

Capstone Project: Create a Testing Framework for the Sportyshoes

Project objective: To develop a comprehensive QA and test suite for the Medicare website. The QA effort will require the following:

1. Browser-based end-user testing using Selenium WebDriver with TestNG framework.
2. Load testing using JMeter.
3. API testing with Cucumber
4. API testing with Postman and Rest Assured The end-deliverables will be executable scripts and modules that can be run on demand for testing the Medicare website.

Source code

Browser-based end-user testing using Selenium WebDriver with TestNG framework.

Base class:

```
package base;
```

```
import org.openqa.selenium.WebDriver;
```

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

```
public class TestBase {
```

```
    public static WebDriver driver;
```

```
    public static void openBrowser()
```

```
    {
```

```
        driver = new ChromeDriver();
```

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

```
        driver.manage().deleteAllCookies();
```

```
        driver.get("http://localhost:9010/");
```

```
    }
```

```
    public void closebrowser()
```

```

        {
            driver.close();
        }
    }
}

```

Pages:-

```
package pages;
```

```

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

```

```
import base.TestBase;
```

```
public class HomePage extends TestBase {
```

```
    // driver.findElement(locator).methods
```

```
    // @FindBy(loacatorname="value")
```

```
    // WebElement name;
```

```
    @FindBy(linkText="New User? Register Here")
```

```
    WebElement registerlink;
```

```
    public HomePage(WebDriver driver)
```

```
    {
```

```
        PageFactory.initElements(driver, this);
```

```
    }
```

```

// action methods

public void getURL_page()
{
    String URL = driver.getCurrentUrl();
    System.out.println(URL);
}

public void Click_registerlink()
{
    registerlink.click();
}
}

```

Loginpage:-

```

package pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.testng.Assert;

import base.TestBase;

public class loginpage extends TestBase {

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

```

```

        @FindBy(xpath="//*[@id=\"password\"]")
        WebElement psw;

        @FindBy(xpath="/html/body/div[2]/form/button")
        WebElement submit;

        public loginpage(WebDriver driver)
        {
            PageFactory.initElements(driver,this);
        }

        public void userlogin()
        {

            email.sendKeys("sadiqali302@gmail.com");
            psw.sendKeys("admin");
            submit.click();

        }

        public void validateurl()
        {
            String expectedurl="http://localhost:9010/login";
            String url=driver.getCurrentUrl();
            Assert.assertEquals(url, expectedurl);
        }

    }
}

```

Productspage

package 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 ProductsPage {
```

```
    @FindBy(xpath="//*[@id='cart101']")
```

```
    WebElement addtocartBTN;
```

```
    @FindBy(linkText="Home")
```

```
    WebElement HomeBTN;
```

```
    @FindBy(linkText="Cart")
```

```
    WebElement CartBTN;
```

```
    @FindBy(xpath="//*[@id='place-order']")
```

```
    WebElement placeOrderBTN;
```

```
    @FindBy(linkText="Orders")
```

```
    WebElement OrdersBTN;
```

```
    @FindBy(linkText="Logout")
```

```
    WebElement logoutBTN;
```

```
    JavascriptExecutor executor;
```

```
public ProductsPage(WebDriver driver)
{
    PageFactory.initElements(driver, this);
    executor = (JavascriptExecutor) driver;
}
```

```
public void click_addtoCart()
{
    executor.executeScript("arguments[0].click();", addtocartBTN);
}
```

```
public void click_homebtn()
{
    HomeBTN.click();
}
```

```
public void click_cartbtn()
{
    CartBTN.click();
}
```

```
public void click_placeorder()
{
    executor.executeScript("arguments[0].click();", placeOrderBTN);
}
```

```
public void click_Ordersbtn()
{
```

```

        OrdersBTN.click();
    }

    public void click_Logoutbtn()
    {
        logoutBTN.click();
    }

}

Registerpage
package pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.testng.Assert;

import base.TestBase;

public class registerpage extends TestBase {

    @FindBy(xpath="//*[@id=\"name\"]")
    WebElement name;

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

```

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

