

CAPSTONE PROJECT

HUMAN RESOURCE MANAGEMENT (AUTOMATION)

- **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 WHICH OFFERS SERVICES LIKE BELOW:**
 1. Adding the new employees
 2. Modifying existing employees
 3. Deleting existing employees
 4. Searching employees
 5. Raising request for hiring new employees
 6. Approving new request
 7. Submitting time sheet
 8. Approving time sheet etc....
- **MODULES IN SCOPE FOR AUTOMATION TESTING**
 1. Login
 2. Admin

2.1 USER MANAGEMENT OBJECTIVES: As a tester you need to understand below requirement and prepare automation script for the same using Selenium and TestNG.

NOTE: Both scenarios are different.

Below are the requirement for OHRM application:

MODULE 1 : LOGIN

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

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

CODE:

OHRM Project.java:

```
package Scenario1;

import org.testng.annotations.Test;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.markuputils.ExtentColor;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import com.aventstack.extentreports.reporter.ExtentSparkReporter;

import com.aventstack.extentreports.reporter.configuration.Theme;

import Utility.Utility;

import org.testng.annotations.BeforeMethod;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.time.Duration;
```

```
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;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.DataProvider;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.AfterTest;

public class OHRM_Project {

    public static WebDriver driver = new ChromeDriver();

    String fPath =
        "D:\\Ruksar_SeleniumDemos\\MyMavenProject\\ExcelFiles\\LoginData.xlsx";

    File file;

    FileInputStream fis;

    XSSFWorkbook wb;

    XSSFSheet sheet;

    XSSFRow row;

    XSSFCell cell;

    ExtentSparkReporter htmlreport;

    ExtentReports report;

    ExtentTest test;

    String extentreportpath =
        "D:\\Ruksar_SeleniumDemos\\MyMavenProject\\ExtentReport\\report.html";

    @Test(dataProvider = "LoginData")
```

```

public void OhrmLogin(String un, String ps) {

    driver.findElement(By.name("username")).sendKeys(un);

    driver.findElement(By.name("password")).sendKeys(ps);

    driver.findElement(By.xpath("//*[@id=\"app\"]/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();

}

```

@DataProvider

```

public Object[][] LoginData() {

    int totalRows = sheet.getPhysicalNumberOfRows();

    String[][] LoginData = new String[totalRows-1][2];

    for (int i = 0; i< totalRows - 1; i++) {

        row = sheet.getRow(i + 1);

        for (int j = 0; j<2; j++) {

            cell = row.getCell(j);

            LoginData[i][j] = cell.getStringCellValue();

        }

    }

    return LoginData;

}

```

@AfterMethod

```

public void afterMethod() throws Exception {

    String current_url = driver.getCurrentUrl();

    Assert.assertTrue(current_url.contains(current_url),

        "URL does not match");

    if (current_url.contains("dashboard")) {

        System.out.println("Test Case Pass....");

        Thread.sleep(2000);

        Utility.getScreenshot(driver);
    }
}

```

```

test= report.createTest("OHRM Valid Login");

test.log(Status.PASS, MarkupHelper.createLabel("OHRM Login:Pass",
ExtentColor.GREEN));

driver.findElement(By.xpath("//*[@id=\"app\"]/div[1]/div[1]/header/div[1]/div[3]/u
l/li/span/i")).click();

driver.findElement(By.LinkText("Logout")).click();

}

else {

System.out.println("Test Case Fail....");

Thread.sleep(20000);

Utility.getScreenshot(driver);

test = report.createTest("OHRM Invalid Login");

test.log(Status.FAIL, MarkupHelper.createLabel("OHRM Login:Fail",
ExtentColor.RED));

}

}

@BeforeTest

public void beforeTest() throws IOException {

file = new File(fPath);

fis = new FileInputStream(file);

wb = new XSSFWorkbook(fis);

sheet = wb.getSheetAt(0);

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

//Extent Report Initialization

htmlreport = new ExtentSparkReporter(extentreportpath);

report = new ExtentReports();

```

```

report.attachReporter(htmlreport);

//Setting environment for Extent report

report.setSystemInfo("Computer Name", "Lenovo");

report.setSystemInfo("Tester Name", "Ruksar");

report.setSystemInfo("OS", "Windows 11");

report.setSystemInfo("Browser", "Google Chrome");

//Configuration of the report

htmlreport.config().setDocumentTitle("ORHM_Project_Report");

htmlreport.config().setReportName("OHRM_Project_Details");

htmlreport.config().setTheme(Theme.DARK);

htmlreport.config().setTimeStampFormat("EEEE MM DD, YYYY HH:mm:ss");

}

@AfterTest

public void afterTest() throws IOException {

    wb.close();

    fis.close();

    report.flush();

    driver.quit();

}

}

```

Utility.java:

```

package Utility;

import java.io.File;

import java.io.IOException;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

```

```

import org.openqa.selenium.io.FileHandler;

public class Utility {

    public static void getScreenshot (WebDriver driver) {

        TakesScreenshot ss = (TakesScreenshot) driver;

        File src = ss.getScreenshotAs(OutputType.FILE);

        File dest = new File("D:\\Ruksar_SeleniumDemos\\OhrmScreenshot" +
            System.currentTimeMillis() + ".png");

        try {

            FileHandler.copy(src, dest);

        }

        catch(IOException e) {

            //TODO Auto-generated Catch Block

            e.printStackTrace();

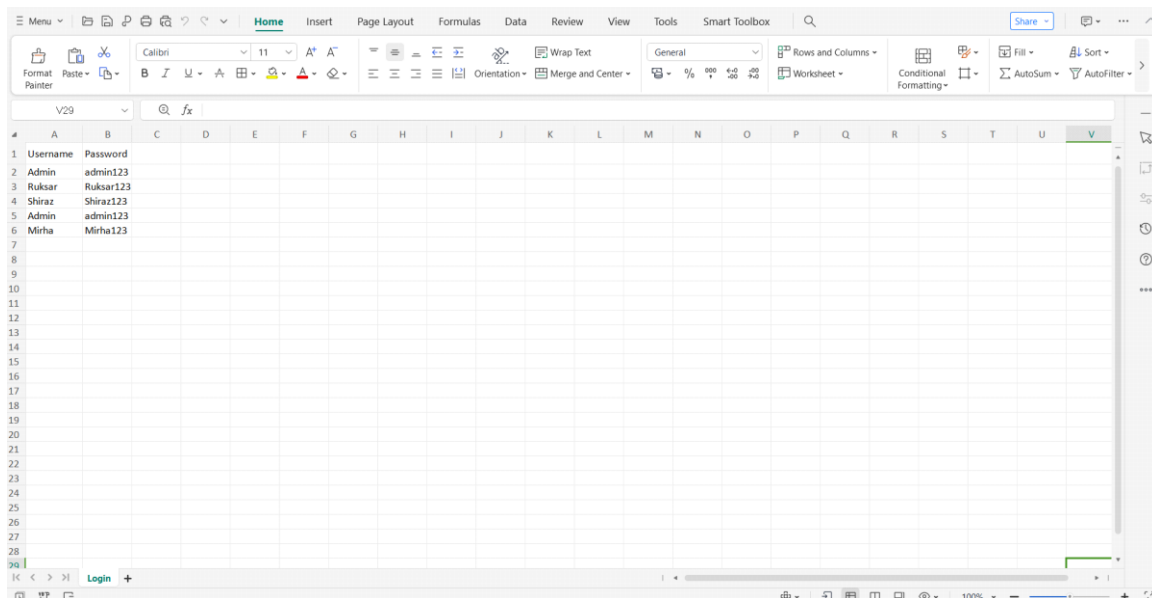
        }

    }

}

