ADP – Shoe Store Test Cases and Automation

Objective of Story 1: I want to be able to view shoes that are being released every month via a link.

Acceptance Criteria:

1. Each month should display a small blurb of each shoe
2. Each month should display an image of each shoe being released
3. Each month should have a suggested price pricing for each shoe

**Test Cases for Story 1**

Test 1 (launchSiteandLogin)

1. Go to http://www.adp.com/newshoereleases/index.php
2. Enter “test1” in the Username field
3. Enter “XXXX” in the Password field
4. Click on the Login button
5. Verify that the title’s text “New Shoe Releases” is present.

Test 2 (displayNewShoeReleases)

1. Click on the “January” link to view new shoe releases
2. Verify that new shoe releases are displayed along with a small blurb of each shoe for this month

Test 3 (checkImageandPrice)

1. Scroll down the page of new shoe releases for the month of January
2. Verify that there is an image and price displayed next to each of the new shoe releases

Test 4 (Logout)

1. Click on the Logout button
2. Verify that the Login button displayed and the user is returned to the main login screen.

Objective of Story 2: I want to be able to submit my email address to be reminded of upcoming shoe releases.

Acceptance Criteria:

1. There should be an area on the website to submit an email address.
2. There should be a confirmation message that reads: “Thanks! We will notify you of our new shoes at this email: *someone@adp.com*” once the email is sent successfully.”

**Test Case for Story 2**

Test 1 (submitEmailandConfirmMessage)

1. Click on the ‘Remind Me’ link at the bottom right hand side of the page
2. Verify that a textbox is visible
3. Enter a valid email address in the textbox
4. Click the “Submit” button
5. Verify that this confirmation message was received: “Thanks! We will notify you of our new shoes at this email: *someone@adp.com* once the email is sent successfully.”

Selenium WebDriver/Java - TestNG Framework

**package**com.newshoereleases.testsuite;

**import static**org.testng.AssertJUnit.assertTrue;

**import static**org.testng.AssertJUnit.assertEquals;

**import**java.util.concurrent.TimeUnit;

**import**org.openqa.selenium.By;

**import**org.openqa.selenium.NoSuchElementException;

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

**import**org.openqa.selenium.firefox.FirefoxDriver;

**import**org.openqa.selenium.chrome.ChromeDriver;

**import**org.openqa.selenium.ie.InternetExplorer.Driver;

**import**org.openqa.selenium.safari.SafariDriver;

**import**org.testng.Assert;

**import**org.testng.ITestContext;

**import**org.testng.annotations.AfterClass;

**import**org.testng.annotations.BeforeClass;

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

**import**org.testng.annotations.BeforeSuite;

**import**org.testng.annotations.AfterSuite;

**public class** { DisplayNewShoeReleases

**private**WebDriver browser;

**private**String baseUrl;

@Test (description="Launches the ADP New Shoe Releases page and Login")

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

    browser.get(baseUrl+"/newshoereleases/index.php

**for**(**int**second = 0; second++) {

**if**(second >= 60) Assert.fail("timeout");

**try**{ **if**(isElementPresent(By.cssSelector("img[alt=\"New Shoe Releases \"]")))

            break; } catch (Exception e) {}

          Thread.sleep(1000);

        }

  browser.findElement(By.id("login\_login\_username")).clear();

  browser.findElement(By.id("login\_login\_username")).sendKeys("user1");

  browser.findElement(By.id("login\_login\_password")).clear();

  browser.findElement(By.id("login\_login\_password")).sendKeys(“\*\*\*\*");

//*password is omitted for security purposes*

  browser.findElement(By.id("login\_submit")).click();

  assertEquals(browser.findElement(By.cssSelector("ul.cr > li > a")).getText(),"New Shoe Releases");

  }

@Test(description="Displays New Shoe Releases")

**public void**displayNewShoeReleases() **throws**

InterruptedException {

    browser.findElement(By.linkText("January")).click();

**for**(**int**second = 0;; second++) {

**if**(second >= 60) Assert.fail("timeout");

**try**{ **if**(isElementPresent(By.id("January ")))

**break**; } **catch**(Exception e) {}

          Thread.sleep(1000);

        }

@Test(description=”Checks Images and Prices”)

Boolean ImagePresent = (Boolean) ((JavascriptExecutor) driver).executeScript("return arguments[0].complete && typeof arguments[0].naturalWidth != \"undefined\" && arguments[0].naturalWidth > 0", ImageFile);

        if (!ImagePresent)

{

System.out.println(“Image not displayed.”);

}

else

{

System.out.println(“Image displayed.”);

}

}

**public void** checkImageandPrice() throws InterrutedException {

WebElement ImageFile = driver.findElement(By.xpath(“//img[contains(@id, ‘Shoe Image’)]””));

Assert.assertEquals(“

driver.findElement(By.xpath("//input[@name='price']")).submit();

String alrt = driver.switchTo().alert().getText();