```
WebElement password;
```

```
@FindBy(xpath="/html/body/div[2]/form/button")
```

```
WebElement submit;
```

```
public registerpage(WebDriver driver)
```

```
{
```

```
    PageFactory.initElements(driver,this);
```

```
}
```

```
public void userregistration()
```

```
{
```

```
    name.sendKeys("sadiq ali");
```

```
    email.sendKeys("sadiqali302@gmail.com");
```

```
    password.sendKeys("admin");
```

```
    submit.click();
```

```
}
```

```
public void validateurl()
```

```
{
```

```
    String expectedurl="http://localhost:9010/register-user";
```

```
    String url=driver.getCurrentUrl();
```

```
    Assert.assertEquals(url, expectedurl);
```

```
}
```

```
}
```

Test


```
package testcase;
```

```
import org.testng.Assert;
```

```
import org.testng.annotations.Test;
```

```
import base.TestBase;
```

```
import pages.HomePage;
```

```
public class HomePagetest extends TestBase {
```

```
    HomePage hp;
```

```
    @Test(priority='1')
```

```
    public void start_browser()
```

```
    {
```

```
        openBrowser();
```

```
        hp = new HomePage(driver);
```

```
    }
```

```
    @Test(priority='2')
```

```
public void test_getURL_method()
{
    hp.getURL_page();
}
```

```
@Test(priority='2')
public void test_registerlink_method()
{
    hp.Click_registerlink();
    String expectedURL =
"http://localhost:9010/register";
    String RegisterPageURL = driver.getCurrentUrl();
    Assert.assertEquals(RegisterPageURL,
expectedURL);
}
```

```
}
```

Login page test

```
package testcase;
```

```
import org.testng.annotations.Test;
```

```
import base.TestBase;
```

```
import pages.loginpage;
```

```
public class loginpagetest extends TestBase {
```

```
    loginpage lp;
```

```
    public void openbrowser()
```

```
    {
```

```
        openBrowser();
```

```
        lp=new loginpage(driver);
```

```
    }
```

```
    @Test(priority='1')
```

```
    public void entereduser()
```

```
    {
```

```
        lp.userlogin();
```

```
    }
```

```
        @Test(priority='2')
        public void validatesuccess()
        {
            lp.validateurl();
        }

    }
```

Product page test

```
package testcase;
```

```
import org.testng.annotations.BeforeClass;
```

```
import org.testng.annotations.Test;
```

```
import base.TestBase;
```

```
import pages.ProductsPage;
```

```
import pages.loginpage;
```

```
public class ProductsTest extends TestBase {

    loginpage lp;
    ProductsPage pp;

    @BeforeClass
    public void start_browser()
    {
        openBrowser();
        lp = new loginpage(driver);
        pp = new ProductsPage(driver);
    }

    @Test(priority='1')
    public void test_user_login() throws
InterruptedException
    {
        Thread.sleep(1000);
        lp.userlogin();
    }

    @Test(priority='2')
```

```
        public void test_product_addtoCart() throws  
InterruptedException
```

```
    {  
        Thread.sleep(1000);  
        pp.click_addtoCart();  
    }
```

```
    @Test(priority='3')
```

```
        public void test_HomeBTN() throws  
InterruptedException
```

```
    {  
        Thread.sleep(1000);  
        pp.click_homebtn();  
    }
```

```
    @Test(priority='4')
```

```
        public void test_CartBTN() throws  
InterruptedException
```

```
    {  
        Thread.sleep(1000);  
        pp.click_cartbtn();  
    }
```

```
}
```

```
@Test(priority='5')
```

```
public void test_placeorderBTN() throws  
InterruptedException
```

```
{
```

```
    Thread.sleep(1000);
```

```
    pp.click_placeorder();
```

```
}
```

```
@Test(priority='6')
```

```
public void test_orders() throws  
InterruptedException
```

```
{
```

```
    Thread.sleep(1000);
```

```
    pp.click_homebtn();
```

```
    pp.click_Ordersbtn();
```

```
}
```

```
@Test(priority='7')
```

```
        public void test_logout() throws  
InterruptedException
```

```
    {  
        Thread.sleep(1000);  
        pp.click_Logoutbtn();  
    }
```

