Scenario1: Automate Orange HRM login and logout with 5 different data set including valid and invalid data set.

1. Save data set 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 set (other than given data) test case should be fail.

5. Capture Screenshot for every login functionality

6. Generate Extent Report for the same.

Base Test Class:

import java.awt.Desktop;

import java.io.File;

import java.io.IOException;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.Reporter;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.BeforeTest;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.reporter.ExtentSparkReporter;

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

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

import Datadriven.Pages.AdminPage;

import Datadriven.Pages.Login;

import Datadriven.Pages.Logout;

import Datadriven.utility.Listener;

import Datadriven.utility.PropertiesConfig;

import Datadriven.utility.Utility;

public class BaseTest{

public PropertiesConfig p1;

public static WebDriver driver;

public Login lp;

public Logout lo;

public AdminPage ad;

@BeforeMethod//For all Pages Browser setup

public void setUpBrowser()

{

p1=new PropertiesConfig();

driver=new ChromeDriver();

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

driver.get(p1.getData("base\_url"));

Reporter.log("Log: Apllication is open",true);

lp=new Login(driver);

lo=new Logout(driver);

ad=new AdminPage(driver);

}

@AfterMethod

public void teardown() {

driver.quit();

}

public static String screenshot() throws IOException {

return Utility.getScreenshot(driver);

}

}

Login Page:

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class LoginOrangeHrm {

private WebDriver driver;

public Login(WebDriver d){

this.driver=d;

}

//locators using By class

private By username =By.name("username");

private By password =By.name("password");

private By loginBtn =By.xpath("//button[contains(@class,' oxd-button--main')]");

private By ErrorMssg=By.xpath("//p[@class='oxd-text oxd-text--p oxd-alert-content-text']");

//actions

public void userName(String un){

if(driver.findElement(username).isDisplayed() && driver.findElement(username).isEnabled())

driver.findElement(username).sendKeys(un);

}

public void password(String pass){

if(driver.findElement(password).isDisplayed() && driver.findElement(password).isEnabled())

driver.findElement(password).sendKeys(pass);

}

public void clickLogin(){

if(driver.findElement(loginBtn).isDisplayed() && driver.findElement(loginBtn).isEnabled())

driver.findElement(loginBtn).click();

}

public String getUrl()

{

return driver.getCurrentUrl();

}

public String getAppTitle()

{

return driver.getTitle();

}

public String getErrorMssg()

{

return driver.findElement(ErrorMssg).getText();

}

public void doLogin(String un,String psw)

{

userName(un);

password(psw);

clickLogin();

}

}

Test case:

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import org.testng.Assert;

public class LoginLogout extends BaseTest{

String expected="Invalid credentials";

@Test(dataProvider = "dp1",dataProviderClass=Datadriven.utility.ExcelDataSupplier.class)

public void LoginLogout(String username,String pw) {

System.out.println(username+""+pw);

lp.doLogin(username,pw);

String curl=lp.getUrl();

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

// Assert.assertTrue(curl.contains("dashboard"),"Fail:Url not matches");

lo.logout();

}else

{

String actual= lp.getErrorMssg();

Assert.assertEquals(actual, expected);

}

}

}

Data Provider Class:

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import org.testng.Assert;

public class Login extends BaseTest{

String expected="Invalid credentials";

@Test(dataProvider = "dp1",dataProviderClass=Datadriven.utility.ExcelDataSupplier.class)

public void LoginLogout(String username,String pw) {

System.out.println(username+""+pw);

lp.doLogin(username,pw);

String curl=lp.getUrl();

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

// Assert.assertTrue(curl.contains("dashboard"),"Fail:Url not matches");

lo.logout();

}else

{

String actual= lp.getErrorMssg();

Assert.assertEquals(actual, expected);

