package StoreObjects;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class BasePage {

public static WebDriver driver;

public String browser = "chrome";

public BasePage() {

if (driver == null) {

if (browser.equals("chrome")) {

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\SARAN\\Desktop\\Chrome driver\\chromedriver.exe");

driver = new ChromeDriver();

} /\*else if (browser.equals("firefox")) {

System.setProperty("webdriver.gecko.driver", "Firefox Driver Server Path");

driver = new FirefoxDriver();

}\*/

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

driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);

driver.get("http://automationpractice.com/index.php");

}

}

public boolean elementFound(WebElement element) {

boolean res = false;

try {

res = element.isDisplayed();

} catch (Exception e) {

e.printStackTrace();

}

return res;

}

public void setText(WebElement element, String name) {

if (name != null) {

element.click();

element.clear();

element.sendKeys(name);

}

}

public String getTitlte() {

return driver.getTitle();

}

public void getintoframe(WebElement fram) {

driver.switchTo().frame(fram);

}

public void getintodefault() {

driver.switchTo().defaultContent();

}

Dresses Objects

package StoreObjects;

import java.util.List;

import java.util.Random;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.FindBys;

import org.openqa.selenium.support.PageFactory;

public class DressesObjects extends BasePage {

public DressesObjects() {

PageFactory.initElements(driver, this);

}

@FindBy(xpath = "\*//div/ul/li[2]//a[@title='Dresses']")

private WebElement dressestab;

public WebElement getdressestab() {

return dressestab;

}

public void clickdresses() {

dressestab.click();

}

@FindBy(xpath = "\*//a[@title='Women']")

private WebElement womenstab;

public WebElement getWomenstab() {

return womenstab;

}

public void clickwomens() {

womenstab.click();

}

@FindBy(xpath = "\*//ul/li[3]/a[@title='T-shirts']")

private WebElement Tshirtstab;

public WebElement getTshirtstab() {

return Tshirtstab;

}

public void clickTshirts() {

Tshirtstab.click();

}

@FindBy(xpath = "//div/ul/li/a[contains(@title,'You are looking for a dress')]")

private WebElement casualdresses;

public WebElement getcasualdresses() {

return casualdresses;

}

public void clickcasualdresses() {

casualdresses.click();

}

@FindBy(xpath = "//label/a[contains(@href,'category#size-s')]")

private WebElement small;

public WebElement getSmall() {

return small;

}

@FindBy(xpath = "//label/a[contains(@href,'category#size-m')]")

private WebElement medium;

public WebElement getMedium() {

return medium;

}

@FindBy(xpath = "//label/a[contains(@href,'category#size-l')]")

private WebElement large;

public WebElement getLarge() {

return large;

}

@FindBy(xpath = "//span[@class='heading-counter']")

private WebElement headcount;

public int getHeadcount() {

String text = headcount.getText();

String arr[] = text.split(" ");

String textnum = arr[2];

int totalcount = Integer.parseInt(textnum);

return totalcount;

//return Integer.parseInt(headcount.getText().split("")[2]);

}

@FindBys(@FindBy(xpath = "//ul[@class='product\_list grid row']/li"))

private List<WebElement> productcount;

public int getProductcount() {

return productcount.size();

}

@FindBy(id="newsletter-input")

private WebElement newsletter;

@FindBy(name = "submitNewsletter")

private WebElement submitnews;

public WebElement getSubscriptiontab() {

return newsletter;

}

public WebElement sendSubscription() {

return submitnews;

}

public void getsubscription() {

Random numb = new Random();

int number = numb.nextInt(50000);

setText(newsletter, "automation" + number + "@gmail.com");

System.out.println(number);

submitnews.click();

}

@FindBy(xpath = "//p[@class='alert alert-success']")

private WebElement subscriptionmessage;

public WebElement getsubscribemessage() {

return subscriptionmessage;

}

public String getsubscriptionmessage() {

return subscriptionmessage.getText();

}

@FindBy(xpath = "\*//img[@title='Faded Short Sleeve T-shirts']")

private WebElement product;

public WebElement getProduct() {

return product;

