1. Page Object Model in Framework – How to Create Test Case in Framework using POM  
     
   🡪Create a Package “com.learnautomation.testcases” in “iNetBankingV4” project.  
   🡪In that package, create class “TestCaseWithoutFramework” and execute.

**package** com.learnautomation.testcases;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

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

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

**import** org.testng.annotations.Test;

@Test

**public** **class** TestCaseWithoutFramework {

**public** **void** test1() **throws** InterruptedException{

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Drivers\\chromedriver.exe");

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

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

driver.manage().timeouts().~~implicitlyWait~~(30, TimeUnit.***SECONDS***);

driver.findElement(By.*name*("username")).sendKeys("Admin");

driver.findElement(By.*name*("password")).sendKeys("admin123");

driver.findElement(By.*xpath*("//button[@class='oxd-button oxd-button--medium oxd-button--main orangehrm-login-button']")).click();

driver.quit();

}

}

Output:  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 14026

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672034696.543][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

PASSED: com.learnautomation.testcases.TestCaseWithoutFramework.test1

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

🡪Now create another package “com.learnautomation.utility”. In this package create a class “BrowserFactory”.  
  
**package** com.learnautomation.utility;

**import** java.util.concurrent.TimeUnit;

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

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

**import** org.openqa.selenium.edge.EdgeDriver;

**public** **class** BrowserFactory {

**public** **static** WebDriver startApplication(WebDriver driver, String browserName, String appURL) {

**if**(browserName.equals("Chrome")) {

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Drivers\\chromedriver.exe");

driver = **new** ChromeDriver();

} **else** **if**(browserName.equals("Edge")) {

System.*setProperty*("webdriver.edge.driver", "C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Drivers\\chromedriver.exe");

driver = **new** EdgeDriver();

} **else** {

System.***out***.println("We do not support this browser");

}

driver.manage().timeouts().~~pageLoadTimeout~~(30, TimeUnit.***SECONDS***);

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

driver.get(appURL);

driver.manage().timeouts().~~implicitlyWait~~(30, TimeUnit.***SECONDS***);

**return** driver;

}

**public** **static** **void** quitBrowser(WebDriver driver) {

driver.quit();

}

}

🡪And create another class “LoginTestHRM” in “com.learnautomation.testcases” package and execute.

**package** com.learnautomation.testcases;

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

**import** org.testng.annotations.Test;

**import** com.learnautomation.utility.BrowserFactory;

**public** **class** LoginTestHRM {

WebDriver driver;

@Test

**public** **void** loginApp() {

driver = BrowserFactory.*startApplication*(driver, "Chrome", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

}

}

Output:  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 43187

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672035812.158][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

🡪Create another package “com.learnautomation.pages”. In this package create a class “LoginPage”.

**package** com.learnautomation.pages;

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

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.support.FindBy;

**public** **class** LoginPage {

WebDriver driver;

**public** LoginPage(WebDriver ldriver) {

**this**.driver = ldriver;

}

@FindBy(name="username") WebElement uname;

@FindBy(name="password") WebElement pass;

@FindBy(xpath="//button[@class='oxd-button oxd-button--medium oxd-button--main orangehrm-login-button']") WebElement loginButton;

**public** **void** loginToHRM(String usernameApplication, String passwordApplication) {

uname.sendKeys(usernameApplication);

pass.sendKeys(passwordApplication);

loginButton.click();

}

}

🡪Now do the modifications in class “LoginTestHRM” and execute.  
  
**package** com.learnautomation.testcases;

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

**import** org.openqa.selenium.support.PageFactory;

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.LoginPage;

**import** com.learnautomation.utility.BrowserFactory;

**public** **class** LoginTestHRM {

WebDriver driver;

@Test

**public** **void** loginApp() {

driver = BrowserFactory.*startApplication*(driver, "Chrome", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

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

BrowserFactory.*quitBrowser*(driver);

}

}

Output:  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 2861

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672039143.891][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

Note: We cannot see the data passing while executing, so we need to use Thread.sleep and surround try catch in a class “LoginPage”.

**package** com.learnautomation.pages;

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

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.support.FindBy;

**public** **class** LoginPage {

WebDriver driver;

**public** LoginPage(WebDriver ldriver) {

**this**.driver = ldriver;

}

@FindBy(name="username") WebElement uname;

@FindBy(name="password") WebElement pass;

@FindBy(xpath="//button[@class='oxd-button oxd-button--medium oxd-button--main orangehrm-login-button']") WebElement loginButton;

**public** **void** loginToHRM(String usernameApplication, String passwordApplication) {

**try** {

Thread.*sleep*(4000);

} **catch** (InterruptedException e) {

}

uname.sendKeys(usernameApplication);

pass.sendKeys(passwordApplication);

loginButton.click();

}

}

Note: Output remains same.

🡪Now use annotations in a class “LoginTestHRM”.  
  
**package** com.learnautomation.testcases;

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

**import** org.openqa.selenium.support.PageFactory;

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

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

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.LoginPage;

**import** com.learnautomation.utility.BrowserFactory;

**public** **class** LoginTestHRM {

WebDriver driver;

@BeforeClass

**public** **void** setup() {

driver = BrowserFactory.*startApplication*(driver, "Chrome", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

}

@Test

**public** **void** loginApp() {

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

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

}

@AfterClass

**public** **void** tearDown() {

BrowserFactory.*quitBrowser*(driver);

}

}

🡪Now create a class “BaseClass” in a package “com.learnautomation.pages”.  
  
**package** com.learnautomation.pages;

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

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

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

**import** com.learnautomation.utility.BrowserFactory;

**public** **class** BaseClass {

WebDriver driver;

@BeforeClass

**public** **void** setup() {

driver = BrowserFactory.*startApplication*(driver, "Chrome", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

}

@AfterClass

**public** **void** tearDown() {

BrowserFactory.*quitBrowser*(driver);

}

}

🡪Now do the modifications in “LoginTestHRM” class.  
  
