```



```
By userNameSearchBox= By.xpath("//div[@class='oxd-input-group__label-wrapper']/following-sibling::div/input");

public String searchByUserName()
{
    String username= ExcelData.getFormattedData("Sample", 2, 5);
    driver.findElement(userNameSearchBox).sendKeys(username);
    clickOnSearchBtn();
    return getCountOfRecords();

}

By userRoleDropdown = By.xpath("(//div[@class='oxd-select-wrapper']/div)[1]");


public String searchByUserRole() {

    String userrole= ExcelData.getFormattedData("Sample", 2, 7);
    driver.findElement(userRoleDropdown).click();
    selectOption(userrole);
    clickOnSearchBtn();
    return getCountOfRecords();
}

By statusDropdown =By.xpath("(//div[@class='oxd-select-wrapper']/div)[1]");

public String searchByUserStatus() {
    String userstatus = ExcelData.getFormattedData("Sample", 2, 9);
    driver.findElement(statusDropdown).click();
    selectOption(userstatus);
    clickOnSearchBtn();
    return getCountOfRecords();
```

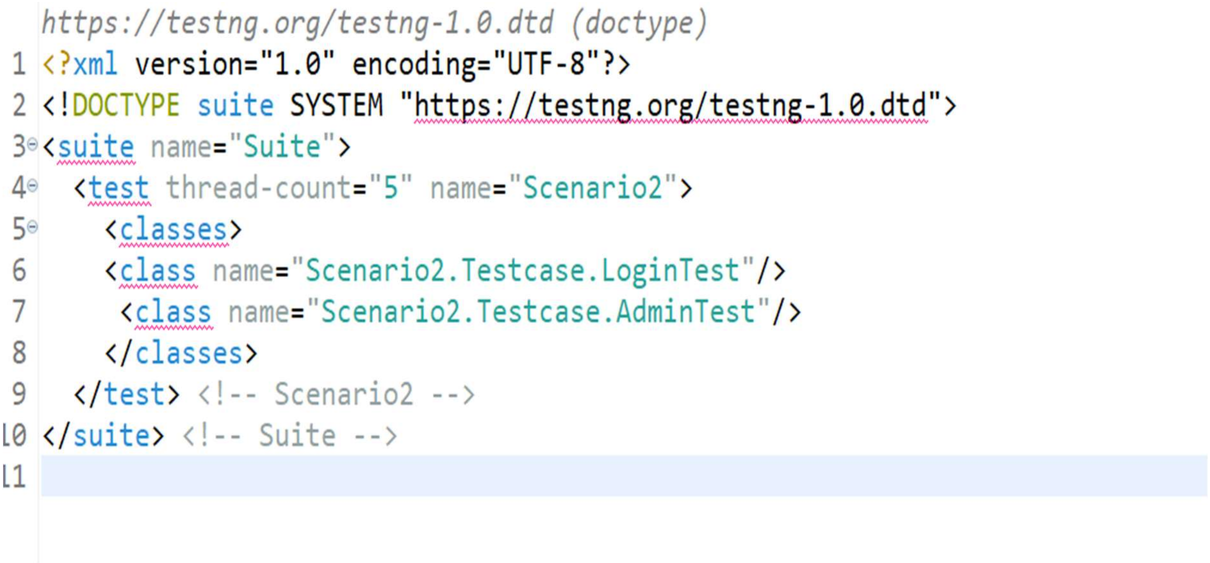


```
}
By records = By.xpath("//div[@class='oxd-table-body']");

public void getRecords() {
    List<WebElement> noOfRecords= driver.findElements(records);
    System.out.println("No of Records are: "+noOfRecords.size());
    for(WebElement i : noOfRecords) {
        System.out.println(i.getText());
    }
}



By searchBtn = By.xpath("//button[text()=' Search ']");

public void clickOnSearchBtn() {
    driver.findElement(searchBtn).click();
}

By countText=By.xpath("//span[@class='oxd-text oxd-text--span']");

public String getCountOfRecords() {

    String count=driver.findElement(countText).getText();
    return count;