```

1. Save data set in Excel File



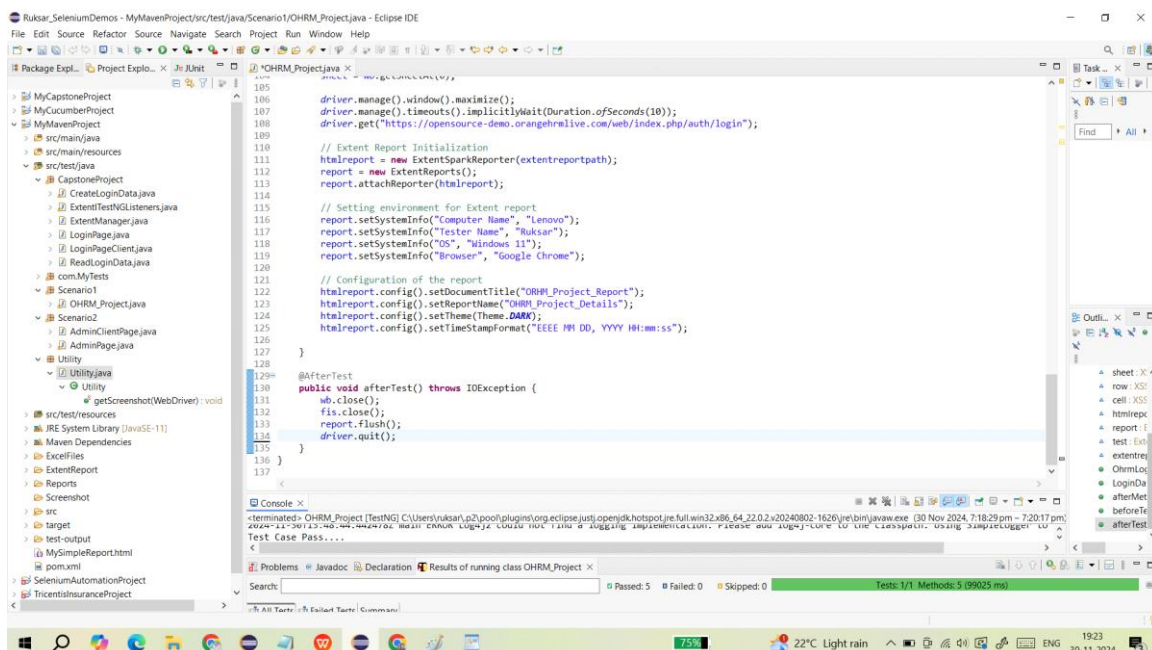
2. Write a script to read data from Excel

```
OHRM_Project.java x
Run All
37 public class OHRM_Project {
38     public static WebDriver driver = new ChromeDriver();
39
40     String fPath = "D:\\Ruksar_SeleniumDemos\\VMyMavenProject\\ExcelFiles\\LoginData.xlsx";
41     File file;
42     FileInputStream fis;
43     XSSFWorkbook wb;
44     XSSFSheet sheet;
45     XSSFRow row;
46     XSSFCell cell;
47
48     ExtentSparkReporter htmReport;
49     ExtentReports report;
50     ExtentTest test;
51     String extentReportPath = "D:\\Ruksar_SeleniumDemos\\VMyMavenProject\\ExtentReport\\report.html";
52
53     @Test(dataProvider = "LoginData")
54
55     Run / Debug
56     public void OHRMLogin(String un, String ps) {
57         driver.findElement(By.name("username")).sendKeys(un);
58         driver.findElement(By.name("password")).sendKeys(ps);
59         driver.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();
60     }
61
62     @DataProvider
63     public Object[][] LoginData() {
64         int totalRows = sheet.getPhysicalNumberOfRows();
65         String[][] LoginData = new String[totalRows-1][2];
66
67         for (int i = 0; i < totalRows - 1; i++) {
68             row = sheet.getRow(i + 1);
69             for (int j = 0; j < 2; j++) {
70                 cell = row.getCell(j);
71                 LoginData[i][j] = cell.getStringCellValue();
72             }
73         }
74         return LoginData;
75     }
76
77     @AfterMethod
78     public void afterMethod() throws Exception {
79         String current_url = driver.getCurrentUrl();
80         Assert.assertTrue(current_url.contains("current_url"), "URL does not match");
81         if (current_url.contains("dashboard")) {
82             System.out.println("Test Case Pass....");
83             Thread.sleep(2000);
84             Utility.getScreenshot(driver);
85             test = report.createTest("OHRM Valid Login");
86             test.log(Status.PASS, MarkupHelper.createLabel("OHRM Login:Pass", ExtentColor.GREEN));
87             driver.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();
88             driver.findElement(By.linkText("Logout")).click();
89         } else {
90             System.out.println("Test Case Fail....");
91             Thread.sleep(20000);
92             Utility.getScreenshot(driver);
93         }
94     }
95 }
```

```
OHRM_Project.java x
61
62     @DataProvider
63     public Object[][] LoginData() {
64         int totalRows = sheet.getPhysicalNumberOfRows();
65         String[][] LoginData = new String[totalRows - 1][2];
66
67         for (int i = 0; i < totalRows - 1; i++) {
68             row = sheet.getRow(i + 1);
69             for (int j = 0; j < 2; j++) {
70                 cell = row.getCell(j);
71                 LoginData[i][j] = cell.getStringCellValue();
72             }
73         }
74         return LoginData;
75     }
76
77     @AfterMethod
78     public void afterMethod() throws Exception {
79         String current_url = driver.getCurrentUrl();
80         Assert.assertTrue(current_url.contains("current_url"), "URL does not match");
81         if (current_url.contains("dashboard")) {
82             System.out.println("Test Case Pass....");
83             Thread.sleep(2000);
84             Utility.getScreenshot(driver);
85             test = report.createTest("OHRM Valid Login");
86             test.log(Status.PASS, MarkupHelper.createLabel("OHRM Login:Pass", ExtentColor.GREEN));
87             driver.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();
88             driver.findElement(By.linkText("Logout")).click();
89         } else {
90             System.out.println("Test Case Fail....");
91             Thread.sleep(20000);
92             Utility.getScreenshot(driver);
93         }
94     }
95 }
```



```
88 driver.findElement(By.xpath("//*[@id='app']/div[1]/div[1]/header/div[1]/div[3]/ul/li/span/1")).click();
89 driver.findElement(By.linkText("Logout")).click();
90 } else {
91     System.out.println("Test Case Fail....");
92     Thread.sleep(20000);
93     Utility.getScreenshot(driver);
94     test = report.createTest("OHRM Invalid Login");
95     test.log(Status.FAIL, MarkupHelper.createLabel("OHRM Login:Fail", ExtentColor.RED));
96 }
97 }
98
99 @BeforeTest
100 public void beforeTest() throws IOException {
101     file = new File(fPath);
102     fis = new FileInputStream(file);
103     wb = new XSSFWorkbook(fis);
104     sheet = wb.getSheetAt(0);
105
106     driver.manage().window().maximize();
107     driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
108     driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");
109
110 // Extent Report Initialization
111 htmlreport = new ExtentSparkReporter(extentreportpath);
112 report = new ExtentReports();
113 report.attachReporter(htmlreport);
114
115 // Setting environment for Extent report
116 report.setSystemInfo("Computer Name", "Lenovo");
117 report.setSystemInfo("Tester Name", "Ruksar");
118 report.setSystemInfo("OS", "Windows 11");
119 report.setSystemInfo("Browser", "Google Chrome");
120 }
```



CONSOLE OUTPUT:

```
Console x
<terminated> OHRM_Project [TestNG] C:\Users\ruksar\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (30 Nov 2024, 7:38:16 pm)
[RemoteTestNG] detected TestNG version 7.10.2
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.
Nov 30, 2024 7:36:41 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 131, returning the closest version; found: 128; Please update to a Selenium version that supports CDP version 131
2024-11-30T14:06:41.785753100Z main ERROR Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger
Test Case Pass
Test Case Fail
Test Case Fail
Test Case Pass
Test Case Fail
PASSED: Scenario1.OHRM_Project.OhrmLogin("Shiraz", "Shiraz123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Admin", "admin123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Mirha", "Mirha123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Ruksar", "Ruksar123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Admin", "admin123")

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

Default suite
Total tests run: 5, Passes: 5, Failures: 0, Skips: 0
=====
```