**package** com.learnautomation.testcases;

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

**import** org.openqa.selenium.support.PageFactory;

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

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

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.BaseClass;

**import** com.learnautomation.pages.LoginPage;

**import** com.learnautomation.utility.BrowserFactory;

**public** **class** LoginTestHRM **extends** BaseClass{

@Test

**public** **void** loginApp() {

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

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

}

}  
  
Note: Change the visibility of WebDriver into public in “BaseClass” class.  
  
**public** WebDriver driver;

🡪Now execute the class “LoginTestHRM”  
  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 30110

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672041247.525][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

🡪Now create on xml file “testing.xml and execute.  
  
<?xml version=*"1.0"* encoding=*"UTF-8"*?>

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

<suite name=*"Suite"*>

<test name=*"Test"*>

<classes>

<class name=*"com.learnautomation.testcases.LoginTestHRM"*/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

🡪Now execute the xml file.

[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 10133

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672044225.728][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

===============================================

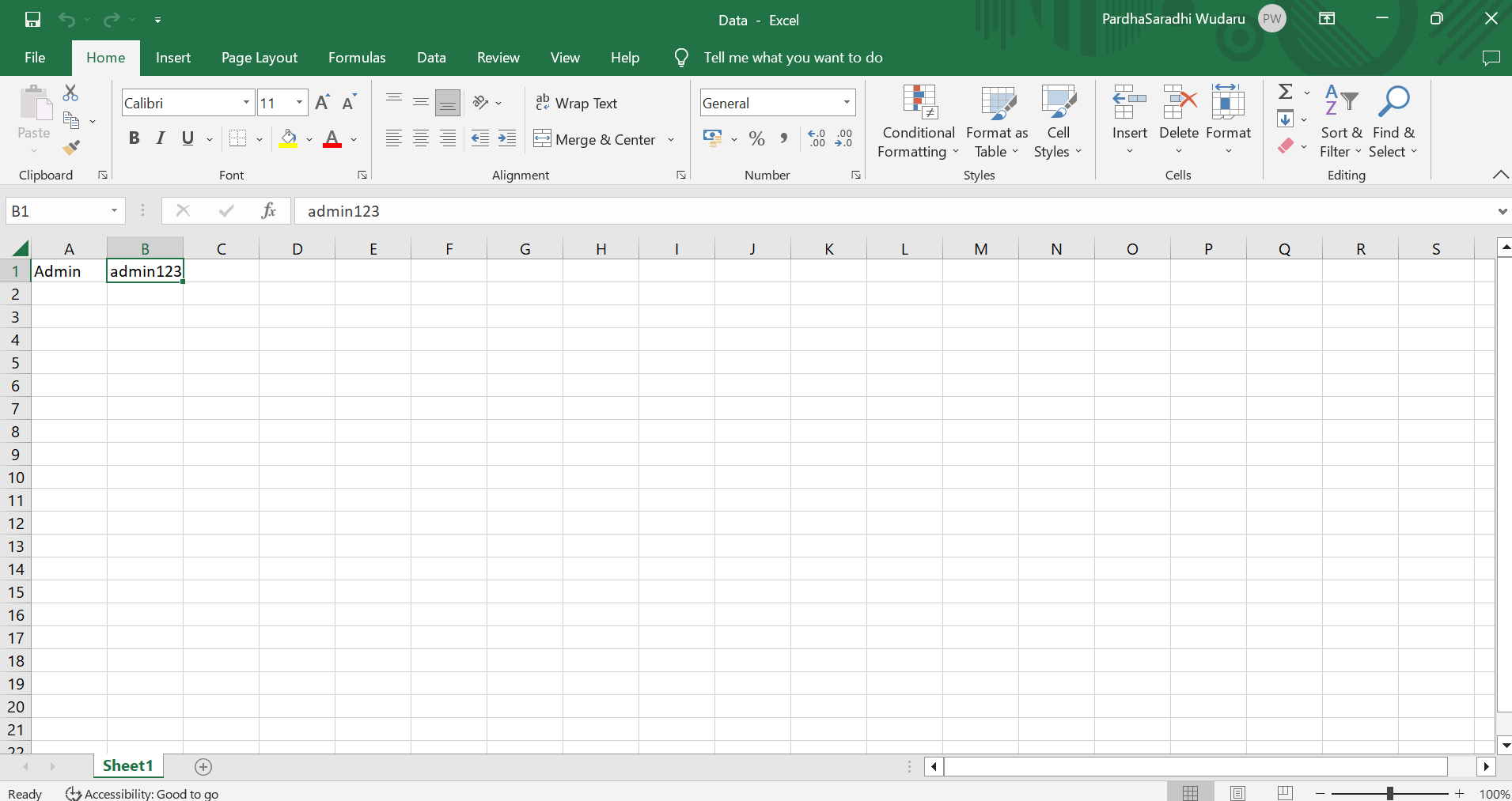
Suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

1. Create Data Provider and Custom Libraries

🡪Now create folder “TestData”. In this folder create on excel file “Data”, add data and save it.



🡪Now create class “ExcelDataProvider” in the package “com.learnautomation.utility.

**package** com.learnautomation.utility;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

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

**public** **class** ExcelDataProvider {

XSSFWorkbook wb;

**public** ExcelDataProvider() {

File src = **new** File("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\TestData\\Data.xlsx");

**try** {

FileInputStream fis = **new** FileInputStream(src);

wb = **new** XSSFWorkbook(fis);

} **catch** (Exception e) {

System.***out***.println(" Unable to read Excel File " + e.getMessage());

}

}

**public** String getStringData(**int** sheetIndex, **int** row, **int** column) {

**return** wb.getSheetAt(sheetIndex).getRow(row).getCell(column).getStringCellValue();

}

**public** String getStringData(String sheetName, **int** row, **int** column) {

**return** wb.getSheet(sheetName).getRow(row).getCell(column).getStringCellValue();

}

**public** **double** getNumericData(String sheetName, **int** row, **int** column) {

**return** wb.getSheet(sheetName).getRow(row).getCell(column).getNumericCellValue();

}

}

🡪Now do the modifications in the class “LoginTestHRM”.  
  