}

}

}

Listener class:

import java.awt.Desktop;

import java.io.File;

import java.io.IOException;

import org.openqa.selenium.WebDriver;

import org.testng.ITestContext;

import org.testng.ITestListener;

import org.testng.ITestResult;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import com.aventstack.extentreports.reporter.ExtentSparkReporter;

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

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

import Datadriven.TestCases.BaseTest;

public class Listener extends BaseTest implements ITestListener {

public ExtentReports extent;

public ExtentTest test;

public ExtentSparkReporter s;

public ExtentSparkReporterConfig c;

public String path;

@Override

public void onTestStart(ITestResult result) {

System.out.println("test started");

test=extent.createTest(result.getMethod().getMethodName()).assignAuthor("kavya").assignCategory("smoke suite").assignDevice("chrome");

}

@Override

public void onTestSuccess(ITestResult result) {

System.out.println("test success");

try {

test

.info("this is a info mssg")

.pass("passed")

.addScreenCaptureFromPath(screenshot(),"home page of orangehrm captured");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@Override

public void onTestFailure(ITestResult result) {

System.out.println("test failed");

try {

test

.info("this is a info mssg")

.fail("failed")

.addScreenCaptureFromPath(screenshot(),"home page of orangehrm captured");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@Override

public void onTestSkipped(ITestResult result) {

System.out.println("testskipped");

}

@Override

public void onTestFailedButWithinSuccessPercentage(ITestResult result) {

}

@Override

public void onTestFailedWithTimeout(ITestResult result) {

System.out.println("test failed with timeout");

}

@Override

public void onStart(ITestContext context) {

System.out.println("execution started");

if(extent==null) {

extent = new ExtentReports();

}

s=new ExtentSparkReporter("Reports/report.html");

extent.setSystemInfo("os", System.getProperty("os.version"));

extent.setSystemInfo("os", System.getProperty("java.version"));

extent.setSystemInfo("os", System.getProperty("os.name"));

c=s.config();

extent.attachReporter(s);

}

@Override

public void onFinish(ITestContext context) {

System.out.println("execution finsihed");

extent.flush();

try {

Desktop.getDesktop().browse(new File("Reports/report.html").toURI());