3. Prepare script for Login and Logout

```
OHRM_Project.java x
52
53 @Test(dataProvider = "LoginData")
54
55 Run | Debug
56 public void OhrmLogin(String un, String ps) {
57     driver.findElement(By.name("username")).sendKeys(un);
58     driver.findElement(By.name("password")).sendKeys(ps);
59     driver.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")).click();
60 }
61
62 @DataProvider
63 public Object[][] LoginData() {
64     int totalRows = sheet.getPhysicalNumberOfRows();
65     String[][] LoginData = new String[totalRows - 1][2];
66
67     for (int i = 0; i < totalRows - 1; i++) {
68         row = sheet.getRow(i + 1);
69         for (int j = 0; j < 2; j++) {
70             cell = row.getCell(j);
71             LoginData[i][j] = cell.getStringCellValue();
72         }
73     }
74     return LoginData;
75 }
76
77
78 @AfterMethod
79 public void afterMethod() throws Exception {
80     String current_url = driver.getCurrentUrl();
81     Assert.assertTrue(current_url.contains(current_url), "URL does not match");
82     if (current_url.contains("dashboard")) {
83         System.out.println("Test Case Pass....");
84     }
85 }
```

```
OHRM_Project.java x
66
67     for (int i = 0; i < totalRows - 1; i++) {
68         row = sheet.getRow(i + 1);
69         for (int j = 0; j < 2; j++) {
70             cell = row.getCell(j);
71             LoginData[i][j] = cell.getStringCellValue();
72         }
73     }
74     return LoginData;
75 }
76
77
78 @AfterMethod
79 public void afterMethod() throws Exception {
80     String current_url = driver.getCurrentUrl();
81     Assert.assertTrue(current_url.contains(current_url), "URL does not match");
82     if (current_url.contains("dashboard")) {
83         System.out.println("Test Case Pass....");
84         Thread.sleep(2000);
85         Utility.getScreenshot(driver);
86         test = report.createTest("OHRM Valid Login");
87         test.log(Status.PASS, MarkupHelper.createLabel("OHRM Login:Pass", ExtentColor.GREEN));
88         driver.findElement(By.xpath("//*[@id='app']/div[1]/div[1]/header/div[1]/div[3]/ul/li/span/i")).click();
89         driver.findElement(By.linkText("Logout")).click();
90     } else {
91         System.out.println("Test Case Fail....");
92         Thread.sleep(2000);
93         Utility.getScreenshot(driver);
94         test = report.createTest("OHRM Invalid Login");
95         test.log(Status.FAIL, MarkupHelper.createLabel("OHRM Login:Fail", ExtentColor.RED));
96     }
97 }
98
```

4.

Perform Assertion for valid data set(use Username: Admin and Password: admin123) test case should be pass and invalid data (Other than given data) test case should be fail.

```
OHRM_Project.java x
76
77 }
78
79 @AfterMethod
80 public void afterMethod() throws Exception {
81     String current_url = driver.getCurrentUrl();
82     Assert.assertTrue(current_url.contains(current_url), "URL does not match");
83     if (current_url.contains("dashboard")) {
84         System.out.println("Test Case Pass....");
85         Thread.sleep(2000);
86         Utility.getScreenshot(driver);
87         test = report.createTest("OHRM Valid Login");
88         test.log(Status.PASS, MarkupHelper.createLabel("OHRM Login:Pass", ExtentColor.GREEN));
89         driver.findElement(By.xpath("//*[@id='app']/div[1]/div[1]/header/div[1]/div[3]/ul/li/span/i")).click();
90         driver.findElement(By.linkText("Logout")).click();
91     } else {
92         System.out.println("Test Case Fail....");
93         Thread.sleep(2000);
94         Utility.getScreenshot(driver);
95         test = report.createTest("OHRM Invalid Login");
96         test.log(Status.FAIL, MarkupHelper.createLabel("OHRM Login:Fail", ExtentColor.RED));
97     }
98 }
99
100 @BeforeTest
101 public void beforeTest() throws IOException {
102     file = new File(filePath);
103     fis = new FileInputStream(file);
104     wb = new XSSFWorkbook(fis);
105     sheet = wb.getSheetAt(0);
106
107     driver.manage().window().maximize();
108     driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
109 }
110
```

CONSOLE OUTPUT:

```
Console x
<terminated>: OHRM_Project [TestNG] C:\Users\ruksar\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (30 Nov 2024, 7:38:16 pm)
[RemoteTestNG] detected TestNG version 7.10.2
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.
Nov 30, 2024 7:36:41 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 131, returning the closest version; found: 128; Please update to a Selenium version that supports CDP version 131
2024-11-30T14:06:41.785753100Z main ERROR Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger
Test Case Pass
Test Case Fail
Test Case Fail
Test Case Pass
Test Case Fail
PASSED: Scenario1.OHRM_Project.OhrmLogin("Shiraz", "Shiraz123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Admin", "admin123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Mihra", "Mihra123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Ruksar", "Ruksar123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Admin", "admin123")

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

Default suite
Total tests run: 5, Passes: 5, Failures: 0, Skips: 0
=====
```

5. Capture Screenshot for every Login functionality

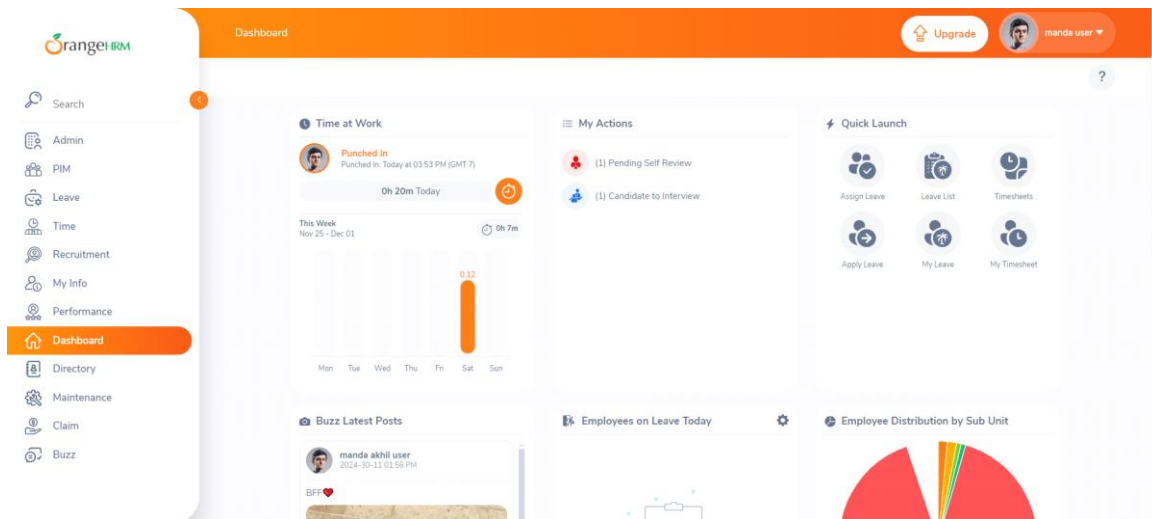
```
Utility.java
package Utility;

import java.io.File;

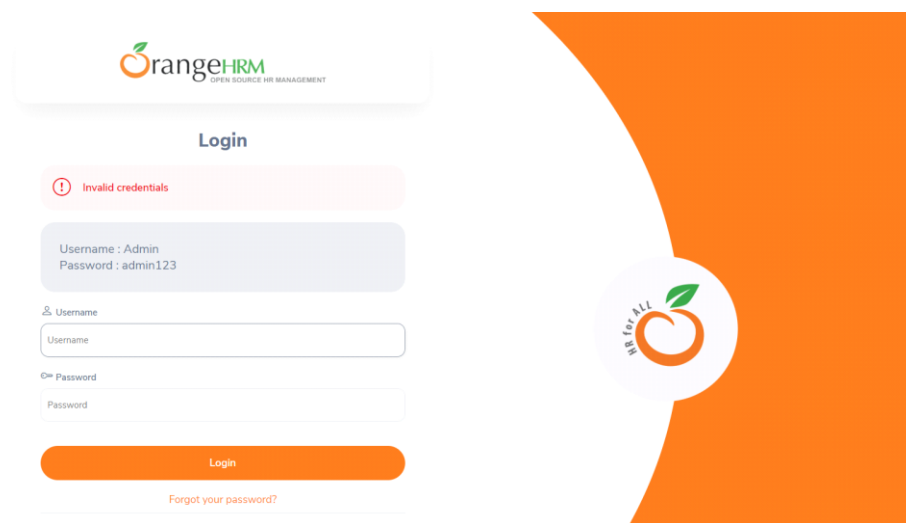
public class Utility {

    public static void getScreenshot(WebDriver driver) {
        TakesScreenshot ss = (TakesScreenshot) driver;
        File src = ss.getScreenshotAs(OutputType.FILE);
        File dest = new File("D:\\Ruksar_SeleniumDemos\\OhrmScreenshot" + System.currentTimeMillis() + ".png");
        try {
            FileHandler.copy(src, dest);
        } catch (IOException e) {
            // TODO Auto-generated Catch Block
            e.printStackTrace();
        }
    }
}
```