**package** com.learnautomation.testcases;

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

**import** org.openqa.selenium.support.PageFactory;

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

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

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.BaseClass;

**import** com.learnautomation.pages.LoginPage;

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ExcelDataProvider;

**public** **class** LoginTestHRM **extends** BaseClass{

@Test

**public** **void** loginApp() {

ExcelDataProvider excel = **new** ExcelDataProvider();

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

loginPage.loginToHRM(excel.getStringData("Sheet1", 0, 0), excel.getStringData("Sheet1", 0, 1));

}

}

Note: In POM.xml remove poi – ooxml schemas and add poi ooxml lite dependencies and execute the class “LoginTestHRM”.

Output:  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 22661

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672047637.979][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

Note: If you run in xml file, it remains same.

🡪You can do same in other format:  
a.  
**package** com.learnautomation.pages;

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

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

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

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ExcelDataProvider;

**public** **class** BaseClass {

**public** WebDriver driver;

**public** ExcelDataProvider excel;

**public** **void** setUpSuite() {

excel = **new** ExcelDataProvider();

}

@BeforeClass

**public** **void** setup() {

driver = BrowserFactory.*startApplication*(driver, "Chrome", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

}

@AfterClass

**public** **void** tearDown() {

BrowserFactory.*quitBrowser*(driver);

}

}

b.

**package** com.learnautomation.testcases;

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

**import** org.openqa.selenium.support.PageFactory;

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

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

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.BaseClass;

**import** com.learnautomation.pages.LoginPage;

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ExcelDataProvider;

**public** **class** LoginTestHRM **extends** BaseClass{

@Test

**public** **void** loginApp() {

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

loginPage.loginToHRM(excel.getStringData("Sheet1", 0, 0), excel.getStringData("Sheet1", 0, 1));

}

}

🡪 Create folder “Config” and a file in that folder “Config.properties”.

Browser=Chrome

qaUrl=https://opensource-demo.orangehrmlive.com/web/index.php/auth/login

🡪Create a class “ConfigDataProvider”.  
  
**package** com.learnautomation.utility;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.util.Properties;

**public** **class** ConfigDataProvider {

Properties pro;

**public** ConfigDataProvider() {

File src = **new** File("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Config\\Config.properties");

**try** {

FileInputStream fis = **new** FileInputStream(src);

pro = **new** Properties();

pro.load(fis);

} **catch** (Exception e) {

System.***out***.println("Not able to load Config File >>>" + e.getMessage());

}

}

**public** String getDataFromConfig(String keyToSearch) {

**return** pro.getProperty(keyToSearch);

}

**public** String getBrowser() {

**return** pro.getProperty("Browser");

}

**public** String getStagingURL() {

**return** pro.getProperty("qaUrl");

}

}

1. Now do the changes in the class “BaseClass”.

**package** com.learnautomation.pages;

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

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

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

**import** org.testng.annotations.BeforeSuite;

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ConfigDataProvider;

**import** com.learnautomation.utility.ExcelDataProvider;

**public** **class** BaseClass {

**public** WebDriver driver;

**public** ExcelDataProvider excel;

**public** ConfigDataProvider config;

@BeforeSuite

**public** **void** setUpSuite() {

excel = **new** ExcelDataProvider();

config = **new** ConfigDataProvider();

}

@BeforeClass

**public** **void** setup() {

driver = BrowserFactory.*startApplication*(driver,config.getBrowser(),config.getStagigURL());

//driver = BrowserFactory.startApplication(driver,"Edge","https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

}

@AfterClass

**public** **void** tearDown() {

BrowserFactory.*quitBrowser*(driver);

}

}

1. Now do the modifications in class “LoginTestHRM” and execute.

**package** com.learnautomation.testcases;

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

**import** org.openqa.selenium.support.PageFactory;

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

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

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.BaseClass;

**import** com.learnautomation.pages.LoginPage;

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ExcelDataProvider;

**public** **class** LoginTestHRM **extends** BaseClass{

@Test

**public** **void** loginApp() {

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

loginPage.loginToHRM(excel.getStringData("Sheet1", 0, 0), excel.getStringData("Sheet1", 0, 1));

}

}

Output:  
  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 34958

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672056065.558][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

How to take a screenshot if test fails:  
🡪Create a class “Helper” in “com.learnautomation.utility” package.  
**package** com.learnautomation.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** Helper {

//Screenshot, alerts, frames, windows, sync issue, javascipt executor

**public** **static** **void** captureScreenshot(WebDriver driver) {

File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

**try** {

FileHandler.*copy*(src, **new** File("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Screenshots\\Login.png"));

System.***out***.println("Screenshot captured");

} **catch** (IOException e) {

System.***out***.println("Unable to capture screenshot " + e.getMessage());

}

}

}

🡪Now do the modifications in the class “BaseClass”.  
**package** com.learnautomation.pages;

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

**import** org.testng.ITestResult;

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

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

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

**import** org.testng.annotations.BeforeSuite;

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ConfigDataProvider;

**import** com.learnautomation.utility.ExcelDataProvider;

**import** com.learnautomation.utility.Helper;

**public** **class** BaseClass {

**public** WebDriver driver;

**public** ExcelDataProvider excel;

**public** ConfigDataProvider config;

@BeforeSuite

**public** **void** setUpSuite() {

excel = **new** ExcelDataProvider();

config = **new** ConfigDataProvider();

}

@BeforeClass

**public** **void** setup() {

driver = BrowserFactory.*startApplication*(driver,config.getBrowser(),config.getStagingURL());

//driver = BrowserFactory.startApplication(driver,"Edge", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

}

@AfterClass

**public** **void** tearDown() {

BrowserFactory.*quitBrowser*(driver);

}

@AfterMethod

**public** **void** tearDownMethod(ITestResult result) {

**if**(result.getStatus()==ITestResult.***FAILURE***) {

Helper.*captureScreenshot*(driver);

}

}

}

🡪Now run the class “LoginTestHRM”. Now the test will pass and it cannot takes the screenshot because it not fails.  
🡪If we change the username as username1 in the class “LoginPage”, then test will fails and takes screen shot.