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

System.out.println(alrt);

}

}

@Test(description="Log out the system")

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

**for**(**int**second = 0;; second++) {

**if**(second >= 60) Assert.fail("timeout");

**try**{ **if**(isElementPresent(By.linkText("Logout")))

**break**; } **catch**(Exception e) {}

        Thread.sleep(1000);

      }

    browser.findElement(By.linkText("Logout")).click();

    assertTrue(isElementPresent(By.id("login\_login\_username")));

  }

  @BeforeClass(alwaysRun = **true**)

**public void**setupBeforeSuite(ITestContext context) {

     baseUrl = " http://www.adp.com/newshoereleases ";

**try**{

     browser=**new**FirefoxDriver();

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

     } **catch**(Exception e) {

**throw new**IllegalStateException("Can't start Web Driver", e);

    }

  }

  @AfterClass

**public void**setupAfterSuite() {

    browser.close();

    browser.quit();

  }

**private boolean**isElementPresent(By by) {

**try**{

          browser.findElement(by);

**return true**;

        } **catch**(NoSuchElementException e) {

**return false**;

        }

      }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TestNG Suite (xml file)

<?xml version=”1.0” encoding=”UTF-8”?>

<suite name=”Suite” parallel=””false”>

<test name=”Test” preserve-order=”true”>

<classes>

<class name=”com.newshoereleases.testsuite.DisplayNewShoes”>

<methods>

<include name=”LaunchSiteAndLogin” />

<include name=”displayNewShoeReleases” />

<include name=” checkImageandPrice” />

<include name=” Logout” />

</methods>

</class>

</classes>

</test> <!-- Test -- >

</suite> <!-- Suite -- >

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NewShoeReleasesMasterLogin.java

Note: Execute the Selenium server from the console or Mac Terminal

**package**com.seleniummaster.differentbrowsers;

**import static**org.junit.Assert.\*;

**import**java.util.Arrays;

**import**java.util.Collection;

**import**org.junit.After;

**import**org.junit.Before;

**import**org.junit.Test;

**import**org.junit.runner.RunWith;

**import**org.junit.runners.Parameterized;

**import**com.thoughtworks.selenium.DefaultSelenium;

**import**com.thoughtworks.selenium.Selenium;

//run with Parameterized class in JUnit

@RunWith(Parameterized.**class**)

**public class**TestSeleniumMasterLogin{

  //define browser

**private**String browser;

**private**Selenium selenium;

  //create a constructor with browser

**public**displayNewShoeReleases (String browser){

**this**.browser=browser;

  }

  //define each browser as a parameter

  @Parameterized.Parameters

**public static**Collection data() {

**return**Arrays.asList(**new**Object[][]{{"\*chrome"},{"\*googlechrome"},{“\*internetexplorer”},{"\*firefox"}});

   }

   @Before

**public void**setUp() **throws**Exception {

   selenium = **new**DefaultSelenium("localhost", 8080, browser, "http://www.seleniummaster.com");

   selenium.start();

   selenium.windowMaximize();

  }

  @Test

**public void**LoginTest() **throws**Exception {

   selenium.open("/seleniummastertestapp/index.php");

   selenium.type("id=login\_login\_username", "user1");

   selenium.type("id=login\_login\_password", "\*\*\*\*"); //password is omitted for security purposes

   selenium.click("id=login\_submit");

**for**(**int**second = 0;; second++) {

**if**(second >= 60) fail("timeout");

**try**{

      assertEquals("Online users", selenium.getText("css=#sb-onlineusers > h3"));

**if**(selenium.isElementPresent("//div[@id='sb-onlineusers']/h3[text()='Online users']"))

**break**; }

**catch**(Exception e) {}

    Thread.sleep(1000);

   }

  }

  @After

**public void**tearDown() **throws**Exception {

   selenium.stop();

  }

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Test ADP Shoe Store Test Cases in Multiple Browsers

1. Tests include using Safari, Chrome and Firefox on a Mac and on Windows
2. Test data should include testing with different user profiles
3. Password is encrypted for security purposes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Action** | **Expected Result** | **Test Data** | **Actual Result** | **Test Result** | **Note** |
| 1 | Navigate to ADP New Shoe Releases Application URL: http://www.adp.com/newshoereleases/index.php | Log in page displays | http://www.adp.com/newshoereleases/index.php |  | **PASS** |  |
| 2 | Enter username in the Username field | Username should be entered and displayed in the username field | User1 |  | **PASS** |  |
| 3 | Enter user password in the Password field | Password should be entered in the password field | \*\*\*\*\* |  | **PASS** | *Password is encrypted for security purposes* |
| 4 | Click on the Log in button | Button should be clicked |  |  | **PASS** |  |
| 5 | Verify that "New Shoe Releases" page is displayed after logging in | The “New Shoe Releases” page should be displayed. |  | The “New Shoe Releases” page was displayed. | **PASS** |  |