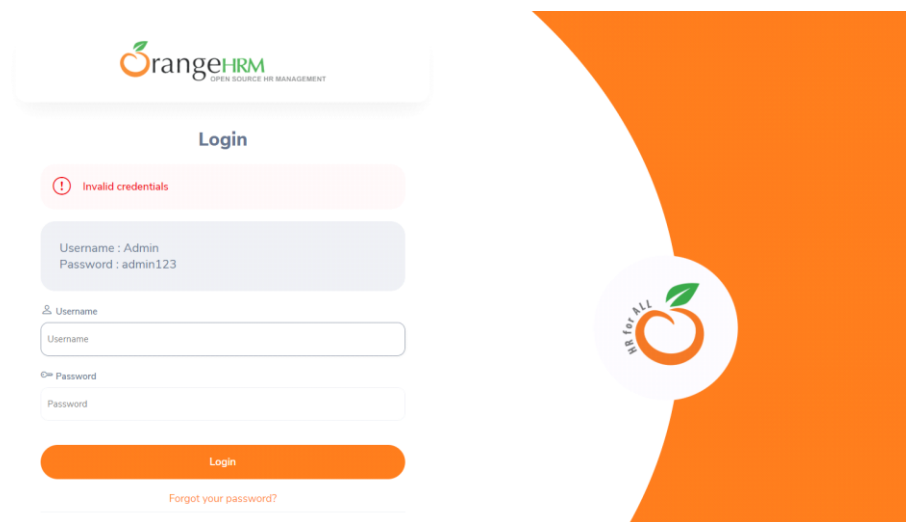
LOGIN 1: Valid data (Username : Admin and Password: admin123)



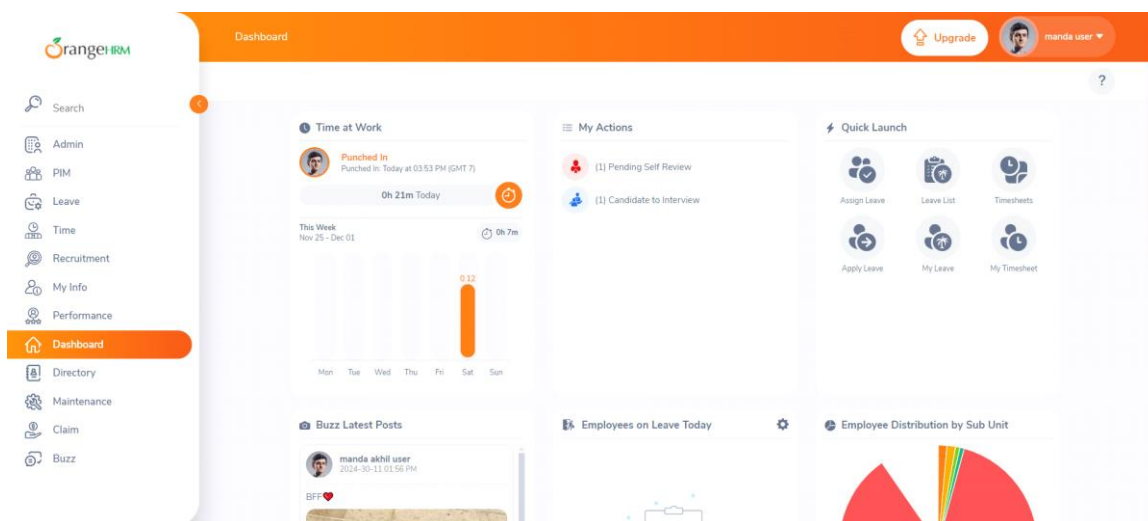
LOGIN 2: Invalid data (Username: Ruksar and Password: Ruksar123)




LOGIN 3: Invalid data (Username: Shiraz and Password: Shiraz123)



LOGIN 4: Valid data (Username : Admin and Password: admin123)



LOGIN 5: Invalid data (Username: Mirha and Password: Mirha123)



Login

Invalid credentials

Username : Admin
Password : admin123

Username

Password

Login

[Forgot your password?](#)



6. Generate Extent Report for the same.

```

OHRM_Project.java x
Run ALL
37 public class OHRM_Project {
38     public static WebDriver driver = new ChromeDriver();
39
40     String fPath = "D:\\Ruksar_SeleniumDemos\\VMyMavenProject\\ExcelFiles\\LoginData.xlsx";
41     File file;
42     FileInputStream fis;
43     XSSFWorkbook wb;
44     XSSFSheet sheet;
45     XSSFForm row;
46     XSSFCell cell;
47
48     ExtentSparkReporter htmlreport;
49     ExtentReports report;
50     ExtentTest test;
51     String extentreportpath = "D:\\Ruksar_SeleniumDemos\\VMyMavenProject\\ExtentReport\\report.html";
52
53     @Test(dataProvider = "LoginData")
54
55     Run / Debug
56     public void OHRMLogin(String un, String ps) {
57         driver.findElement(By.name("username")).sendKeys(un);
58         driver.findElement(By.name("password")).sendKeys(ps);
59         driver.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[2]/form/div[3]/button")).click();
60     }
61
62     @DataProvider
63     public Object[][] LoginData() {
64         int totalRows = sheet.getPhysicalNumberOfRows();
65         String[][] LoginData = new String[totalRows - 1][2];
66
67         for (int i = 0; i < totalRows - 1; i++) {
68             row = sheet.getRow(i + 1);
69             for (int j = 0; j < 2; j++) {
70                 cell = row.getCell(j);
71                 LoginData[i][j] = cell.getStringCellValue();
72             }
73         }
74         return LoginData;
75     }
76
77     @AfterMethod
78     public void afterMethod() throws Exception {
79         String current_url = driver.getCurrentUrl();
80         Assert.assertTrue(current_url.contains(current_url), "URL does not match");
81         if (current_url.contains("dashboard")) {
82             System.out.println("Test Case Pass....");
83             Thread.sleep(2000);
84             Utility.getScreenshot(driver);
85             test = report.createTest("OHRM Valid Login");
86             test.log(Status.PASS, MarkupHelper.createLabel("OHRM Login:Pass", ExtentColor.GREEN));
87             driver.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[3]/ul/li/span/i")).click();
88             driver.findElement(By.linkText("Logout")).click();
89         } else {
90             System.out.println("Test Case Fail....");
91             Thread.sleep(2000);
92             Utility.getScreenshot(driver);
93         }
94     }
95 }

```

```

OHRM_Project.java x
61
62 @DataProvider
63 public Object[][] LoginData() {
64     int totalRows = sheet.getPhysicalNumberOfRows();
65     String[][] LoginData = new String[totalRows - 1][2];
66
67     for (int i = 0; i < totalRows - 1; i++) {
68         row = sheet.getRow(i + 1);
69         for (int j = 0; j < 2; j++) {
70             cell = row.getCell(j);
71             LoginData[i][j] = cell.getStringCellValue();
72         }
73     }
74     return LoginData;
75 }
76
77 @AfterMethod
78 public void afterMethod() throws Exception {
79     String current_url = driver.getCurrentUrl();
80     Assert.assertTrue(current_url.contains(current_url), "URL does not match");
81     if (current_url.contains("dashboard")) {
82         System.out.println("Test Case Pass....");
83         Thread.sleep(2000);
84         Utility.getScreenshot(driver);
85         test = report.createTest("OHRM Valid Login");
86         test.log(Status.PASS, MarkupHelper.createLabel("OHRM Login:Pass", ExtentColor.GREEN));
87         driver.findElement(By.xpath("//*[@id='app']/div[1]/div/div[1]/div/div[3]/ul/li/span/i")).click();
88         driver.findElement(By.linkText("Logout")).click();
89     } else {
90         System.out.println("Test Case Fail....");
91         Thread.sleep(2000);
92         Utility.getScreenshot(driver);
93     }
94 }
95 }

