|  |  |
| --- | --- |
| **Capstone Project** | **Student Name: Vijina P** |
| **Human Resource Management Domain Project** | |

**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

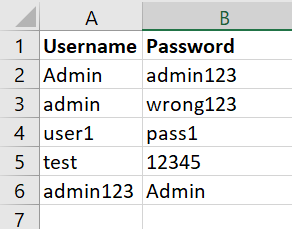
3. Prepare a scriptfor Login and logout

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 forthe same.

Ans:

**Data Input Excel Sheet: LoginData.xlsx**



**Pom.xml:**

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>com.orangehrm</groupId>

<artifactId>OrangeHRM\_Automation</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.21.0</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.10.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

<version>5.1.1</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.5</version>

</dependency>

</dependencies>

</project>

**testng.xml:**

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >

<suite name="OrangeHRM Automation">

<test name="Login Tests">

<classes>

<class name="tests.LoginTest" />

</classes>

</test>

</suite>

**\src\test\java\utils\ExcelUtils.java**

package utils;

import org.apache.poi.ss.usermodel.\*;

import org.apache.poi.xssf.usermodel.XSSFWorkbook;

import java.io.FileInputStream;

public class ExcelUtils {

public static Object[][] getData(String filePath, String sheetName) throws Exception {

FileInputStream fis = new FileInputStream(filePath);

Workbook wb = new XSSFWorkbook(fis);

Sheet sheet = wb.getSheet(sheetName);

int rowCount = sheet.getPhysicalNumberOfRows();

int colCount = sheet.getRow(0).getPhysicalNumberOfCells();

Object[][] data = new Object[rowCount - 1][colCount];

for (int i = 1; i < rowCount; i++) {

Row row = sheet.getRow(i);

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

data[i - 1][j] = row.getCell(j).toString();

}

}

wb.close();

return data;

}

}

**\src\test\java\base\BaseTest.java**

package base;

import com.aventstack.extentreports.\*;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.\*;

import java.io.File;

import java.io.FileOutputStream;

import java.text.SimpleDateFormat;

import java.util.Date;

public class BaseTest {

public WebDriver driver;

public ExtentReports extent;

public ExtentTest test;

@BeforeSuite

public void setupReport() {

ExtentHtmlReporter htmlReporter = new ExtentHtmlReporter("reports/ExtentReport.html");

extent = new ExtentReports();

extent.attachReporter(htmlReporter);

}

@BeforeMethod

public void setup() {

driver = new ChromeDriver();

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

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

}

@AfterMethod

public void tearDown() {

driver.quit();

}

@AfterSuite

public void flushReport() {

extent.flush();

}

public String captureScreenshot(String testName) {

String filePath = "screenshots/" + testName + "\_" + timestamp() + ".png";

try {

TakesScreenshot ts = (TakesScreenshot) driver;

byte[] src = ts.getScreenshotAs(OutputType.BYTES);

FileOutputStream fos = new FileOutputStream(new File(filePath));

fos.write(src);

fos.close();

} catch (Exception e) {

e.printStackTrace();

}

return filePath;

}

private String timestamp() {

return new SimpleDateFormat("yyyyMMddHHmmss").format(new Date());

}

}

**\src\test\java\pages\DashboardPage.java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class DashboardPage {

WebDriver driver;

By dashboardHeader = By.xpath("//h6[text()='Dashboard']");

public DashboardPage(WebDriver driver) {

this.driver = driver;

}

public boolean isDashboardDisplayed() {

return driver.findElements(dashboardHeader).size() > 0;

}

}

**\src\test\java\pages\LoginPage.java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class LoginPage {

WebDriver driver;

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

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

By loginBtn = By.xpath("//button[@type='submit']");

public LoginPage(WebDriver driver) {

this.driver = driver;

}

public void login(String user, String pass) {

driver.findElement(username).clear();

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

driver.findElement(password).clear();

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

driver.findElement(loginBtn).click();

}

}

**\src\test\java\tests\LoginTest.java**

package tests;

import base.BaseTest;

import org.testng.Assert;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import pages.DashboardPage;

import pages.LoginPage;

import utils.ExcelUtils;

public class LoginTest extends BaseTest {

@Test(dataProvider = "loginData")

public void loginTest(String username, String password) {

test = extent.createTest("Login Test with user: " + username);

LoginPage loginPage = new LoginPage(driver);

loginPage.login(username, password);

DashboardPage dashboardPage = new DashboardPage(driver);

String screenshotPath = captureScreenshot(username);

if (username.equals("Admin") && password.equals("admin123")) {

Assert.assertTrue(dashboardPage.isDashboardDisplayed());

test.pass("Valid login successful").addScreenCaptureFromPath(screenshotPath);