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

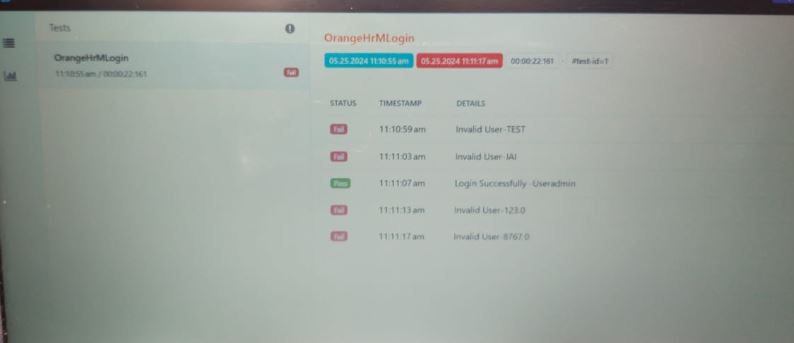
}

}

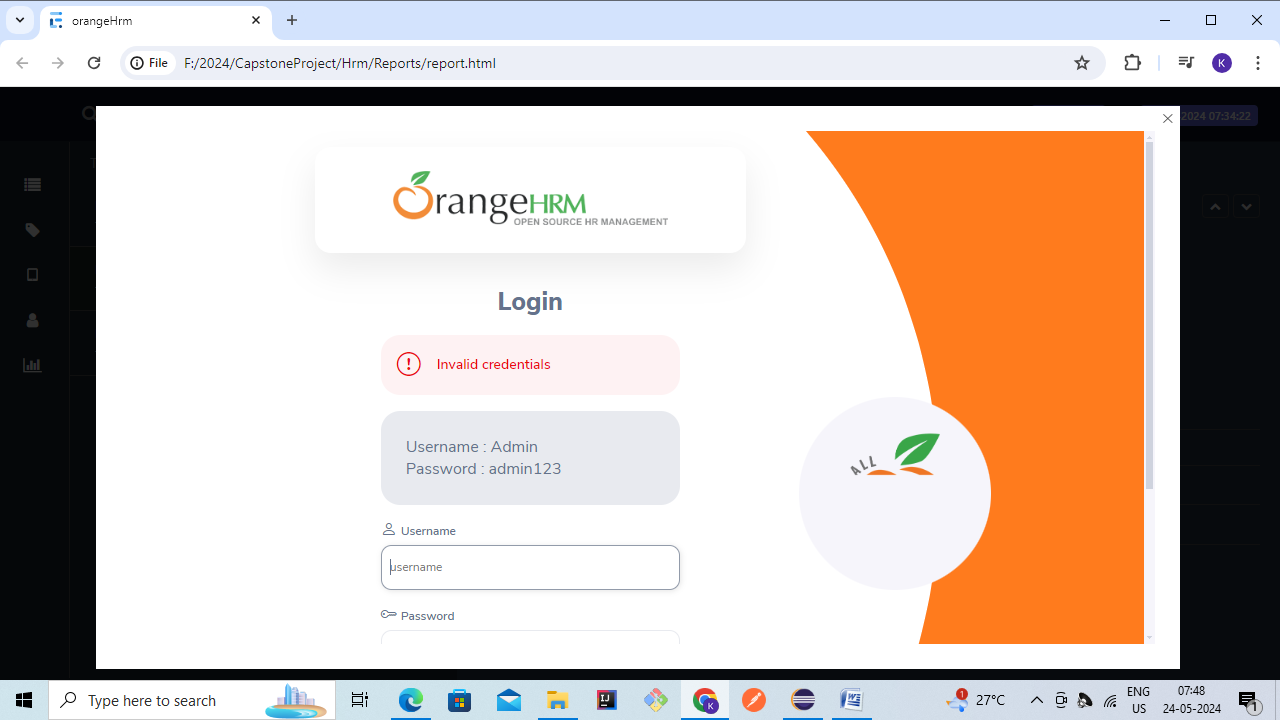
}

}

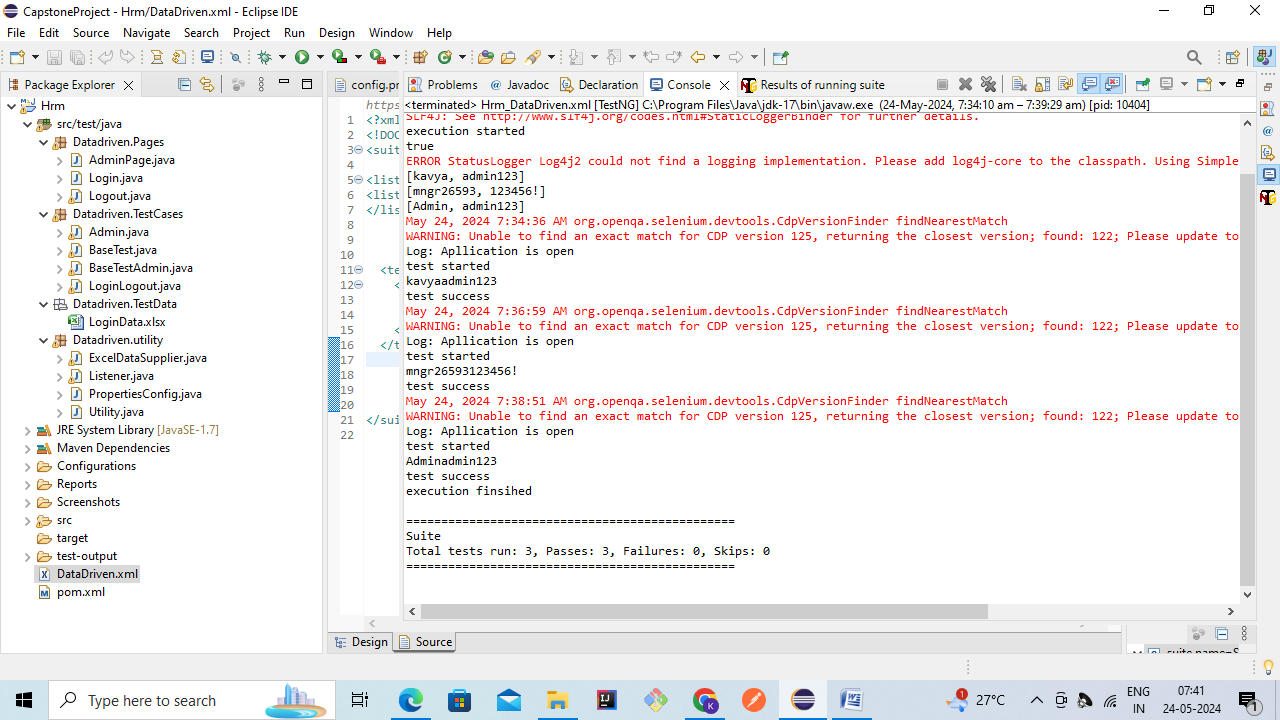
Extent Report:



Screenshot:

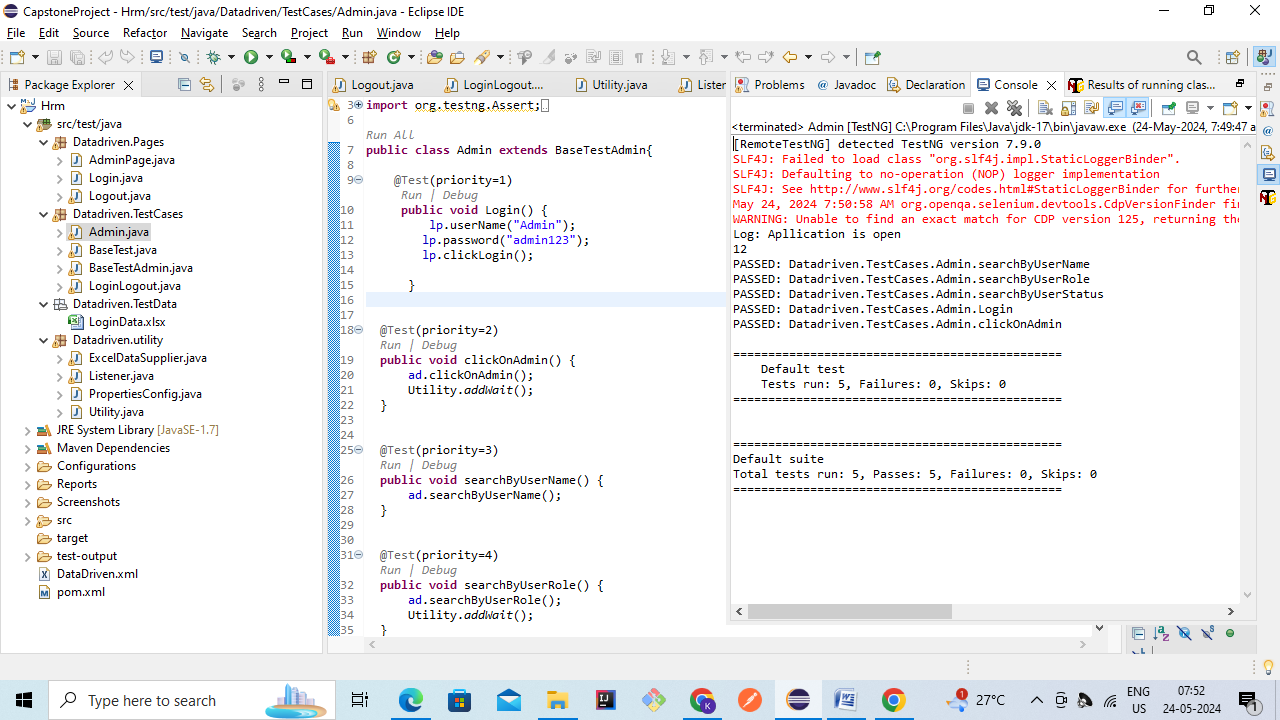


Console output:



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 classfor 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

Console output:



Admin page:

import java.util.List;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class AdminPageclass {

WebDriver d;

public AdminPage(WebDriver d) {

super();

this.d = d;

PageFactory.initElements(d, this);

}

@FindBy(xpath="//ul[@class='oxd-main-menu']/li/a/span")

private List<WebElement> menu;

@FindBy(xpath="(//\*[@class='oxd-input oxd-input--active'])[2]")

private WebElement textbox;

@FindBy(xpath="//\*[@type='submit']")

private WebElement searchBox;

@FindBy(xpath="//\*[@class='oxd-select-text--after']/i")

private WebElement userRoleDropdownIcon;

@FindBy(xpath="//\*[@class='oxd-select-option']/span")

private WebElement AdminRoleOption;

@FindBy(xpath="(//\*[@class='oxd-select-text--after'])[2]")

private WebElement statusDropdownIcon;

@FindBy(xpath="//\*[@class='oxd-select-text-input']")

private WebElement disabledOption;

public void clickOnAdmin() {

System.out.println(menu.size());

for(WebElement a:menu) {

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

a.click();

break;

}

}

}

public void searchByUserName() {

textbox.sendKeys("Admin");

searchBox.click();

d.navigate().refresh();

}

public void searchByUserRole() {

userRoleDropdownIcon.click();

AdminRoleOption.click();

searchBox.click();

d.navigate().refresh();

}

public void searchByUserStatus() {

statusDropdownIcon.click();

disabledOption.click();

searchBox.click();

//d.navigate().refresh();

}

}

Base test:

**import** java.time.Duration;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.Reporter;

**import** org.testng.annotations.AfterClass;

**import** org.testng.annotations.AfterMethod;

**import** org.testng.annotations.BeforeClass;

**import** org.testng.annotations.BeforeMethod;

**import** Datadriven.Pages.AdminPage;

**import** Datadriven.Pages.Login;

**import** Datadriven.Pages.Logout;

**import** Datadriven.utility.PropertiesConfig;

**import** Datadriven.utility.\*;

**public** **class** BaseTestAdmin {

**public** PropertiesConfig p1;

**public** **static** WebDriver *driver*;

**public** Login lp;

**public** Logout lo;

**public** AdminPage ad;

@BeforeClass

**public** **void** setUpBrowser()

{

p1=**new** PropertiesConfig();

*driver*=**new** ChromeDriver();

*driver*.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(20));

*driver*.get(p1.getData("base\_url"));

Reporter.*log*("Log: Apllication is open",**true**);

lp=**new** Login(*driver*);

ad=**new** AdminPage(*driver*);

}

@AfterClass(enabled=**false**)

**public** **void** teardown() {

*driver*.quit();

}

}

Test file:

import org.testng.Assert;

import org.testng.annotations.Test;

import Datadriven.utility.\*;

public class Admin extends BaseTest{

@Test(priority=1)

public void Login() {

lp.userName("Admin");

lp.password("admin123");

lp.clickLogin();

}

@Test(priority=2)

public void clickOnAdmin() {

ad.clickOnAdmin();

Utility.addWait();

}

@Test(priority=3)

public void searchByUserName() {

ad.searchByUserName();

}

@Test(priority=4)

public void searchByUserRole() {

ad.searchByUserRole();

Utility.addWait();

}

@Test(priority=5)

public void searchByUserStatus() {

ad.searchByUserStatus();

}

}