```

```

108 driver.findElement(By.xpath("//*[@id=\"app\"]/div[1]/div[1]/header/div[1]/div[3]/ul/li/span/i")).click();
109 driver.findElement(By.linkText("Logout")).click();
110 } else {
111     System.out.println("Test Case Fail....");
112     Thread.sleep(20000);
113     Utility.getScreenshot(driver);
114     test = report.createTest("OHRM Invalid Login");
115     test.log(Status.FAILED, MarkupHelper.createLabel("OHRM Login:Fail", ExtentColor.RED));
116 }
117 }
118
119 @BeforeTest
120 public void beforeTest() throws IOException {
121     file = new File(filePath);
122     fis = new FileInputStream(file);
123     wb = new XSSFWorkbook(fis);
124     sheet = wb.getSheetAt(0);
125
126     driver.manage().window().maximize();
127     driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
128     driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");
129
130     // Extent Report Initialization
131     htmlReport = new ExtentSparkReporter(extentReportPath);
132     report = new ExtentReports();
133     report.attachReporter(htmlReport);
134
135     // Setting environment for Extent report
136     report.setSystemInfo("Computer Name", "Lenovo");
137     report.setSystemInfo("Tester Name", "Ruksar");
138     report.setSystemInfo("OS", "Windows 11");
139     report.setSystemInfo("Browser", "Google Chrome");
140 }

```

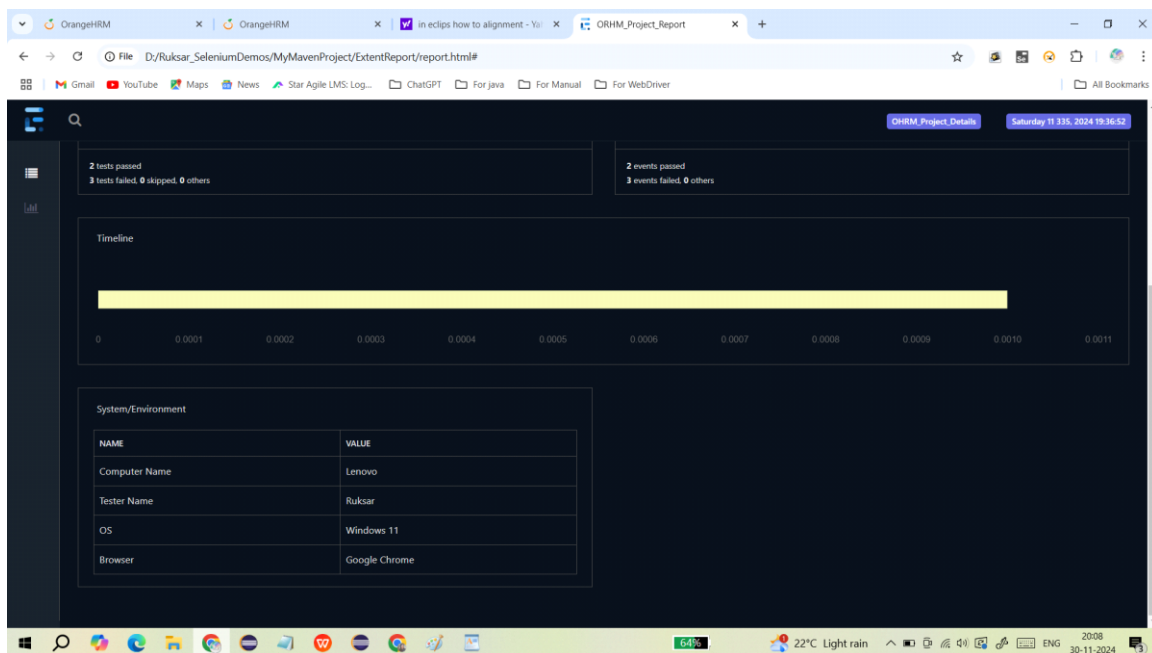
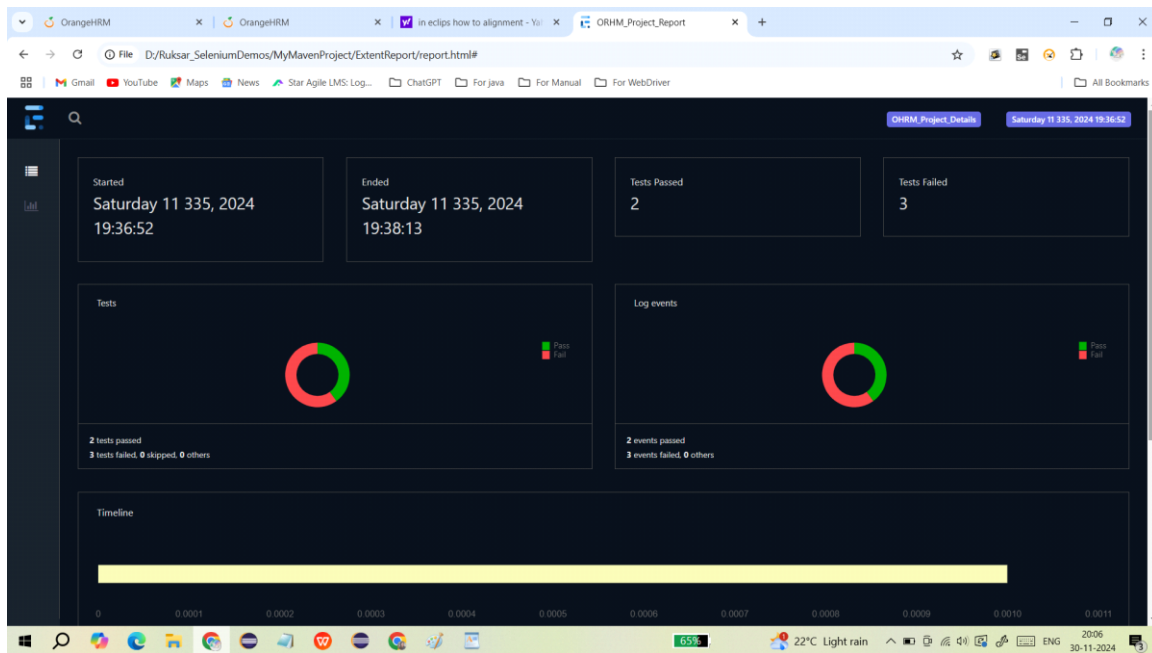
OrangeHRM x OrangeHRM x in eclipse how to alignment - Yo x OHRM_Project_Report x

D:\Ruksar_SeleniumDemos\MyMavenProject\ExtentReport\report.html

Gmail YouTube Maps News Star Agile LMS: Log... ChatGPT For Java For Manual For WebDriver All Bookmarks

OHRM_Project_Details Saturday 11/30/2024 19:36:52

Tests	OHRM Valid Login
OHRM Valid Login 7:36:57 pm / 00:00:00:030	11/30/2024 7:36:57 pm 11/30/2024 7:36:58 pm 00:00:00:030 #test-id=1
OHRM Invalid Login 7:37:22 pm / 00:00:00:001	Pass 7:36:58 pm OHRM Login:Pass
OHRM Invalid Login 7:37:44 pm / 00:00:00:000	
OHRM Valid Login 7:37:49 pm / 00:00:00:000	
OHRM Invalid Login 7:38:13 pm / 00:00:00:001	



CONSOLE OUTPUT:

```
Console x
<-terminated> OHRM_Project [TestNG] C:\Users\ruksar\p2\pool\plugins\org.eclipse.just.openjdk.hotspot.jre.full.win32.x86_64.22.0.2\j220240802-1626\jre\bin\java.exe (30 Nov 2024, 7:38:16 pm)
[RemoteTestNG] detected TestNG version 7.10.2
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.
Nov 30, 2024 7:36:41 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 131, returning the closest version; found: 128; Please update to a Selenium version that supports CDP version 131
2024-11-30T14:06:41.785753100Z main ERROR Log4j2 could not find a logging implementation. Please add log4j-core to the classpath. Using SimpleLogger
Test Case Pass
Test Case Fail
Test Case Fail
Test Case Pass
Test Case Fail
PASSED: Scenario1.OHRM_Project.OhrmLogin("Shiraz", "Shiraz123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Admin", "admin123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Mirha", "Mirha123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Ruksar", "Ruksar123")
PASSED: Scenario1.OHRM_Project.OhrmLogin("Admin", "admin123")

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

Default suite
Total tests run: 5, Passes: 5, Failures: 0, Skips: 0
=====
```

RESULTS:

```
Problems Javadoc Declaration Results of running class OHRM_Project x
Search: Passed: 5 Failed: 0 Skipped: 0 Tests: 1/1 Methods: 5 (100060 ms)
All Tests Failed Tests Summary
v Default suite (5/0/0) (7.203 s)
  v Default test (7.203 s)
    v Scenario1.OHRM_Project
      v OhrmLogin (2.058 s)
        v "Admin","admin123" (2.058 s)
        v OhrmLogin (2.083 s)
        v "Ruksar","Ruksar123" (2.083 s)
        v OhrmLogin (0.502 s)
        v "Shiraz","Shiraz123" (0.502 s)
        v OhrmLogin (0.478 s)
        v "Admin","admin123" (0.478 s)
        v OhrmLogin (2.082 s)
        v "Mirha","Mirha123" (2.082 s)
```

Scenario 2: Create a Page Object Model for two pages LoginPage and Admin Page and write automation script for Login functionality and Admin search feature

Create a page and Class for LoginPage and automate the functionality of OrangeHRM 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 testcase to get all 12 options from leftside menu and print the count which should be 12

from that list click on Admin then Admin page get open.