Output after changing username as username1 in “LoginPage” class:  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 35916

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672058273.689][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

Screenshot captured

===============================================

Suite

Total tests run: 1, Passes: 0, Failures: 1, Skips: 0

===============================================

Capturing Screenshot with Current date and time:  
🡪Add the method and do the modifications in the class “Helper”.  
**package** com.learnautomation.utility;

**import** java.io.File;

**import** java.io.IOException;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.TakesScreenshot;

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

**import** org.openqa.selenium.io.FileHandler;

**public** **class** Helper {

//Screenshot, alerts, frames, windows, sync issue, javascipt executor

**public** **static** **void** captureScreenshot(WebDriver driver) {

File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.***FILE***);

**try** {

FileHandler.*copy*(src, **new** File("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Screenshots\\OrangeHRM\_" + *getCurrentDateTime*()+ ".png"));

System.***out***.println("Screenshot captured");

} **catch** (IOException e) {

System.***out***.println("Unable to capture screenshot " + e.getMessage());

}

}

**public** **static** String getCurrentDateTime() {

DateFormat customformat = **new** SimpleDateFormat("MM\_dd\_yyyy\_HH\_mm\_ss");

Date currentDate = **new** Date();

**return** customformat.format(currentDate);

}

}

🡪In class “LoginPage” change the username as username1 and execute the xml file with TestNGSuite. It fails and takes the screenshot.

[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 4201

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672060063.135][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

Screenshot captured

===============================================

Suite

Total tests run: 1, Passes: 0, Failures: 1, Skips: 0

===============================================

1. Integrate Extent Report and Logs in Framework

🡪Create one folder “Reports” to the project.

🡪Do the modifications in the class “BaseClass”.  
**package** com.learnautomation.pages;

**import** java.io.File;

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

**import** org.testng.ITestResult;

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

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

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

**import** org.testng.annotations.BeforeSuite;

**import** com.aventstack.extentreports.ExtentReports;

**import** com.aventstack.extentreports.ExtentTest;

**import** com.aventstack.extentreports.reporter.ExtentHtmlReporter;

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ConfigDataProvider;

**import** com.learnautomation.utility.ExcelDataProvider;

**import** com.learnautomation.utility.Helper;

**public** **class** BaseClass {

**public** WebDriver driver;

**public** ExcelDataProvider excel;

**public** ConfigDataProvider config;

**public** ExtentReports report;

**public** ExtentTest logger;

@BeforeSuite

**public** **void** setUpSuite() {

excel = **new** ExcelDataProvider();

config = **new** ConfigDataProvider();

ExtentHtmlReporter extent = **new** ExtentHtmlReporter("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Reports\\OrangeHRM.html");

report = **new** ExtentReports();

report.attachReporter(extent);

}

@BeforeClass

**public** **void** setup() {

driver = BrowserFactory.*startApplication*(driver,config.getBrowser(),config.getStagingURL());

//driver = BrowserFactory.startApplication(driver,"Edge", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

}

@AfterClass

**public** **void** tearDown() {

BrowserFactory.*quitBrowser*(driver);

}

@AfterMethod

**public** **void** tearDownMethod(ITestResult result) {

**if**(result.getStatus()==ITestResult.***FAILURE***) {

Helper.*captureScreenshot*(driver);

}

report.flush();

}

}

🡪Now do the modifications in the class “LoginTestHRM” and execute.  
**package** com.learnautomation.testcases;

**import** org.openqa.selenium.support.PageFactory;

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.BaseClass;

**import** com.learnautomation.pages.LoginPage;

**public** **class** LoginTestHRM **extends** BaseClass{

@Test

**public** **void** loginApp() {

logger = report.createTest("Login to HRM");

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

logger.info("Starting Aplication");

loginPage.loginToHRM(excel.getStringData("Sheet1", 0, 0), excel.getStringData("Sheet1", 0, 1));

logger.pass("Login Success");

}

}

Output:

[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 20348

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672134843.488][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp

===============================================

Default test

Tests run: 1, Failures: 0, Skips: 0

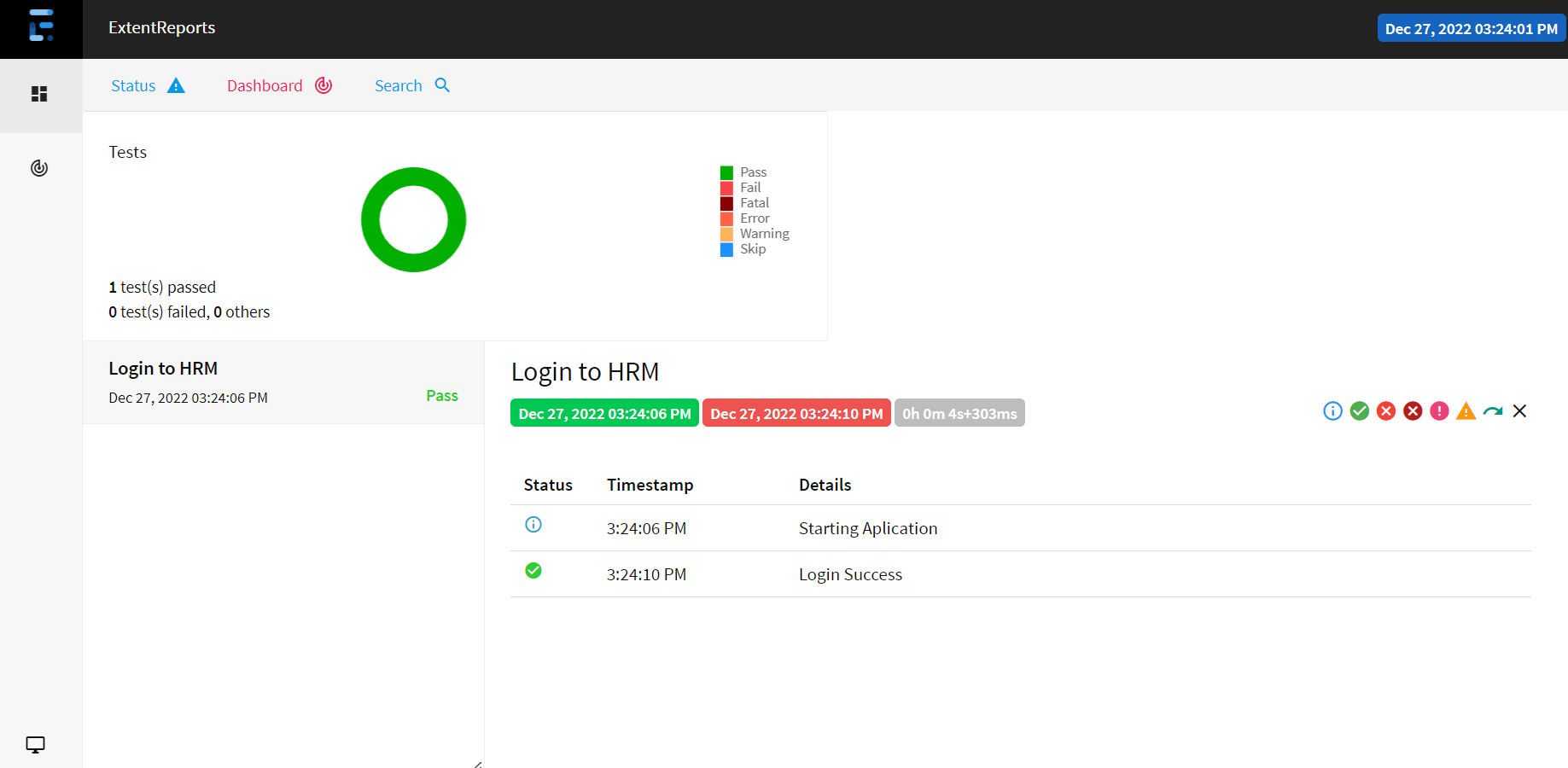
===============================================