}

public void clickproduct() {

product.click();

}

@FindBy(xpath = "\*//button[contains(@onclick,'socialsharing\_twitter\_click')]")

private WebElement tweet;

public WebElement getTweet() {

return tweet;

}

@FindBy(xpath = "\*//button[contains(@onclick,'socialsharing\_facebook\_click();')]")

private WebElement facebook;

public WebElement getFacebook() {

return facebook;

}

@FindBy(id = "short\_description\_content")

private WebElement productdescription;

public WebElement getProductDes() {

return productdescription;

}

@FindBy(xpath = "\*//button[@class='exclusive']")

private WebElement addtocart;

public WebElement addproductTocart() {

return addtocart;

}

public void clickaddtocart() {

addtocart.click();

}

@FindBy(xpath = "\*//i[@class='icon-ok']")

private WebElement successfulmessage;

public WebElement getsuccessmessage() {

return successfulmessage;

}

@FindBy(xpath = "\*//iframe[contains(@id,'fancybox-frame')]")

private WebElement productframe;

public WebElement getproductframe() {

return productframe;

}

public void gotoFrame() {

getintoframe(productframe);

}

}

Dresses Test Page

package StoreTestScripts;

import StoreObjects.BasePage;

import StoreObjects.DressesObjects;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.testng.Assert;

import org.testng.annotations.Test;

public class DressesTestScripts extends BasePage {

DressesObjects dresses;

BasePage bp;

public DressesTestScripts() {

dresses = new DressesObjects();

bp = new BasePage();

}

@Test(enabled = true)

public void verifytabs() {

Assert.assertTrue(bp.elementFound(dresses.getdressestab()));

Assert.assertTrue(bp.elementFound(dresses.getWomenstab()));

Assert.assertTrue(bp.elementFound(dresses.getTshirtstab()));

}

@Test(enabled = true)

public void verifydressestab() {

dresses.clickdresses();

Assert.assertTrue(bp.getTitlte().equals("Dresses - My Store"));

}

@Test(enabled = false)

public void verifywomenstab() {

dresses.clickwomens();

Assert.assertTrue(bp.getTitlte().equals("Women - My Store"));

}

@Test(enabled = false)

public void verifyTshirtstab() {

dresses.clickTshirts();

Assert.assertTrue(bp.getTitlte().equals("T-shirts - My Store"));

}

@Test(enabled = false)

public void verifycasualdresses() {

dresses.clickdresses();

dresses.clickcasualdresses();

Assert.assertTrue(bp.getTitlte().equals("Casual Dresses - My Store"));

}

@Test(enabled = false)

public void verifysize() {

Assert.assertTrue(bp.elementFound(dresses.getSmall()));

Assert.assertTrue(bp.elementFound(dresses.getMedium()));

Assert.assertTrue(bp.elementFound(dresses.getLarge()));

}

@Test(enabled = false)

public void verifyproductcount() {

int totalnum = dresses.getHeadcount();

int prodcount = dresses.getProductcount();

Assert.assertTrue(totalnum == prodcount);

}

@Test(enabled = false)

public void verifysubscription() {

dresses.getsubscription();

Assert.assertTrue(dresses.getsubscriptionmessage().contains("successfully subscribed"));

}

@Test(enabled = false)

public void verifysociallink() {

dresses.clickwomens();

dresses.clickproduct();

dresses.gotoFrame();

Assert.assertTrue(bp.elementFound(dresses.getProduct()));

Assert.assertTrue(bp.elementFound(dresses.getFacebook()));

Assert.assertTrue(bp.elementFound(dresses.getTweet()));

Assert.assertTrue(bp.elementFound(dresses.getProductDes()));

}

@Test(enabled = false)

public void verifyaddtocart() {

dresses.clickwomens();

dresses.clickproduct();

dresses.gotoFrame();

dresses.clickaddtocart();

Assert.assertTrue(bp.elementFound(dresses.getsuccessmessage()));

}

}