package com.ecommerce.test;

import java.net.MalformedURLException;

import java.net.URL;

import org.openqa.selenium.By;

import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

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

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import org.testng.asserts.SoftAssert;

public class ParalellTest {

public String baseUrl = "https://www.facebook.com/r.php?locale=en\_GB&display=page";

private WebDriver driver;

SoftAssert softassert = new SoftAssert();

@Test(groups = {"Account Creation"})

public void facebookAccCreation() {

System.out.println("Inside facebookAccCreation");

System.out.println("Thread ID " +Thread.currentThread().getId());

String baseUrl = "https://www.facebook.com/r.php?locale=en\_GB&display=page";

driver.get(baseUrl);

String cssDay = "#day";

WebElement cssDaySelect = driver.findElement(By.cssSelector(cssDay));

softassert.assertNotNull(cssDaySelect);

Select daySelect = new Select(cssDaySelect);

daySelect.selectByVisibleText("11");

WebElement cssGenderRadio = driver.findElement(By.cssSelector("span > span > input[type='radio'][value='2']"));

cssGenderRadio.click();

softassert.assertTrue(cssGenderRadio.isSelected());

System.out.println("Gender is enabled = " + cssGenderRadio.isSelected());

softassert.assertAll(" Day or/and Gender assertion failed");

}

@Test(groups = "Account Creation", dependsOnGroups = { "Launch" },

dataProvider = "googleUserData")

public void googleAccCreation(String fName, String lName) {

System.out.println("Inside googleAccCreation");

System.out.println("Thread ID " +Thread.currentThread().getId());

String baseUrl = "https://accounts.google.com/signup/v2?biz=true&flowEntry=SignUp";

driver.get(baseUrl);

// Let's locate the first name text field by its id.

WebElement firstNameTF = driver.findElement(By.id("firstName"));

firstNameTF.sendKeys(fName) ;//("Myname");

// Let's locate the surname name text field by its name.

WebElement surNameTF = driver.findElement(By.name("lastName"));

surNameTF.sendKeys(lName) ;//("latName");

// The Next button using its class

WebElement nextButton = driver.findElement(By.className("VfPpkd-vQzf8d"));

System.out.println("Text on the button is " + nextButton.getText());

softassert.assertEquals("Next", nextButton.getText());

softassert.assertAll(" Google Next button text did not match");

}

@Test(groups = "Launch")

public void googleWebsiteTitle() {

System.out.println("Inside googleWebsiteTitle");

System.out.println("Thread ID " +Thread.currentThread().getId());

String baseUrl = "https://www.google.com";

driver.get(baseUrl);

System.out.println("Google Title is " + driver.getTitle());

softassert.assertEquals("Google", driver.getTitle());

softassert.assertAll(" Google title did not match");

}

@Test

public void f1() {

System.out.println("Inside f1 in Thread ID " +Thread.currentThread().getId());

}

@Test

public void f2() {

System.out.println("Inside f2 in Thread ID " +Thread.currentThread().getId());

}

@DataProvider(name = "googleUserData")

public Object[][] createData1() {

return new Object[][] {

{"fname1","lname1"},

{"fname2","lname2"}

};

}

@BeforeClass

public void beforeClass() {

//System.setProperty("webdriver.chrome.driver","C:\\Users\\HomeWk\\Downloads\\chromedriver\_win32\\chromedriver.exe");

DesiredCapabilities caps = new DesiredCapabilities();

caps.setPlatform(Platform.WIN10);

caps.setBrowserName("firefox");

try {

driver = new RemoteWebDriver(new URL("http://192.168.99.1:4444"), caps);

} catch (MalformedURLException e) {

e.printStackTrace();

}

//driver = new ChromeDriver();

}

@AfterClass

public void afterClass() {

driver.quit();

}

}