} else {

Assert.assertFalse(dashboardPage.isDashboardDisplayed());

test.fail("Invalid login.").addScreenCaptureFromPath(screenshotPath);

}

}

@DataProvider(name = "loginData")

public Object[][] getLoginData() throws Exception {

return ExcelUtils.getData("data/LoginData.xlsx", "Sheet1");

}

}

**Scenario 2:**

**Create a Page Object Model for two pages**

1. **Login Page**
2. **Admin Page**

**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 ValidTest Data:**

* **Username: Admin**
* **Password: admin123.**

**Ans:**

**Pom.xml**

<project>

<modelVersion>4.0.0</modelVersion>

<groupId>com.orangehrm</groupId>

<artifactId>OrangeHRM\_LoginTest</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>4.21.0</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.10.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**BaseTest.java**

package base;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.\*;

public class BaseTest {

public WebDriver driver;

@BeforeMethod

public void setUp() {

driver = new ChromeDriver();

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

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

}

@AfterMethod

public void tearDown() {

if (driver != null) {

driver.quit();

}

}

}

**LoginPage.java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class LoginPage {

WebDriver driver;

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

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

By loginBtn = By.xpath("//button[@type='submit']");

By dashboardHeader = By.xpath("//h6[text()='Dashboard']");

public LoginPage(WebDriver driver) {

this.driver = driver;

}

public void enterUsername(String user) {

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

}

public void enterPassword(String pass) {

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

}

public void clickLogin() {

driver.findElement(loginBtn).click();

}

public boolean isDashboardDisplayed() {

return driver.findElements(dashboardHeader).size() > 0;

}

public void login(String user, String pass) {

enterUsername(user);

enterPassword(pass);

clickLogin();

}

}

**LoginTest.java**

package tests;

import base.BaseTest;

import org.testng.Assert;

import org.testng.annotations.Test;

import pages.LoginPage;

public class LoginTest extends BaseTest {

@Test

public void testValidLogin() {

LoginPage loginPage = new LoginPage(driver);

loginPage.login("Admin", "admin123");

Assert.assertTrue(loginPage.isDashboardDisplayed(), "Something wrong, dashboard is not displayed.");

}

}

**testng.xml**

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd" >

<suite name="OrangeHRM Login Suite">

<test name="Valid Login Test">

<classes>

<class name="tests.LoginTest"/>

</classes>

</test>

</suite>

**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.

**Test Scenario**

1. Login to OrangeHRM
2. Get all 12 left side menu items
3. Print the count of menu items(12)
4. Click on “Admin” menu link.

**Ans:**

**DashboardPage.java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import java.util.List;

public class DashboardPage {

WebDriver driver;

By sideMenuOptions = By.cssSelector("ul.oxd-main-menu li span.oxd-text");

By adminPageTitle = By.xpath("//h6[text()='User Management']");

public DashboardPage(WebDriver driver) {

this.driver = driver;

}

public List<WebElement> getAllMenuOptions() {

return driver.findElements(sideMenuOptions);

}

public void clickOnMenuOption(String optionText) {

for (WebElement option : getAllMenuOptions()) {

if (option.getText().equalsIgnoreCase(optionText)) {

option.click();

break;

}

}

}

public boolean isAdminPageDisplayed() {

return driver.findElements(adminPageTitle).size() > 0;

}

}

**MenuTest.java**

package tests;

import base.BaseTest;

import org.openqa.selenium.WebElement;

import org.testng.Assert;

import org.testng.annotations.Test;

import pages.LoginPage;

import pages.DashboardPage;

import java.util.List;

public class MenuTest extends BaseTest {

@Test

public void testLeftMenuOptions() throws InterruptedException {

// Login

LoginPage loginPage = new LoginPage(driver);

loginPage.login("Admin", "admin123");

DashboardPage dashboard = new DashboardPage(driver);

Thread.sleep(2000); // wait for menu to load

// 1. Get all menu options

List<WebElement> menuOptions = dashboard.getAllMenuOptions();

// 2. Print count and menu names

System.out.println("Total menu items: " + menuOptions.size());

for (WebElement menu : menuOptions) {

System.out.println("Menu: " + menu.getText());

}

// Assert that there are exactly 12 items

Assert.assertEquals(menuOptions.size(), 12, "Menu count is not 12");

// 3. Click on "Admin" menu and verify

dashboard.clickOnMenuOption("Admin");

Thread.sleep(2000); // wait for page load

Assert.assertTrue(dashboard.isAdminPageDisplayed(), "Admin page is not displayed");

}

}

1. 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.

