**AdressessPage:**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class AddressesPage {

@FindBy(xpath = "//\*[@id='center\_column']/form/p/button/span")

private WebElement ProceedToCheckoutBtn;

WebDriver driver = null;

public AddressesPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public ShippingPage clickOnProceedBtn(){

ProceedToCheckoutBtn.click();

return new ShippingPage(driver);

}

}

**AuthenticationPage**

package com.project.pages;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class AuthenticationPage {

@FindBy(xpath = "//\*[@id='email\_create']")

private WebElement EmailAddressTextBox;

@FindBy(xpath = "//\*[@id='SubmitCreate']/span")

private WebElement CreateAnAccountBtn;

@FindBy(xpath = "//\*[@id='email']")

private WebElement SignInEmail;

@FindBy(xpath = "//\*[@id='passwd']")

private WebElement SignInPassword;

@FindBy(xpath = "//\*[@id='SubmitLogin']/span")

private WebElement SignInBtn;

@FindBy(xpath = "//\*[@id='login\_form']/div/p[1]/a")

private WebElement ForgotPasswordLink;

@FindBy(xpath = "//\*[@id='center\_column']/div[1]")

public WebElement ErrorBox;

@FindBy(xpath = "//\*[@id='order\_step']/li[1]/a")

private WebElement ProgressBar;

@FindBy(xpath = "//\*[@id='header\_logo']/a/img")

private WebElement HomeLogo;

WebDriver driver = null;

public AuthenticationPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public void enterEmailAddress(String email){

EmailAddressTextBox.clear();

EmailAddressTextBox.sendKeys(email);

}

public CreateAnAccountPage clickOnCreateAnAccountBtn(){

CreateAnAccountBtn.click();

return new CreateAnAccountPage(driver);

}

public void SignInEmail(String email){

SignInEmail.clear();

SignInEmail.sendKeys(email);

}

public void SignInPassword(String password){

SignInPassword.clear();

SignInPassword.sendKeys(password);

}

public MyAccountPage clickSignInBtn(){

SignInBtn.click();

return new MyAccountPage(driver);

}

public IndexPage clickOnHomeLogo(){

HomeLogo.click();

return new IndexPage(driver);

}

public Boolean ErrorBoxContains(String error){

return ErrorBox.getText().contains(error);

}

public Boolean titleContains(String message){

return driver.getTitle().equalsIgnoreCase(message);

}

}

**CartSummaryPage**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class CartSummaryPage {

@FindBy(xpath = "//\*[@id='center\_column']/p[2]/a[1]/span")

private WebElement ProceedToCheckoutBtn;

WebDriver driver = null;

public CartSummaryPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public AddressesPage clickOnProceedBtnSignedIn(){

ProceedToCheckoutBtn.click();

return new AddressesPage(driver);

}

}

**CreateAnAccountPage:**

package com.project.pages;

import java.util.List;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

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

public class CreateAnAccountPage {

@FindBy(xpath="//\*[@id='id\_gender1']")

private WebElement TitleMr;

@FindBy(xpath = "//\*[@id='id\_gender2']")

private WebElement TitleMrs;

@FindBy(id = "customer\_firstname")

private WebElement FirstNameTextBox;

@FindBy(id = "customer\_lastname")

private WebElement LastNameTextBox;

@FindBy(id = "email")

private WebElement EmailTextBox;

@FindBy(id = "passwd")

private WebElement PasswordTextBox;

@FindBy(xpath = "//\*[@id='days']")

private WebElement DOB\_date;

@FindBy(xpath = "//\*[@id='months']")

private WebElement DOB\_month;

@FindBy(xpath = "//\*[@id='years']")

private WebElement DOB\_year;

@FindBy(xpath = "//\*[@id='newsletter']")

private WebElement NewsLetterCheckbox;

@FindBy(xpath = "//\*[@id='optin']")

private WebElement SpecialOffersCheckbox;

@FindBy(xpath = "//\*[@id='address1']")

private WebElement AddressTextBox;

@FindBy(xpath = "//\*[@id='address2']")

private WebElement AddressLn2TextBox;

@FindBy(xpath = "//\*[@id='city']")

private WebElement CityTextBox;

@FindBy(xpath = "//\*[@id='id\_state']")

private WebElement StateDropDown;

@FindBy(xpath = "//\*[@id='postcode']")

private WebElement ZipCodeTextBox;

@FindBy(xpath = "//\*[@id='id\_country']")

private WebElement CountryDropDown;

@FindBy(xpath = "//\*[@id='other']")

private WebElement AdditionalInfoTextBox;

@FindBy(xpath = "//\*[@id='phone']")

private WebElement HomePhoneTextBox;

@FindBy(xpath = "//\*[@id='phone\_mobile']")

private WebElement MobileTextBox;

@FindBy(xpath = "//\*[@id='alias']")

private WebElement AliasAddressTextBox;

@FindBy(xpath = "//\*[@id='submitAccount']/span")

private WebElement RegisterBtn;

@FindBy(xpath = "//\*[@id='center\_column']/div")

private WebElement ErrorBox;

WebDriver driver = null;

public CreateAnAccountPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public void selectTitle(String title) throws InterruptedException{

if(title.equals("Mr.")){

TitleMr.click();

}

else

TitleMrs.click();

// for(WebElement Title : TitleList){

// System.out.println("Title is " + Title);

// if(Title.getAttribute("value").trim().equals(title)){

// if(!Title.isSelected()){

// Title.click();

// break;