===============================================

Default suite

Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

===============================================

Report:  


🡪If we write two test methods in class “LoginTestHRM” and execute:  
**package** com.learnautomation.testcases;

**import** org.openqa.selenium.support.PageFactory;

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.BaseClass;

**import** com.learnautomation.pages.LoginPage;

**public** **class** LoginTestHRM **extends** BaseClass{

@Test(priority=1)

**public** **void** loginApp() {

logger = report.createTest("Login to HRM");

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

logger.info("Starting Aplication");

loginPage.loginToHRM(excel.getStringData("Sheet1", 0, 0), excel.getStringData("Sheet1", 0, 1));

logger.pass("Login Success");

}

@Test(priority=2)

**public** **void** loginApp1() {

logger = report.createTest("Logout");

logger.fail("Login Failed");

}

}

Output:  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 2663

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672135268.881][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp

PASSED: com.learnautomation.testcases.LoginTestHRM.loginApp1

===============================================

Default test

Tests run: 2, Failures: 0, Skips: 0

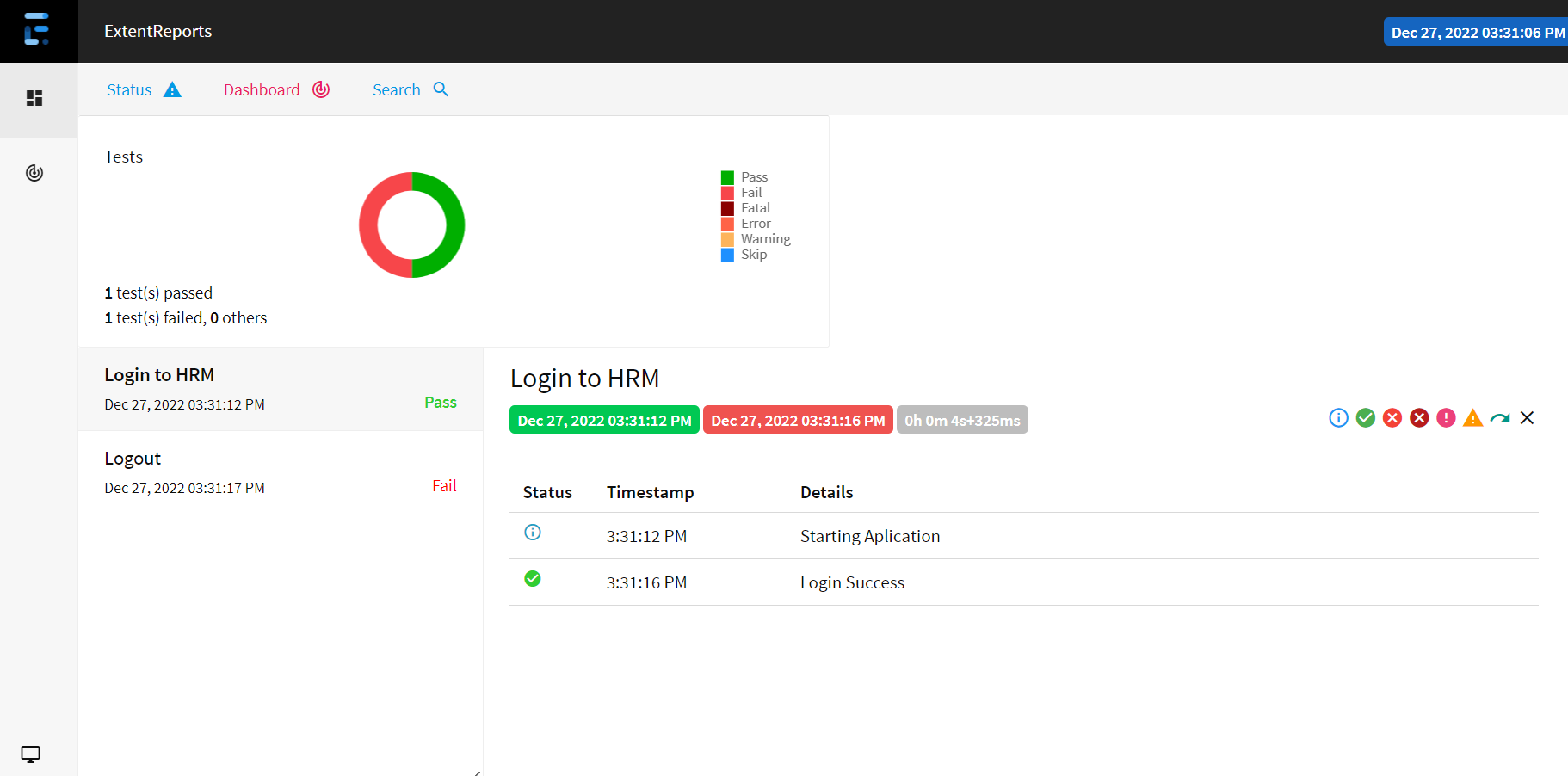
===============================================

