Page Object FrameWorks

HOMEPAGE:

package com.test1.pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class HomePage

{

private WebDriver driver;

public HomePage(WebDriver driver)

{

this.driver=driver;

}

public String getTitle()

{

return driver.getTitle();

}

public MyAccountPage clickMyAccountLink()

{

driver.findElement(By.linkText("My Account")).click();

return new MyAccountPage(driver);

}

public ShopPage clickShopLink()

{

driver.findElement(By.linkText("Shop")).click();

return new ShopPage(driver);

}

}

MYACCOUNT PAGE:

package com.test1.pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class MyAccountPage

{

private WebDriver driver;

public MyAccountPage(WebDriver driver)

{

this.driver=driver;

}

public String getTitle()

{

return driver.getTitle();

}

public MyAccountPage loginAs(String username,String password)

{

driver.findElement(By.id("username")).sendKeys(username);

driver.findElement(By.id("password")).sendKeys(password);

driver.findElement(By.name("login")).click();

return new MyAccountPage(driver);

}

public boolean isLoginSuccessful(String username)

{

return driver.findElement(By.id("user\_info")).getText().contains(username);

}

}

SHOPPAGE:

package com.test1.pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

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

public class ShopPage

{

private WebDriver driver;

public ShopPage(WebDriver driver)

{

this.driver=driver;

}

public String getTitle()

{

return driver.getTitle();

}

public ShopPage setSortOrder(String sortOrder)

{

Select sortList=new Select(driver.findElement(By.name("orderby")));

sortList.selectByVisibleText(sortOrder);

return new ShopPage(driver);

}

public String getSortOrder()

{

Select sortList=new Select(driver.findElement(By.name("orderby")));

return sortList.getFirstSelectedOption().getText();

}

}

LOGINTESTS:

package com.test1.tests;

import org.testng.annotations.Test;

import com.test1.pages.HomePage;

import org.testng.annotations.BeforeMethod;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

public class LoginTests

{

private WebDriver driver;

@BeforeMethod

public void beforeMethod()

{

System.setProperty("webdriver.gecko.driver","C:\\Users\\Sivachaitanya\\Downloads\\Compressed\\geckodriver-v0.13.0-win64\\geckodriver.exe");

driver = new FirefoxDriver();

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

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

driver.get("http://test1.absofttrainings.com");

}

@Test

public void testSuccessfulLogin()

{

HomePage homePage = new HomePage(driver);

boolean testResult = homePage.clickMyAccountLink()

.loginAs("testuser1", "testpwd1")

.isLoginSuccessful("testuser1");

Assert.assertTrue(testResult,"Login is not successful for user testuser1");

}

@AfterMethod

public void afterMethod()

{

//driver.quit();

}

}

SHOPPAGETEST:

package com.test1.tests;

import org.testng.annotations.Test;

import com.test1.pages.HomePage;

import org.testng.annotations.BeforeMethod;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

public class ShopPageTest {

private WebDriver driver;

@BeforeMethod

public void beforeMethod()

{

System.setProperty("webdriver.gecko.driver","C:\\Users\\Sivachaitanya\\Downloads\\Compressed\\geckodriver-v0.13.0-win64\\geckodriver.exe");

driver = new FirefoxDriver();

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

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

driver.get("http://test1.absofttrainings.com");

}

@Test

public void testApplyingSortOrder()

{

HomePage homePage=new HomePage(driver);

String sortOrder=homePage.clickShopLink().setSortOrder("Sort by popularity").getSortOrder();

Assert.assertTrue(sortOrder.equals("Sort by popularity"),"Sort order is not applied properly");

}

@AfterMethod

public void afterMethod()

{

//driver.quit();

}

}