// }

// }

//

// }

}

public void setFirstName(String FirstName) {

FirstNameTextBox.clear();

FirstNameTextBox.sendKeys(FirstName);

}

public void setLastName(String LastName) {

LastNameTextBox.clear();

LastNameTextBox.sendKeys(LastName);

}

public void setEmailTextBox(String email) {

EmailTextBox.clear();

EmailTextBox.sendKeys(email);

}

public void setPassword(String Password) {

PasswordTextBox.clear();

PasswordTextBox.sendKeys(Password);

}

public void setDOB\_date(String date){

Select dateDropDown = new Select(DOB\_date);

List<WebElement> j = dateDropDown.getOptions();

for(WebElement i : j){

if(i.getText().trim().equalsIgnoreCase(date))

i.click();

// dateDropDown.selectByVisibleText(i.getText());

}

}

public void setDOB\_month(String month){

Select monthDropDown = new Select(DOB\_month);

List<WebElement> j = monthDropDown.getOptions();

for(WebElement i : j){

if(i.getText().trim().equalsIgnoreCase(month))

i.click();

}

}

public void setDOB\_year(String year){

Select yearDropDown = new Select(DOB\_year);

List<WebElement> j = yearDropDown.getOptions();

for(WebElement i : j){

if(i.getText().trim().equalsIgnoreCase(year))

i.click();

}

}

public void setNewsLetterCheckbox() {

if(!NewsLetterCheckbox.isSelected())

NewsLetterCheckbox.click();

}

public void setSpecialOffersCheckbox() {

if(!SpecialOffersCheckbox.isSelected())

SpecialOffersCheckbox.click();

}

public void setAddress(String address) {

AddressTextBox.clear();

AddressTextBox.sendKeys(address);

}

public void setAddressLn2(String addressLn2) {

AddressLn2TextBox.clear();

AddressLn2TextBox.sendKeys(addressLn2);

}

public void setCity(String city) {

CityTextBox.clear();

CityTextBox.sendKeys(city);

}

public void setState(String state) {

Select stDropDown = new Select(StateDropDown);

List<WebElement> j = stDropDown.getOptions();

for(WebElement i : j){

if(i.getText().trim().equalsIgnoreCase(state))

i.click();

}

}

public void setZipCode(String zipCode) {

ZipCodeTextBox.clear();

ZipCodeTextBox.sendKeys(zipCode);

}

public void setCountry(String country) {

Select ctryDropDown = new Select(CountryDropDown);

List<WebElement> j = ctryDropDown.getOptions();

for(WebElement i : j){

if(i.getText().trim().equalsIgnoreCase(country))

i.click();

}

}

public void setMobile(String mobile) {

MobileTextBox.clear();

MobileTextBox.sendKeys(mobile);

}

public MyAccountPage clickRegisterBtn() {

RegisterBtn.click();

return new MyAccountPage(driver);

}

public boolean ErrorBoxContains(String error){

return ErrorBox.getText().contains(error);

}

}

**FacebookPage:**

package com.project.pages;

import java.util.Iterator;

import java.util.Set;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class FacebookPage {

@FindBy(xpath = "//\*[@id='email']")

private WebElement EmailTextBox;

@FindBy(xpath = "//\*[@id='pass']")

private WebElement PasswordTextBox;

WebDriver driver = null;

public FacebookPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public void setEmail(String email){

EmailTextBox.clear();

EmailTextBox.sendKeys(email);

}

public void setPassword(String password){

PasswordTextBox.clear();

PasswordTextBox.sendKeys(password);

}

public String getEmail(){

return EmailTextBox.getAttribute("value");

}

public String getPassword(){

return PasswordTextBox.getAttribute("value");

}

// public ProductPage switchToProductPage(){

//

// Set<String> s = driver.getWindowHandles();

// Iterator<String> i1 = s.iterator();

// String ChildWindow = null;

// while(i1.hasNext()){

// ChildWindow = i1.next();

// }

//

// System.out.println("current window handle : "+driver.getWindowHandle());

// driver.switchTo().window(ChildWindow);

//

// return new ProductPage(driver);

// }

public ProductPage closeWindow(){

driver.close();

return new ProductPage(driver);

}

}

**GooglePlusPage:**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class GooglePlusPage {

@FindBy(xpath = "//\*[@id='identifierId']")

private WebElement EmailTextBox;

WebDriver driver = null;

public GooglePlusPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public void setEmail(String email){

EmailTextBox.clear();

EmailTextBox.sendKeys(email);

}

public String getEmail(){

return EmailTextBox.getAttribute("value");

}

}

**IndexPage:**

package com.project.pages;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class IndexPage {

Actions actions = null;

@FindBy(xpath = "//\*[@id='header']/div[2]/div/div/nav/div[1]/a")

private WebElement SignIn;

@FindBy(xpath = "//\*[@id='search\_query\_top']")

private WebElement SearchTextBox;

@FindBy(xpath = "//\*[@id='searchbox']/button")

private WebElement SearchIcon;

@FindBy(xpath = "//\*[@id='homefeatured']/li[1]/div/div[2]/h5/a")

private WebElement FeaturedItem;

@FindBy(xpath = "//\*[@id='block\_top\_menu']/ul/li[2]/a")

private WebElement DressesLink;

@FindBy(xpath = "//\*[@id='center\_column']/ul/li[1]/div/div[2]/div[2]/a[1]/span")

private WebElement AddToCartBtn;

