**BaseClass:**

package executePageClasses;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

import org.testng.annotations.BeforeMethod;

import java.time.Duration;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterMethod;

public class BaseClass {

WebDriver driver;

@BeforeMethod

public void beforeMethod() {

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://qabible.in/payrollapp/site/login");

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

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

}

@AfterMethod

public void afterMethod() {

driver.quit();

}

}

**LoginPageClass:**

package pageClasses;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

import utilities.GeneralUtilities;

public class LoginPageClass {

WebDriver driver;

GeneralUtilities gl = new GeneralUtilities();

public LoginPageClass(WebDriver driver) {

this.driver = driver;

PageFactory.initElements(driver, this);

}

@FindBy(id = "loginform-username")

WebElement userName;

@FindBy(id = "loginform-password")

WebElement password;

@FindBy(xpath = "//button[@name='login-button']")

WebElement loginButton;

@FindBy(xpath = "//p[contains(text(),'Welcome to Payroll')]")

WebElement welcomeMessage;

@FindBy(xpath = "//input[@id='loginform-rememberme']")

WebElement rememberMe;

@FindBy(xpath = "//\*[contains(text(),'Incorrect username or')]")

WebElement errorMessage;

public void login(String uname, String pswd) {

gl.typeElement(userName, uname);

gl.typeElement(password, pswd);

gl.clickElement(loginButton);

}

public boolean isRememberMeSelected() {

return gl.isSelectedMethod(rememberMe);

}

public String getTextOfWelcomeMessage() {

return gl.getTextofElement(welcomeMessage);

}

public boolean isWelcomeMessageDisplayed() {

return gl.isDisplayedMethod(welcomeMessage);

}

public String getTextofErrorMessage() {

return gl.getTextofElement(errorMessage);

}

}

**LoginPagreTestClass:**

**package** executePageClasses;

**import** org.testng.Assert;

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

**import** pageClasses.LoginPageClass;

**public** **class** LoginPageTestClass **extends** BaseClass {

LoginPageClass lp;

@Test

**public** **void** verifySuccessfulLogin() {

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

lp.login("carol", "1q2w3e4r");

String actual = lp.getTextOfWelcomeMessage();

String expected = "Welcome to Payroll Application";

Assert.*assertEquals*(actual, expected);

}

@Test

**public** **void** verifyUnSuccessfulLogin() {

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

lp.login("carol", "welcome");

String actualResult = lp.getTextofErrorMessage();

String expectedResult = "Incorrect username or password.";

Assert.*assertEquals*(actualResult, expectedResult);

}

@Test

**public** **void** verifyRememberMeCheckBoxIsSelectedByDefaultOrNot() {

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

lp.login("carol", "1q2w3e4r");

**boolean** actualResult = lp.isRememberMeSelected();

Assert.*assertTrue*(actualResult);

}

}

**HomePageClass:**

**package** pageClasses;

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

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

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

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

**import** utilities.GeneralUtilities;

**public** **class** HomePageClass {

WebDriver driver;

GeneralUtilities gl = **new** GeneralUtilities();

**public** HomePageClass(WebDriver driver) {

**this**.driver = driver;

PageFactory.*initElements*(driver, **this**);

}

@FindBy(id = "loginform-username")

WebElement userName;

@FindBy(id = "loginform-password")

WebElement password;

@FindBy(xpath = "//button[@name='login-button']")

WebElement loginButton;

@FindBy(xpath = "//p[contains(text(),'Welcome to Payroll')]")

WebElement welcomeMessage;

@FindBy(xpath = "//img[@src='/payrollapp/images/logo.png']")

WebElement logoImage;

@FindBy(xpath = "//h1[contains(text(),'Payroll Application')]")

WebElement dashboardDisplay;

@FindBy(xpath = "//a[@href='/payrollapp/user']")

WebElement settingsTab;

@FindBy(xpath = "//a[@class='dropdown-toggle']")

WebElement logout;

**public** **void** login(String uname, String pswd) {

gl.typeElement(userName, uname);

gl.typeElement(password, pswd);

gl.clickElement(loginButton);

}

**public** String getTextOfWelcomeMessage() {

**return** gl.getTextofElement(welcomeMessage);

}

**public** **boolean** isLogoDisplayed() {

**return** gl.isDisplayedMethod(logoImage);

}

**public** String getTextofHomePage() {

**return** gl.getTextofElement(dashboardDisplay);

}

**public** **boolean** isSettingsTabDisplayed() {

**return** gl.isDisplayedMethod(settingsTab);

}

**public** **boolean** isLogoutMenuDisplayed() {

**return** gl.isDisplayedMethod(logout);

}

}

**HomePageTestClass:**

**package** executePageClasses;

**import** org.testng.Assert;

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

**import** pageClasses.HomePageClass;

**public** **class** HomePageTestClass **extends** BaseClass {

HomePageClass hp;

@Test

**public** **void** verifyIfTheLogoImageIsDisplayedOrNot() {

hp = **new** HomePageClass(driver);

hp.login("carol", "1q2w3e4r");

String actual = hp.getTextOfWelcomeMessage();

String expected = "Welcome to Payroll Application";

Assert.*assertEquals*(actual, expected);

**boolean** actualRes = hp.isLogoDisplayed();

Assert.*assertTrue*(actualRes);