**Test Scenario**

1. Login to OrangeHRM
2. Navigate to **Admin** section
3. Search for user with username Admin
4. Click **Search**
5. Display total records found
6. Refresh the page

Ans:

**AdminPage.java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class AdminPage {

WebDriver driver;

By usernameInput = By.xpath("//label[text()='Username']/../following-sibling::div/input");

By searchButton = By.xpath("//button[normalize-space()='Search']");

By resultRows = By.cssSelector(".oxd-table-body .oxd-table-card");

By resetButton = By.xpath("//button[normalize-space()='Reset']");

public AdminPage(WebDriver driver) {

this.driver = driver;

}

public void searchByUserName(String username) {

driver.findElement(usernameInput).clear();

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

}

public void clickSearch() {

driver.findElement(searchButton).click();

}

public int getTotalRecords() {

return driver.findElements(resultRows).size();

}

public void refreshPage() {

driver.navigate().refresh();

}

}

**EmployeeSearchTest.java**

package tests;

import base.BaseTest;

import org.testng.Assert;

import org.testng.annotations.Test;

import pages.LoginPage;

import pages.DashboardPage;

import pages.AdminPage;

public class EmployeeSearchTest extends BaseTest {

@Test

public void testSearchByUsername() throws InterruptedException {

// Step 1: Login

LoginPage loginPage = new LoginPage(driver);

loginPage.login("Admin", "admin123");

// Step 2: Click on Admin menu

DashboardPage dashboardPage = new DashboardPage(driver);

Thread.sleep(2000); // wait for menu

dashboardPage.clickOnMenuOption("Admin");

// Step 3–6: Search and validate

AdminPage adminPage = new AdminPage(driver);

Thread.sleep(2000); // wait for Admin page

adminPage.searchByUserName("Admin");

adminPage.clickSearch();

Thread.sleep(2000); // wait for results

int recordCount = adminPage.getTotalRecords();

System.out.println("Total records found: " + recordCount);

Assert.assertTrue(recordCount > 0, "No records found!");

// Step 7: Refresh page

adminPage.refreshPage();

}

}

1. **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.**

**Test Scenario**

1. Open OrangeHRM dashboard (after login)
2. Navigate to Admin page
3. Use SearchByUserRole("Admin") method to select user role from dropdown
4. Click Search
5. Display total records found
6. Refresh the page

**AdminPage.java**

package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.ui.\*;

import java.time.Duration;

public class AdminPage {

WebDriver driver;

By userRoleDropdown = By.xpath("//label[text()='User Role']/../following-sibling::div//div[contains(@class,'select-text')]");

By dropdownOptions = By.xpath("//div[@role='listbox']//span");

By searchButton = By.xpath("//button[normalize-space()='Search']");

By resultRows = By.cssSelector(".oxd-table-body .oxd-table-card");

public AdminPage(WebDriver driver) {

this.driver = driver;

}

public void searchByUserRole(String roleName) {

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

driver.findElement(userRoleDropdown).click();

wait.until(ExpectedConditions.visibilityOfAllElementsLocatedBy(dropdownOptions));

for (WebElement option : driver.findElements(dropdownOptions)) {

if (option.getText().equalsIgnoreCase(roleName)) {

option.click();

break;

}

}

}

public void clickSearch() {

driver.findElement(searchButton).click();

}

public int getTotalRecords() {

return driver.findElements(resultRows).size();

}

public void refreshPage() {

driver.navigate().refresh();

}

}

**UserRoleSearchTest.java**

package tests;

import base.BaseTest;

import org.testng.Assert;

import org.testng.annotations.Test;

import pages.AdminPage;

import pages.DashboardPage;

import pages.LoginPage;

public class UserRoleSearchTest extends BaseTest {

@Test

public void testSearchByUserRole() throws InterruptedException {

// Step 1: Login

LoginPage loginPage = new LoginPage(driver);

loginPage.login("Admin", "admin123");

// Step 2: Navigate to Admin

DashboardPage dashboardPage = new DashboardPage(driver);

Thread.sleep(2000);

dashboardPage.clickOnMenuOption("Admin");

// Step 3: Search by User Role

AdminPage adminPage = new AdminPage(driver);

Thread.sleep(2000);

adminPage.searchByUserRole("Admin");

adminPage.clickSearch();

// Step 4: Display results

Thread.sleep(2000);

int recordCount = adminPage.getTotalRecords();

System.out.println("Total records found for role 'Admin': " + recordCount);

Assert.assertTrue(recordCount > 0, "No records found for Admin role");

// Step 5: Refresh the page

adminPage.refreshPage();

}

}