@FindBy(xpath = "//\*[@id='subcategories']/ul/li[1]/div[1]/a/img")

private WebElement CasualDressesCat;

@FindBy(xpath = "//\*[@id='subcategories']/ul/li[2]/div[1]/a/img")

private WebElement EveningDressesCat;

@FindBy(xpath = "//\*[@id='subcategories']/ul/li[3]/div[1]/a/img")

private WebElement SummerDressesCat;

@FindBy(xpath = "//\*[@id='layer\_cart']/div[1]/div[2]/div[4]/span/span")

private WebElement ContinueShoppingBtn;

@FindBy(xpath = "//\*[@id='columns']/div[1]/a[3]")

private WebElement DressesNavLink;

@FindBy(xpath = "//\*[@id='center\_column']/ul/li[1]/div/div[1]/div/a[1]/img")

private WebElement ToBeHoveredForCartBtn;

@FindBy(xpath = "//\*[@id='layer\_cart']/div[1]/div[2]/div[4]/a/span")

private WebElement ProceedToCheckoutBtn;

WebDriver driver;

public IndexPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public AuthenticationPage clickSignIn(){

SignIn.click();

return new AuthenticationPage(driver);

}

public void setSearchTextBox(String searchQuery){

SearchTextBox.clear();

SearchTextBox.sendKeys(searchQuery);

}

public SearchedPage clickOnSearchBtn(){

SearchIcon.click();

return new SearchedPage(driver);

}

public ProductPage clickOnFeaturedItem(){

FeaturedItem.click();

return new ProductPage(driver);

}

public void clickOnDresses(){

DressesLink.click();

}

public void clickOnDressesNav(){

DressesNavLink.click();

}

public void clickOnCasualDressesLink(){

CasualDressesCat.click();

}

public void clickOnEveningDressesLink(){

EveningDressesCat.click();

}

public void clickOnSummerDressesLink(){

SummerDressesCat.click();

}

public void AddCartBtn(){

AddToCartBtn.click();

}

public void mouseHover() throws InterruptedException{

actions = new Actions(driver);

Thread.sleep(2000);

JavascriptExecutor js = (JavascriptExecutor) driver;

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

actions.moveToElement(ToBeHoveredForCartBtn).build().perform();

}

public void clickOnContinueShoppingBtn(){

ContinueShoppingBtn.click();

}

public CartSummaryPage clickOnProceedBtn(){

ProceedToCheckoutBtn.click();

return new CartSummaryPage(driver);

}

}

**MyAccountPage:**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class MyAccountPage {

WebDriver driver;

@FindBy(xpath = "//\*[@id='header']/div[2]/div/div/nav/div[2]/a")

private WebElement Signout;

@FindBy(xpath = "//\*[@id='center\_column']/p")

private WebElement AccountInfo;

@FindBy(xpath = "//\*[@id='center\_column']/ul/li/a/span")

private WebElement HomeBtn;

public MyAccountPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public Boolean isSignOutDisplayed(){

return Signout.isDisplayed();

}

public AuthenticationPage clickOnSigOut(){

Signout.click();

return new AuthenticationPage(driver);

}

public Boolean AccountInfoContains(String message){

return AccountInfo.getText().contains(message);

}

public void clickOnHomeBtn(){

HomeBtn.click();

}

public Boolean titleContains(String message){

return driver.getTitle().equalsIgnoreCase(message);

}

}

**OrderConfirmationPage:**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class OrderConfirmationPage {

@FindBy(xpath = "//\*[@id='center\_column']/p[1]")

private WebElement ConfirmationMessage;

WebDriver driver = null;

public OrderConfirmationPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public Boolean ConfirmationMessageContains(String message){

return ConfirmationMessage.getText().toUpperCase().contains(message.toUpperCase());

}

}

**OrderSummaryPage**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class OrderSummaryPage {

@FindBy(xpath = "//\*[@id='cart\_navigation']/button/span")

private WebElement IConfirmBtn;

WebDriver driver = null;

public OrderSummaryPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public OrderConfirmationPage clickOnIConfirmBtn(){

IConfirmBtn.click();

return new OrderConfirmationPage(driver);

}

}

**PaymentPage:**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class PaymentPage {

@FindBy(xpath = "//\*[@id='HOOK\_PAYMENT']/div[1]/div/p/a")

private WebElement PayByWire;

@FindBy(xpath = "//\*[@id='HOOK\_PAYMENT']/div[2]/div/p/a")

private WebElement PayByCheck;

WebDriver driver = null;

public PaymentPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public OrderSummaryPage clickOnPayByWire(){

PayByWire.click();

return new OrderSummaryPage(driver);

}

public OrderSummaryPage clickOnPayByCheck(){

PayByCheck.click();

return new OrderSummaryPage(driver);

}

}

**PinterestPage**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class PinterestPage {

@FindBy(xpath = "//\*[@id='\_\_PWS\_ROOT\_\_']/div[1]/div/div/div[1]/div[1]/div[1]/div[2]/div[2]/button/div")

private WebElement LogInBtn;

@FindBy(xpath = "//\*[@id='email']")

private WebElement EmailTextBox;

@FindBy(xpath = "//\*[@id='password']")

private WebElement PasswordTextBox;

WebDriver driver = null;

public PinterestPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public void clickLoginBtn(){

LogInBtn.click();

}

public void setEmail(String email){

EmailTextBox.clear();

EmailTextBox.sendKeys(email);

}

public void setPassword(String password){

PasswordTextBox.clear();

PasswordTextBox.sendKeys(password);