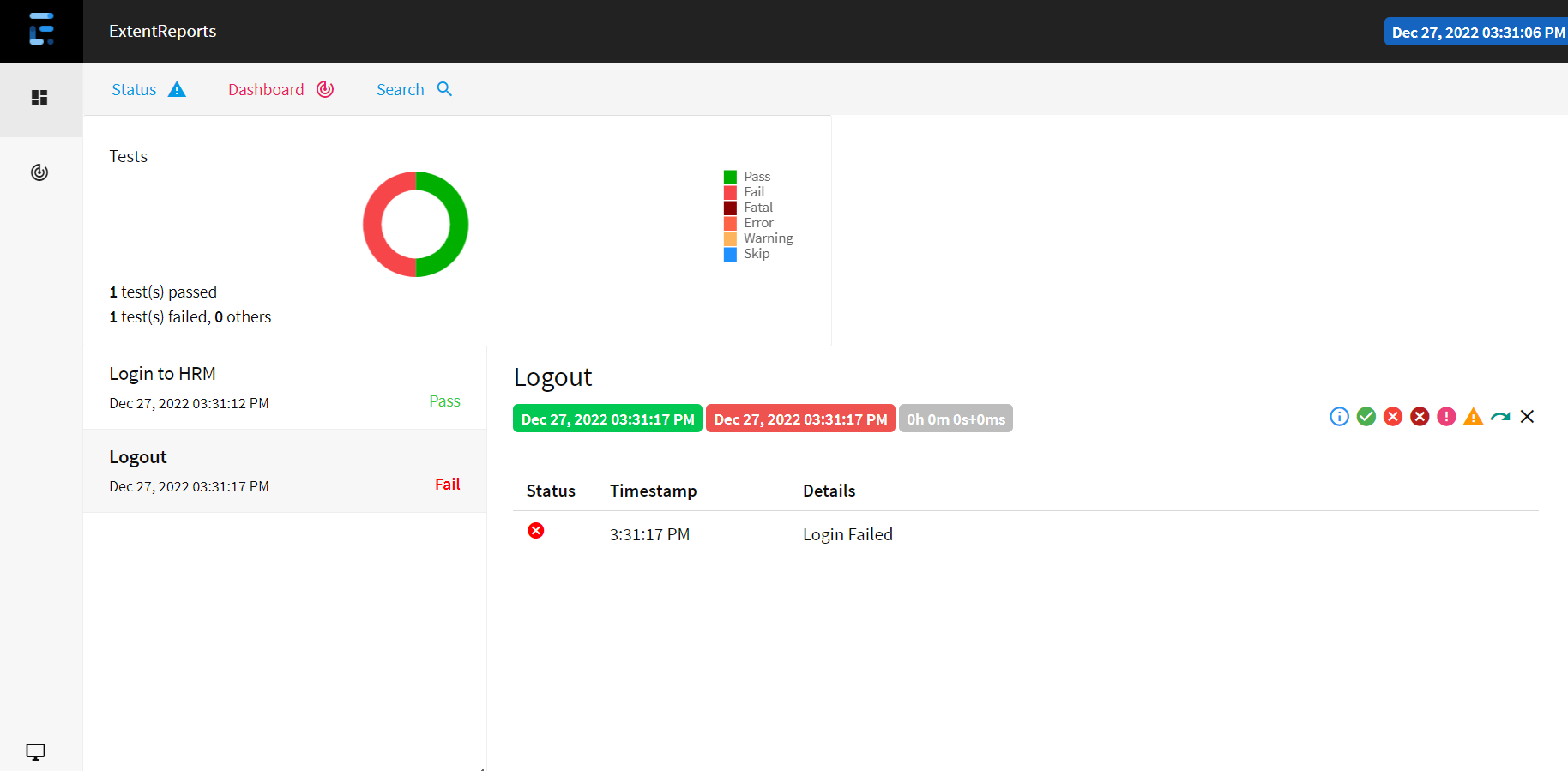
===============================================

Default suite

Total tests run: 2, Passes: 2, Failures: 0, Skips: 0

===============================================

Report:  




🡪If we change the path format in class “BaseClass” like this:  
ExtentHtmlReporter extent = **new** ExtentHtmlReporter("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Reports\\OrangeHRM" + Helper.*getCurrentDateTime*() + ".html");

We create a report in the format, OrangeHRM12\_27\_2022\_15\_45\_59.html.

To get Screenshot and Report at a single time:  
🡪Now do the modifications in the class “Helper”

package com.learnautomation.utility;

import java.io.File;

import java.io.IOException;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.io.FileHandler;

public class Helper {

//Screenshot, alerts, frames, windows, sync issue, javascipt executor

public static String captureScreenshot(WebDriver driver) {

File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

try {

FileHandler.copy(src, new File("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Screenshots\\OrangeHRM1\_" + getCurrentDateTime()+ ".png"));

System.out.println("Screenshot captured");

} catch (IOException e) {

System.out.println("Unable to capture screenshot " + e.getMessage());

}

return null;

}

public static String getCurrentDateTime() {

DateFormat customformat = new SimpleDateFormat("MM\_dd\_yyyy\_HH\_mm\_ss");

Date currentDate = new Date();

return customformat.format(currentDate);

}

}

🡪Now the class “BaseClass” should be:

**package** com.learnautomation.pages;

**import** java.io.File;

**import** java.io.IOException;

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

**import** org.testng.ITestResult;

**import** org.testng.Reporter;

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

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

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

**import** org.testng.annotations.BeforeSuite;

**import** com.aventstack.extentreports.ExtentReports;

**import** com.aventstack.extentreports.ExtentTest;

**import** com.aventstack.extentreports.MediaEntityBuilder;

**import** com.aventstack.extentreports.reporter.ExtentHtmlReporter;

**import** com.learnautomation.utility.BrowserFactory;

**import** com.learnautomation.utility.ConfigDataProvider;

**import** com.learnautomation.utility.ExcelDataProvider;

**import** com.learnautomation.utility.Helper;

**public** **class** BaseClass {

**public** WebDriver driver;

**public** ExcelDataProvider excel;

**public** ConfigDataProvider config;

**public** ExtentReports report;

**public** ExtentTest logger;

@BeforeSuite

**public** **void** setUpSuite() {

Reporter.*log*("Setting up reports and Test is getting ready", **true**);

excel = **new** ExcelDataProvider();

config = **new** ConfigDataProvider();

ExtentHtmlReporter extent = **new** ExtentHtmlReporter("C:\\Users\\ADMIN\\eclipse-workspace\\iNetBankingV4\\Reports\\OrangeHRM" + Helper.*getCurrentDateTime*() + ".html");

report = **new** ExtentReports();

report.attachReporter(extent);

Reporter.*log*("Setting done - Test can be started", **true**);

}

@BeforeClass

**public** **void** setup() {

Reporter.*log*("Trying to start browser and getting application ready", **true**);

driver = BrowserFactory.*startApplication*(driver,config.getBrowser(),config.getStagingURL());

//driver = BrowserFactory.startApplication(driver,"Edge", "https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

System.***out***.println(driver.getTitle());

Reporter.*log*("Browser and application is up and running", **true**);

}

@AfterClass

**public** **void** tearDown() {

BrowserFactory.*quitBrowser*(driver);

}

@AfterMethod

**public** **void** tearDownMethod(ITestResult result) **throws** IOException {

Reporter.*log*("Test is about to end", **true**);

**if**(result.getStatus()==ITestResult.***FAILURE***) {

Helper.*captureScreenshot*(driver);

//logger.fail("Test Failed", MediaEntityBuilder.createScreenCaptureFromPath(Helper.captureScreenshot(driver)).build());

}

report.flush();

Reporter.*log*("Tests Completed >>> Reports Generated", **true**);

}

}

🡪Now do the modifications in class “LoginTestHRM” and execute.

**package** com.learnautomation.testcases;

**import** org.openqa.selenium.support.PageFactory;

**import** org.testng.annotations.Test;

**import** com.learnautomation.pages.BaseClass;

**import** com.learnautomation.pages.LoginPage;

**public** **class** LoginTestHRM **extends** BaseClass{

@Test

**public** **void** loginApp() {

logger = report.createTest("Login to HRM");

LoginPage loginPage = PageFactory.*initElements*(driver, LoginPage.**class**);

logger.info("Starting Aplication");

loginPage.loginToHRM(excel.getStringData("Sheet1", 0, 0), excel.getStringData("Sheet1", 0, 1));

logger.pass("Login Success");

}

}

Output:  
[RemoteTestNG] detected TestNG version 7.7.0

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.

Setting up reports and Test is getting ready

Setting done - Test can be started

Trying to start browser and getting application ready

Starting ChromeDriver 107.0.5304.62 (1eec40d3a5764881c92085aaee66d25075c159aa-refs/branch-heads/5304@{#942}) on port 39509

Only local connections are allowed.

Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.

ChromeDriver was started successfully.

[1672203082.728][WARNING]: This version of ChromeDriver has not been tested with Chrome version 108.

OrangeHRM

Browser and application is up and running

Test is about to end

Screenshot captured

Tests Completed >>> Reports Generated

FAILED: com.learnautomation.testcases.LoginTestHRM.loginApp

org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element: {"method":"css selector","selector":"\*[name='username1']"}

(Session info: chrome=108.0.5359.125)

For documentation on this error, please visit: https://selenium.dev/exceptions/#no\_such\_element

Build info: version: '4.7.2', revision: '4d4020c3b7'

System info: os.name: 'Windows 10', os.arch: 'amd64', os.version: '10.0', java.version: '17.0.5'

Driver info: org.openqa.selenium.chrome.ChromeDriver

Command: [be800deaccbaef54de1276c436a46e1b, findElement {using=name, value=username1}]

Capabilities {acceptInsecureCerts: false, browserName: chrome, browserVersion: 108.0.5359.125, chrome: {chromedriverVersion: 107.0.5304.62 (1eec40d3a576..., userDataDir: C:\Users\ADMIN\AppData\Loca...}, goog:chromeOptions: {debuggerAddress: localhost:55114}, networkConnectionEnabled: false, pageLoadStrategy: normal, platformName: WINDOWS, proxy: Proxy(), se:cdp: ws://localhost:55114/devtoo..., se:cdpVersion: 108.0.5359.125, setWindowRect: true, strictFileInteractability: false, timeouts: {implicit: 0, pageLoad: 300000, script: 30000}, unhandledPromptBehavior: dismiss and notify, webauthn:extension:credBlob: true, webauthn:extension:largeBlob: true, webauthn:virtualAuthenticators: true}

Session ID: be800deaccbaef54de1276c436a46e1b

at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:77)

at java.base/jdk.internal.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)

at java.base/java.lang.reflect.Constructor.newInstanceWithCaller(Constructor.java:499)

at java.base/java.lang.reflect.Constructor.newInstance(Constructor.java:480)

at org.openqa.selenium.remote.codec.w3c.W3CHttpResponseCodec.createException(W3CHttpResponseCodec.java:200)

at org.openqa.selenium.remote.codec.w3c.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:133)

at org.openqa.selenium.remote.codec.w3c.W3CHttpResponseCodec.decode(W3CHttpResponseCodec.java:53)

at org.openqa.selenium.remote.HttpCommandExecutor.execute(HttpCommandExecutor.java:184)

at org.openqa.selenium.remote.service.DriverCommandExecutor.invokeExecute(DriverCommandExecutor.java:167)

at org.openqa.selenium.remote.service.DriverCommandExecutor.execute(DriverCommandExecutor.java:142)

at org.openqa.selenium.remote.RemoteWebDriver.execute(RemoteWebDriver.java:535)