}

@Test

**public** **void** verifyIfTheDashboardMessageIsDisplayed() {

hp = **new** HomePageClass(driver);

hp.login("carol", "1q2w3e4r");

String act = hp.getTextofHomePage();

String exp = "PAYROLL APPLICATION";

Assert.*assertEquals*(act, exp);

}

@Test

**public** **void** verifyIfTheSettingsTabIsDisplayed() {

hp = **new** HomePageClass(driver);

hp.login("carol", "1q2w3e4r");

**boolean** actualRes = hp.isSettingsTabDisplayed();

Assert.*assertTrue*(actualRes);

}

@Test

**public** **void** verifyIfLogoutOptionIsViewableInHomepageDashboard() {

hp = **new** HomePageClass(driver);

hp.login("carol", "1q2w3e4r");

**boolean** actualResult = hp.isLogoutMenuDisplayed();

Assert.*assertTrue*(actualResult);

}

}

**ClientsPageClass:**

**package** pageClasses;

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

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

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

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

**import** utilities.GeneralUtilities;

**public** **class** ClientsPageClass {

WebDriver driver;

GeneralUtilities gl = **new** GeneralUtilities();

**public** ClientsPageClass(WebDriver driver) {

**this**.driver = driver;

PageFactory.*initElements*(driver, **this**);

}

@FindBy(id = "loginform-username")

WebElement userName;

@FindBy(id = "loginform-password")

WebElement password;

@FindBy(xpath = "//button[@name='login-button']")

WebElement loginButton;

@FindBy(xpath = "//a[@href='/payrollapp/client/index']")

WebElement clientsTab;

@FindBy(xpath = "//a[@href='/payrollapp/client/create']")

WebElement createClientField;

@FindBy(xpath = "//\*[@id='clientsearch-client\_name']")

WebElement clientName;

@FindBy(xpath = "//input[@id='clientsearch-id']")

WebElement clientId;

@FindBy(xpath = "//button[@class='btn btn-primary']")

WebElement searchField;

@FindBy(xpath = "//\*[@class='btn btn-default']")

WebElement resetTab;

**public** **void** login(String uname, String pswd) {

gl.typeElement(userName, uname);

gl.typeElement(password, pswd);

gl.clickElement(loginButton);

}

**public** **void** clientsPage() {

gl.clickElement(clientsTab);

}

**public** **boolean** iscreateClientFieldDisplayed() {

**return** gl.isDisplayedMethod(createClientField);

}

**public** **void** sendKeystoElement(String value) {

gl.sendKeysClickElement(clientName, value);

}

**public** **void** clickSearchElement() {

gl.clickElement(searchField);

}

}

**ClientsPageTestClass:**

**package** executePageClasses;

**import** org.testng.Assert;

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

**import** pageClasses.ClientsPageClass;

**public** **class** ClientsPageTestClass **extends** BaseClass {

ClientsPageClass cp;

@Test

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

cp = **new** ClientsPageClass(driver);

cp.login("carol", "1q2w3e4r");

cp.clientsPage();

**boolean** ar = cp.iscreateClientFieldDisplayed();

Assert.*assertTrue*(ar);

cp.sendKeystoElement("Sabitha");

cp.clickSearchElement();

}

}

**GeneralUtilities:**

**package** utilities;

**import** org.openqa.selenium.JavascriptExecutor;

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

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

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

**public** **class** GeneralUtilities {

**public** **void** clickElement(WebElement element) {

element.click();

}

**public** **void** clearElement(WebElement element) {

element.clear();

}

**public** **void** typeElement(WebElement element, String text) {

element.sendKeys(text);

}

**public** **void** sendKeysClickElement(WebElement element, String click) {

element.sendKeys(click);

}

**public** String getTextofElement(WebElement element) {

**return** element.getText();

}

**public** String getTitleofElement(WebDriver driver) {

**return** driver.getTitle();

}

**public** String getCurrenturl(WebDriver driver) {

**return** driver.getCurrentUrl();

}

**public** **boolean** isDisplayedMethod(WebElement element) {

**return** element.isDisplayed();

}

**public** **boolean** isSelectedMethod(WebElement element) {

**return** element.isSelected();

}

**public** **boolean** isEnabledMethod(WebElement element) {

**return** isEnabledMethod(element);

}

**public** **void** clickElementJS(WebElement element, WebDriver driver) {

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click();", element);

}

**public** **void** scrollToAnElement(WebElement element, WebDriver driver) {

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].scrollIntoView();", element);

}

**public** **void** selectByIndex(WebElement element, **int** index) {

Select sel = **new** Select(element);

sel.selectByIndex(index);

}

**public** **void** selectByVisibleText(WebElement element, String value) {

Select select = **new** Select(element);

select.selectByVisibleText(value);

}

**public** **void** selectByValue(WebElement element, String value) {

Select sel = **new** Select(element);

sel.selectByValue(value);

}

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

Thread.*sleep*(2000);

}

**public** **void** alertAccept(WebDriver driver) {

driver.switchTo().alert().accept();

}

**public** **void** alertDismiss(WebDriver driver) {

driver.switchTo().alert().dismiss();

}

**public** **void** alertGetText(WebDriver driver) {

driver.switchTo().alert().getText();

}

**public** **void** alertSendKeys(WebDriver driver, String text) {

driver.switchTo().alert().sendKeys(text);

}

}