}

public String getEmail(){

return EmailTextBox.getAttribute("value");

}

public String getPassword(){

return PasswordTextBox.getAttribute("value");

}

}

**ProductPage:**

package com.project.pages;

import java.util.Iterator;

import java.util.List;

import java.util.Set;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class ProductPage {

WebDriver driver = null;

@FindBy(xpath = "//\*[@id='product\_comments\_block\_extra']/ul/li/a")

private WebElement WriteAReviewLink;

@FindBy(className = "star")

private List<WebElement> QualityRating;

@FindBy(xpath = "//\*[@id='comment\_title']")

private WebElement ReviewTitleTextBox;

@FindBy(xpath = "//\*[@id='content']")

private WebElement ReviewCommentTextBox;

@FindBy(xpath = "//\*[@id='submitNewMessage']/span")

private WebElement SendBtn;

@FindBy(xpath = "//\*[@id='new\_comment\_form\_footer']/p[2]/a")

private WebElement CancelLink;

@FindBy(xpath = "//\*[@id='product']/div[2]/div/div/div/p[1]")

private WebElement PopupMessagePostReviewSubmit;

@FindBy(xpath = "//\*[@id='center\_column']/div/div/div[3]/p[7]/button[1]")

private WebElement TweetBtn;

@FindBy(xpath = "//\*[@id='center\_column']/div/div/div[3]/p[7]/button[2]")

private WebElement FacebookShareBtn;

@FindBy(xpath = "//\*[@id='center\_column']/div/div/div[3]/p[7]/button[3]")

private WebElement GooglePlusShareBtn;

@FindBy(xpath = "//\*[@id='center\_column']/div/div/div[3]/p[7]/button[4]")

private WebElement PinterestBtn;

public ProductPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public void clikOnReviewLink(){

WriteAReviewLink.click();

}

public void setReviewTitle(String title){

ReviewTitleTextBox.clear();

ReviewTitleTextBox.sendKeys(title);

}

public void setReviewComment(String comment){

ReviewCommentTextBox.clear();

ReviewCommentTextBox.sendKeys(comment);

}

public void qualityRating(String rating){

for(WebElement star : QualityRating){

if(star.getText().trim().equalsIgnoreCase(rating))

star.click();

}

}

public void clickSendBtn(){

SendBtn.click();

}

public Boolean PopupContains(String message){

return PopupMessagePostReviewSubmit.getText().toLowerCase().contains(message.toLowerCase());

}

public void clickTweetBtn(){

TweetBtn.click();

}

public void clickFShareBtn(){

FacebookShareBtn.click();

}

public void clickGooglePlusBtn(){

GooglePlusShareBtn.click();

}

public void clickPinterestBtn(){

PinterestBtn.click();

// return new PinterestPage(driver);

}

static String ParentWindow = null;

public PinterestPage switchToPinterestPage(){

Set<String> s = driver.getWindowHandles();

Iterator<String> i1 = s.iterator();

ParentWindow = driver.getWindowHandle();

String ChildWindow = null;

while(i1.hasNext()){

ChildWindow = i1.next();

if(!ParentWindow.equals(ChildWindow))

{

// System.out.println("current window handle : "+driver.getWindowHandle());

driver.switchTo().window(ChildWindow);

break;

}

}

return new PinterestPage(driver);

}

public GooglePlusPage switchToGooglePlusPage(){

Set<String> s = driver.getWindowHandles();

Iterator<String> i1 = s.iterator();

ParentWindow = driver.getWindowHandle();

String ChildWindow = null;

while(i1.hasNext()){

ChildWindow = i1.next();

if(!ParentWindow.equals(ChildWindow))

{

// System.out.println("current window handle : "+driver.getWindowHandle());

driver.switchTo().window(ChildWindow);

break;

}

}

return new GooglePlusPage(driver);

}

public FacebookPage switchToFacebookPage(){

Set<String> s = driver.getWindowHandles();

Iterator<String> i1 = s.iterator();

ParentWindow = driver.getWindowHandle();

// System.out.println("Parent : "+ParentWindow);

String ChildWindow = null;

while(i1.hasNext()){

ChildWindow = i1.next();

if(!ParentWindow.equals(ChildWindow))

{

// System.out.println("current window handle : "+driver.getWindowHandle());

driver.switchTo().window(ChildWindow);

break;

}

}

return new FacebookPage(driver);

}

public void getHandleBack(){

System.out.println("printing in get Handle function :"+ParentWindow);

driver.switchTo().window(ParentWindow);

ParentWindow = null;

// return new ProductPage(driver);

}

public Boolean verifyUrl(String str){

return driver.getCurrentUrl().contains(str);

}

}

**SearchedPage:**

package com.project.pages;

import java.util.List;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class SearchedPage {

@FindBy(xpath = "//\*[@id='center\_column']/h1/span[2]")

private WebElement NumberOfItems;

// @FindBy(className = "product-image-container")

@FindBy(xpath="//\*[@id='center\_column']/ul/li")

private List<WebElement> ItemContainerByClass;

@FindBy(xpath = "//\*[@id='manufacturers\_block\_left']/div/ul/li/a")

private WebElement ManufacturerLink;

@FindBy(xpath = "////\*[@id='center\_column']/p")

private WebElement NoResultsMessage;

@FindBy(xpath = "//\*[@id='center\_column']/h1")

private WebElement ProductsByManufacturerHeader;