at org.openqa.selenium.remote.ElementLocation$ElementFinder$2.findElement(ElementLocation.java:162)

at org.openqa.selenium.remote.ElementLocation.findElement(ElementLocation.java:66)

at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:351)

at org.openqa.selenium.remote.RemoteWebDriver.findElement(RemoteWebDriver.java:343)

at org.openqa.selenium.support.pagefactory.DefaultElementLocator.findElement(DefaultElementLocator.java:70)

at org.openqa.selenium.support.pagefactory.internal.LocatingElementHandler.invoke(LocatingElementHandler.java:39)

at jdk.proxy2/jdk.proxy2.$Proxy21.sendKeys(Unknown Source)

at com.learnautomation.pages.LoginPage.loginToHRM(LoginPage.java:26)

at com.learnautomation.testcases.LoginTestHRM.loginApp(LoginTestHRM.java:17)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)

at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.base/java.lang.reflect.Method.invoke(Method.java:568)

at org.testng.internal.invokers.MethodInvocationHelper.invokeMethod(MethodInvocationHelper.java:139)

at org.testng.internal.invokers.TestInvoker.invokeMethod(TestInvoker.java:677)

at org.testng.internal.invokers.TestInvoker.invokeTestMethod(TestInvoker.java:221)

at org.testng.internal.invokers.MethodRunner.runInSequence(MethodRunner.java:50)

at org.testng.internal.invokers.TestInvoker$MethodInvocationAgent.invoke(TestInvoker.java:969)

at org.testng.internal.invokers.TestInvoker.invokeTestMethods(TestInvoker.java:194)

at org.testng.internal.invokers.TestMethodWorker.invokeTestMethods(TestMethodWorker.java:148)

at org.testng.internal.invokers.TestMethodWorker.run(TestMethodWorker.java:128)

at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)

at org.testng.TestRunner.privateRun(TestRunner.java:829)

at org.testng.TestRunner.run(TestRunner.java:602)

at org.testng.SuiteRunner.runTest(SuiteRunner.java:437)

at org.testng.SuiteRunner.runSequentially(SuiteRunner.java:431)

at org.testng.SuiteRunner.privateRun(SuiteRunner.java:391)

at org.testng.SuiteRunner.run(SuiteRunner.java:330)

at org.testng.SuiteRunnerWorker.runSuite(SuiteRunnerWorker.java:52)

at org.testng.SuiteRunnerWorker.run(SuiteRunnerWorker.java:95)

at org.testng.TestNG.runSuitesSequentially(TestNG.java:1256)

at org.testng.TestNG.runSuitesLocally(TestNG.java:1176)

at org.testng.TestNG.runSuites(TestNG.java:1099)

at org.testng.TestNG.run(TestNG.java:1067)

at org.testng.remote.AbstractRemoteTestNG.run(AbstractRemoteTestNG.java:115)

at org.testng.remote.RemoteTestNG.initAndRun(RemoteTestNG.java:251)

at org.testng.remote.RemoteTestNG.main(RemoteTestNG.java:77)

===============================================

Default test

Tests run: 1, Failures: 1, Skips: 0

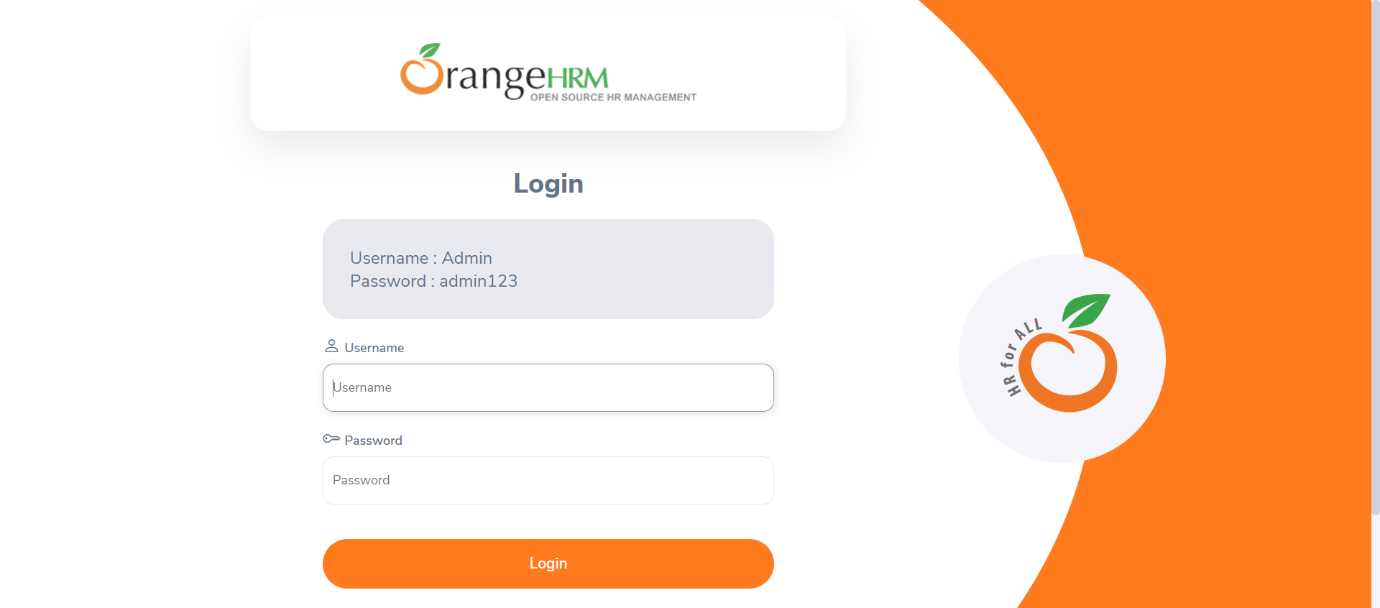
===============================================

===============================================

Default suite

Total tests run: 1, Passes: 0, Failures: 1, Skips: 0

===============================================

Screenshot: OrangeHRM1\_12\_28\_2022\_10\_22\_01.png  


Report: OrangeHRM12\_28\_2022\_10\_21\_20.html  