2. Create test case for search For Existing Employee searchByUsername() : here send username

Admin to username text box and click on search button and display total record found and refresh page.

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

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

CODE:

Login Page.java

```
package Scenario2;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class Login_Page {

    WebDriver driver;

    public Login_Page(WebDriver d) {

        this.driver = d;

        PageFactory.initElements(driver, this);

    }
```

```
@FindBy(name = "username")

private WebElement username;

@FindBy(name = "password")

private WebElement password;

@FindBy(xpath =
"//*[@id=\"app\"]/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")

private WebElement submit;

public WebElement getUsername() {

return username;

}

public WebElement getPassword() {

return password;

}

public WebElement getSubmit() {

return submit;

}

@Test

public void Login(String un, String ps) {

getUsername().sendKeys(un);

getPassword().sendKeys(ps);

try {

Thread.sleep(2000);

} catch (InterruptedException e) {

//TODO Auto-generated catch block

e.printStackTrace();

}
```

```

}

getSubmit().submit();

System.out.println("Login Successful");

}

}

```

Admin Page.java

```

package Scenario2;

import java.util.List;

import org.junit.jupiter.api.Test;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindAll;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import org.testng.Assert;

public class Admin_Page {

    private WebDriver driver;

    public Admin_Page(WebDriver d) {

        this.driver = d;

        PageFactory.initElements(driver, this);

    }

    @FindBy (xpath = "//*[@id=\\\\"app\\\\"]/div[1]/div[1]/aside/nav/div[2]/ul")

    private List<WebElement> all_menu;

    @FindBy (xpath = "/html/body/div/div[1]/div[1]/aside/nav/div[2]/ul/li")

    private List<WebElement> menu_items;

```

```

@FindBy (xpath = "//*[@id=\"app\"]/div[1]/div[1]/aside/nav/div[2]/ul/li[1]/a")

private WebElement admin_Link;

@FindBy (xpath =
"//*[@id=\"app\"]/div[1]/div[2]/div[2]/div/div[1]/div[2]/form/div[1]/div/div[1]/div/div[2]/input")

private WebElement username;

@FindBy (xpath =
"//*[@id=\"app\"]/div[1]/div[2]/div[2]/div/div[1]/div[2]/form/div[2]/button[2]")

private WebElement search_button;

@FindBy (xpath =
"//*[@id=\"app\"]/div[1]/div[2]/div[2]/div/div[2]/div[2]/div/span")

private WebElement records_found;

@FindBy (xpath =
"//*[@id=\"app\"]/div[1]/div[2]/div[2]/div/div[1]/div[2]/form/div[1]/div/div[2]/div/div[2]/div/div/div[1]")

private List<WebElement> drop_down;

@FindBy (xpath =
"//*[@id=\"app\"]/div[1]/div[2]/div[2]/div/div[1]/div[2]/form/div[1]/div/div[2]/div/div[2]/div/div/div[2]/i")

private List<WebElement> drop_down_items;

@FindBy (xpath =
"//*[@id=\"app\"]/div[1]/div[2]/div[2]/div/div[1]/div[2]/form/div[1]/div/div[4]/div/div[2]/div/div/div[1]")

private List<WebElement> user_status_items;

public List<WebElement> getUser_status_items() {

return user_status_items;

}

public List<WebElement> getDrop_down() {

return drop_down;

```

```

}

public List<WebElement> getdrop_down_items() {

return drop_down_items;

}

public WebElement getRecords_found() {

return records_found;

}

public WebElement getUsername() {

return username;

}

public WebElement getSearch_button() {

return search_button;

}

public List<WebElement> getMenu_items() {

return menu_items;

}

public WebElement getAdmin_link() {

return admin_Link;

}

@Test

public void menu_options() {

System.out.println("Total number of Menu items:" + getMenu_items().size());

getAdmin_link().click();

String url = driver.getCurrentUrl();

Assert.assertTrue(url.contains("admin"), "URL does not match");

```

```

System.out.println("Logged into Admin Page");

}

@Test

public void searchByUserName(String un) {

    getUsername().sendKeys(un);

    getSearch_button().click();

    try {

        Thread.sleep(2000);

    } catch (InterruptedException e) {

        //TODO Auto-Generated Catch Block

        e.printStackTrace();

    }

    System.out.println("Total records with username: " + un + " is: " +
        getRecords_found().getText());

    driver.navigate().refresh();

}

@Test

public void searchByUserRole() {

    getDrop_down().get(0).click();

    for (WebElement i : getdrop_down_items()) {

        if(i.getText().contains("Admin")) {

            i.click();

            break;

        }

    }

}

```



```
getSearch_button().click();

try {

Thread.sleep(2000);

} catch (InterruptedException e) {

//TODO Auto-Generated Catch Block

e.printStackTrace();

}

System.out.println("Total records with User Role Admin: " +
getRecords_found().getText());

driver.navigate().refresh();

}

public void searchByUserStatus() {

try {

Thread.sleep(2000);

} catch (InterruptedException e) {

//TODO Auto-Generated Catch Block

e.printStackTrace();

}

getDrop_down().get(0).click();

for(WebElement i : getdrop_down_items()) {

if(i.getText().contains("Enabled"))

i.click();

break;

}

getSearch_button().click();
```

```

try {

Thread.sleep(2000);

} catch (InterruptedException e) {

//TODO Auto-Generated Catch Block

e.printStackTrace();

}

System.out.println("Total records enabled: " + getRecords_found().getText());

}

}