WebDriver driver =null;

public SearchedPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public int getNumberOfExpectedItems(){

try{

if(NoResultsMessage.isDisplayed())

return 0;

}

catch(Exception e)

{

// e.printStackTrace();

}

return Integer.parseInt(NumberOfItems.getText().replaceAll("[^0-9]", ""));

}

public int countNumberOfVisibleItems(){

System.out.println("list : "+ItemContainerByClass);

// for(WebElement i : ItemContainerByClass){

// System.out.println(i.getText());

// }

return ItemContainerByClass.size();

}

public String clickOnManufacturerLink(){

ManufacturerLink.click();

return ManufacturerLink.getText();

}

public Boolean HeaderContains(String header){

return ProductsByManufacturerHeader.getText().toUpperCase().contains(header.toUpperCase());

}

}

**SippingPage:**

package com.project.pages;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class ShippingPage {

@FindBy(xpath = "//\*[@id='form']/p/button/span")

private WebElement ProceedToCheckoutBtn;

@FindBy(xpath = "//\*[@id='cgv']")

private WebElement TOSCheckBox;

WebDriver driver = null;

public ShippingPage(WebDriver driver){

this.driver = driver;

PageFactory.initElements(driver, this);

}

public PaymentPage clickOnProceedBtn(){

ProceedToCheckoutBtn.click();

return new PaymentPage(driver);

}

public void clickOnTncCheckBox(){

TOSCheckBox.click();

}

}

**Account\_1**

package com.project.testcases;

import java.io.IOException;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.AuthenticationPage;

import com.project.pages.CreateAnAccountPage;

import com.project.pages.IndexPage;

import com.project.pages.MyAccountPage;

import com.project.utils.TempMailGetter;

public class Account\_1 extends BaseClass {

@Test(dataProvider = "TestData")

public void RegisterAccount(String... data) throws InterruptedException, IOException

{

IndexPage ip = new IndexPage(driver);

AuthenticationPage ap = ip.clickSignIn();

ip = null;

ap.enterEmailAddress(TempMailGetter.getTempMail());

// ap.enterEmailAddress("wabayex3284443@aieen.com");

CreateAnAccountPage caap = ap.clickOnCreateAnAccountBtn();

caap.selectTitle(data[0]);

caap.setFirstName(data[1]);

caap.setLastName(data[2]);

caap.setPassword(data[3]);

caap.setDOB\_date(data[4]);

caap.setDOB\_month(data[5]);

caap.setDOB\_year(data[6]);

// JavascriptExecutor js = ((JavascriptExecutor) driver);

// js.executeScript("window.scrollTo(0, document.body.scrollHeight/2)");

caap.setAddress(data[7]);

caap.setCity(data[8]);

caap.setState(data[9]);

caap.setZipCode(data[10]);

caap.setCountry(data[11]);

caap.setMobile(data[12]);

MyAccountPage mp = caap.clickRegisterBtn();

caap = null;

logger = extent.createTest("Registering an account");

Assert.assertTrue(mp.AccountInfoContains(data[13]));

}

}

**Account\_2**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.AuthenticationPage;

import com.project.pages.CreateAnAccountPage;

import com.project.pages.IndexPage;

public class Account\_2 extends BaseClass{

@Test(dataProvider = "TestData")

public void RegisterAccount(String... data) {

IndexPage ip = new IndexPage(driver);

AuthenticationPage ap = ip.clickSignIn();

ip = null;

// ap.enterEmailAddress(TempMailGetter.getTempMail());

ap.enterEmailAddress(data[0]);

CreateAnAccountPage caap = ap.clickOnCreateAnAccountBtn();

try {

caap.selectTitle(data[1]);

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

caap.setFirstName(data[2]);

caap.setLastName(data[3]);

caap.setPassword(data[4]);

caap.setDOB\_date(data[5]);

caap.setDOB\_month(data[6]);

caap.setDOB\_year(data[7]);

caap.setAddress(data[8]);

caap.setCity(data[9]);

caap.setState(data[10]);

caap.setZipCode(data[11]);

caap.setCountry(data[12]);

caap.setMobile(data[13]);

caap.clickRegisterBtn();

logger = extent.createTest("Registering account with less than 5 char");

Assert.assertTrue(caap.ErrorBoxContains(data[14]));

}

}

**BaseClass**

package com.project.testcases;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Iterator;

import java.util.List;

import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.testng.ITestNGMethod;

import org.testng.ITestResult;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.DataProvider;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.markuputils.ExtentColor;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.Theme;

import com.project.utils.PropertyReader;

import au.com.bytecode.opencsv.CSVReader;

public class BaseClass //extends TestListenerAdapter

{

WebDriver driver = null;

public static ExtentHtmlReporter htmlReporter = null;

public static ExtentReports extent = null;

public static ExtentTest logger = null;

@BeforeSuite

public void startReport(){

htmlReporter = new ExtentHtmlReporter(System.getProperty("user.dir") +"/test-output/STMExtentReport.html");

extent = new ExtentReports();

extent.attachReporter(htmlReporter);

extent.setSystemInfo("Host Name", "SoftwareTesting");

extent.setSystemInfo("Environment", "Automation Testing");

extent.setSystemInfo("User Name", "Alok");

htmlReporter.config().setDocumentTitle("Automation Report");

htmlReporter.config().setReportName("Selenium Regression Test Suite");

htmlReporter.config().setTheme(Theme.STANDARD);

}

@BeforeMethod

public void initiateDriver() throws IOException{

String browser = PropertyReader.ReadProperty("browser");