}

public void refresh()
{
    driver.navigate().refresh();
}
```



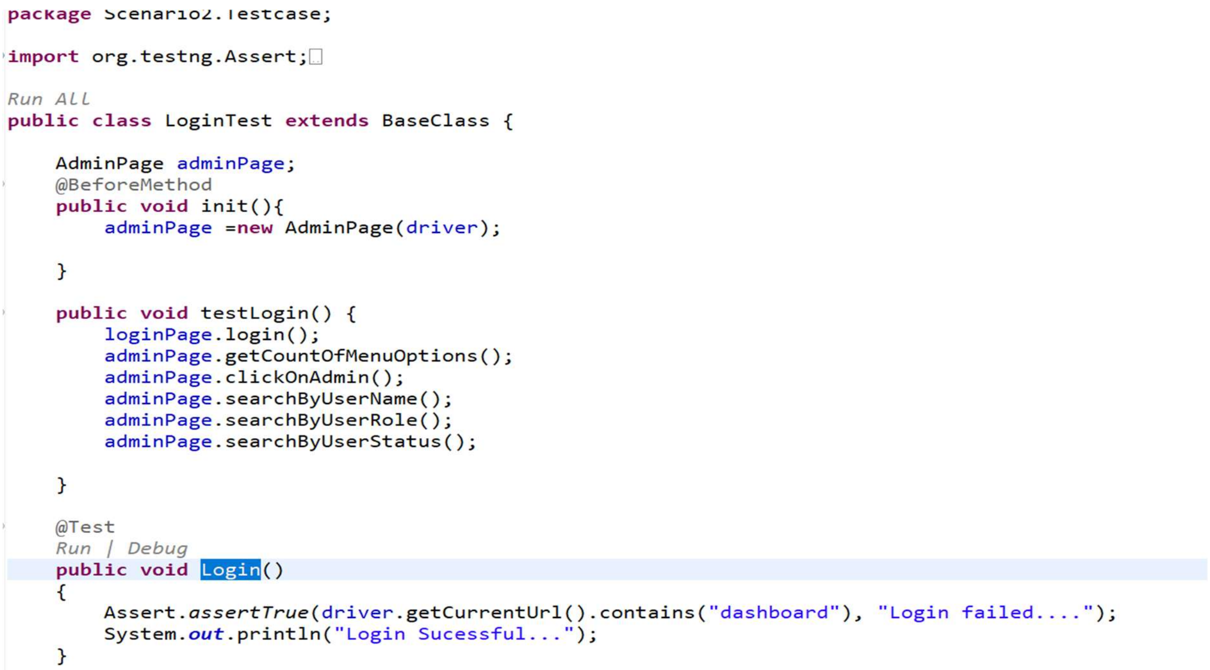
```
    By dropdownOptions =By.xpath("//div[contains(@class,'oxd-select-dropdown')]");

    public void selectOption(String text) {
        List<WebElement> userRoleOptions = driver.findElements(dropdownOptions);
        for (WebElement option : userRoleOptions)
        {
            if(option.getText().equals(text)) {
                option.click();
                break;
            }

        }

    }
}
```

XML File:



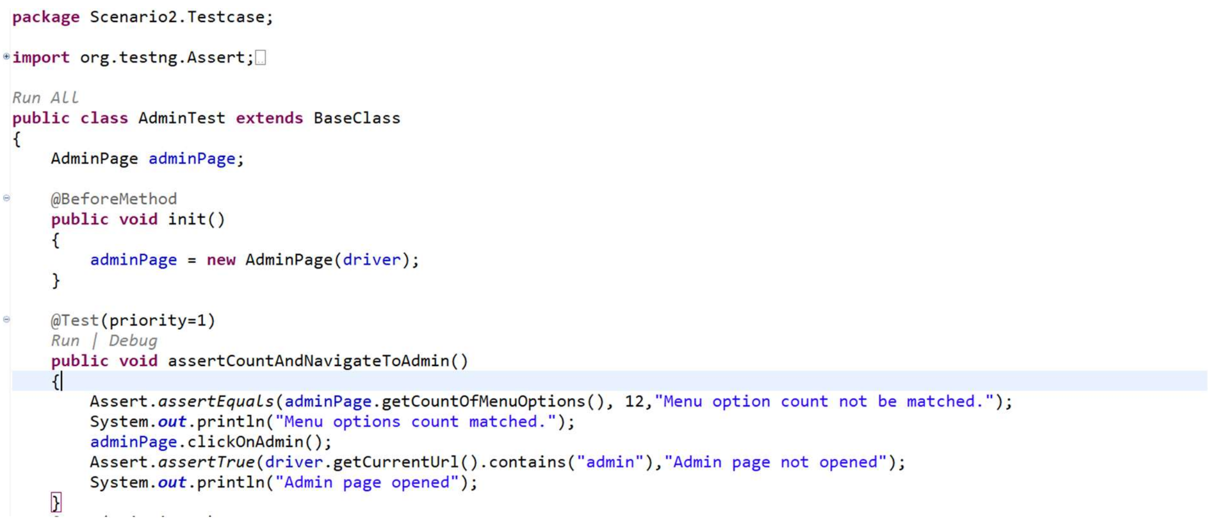
```
https://testng.org/testng-1.0.dtd (doctype)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Scenario2">
    <classes>
    <class name="Scenario2.Testcase.LoginTest"/>
     <class name="Scenario2.Testcase.AdminTest"/>
    </classes>
  </test> <!-- Scenario2 -->
</suite> <!-- Suite -->
```

LoginTest:



```
package Scenario2.Testcase;

import org.testng.Assert;□

Run All
public class LoginTest extends BaseClass {

    AdminPage adminPage;
    @BeforeMethod
    public void init(){
        adminPage =new AdminPage(driver);

    }

    public void testLogin() {
        loginPage.login();
        adminPage.getCountOfMenuOptions();
        adminPage.clickOnAdmin();
        adminPage.searchByUserName();
        adminPage.searchByUserRole();
        adminPage.searchByUserStatus();

    }

    @Test
    Run | Debug
    public void Login()
    {
        Assert.assertTrue(driver.getCurrentUrl().contains("dashboard"), "Login failed....");
        System.out.println("Login Sucessful...");
    }
```

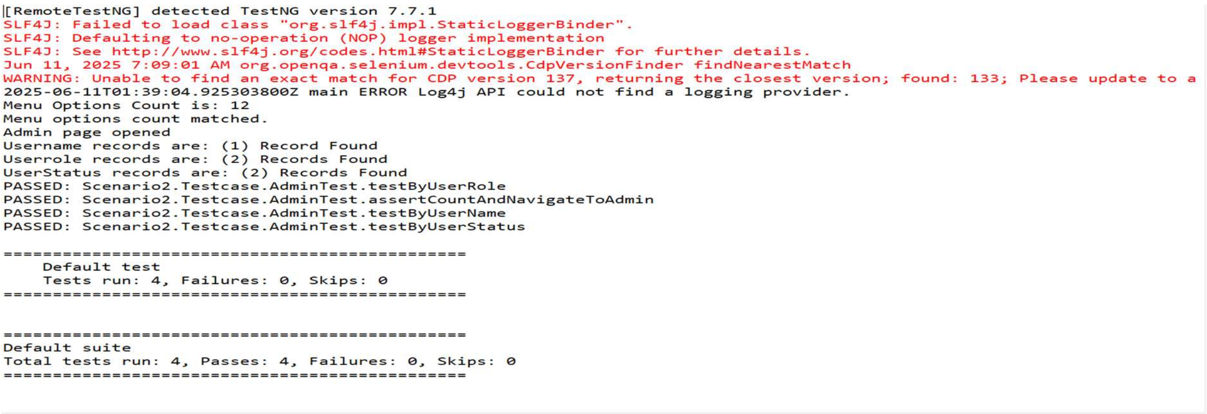
AdminTest:



```
package Scenario2.Testcase;

import org.testng.Assert;

Run All
public class AdminTest extends BaseClass
{
    AdminPage adminPage;

    @BeforeMethod
    public void init()
    {
        adminPage = new AdminPage(driver);
    }

    @Test(priority=1)
    Run | Debug
    public void assertCountAndNavigateToAdmin()
    {
        Assert.assertEquals(adminPage.getCountOfMenuOptions(), 12,"Menu option count not be matched.");
        System.out.println("Menu options count matched.");
        adminPage.clickOnAdmin();
        Assert.assertTrue(driver.getCurrentUrl().contains("admin"),"Admin page not opened");
        System.out.println("Admin page opened");
    }
```

```
    @Test(priority=2)
    Run | Debug
    public void testByUserName()
    {
        adminPage.clickOnAdmin();
        String records = adminPage.searchByUserName();
        System.out.println("Username records are: "+records);
        adminPage.refresh();
    }

    @Test(priority=3)

Run | Debug
public void testByUserRole()
{
        adminPage.clickOnAdmin();
        String records=adminPage.searchByUserRole();
        System.out.println("Userrole records are: "+records);
        adminPage.refresh();
}
    @Test(priority=4)
    Run | Debug
    public void testByUserStatus()
    {
        adminPage.clickOnAdmin();
        String records=adminPage.searchByUserStatus();
        System.out.println("UserStatus records are: "+records);
        adminPage.refresh();
    }

}
```

Output:



```
[RemoteTestNG] detected TestNG version 7.7.1
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Jun 11, 2025 7:09:01 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 137, returning the closest version; found: 133; Please update to a
2025-06-11T01:39:04.925303800Z main ERROR Log4j API could not find a logging provider.
Menu Options Count is: 12
Menu options count matched.
Admin page opened
Username records are: (1) Record Found
Userrole records are: (2) Records Found
UserStatus records are: (2) Records Found
PASSED: Scenario2.Testcase.AdminTest.testByUserRole
PASSED: Scenario2.Testcase.AdminTest.assertCountAndNavigateToAdmin
PASSED: Scenario2.Testcase.AdminTest.testByUserName
PASSED: Scenario2.Testcase.AdminTest.testByUserStatus

===============================================
    Default test
    Tests run: 4, Failures: 0, Skips: 0
===============================================


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