```

AdminClientPage.java

```

package Scenario2;

import org.testng.annotations.Test;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

public class AdminClientPage {

WebDriver driver;

Admin_Page ap;

Login_Page lp;

@BeforeTest

public void beforeTest() {

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");

ap = new Admin_Page(driver);

lp= new Login_Page(driver);

}

@Test

public void testLogin_AdminPage() {

lp.Login("Admin", "admin123");

ap.menu_options();

ap.searchByUserName("Admin");

ap.searchByUserRole();

ap.searchByUserStatus();

}

@AfterTest

public void afterTest() {

driver.close();

}

}

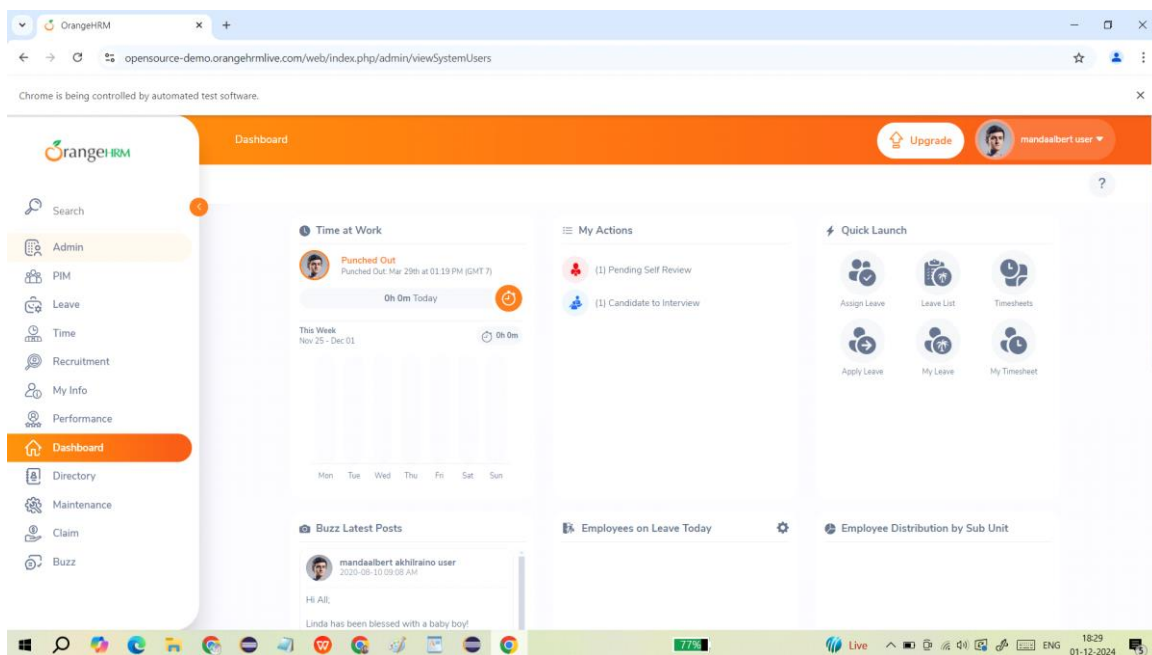
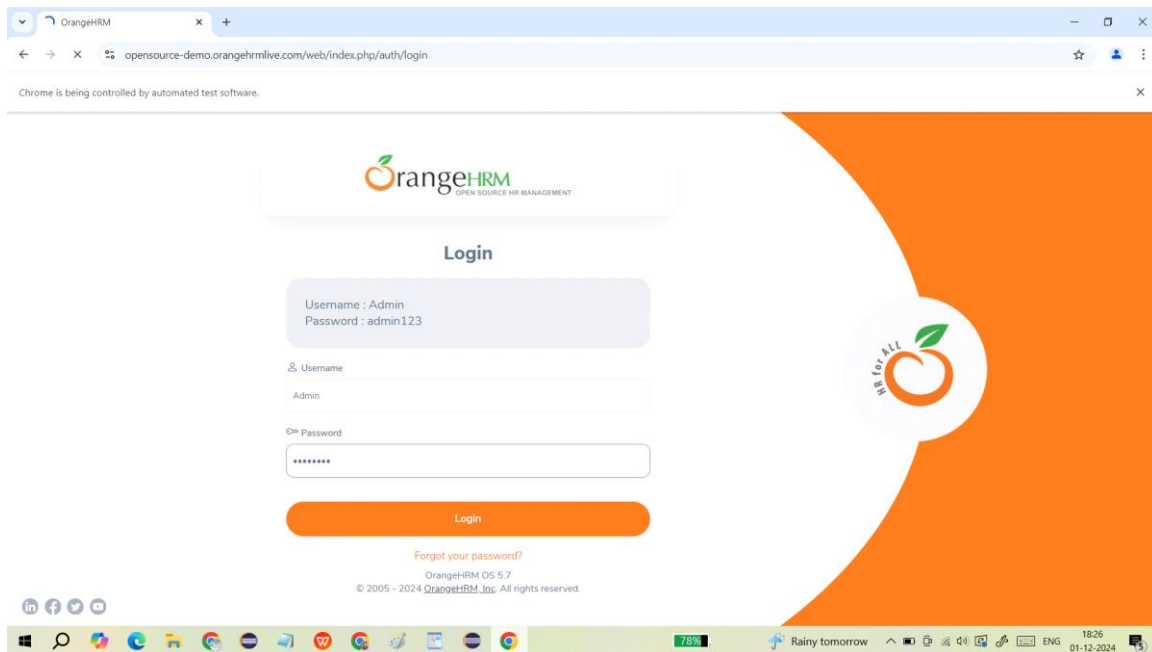
```

Create a Page and Class for Login and Automate the functionality of OrangeHRM login for Valid Test Data: (Username : Admin and Password : admin123).

```
1 package Scenario2;
2
3 import java.util.List;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.WebElement;
7 import org.openqa.selenium.support.FindBy;
8 import org.openqa.selenium.support.PageFactory;
9 import org.testng.annotations.BeforeTest;
10 import org.testng.annotations.Test;
11
12 Run ALL
13 public class Login_Page {
14     WebDriver driver;
15     public Login_Page(WebDriver d) {
16         this.driver = d;
17         PageFactory.initElements(driver, this);
18     }
19     @FindBy(name = "username")
20     private WebElement username;
21     @FindBy(name = "password")
22     private WebElement password;
23     @FindBy(xpath = "//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")
24     private WebElement submit; ////*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button
25
26     public WebElement getUsername() {
27         return username;
28     }
29
30     public WebElement getPassword() {
31         return password;
32     }
33     public WebElement getSubmit() {
```

```
18
19     @FindBy(name = "username")
20     private WebElement username;
21     @FindBy(name = "password")
22     private WebElement password;
23     @FindBy(xpath = "//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")
24     private WebElement submit; ////*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button
25
26     public WebElement getUsername() {
27         return username;
28     }
29
30     public WebElement getPassword() {
31         return password;
32     }
33     public WebElement getSubmit() {
34         return submit;
35     }
36     @Test
37     Run / Debug
38     public void Login(String un, String ps) {
39         getUsername().sendKeys(un);
40         getPassword().sendKeys(ps);
41         try {
42             Thread.sleep(2000);
43         } catch (InterruptedException e) {
44             //TODO Auto-generated catch block
45             e.printStackTrace();
46         }
47         getSubmit().submit();
48         System.out.println("Login Successful");
49     }
50 }
```

```
1 AdminClientPage.java X
2
3 package Scenario2;
4
5 import org.testng.annotations.Test;
6
7 Run ALL
8 public class AdminClientPage {
9     WebDriver driver;
10     Admin_Page ap;
11     Login_Page lp;
12
13     @BeforeTest
14     public void beforeTest() {
15         driver = new ChromeDriver();
16         driver.manage().window().maximize();
17         driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
18         driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");
19
20         ap = new Admin_Page(driver);
21         lp = new Login_Page(driver);
22     }
23
24     @Test
25     Run / Debug
26     public void testLogin_AdminPage() {
27         lp.Login("Admin", "admin123");
28         ap.menu_options();
29         ap.searchByUserName("Admin");
30         ap.searchByUserRole();
31         ap.searchByUserStatus();
32     }
33
34     @AfterTest
35     public void afterTest() {
36         driver.close();
37     }
38 }
39
40
41
42
43
44
```



1. Create testcase to get all 12 options from left side meny and print the count which should be 12 from that list click on Admin then Admin page get open
2. Create test case for search for existing employee searchByUsername() : here send username Admin to username text box and click on search button and display total record found and refresh page.

3. Create test case for search for existing employee searchByUserRole() : here automate dropdown and select role Admin and Click on search button and display total record found and refresh page.

4. Create test case for search for existing employee searchByUserStatus() : here automate dropdown and select status Enabled or Disabled the click on search button and display the total record found.

```
1 package Scenario2;
2
3 import java.util.List;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.WebElement;
7 import org.openqa.selenium.support.FindBy;
8 import org.openqa.selenium.support.PageFactory;
9 import org.testng.annotations.BeforeTest;
10 import org.testng.annotations.Test;
11
12 public class Login_Page {
13     WebDriver driver;
14     public Login_Page(WebDriver d) {
15         this.driver = d;
16         PageFactory.initElements(driver, this);
17     }
18
19     @FindBy(name = "username")
20     private WebElement username;
21     @FindBy(name = "password")
22     private WebElement password;
23     @FindBy(xpath = "//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")
24     private WebElement submit; ////*[@id="app"]/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button
25
26     public WebElement getUsername() {
27         return username;
28     }
29
30     public WebElement getPassword() {
31         return password;
32     }
33     public WebElement getSubmit() {
```

```
18
19     @FindBy(name = "username")
20     private WebElement username;
21     @FindBy(name = "password")
22     private WebElement password;
23     @FindBy(xpath = "//*[@id='app']/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button")
24     private WebElement submit; ////*[@id="app"]/div[1]/div/div[1]/div/div[2]/div[2]/form/div[3]/button
25
26     public WebElement getUsername() {
27         return username;
28     }
29
30     public WebElement getPassword() {
31         return password;
32     }
33     public WebElement getSubmit() {
34         return submit;
35     }
36     @Test
37     public void Login(String un, String ps) {
38         getUsername().sendKeys(un);
39         getPassword().sendKeys(ps);
40         try {
41             Thread.sleep(2000);
42         } catch (InterruptedException e) {
43             //TODD Auto-generated catch block
44             e.printStackTrace();
45         }
46         getSubmit().submit();
47         System.out.println("Login Successful");
48     }
49 }
50
```



```

D:\Admin_Page.java x
69 public WebElement getSearch_button() {
70     return search_button;
71 }
72
73 public List<WebElement> getMenu_items() {
74     return menu_items;
75 }
76
77 public WebElement getAdmin_link() {
78     return admin_link;
79 }
80
81
82 @Test
83 public void menu_options() {
84     System.out.println("Total number of Menu items:" + getMenu_items().size());
85     getAdmin_link().click();
86     String url = driver.getCurrentUrl();
87     Assert.assertTrue(url.contains("admin"), "URL does not match");
88     System.out.println("Logged into Admin Page");
89 }
90
91 @Test
92 public void searchByUsername(String un) {
93     getUsername().sendKeys(un);
94     getSearch_button().click();
95     try {
96         Thread.sleep(2000);
97     } catch (InterruptedException e) {
98         //T000 Auto-Generated Catch Block
99         e.printStackTrace();
100     }
101     System.out.println("Total records with username: " + un + " is: " + getRecords_found().getText());
102     driver.navigate().refresh();