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

{

System.setProperty("webdriver.chrome.driver", PropertyReader.ReadProperty("chromedriverpath"));

driver = new ChromeDriver();

}else if(browser.trim().equals("firefox"))

{

System.setProperty("webdriver.gecko.driver", PropertyReader.ReadProperty("firefoxdriverpath"));

driver = new FirefoxDriver();

}else if(browser.trim().equals("ie"))

{

System.setProperty("webdriver.ie.driver", PropertyReader.ReadProperty("iedriverpath"));

driver = new InternetExplorerDriver();

}

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

driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);

driver.get(PropertyReader.ReadProperty("appurl"));

}

public String getScreenshot(WebDriver driver, String ScreenshotName) throws IOException{

//for that we need to reference another jar files

//one for the extent report and another is for snapshot

//common-io jars - reference it in the project

TakesScreenshot ts = (TakesScreenshot)driver;

File source= ts.getScreenshotAs(OutputType.FILE);

String dateName = new SimpleDateFormat("yyyyMMddhhmmss").format(new Date());

String destination = System.getProperty("user.dir")+"/Screenshots/" + ScreenshotName + dateName + ".png";

//added common-io jars

FileUtils.copyFile(source, new File(destination));

return destination;

}

ITestNGMethod testContext = null;

@DataProvider(name = "TestData")

public String[][] getData(ITestNGMethod testContext) throws IOException {

this.testContext = testContext;

System.out.println("|"+testContext.getRealClass().getSimpleName()+"|");

CSVReader reader = null;

try{

// reader = new CSVReader(new FileReader("./src/com/project/testdata/Signin\_Invalid.csv"));

reader = new CSVReader(new FileReader(PropertyReader.ReadProperty(testContext.getRealClass().getSimpleName())));

}

catch (NullPointerException e) {

e.printStackTrace();

}

// this will load content into list

List<String[]> li=reader.readAll();

// create Iterator reference

Iterator<String[]> iter= li.iterator();

System.out.println(li.size());

String[][] creds = new String[li.size()-1][iter.next().length];

for(int i=0; iter.hasNext(); i++){

creds[i] = iter.next();

}

return creds;

}

@AfterMethod

public void CloseBrowser(ITestResult result) throws IOException {

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

//log the results

logger.log(Status.FAIL, MarkupHelper.createLabel(result.getName() + " - Test Case Failed ", ExtentColor.RED));

logger.log(Status.FAIL, MarkupHelper.createLabel(result.getThrowable() + " - Test Case Failed ", ExtentColor.RED));

logger.fail("Test Case Failed Snapshot is below - " + logger.addScreenCaptureFromPath(getScreenshot(driver,result.getName())));

}

else if(result.getStatus() == ITestResult.SKIP){

logger.log(Status.SKIP, MarkupHelper.createLabel(result.getName() + " - Test Case Skipped ", ExtentColor.ORANGE));

logger.log(Status.SKIP,"Snapshot is below - " + logger.addScreenCaptureFromPath(getScreenshot(driver,result.getName())));

}

else if(result.getStatus() == ITestResult.SUCCESS){

logger.log(Status.PASS, MarkupHelper.createLabel(result.getName() + " - Test Case Passed ", ExtentColor.GREEN));

logger.log(Status.PASS,"Snapshot is below - " + logger.addScreenCaptureFromPath(getScreenshot(driver,result.getName())));

}

// logic for closing driver

driver.quit();

driver=null;

}

@AfterSuite

public void stopTest() {

extent.flush(); //at the end of all tests - it will append the test results to the htmlreport when flushed

}

}

**LogOff**

package com.project.testcases;

import org.testng.annotations.Test;

import org.testng.asserts.SoftAssert;

import com.project.pages.AuthenticationPage;

import com.project.pages.IndexPage;

import com.project.pages.MyAccountPage;

public class LogOff extends BaseClass {

@Test(dataProvider = "TestData")

public void Logoff(String... data){

SoftAssert softAssert = new SoftAssert();

IndexPage ip = new IndexPage(driver);

AuthenticationPage ap = ip.clickSignIn();

ip=null;

ap.SignInEmail(data[0]);

ap.SignInPassword(data[1]);

MyAccountPage mp = ap.clickSignInBtn();

//checking whether logged in by checking page title

softAssert.assertTrue(mp.titleContains(data[2]));

ap = mp.clickOnSigOut();

//checking whether logged out by checking page title

logger = extent.createTest("LogOff test case");

softAssert.assertTrue(ap.titleContains(data[3]));

softAssert.assertAll();

}

}

**Manufactures**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.IndexPage;

import com.project.pages.SearchedPage;

public class Manufacturers extends BaseClass {

// WebDriver driver = null;

@Test(dataProvider = "TestData")

public void SearctTestCase(String... data){

IndexPage ip = new IndexPage(driver);

ip.setSearchTextBox(data[0]); //fashion

SearchedPage sp = ip.clickOnSearchBtn();

String header = sp.clickOnManufacturerLink();

// logger = extent.createTest("Manufacturers");

logger = extent.createTest(testContext.getRealClass().getSimpleName());

Assert.assertTrue(sp.HeaderContains(header));

}

}

**PlacingOrder**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.AddressesPage;

import com.project.pages.AuthenticationPage;

import com.project.pages.CartSummaryPage;

import com.project.pages.IndexPage;

import com.project.pages.OrderConfirmationPage;

import com.project.pages.OrderSummaryPage;

import com.project.pages.PaymentPage;

import com.project.pages.ShippingPage;

public class PlacingOrder extends BaseClass{

@Test(dataProvider = "TestData")

public void PlacingOrder1(String... data) throws InterruptedException {

IndexPage ip = new IndexPage(driver);

AuthenticationPage ap = ip.clickSignIn();

ap.SignInEmail("fibefi9357@adeata.com");

ap.SignInPassword("password");

ap.clickSignInBtn();

ip = ap.clickOnHomeLogo();

ip.clickOnDresses();

ip.clickOnCasualDressesLink();

ip.mouseHover();

ip.AddCartBtn();

ip.clickOnContinueShoppingBtn();

ip.clickOnDressesNav();

ip.clickOnEveningDressesLink();

ip.mouseHover();

ip.AddCartBtn();

ip.clickOnContinueShoppingBtn();

ip.clickOnDressesNav();

ip.clickOnSummerDressesLink();

ip.mouseHover();

ip.AddCartBtn();

CartSummaryPage csp = ip.clickOnProceedBtn();

AddressesPage add = csp.clickOnProceedBtnSignedIn();

ShippingPage ship = add.clickOnProceedBtn();

ship.clickOnTncCheckBox();

PaymentPage pay = ship.clickOnProceedBtn();

OrderSummaryPage osp = pay.clickOnPayByCheck();

OrderConfirmationPage ocp = osp.clickOnIConfirmBtn();

logger = extent.createTest("Placing order1");

Assert.assertTrue(ocp.ConfirmationMessageContains("Your order on My Store is complete"));

}

@Test(dataProvider = "TestData")

public void PlacingOrder2(String... data) throws InterruptedException {

IndexPage ip = new IndexPage(driver);

AuthenticationPage ap = ip.clickSignIn();

ap.SignInEmail(data[0]);

ap.SignInPassword(data[1]);

ap.clickSignInBtn();

ip = ap.clickOnHomeLogo();

ip.clickOnDresses();

ip.clickOnCasualDressesLink();

ip.mouseHover();

ip.AddCartBtn();

ip.clickOnContinueShoppingBtn();

ip.clickOnDressesNav();

ip.clickOnEveningDressesLink();

ip.mouseHover();

ip.AddCartBtn();

ip.clickOnContinueShoppingBtn();

ip.clickOnDressesNav();

ip.clickOnSummerDressesLink();

ip.mouseHover();

ip.AddCartBtn();

CartSummaryPage csp = ip.clickOnProceedBtn();

AddressesPage add = csp.clickOnProceedBtnSignedIn();

ShippingPage ship = add.clickOnProceedBtn();

ship.clickOnTncCheckBox();

PaymentPage pay = ship.clickOnProceedBtn();

OrderSummaryPage osp = pay.clickOnPayByWire();

OrderConfirmationPage ocp = osp.clickOnIConfirmBtn();

logger = extent.createTest("Placing order");

Assert.assertTrue(ocp.ConfirmationMessageContains(data[2]));

}

}

**Review**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.AuthenticationPage;

import com.project.pages.IndexPage;

import com.project.pages.MyAccountPage;

import com.project.pages.ProductPage;

public class Review extends BaseClass{

// WebDriver driver = null;

@Test(dataProvider = "TestData")

public void WriteAReview(String... data){

IndexPage ip = new IndexPage(driver);

AuthenticationPage ap = ip.clickSignIn();

ap.SignInEmail(data[0]);

ap.SignInPassword(data[1]);

MyAccountPage mp = ap.clickSignInBtn();

mp.clickOnHomeBtn();

ProductPage pp = ip.clickOnFeaturedItem();

pp.clikOnReviewLink();

pp.qualityRating(data[2]); // Cancel Rating

pp.setReviewTitle(data[3]);

pp.setReviewComment(data[4]);

pp.clickSendBtn();

logger = extent.createTest("Review");

logger.createNode("Error Pop-up asking for more than 50 char");

Assert.assertTrue(pp.PopupContains(data[5]));

}

}

**Search**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.IndexPage;

import com.project.pages.SearchedPage;

public class Search extends BaseClass {

// WebDriver driver = null;

@Test(dataProvider = "TestData")

public void SearctTestCase(String... data){

IndexPage ip = new IndexPage(driver);

ip.setSearchTextBox(data[0]);

SearchedPage sp = ip.clickOnSearchBtn();

logger = extent.createTest("Search");

// System.out.println("visible items : "+sp.countNumberOfVisibleItems());

// System.out.println("Expected Number of Items : "+ sp.getNumberOfExpectedItems());

Assert.assertEquals(sp.countNumberOfVisibleItems(), sp.getNumberOfExpectedItems());

}

}

**Share\_Product\_Facebook**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.FacebookPage;

import com.project.pages.IndexPage;

import com.project.pages.ProductPage;

public class ShareProduct\_Facebook extends BaseClass{

@Test(dataProvider = "TestData")

public void ShareProductFacebook(String... data) {

IndexPage ip = new IndexPage(driver);

ProductPage pp = ip.clickOnFeaturedItem();

pp.clickFShareBtn();

FacebookPage fp = pp.switchToFacebookPage();

pp = fp.closeWindow();

pp.getHandleBack();

logger = extent.createTest("Share Product Facebook");

Assert.assertTrue(pp.verifyUrl(data[0]));