```

```

D:\Admin_Page.java x
99     e.printStackTrace();
100 }
101 System.out.println("Total records with username: " + un + " is: " + getRecords_found().getText());
102 driver.navigate().refresh();
103 }
104 @Test
105 public void searchByUserRole() {
106     getDrop_down().get(0).click();
107     for (WebElement i : getdrop_down_items()) {
108         if(i.getText().contains("Admin")) {
109             i.click();
110             break;
111         }
112     }
113
114     getSearch_button().click();
115     try {
116         Thread.sleep(2000);
117     } catch (InterruptedException e) {
118         //T000 Auto-Generated Catch Block
119         e.printStackTrace();
120     }
121     System.out.println("Total records with User Role Admin: " + getRecords_found().getText());
122     driver.navigate().refresh();
123 }
124
125
126 public void searchByUserStatus() {
127     try {
128         Thread.sleep(2000);
129     } catch (InterruptedException e) {
130         //T000 Auto-Generated Catch Block
131         e.printStackTrace();

```

```

D:\Admin_Page.java x
120     }
121     System.out.println("Total records with User Role Admin: " + getRecords_found().getText());
122     driver.navigate().refresh();
123 }
124
125
126 public void searchByUserStatus() {
127     try {
128         Thread.sleep(2000);
129     } catch (InterruptedException e) {
130         //T000 Auto-Generated Catch Block
131         e.printStackTrace();
132     }
133
134     getDrop_down().get(0).click();
135     for (WebElement i : getdrop_down_items()) {
136         if(i.getText().contains("Enabled"))
137             i.click();
138         break;
139     }
140
141     getSearch_button().click();
142     try {
143         Thread.sleep(2000);
144     } catch (InterruptedException e) {
145         //T000 Auto-Generated Catch Block
146         e.printStackTrace();
147     }
148
149     System.out.println("Total records enabled: " + getRecords_found().getText());
150 }
151 }
152 }
153

```


OrangeHRM Admin / User Management

System Users

Username: Admin User Role: -- Select -- Employee Name: Type for hints... Status: -- Select --

Reset Search

+ Add

(1) Record Found

Username	User Role	Employee Name	Status	Actions
Admin	Admin	albert mars	Enabled	

Taskbar: 21°C Rain showers, 18:59, 01-12-2024

OrangeHRM Admin / User Management

System Users

Username: User Role: -- Select -- Employee Name: Type for hints... Status: -- Select --

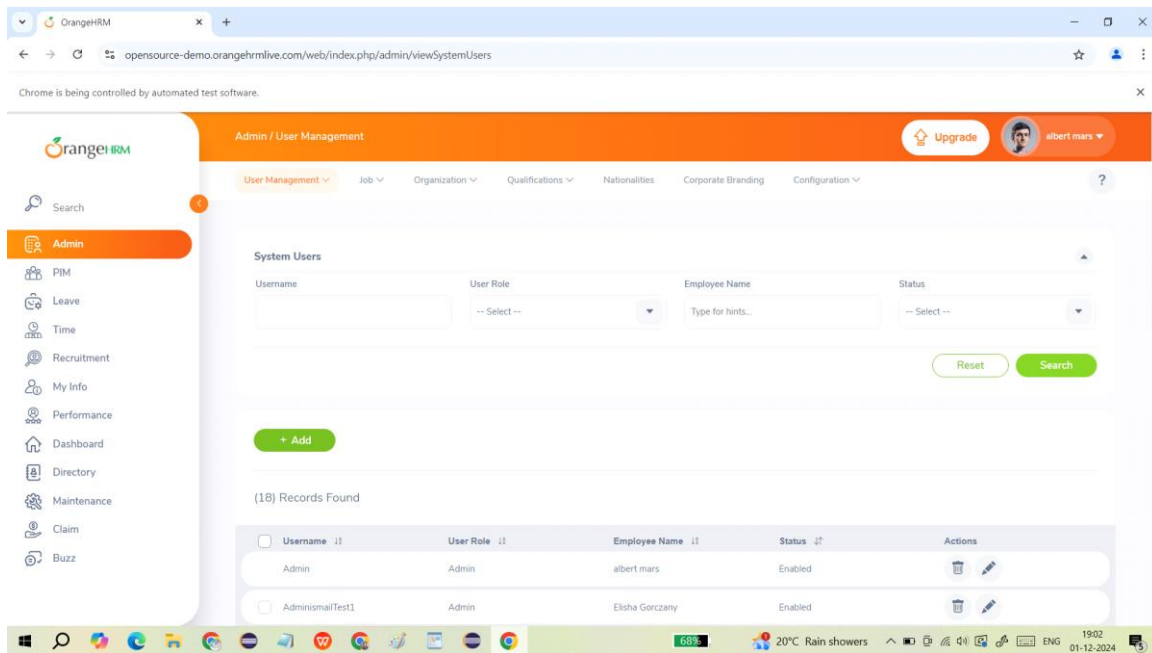
Reset Search

+ Add

(13) Records Found

Username	User Role	Employee Name	Status	Actions
Admin	Admin	albert mars	Enabled	
Admin@mailTest1	Admin	Elisha Gorzany	Enabled	

Taskbar: GBP/INR +0.53%, 19:01, 01-12-2024



CONSOLE OUTPUT:

```

<terminated> AdminClientPage [TestNG] C:\Users\yuka\p2\pool\plugins\org.eclipse.just.openjdk.hotspot.jre.full.win32.x86_64.22.0.2\j20240802-1626\jre\bin\javaw.exe (1 Dec 2024, 7:02:07 pm - 7:02:43 pm)
[RemoteTestNG] detected TestNG version 7.10.2
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.
Dec 01, 2024 7:02:15 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 131, returning the closest version; found: 128; Please update to a Selenium version that supports CDP version 131.
Login Successful
Total number of Menu items:12
Logged into Admin Page
Total records with username Admin is: (1) Record Found
Total records with User Role Admin: (17) Records Found
Total records enabled: (18) Records Found
Dec 01, 2024 7:02:43 PM org.openqa.selenium.remote.http.WebSocket$Listener onError
WARNING: Connection reset
java.net.SocketException: Connection reset
    at java.base/sun.nio.ch.SocketChannelImpl.throwConnectionReset(SocketChannelImpl.java:403)
    at java.base/sun.nio.ch.SocketChannelImpl.implRead(SocketChannelImpl.java:435)
    at java.base/sun.nio.ch.SocketChannelImpl.read(SocketChannelImpl.java:493)
  
```

```
Console x
terminated> AdminClientPage [TestNG] C:\Users\yusar\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.22.0.2\20240802-1626\jre\bin\java.exe (1 Dec 2024, 7:02:07 pm - 7:02:43 pm)
at java.net.http.jdk.internal.net.http.HttpClientImpl$SelectorManager.lambda$run$3(HttpClientImpl.java:1421)
at java.base/java.util.ArrayList.forEach(ArrayList.java:1597)
at java.net.http.jdk.internal.net.http.HttpClientImpl$SelectorManager.run(HttpClientImpl.java:1421)

PASSED: Scenario2.AdminClientPage.testLogin_AdminPage

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

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

Javadoc Declaration Results of running class AdminClientPage x Error Log

Search: Passed: 1 Failed: 0 Skipped: 0 Tests: 1/1 Methods: 1 (32885 ms)

All Tests Failed Tests Summary

- Default suite (1/0/0/0) (20.632 s)
 - Default test (20.632 s)
 - Scenario2.AdminClientPage
 - testLogin_AdminPage (20.632 s)

Failure Exception