}

}

**Share\_Product\_googlePlus**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.GooglePlusPage;

import com.project.pages.IndexPage;

import com.project.pages.ProductPage;

public class ShareProduct\_GooglePlus extends BaseClass{

@Test(dataProvider = "TestData")

public void ShareProductGooglePlus(String... data) {

IndexPage ip = new IndexPage(driver);

ProductPage pp = ip.clickOnFeaturedItem();

pp.clickGooglePlusBtn();

GooglePlusPage gpp= pp.switchToGooglePlusPage();

gpp.setEmail(data[0]);

logger = extent.createTest("Share product - G+ ");

Assert.assertEquals(gpp.getEmail(), data[0]);

}

}

Share\_product\_Pinterest

package com.project.testcases;

import org.testng.annotations.Test;

import org.testng.asserts.SoftAssert;

import com.project.pages.IndexPage;

import com.project.pages.PinterestPage;

import com.project.pages.ProductPage;

public class ShareProduct\_Pinterest extends BaseClass{

@Test(dataProvider = "TestData")

public void ShareProductPinIt(String... data) {

SoftAssert softAssert = new SoftAssert();

IndexPage ip = new IndexPage(driver);

ProductPage pp = ip.clickOnFeaturedItem();

pp.clickPinterestBtn();

PinterestPage pin = pp.switchToPinterestPage();

pin.clickLoginBtn();

pin.setEmail(data[0]);

pin.setPassword(data[1]);

// logger = extent.createTest("Share Product - PinIt");

softAssert.assertEquals(pin.getEmail(), data[0]);

softAssert.assertEquals(pin.getPassword(), data[1]);

softAssert.assertAll();

}

}

**Signin\_invalid**

package com.project.testcases;

import org.testng.annotations.Test;

import org.testng.asserts.SoftAssert;

import com.project.pages.AuthenticationPage;

import com.project.pages.IndexPage;

public class Signin\_Invalid extends BaseClass{

// 3 sets of test data

// 1. Invallid email ID - "Invalid email address" should be displayed

// 2. password less than 5 char - "Invalid password" should be displayed

// 3. wrong password - "Authentication failed" should be displayed

@Test(dataProvider = "TestData")

public void SignInInvalid(String... data){

SoftAssert softAssert = new SoftAssert();

IndexPage ip = new IndexPage(driver);

AuthenticationPage ap = ip.clickSignIn();

ap.SignInEmail(data[0]);

ap.SignInPassword(data[1]);

ap.clickSignInBtn();

logger = extent.createTest(testContext.getRealClass().getSimpleName() + " - " + testContext.getCurrentInvocationCount());

logger.createNode("checking colour of error message box");

// System.out.println("Get the css width value of ageTextbox = " + ap.ErrorBox.getCssValue("background-color"));

softAssert.assertEquals(data[2], ap.ErrorBox.getCssValue("background-color"));

logger.createNode("Checking whether the error message is as expected");

softAssert.assertTrue(ap.ErrorBoxContains(data[3])); //Error message

softAssert.assertAll();

}

}

**Signin\_valid**

package com.project.testcases;

import org.testng.Assert;

import org.testng.annotations.Test;

import com.project.pages.AuthenticationPage;

import com.project.pages.IndexPage;

import com.project.pages.MyAccountPage;

public class Signin\_Valid extends BaseClass{

private static int count = 1;

@Test(dataProvider = "TestData")

public void SignIn(String... d){

IndexPage ip = new IndexPage(driver);

ip.clickSignIn();

ip=null;

AuthenticationPage ap = new AuthenticationPage(driver);

// ap.SignInEmail("fibefi9357@adeata.com");

// ap.SignInPassword("password");

ap.SignInEmail(d[0]);

ap.SignInPassword(d[1]);

MyAccountPage mp = ap.clickSignInBtn();

logger = extent.createTest("Signin valid"+ Signin\_Valid.count++);

Assert.assertTrue(mp.isSignOutDisplayed());

}

}

**PropertyReader:**

package com.project.utils;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.Properties;

public class PropertyReader {

public static FileInputStream inputStream= null;

public static Properties prop = null;

public static String ReadProperty(String PropertyName) throws IOException{

String PropertyValue = null;

String ProjectPath = System.getProperty("user.dir");

File file = new File(ProjectPath + "/src/com/project/config/config.properties");

inputStream = new FileInputStream(file);

prop = new Properties();

//for reading a property from the property file, we first need to load the file

prop.load(inputStream);

PropertyValue = prop.getProperty(PropertyName);

return PropertyValue;

}

}

**TempMailGetter:**

package com.project.utils;

//import java.io.FileNotFoundException;

import java.io.IOException;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

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

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

public class TempMailGetter {

public static String getTempMail() throws IOException {

WebDriver driver1 = null;

String ProjectPath = System.getProperty("user.dir");

System.out.println(ProjectPath);

String FilePath = ProjectPath + "/" + "TempMailFile.xlsx";

// File file = new File(FilePath);

// Workbook wb = new XSSFWorkbook();

// FileOutputStream fos = new FileOutputStream(file);

System.setProperty("webdriver.chrome.driver", PropertyReader.ReadProperty("chromedriverpath"));

ChromeOptions options = new ChromeOptions();

// options.addArguments("--headless");

driver1 = new ChromeDriver(options);

driver1.get("https://temp-mail.org/en/api/");

WebDriverWait wait = new WebDriverWait(driver1, 10);

wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//\*[@id='click-to-copy']")));

String str = driver1.findElement(By.xpath("//\*[@id='mail']")).getAttribute("value");

driver1.quit();

return str;

}

}