```
}
```

```
Register page test
```

```
package testcase;
```

```
import org.testng.annotations.BeforeClass;
```

```
import org.testng.annotations.Test;
```

```
import base.TestBase;
```

```
import pages.HomePage;
```

```
import pages.registerpage;
```

```
public class registerpagetest extends TestBase {
```

```
    HomePage hp;
```



```
registerpage rp;
@BeforeClass
public void openbrowser()
{
    openBrowser();
    hp=new HomePage(driver);
    rp=new registerpage(driver);
}
@Test(priority='1')

    public void clickregisterlink() throws
InterruptedException
    {
        Thread.sleep(1000);
        hp.Click_registerlink();

    }
@Test(priority='2')
public void test_registeruser()
{
```

```
rp.userregistration();  
}  
@Test(priority='3')  
public void testurl()  
{  
rp.validateurl();  
}  
  
}
```

Load Testing using Jmeter

API testing with Cucumber

Cucumber feature file Sporty shoes

Feature: Api testing for sportyshoes application

Scenario: Retrieve the list of all products in the store.

Given user sends a get request to get the products

Scenario: Retrieve the list of all registered users

Given user sends a get request to list the users

Scenario: Add the product.

When user gives new products details add the product

Scenario: Delete the product

When user give productid ,delete the product

Scenario: Update the product

When user give productid and update data

Steps definition file

```
package steps;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.restassured.RestAssured;

public class stepssportyshoes {
    @Given("user sends a get request to get the products")
    public void user_sends_a_get_request_to_get_the_products() {
        RestAssured.given()
            .baseUrl("http://localhost:9010")
            .basePath("/get-shoes")
            .when().get().then().statusCode(200).log().all();
    }

    @Given("user sends a get request to list the users")
    public void user_sends_a_get_request_to_list_the_users() {
        RestAssured.given()
            .baseUrl("http://localhost:9010")
            .basePath("/get-users")
            .when().get().then().statusCode(200).log().all();
    }

    @When("user gives new products details add the product")
    public void user_gives_new_products_details_add_the_product() {
        RestAssured.given()
            .baseUrl("http://localhost:9010")
            .basePath("/add-shoe")
            .queryParams("id", "10001")
            .queryParams("image", "sadiq.jpg")
            .queryParams("name", "sadiqali")
            .queryParams("category", "running")
            .queryParams("price", "2000")
            .queryParams("sizes", "9")
            .when().post()
            .then().statusCode(200)
            .log().all()
            ;
    }

    @When("user give productid ,delete the product")
    public void user_give_productid_delete_the_product() {
        // Write code here that turns the phrase above into concrete actions
        RestAssured.given()
            .baseUrl("http://localhost:9010")
            .basePath("/delete-shoe")
    }
```

```

        .queryParams("id", "101")
        .when().delete().then().statusCode(200).log().all();
    }

    @When("user give productid and update data")
    public void user_give_productid_and_update_data() {
        RestAssured.given()
            .baseUrl("http://localhost:9010")
            .basePath("/update-shoe")
            .queryParams("id", "1010")
            .queryParams("image", "sadiq.jpg")
            .queryParams("name", "newadded")
            .queryParams("category", "running")
            .queryParams("price", "2000")
            .queryParams("sizes", "9")
            .when().put().then().statusCode(200).log().all();
    }
}

```

Runner file

```

package runner;

import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;

import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)

@CucumberOptions(features = "C:\\Users\\sadiq\\eclipse-
workspace\\medicare_capstoneproject\\src\\test\\java\\feature\\sportsshoes.feature", glue = {
"steps" }, dryRun = false, plugin = { "pretty",
"html:target/cucumberreport.html" })

public class TestRunner2 {

}

```

Api testing with postman

Get products

```

{
  "code": 101,

```

```
"message": "12 Shoes Fetched Successfully.",
"shoes": [
  {
    "id": 101,
    "image": "1.png",
    "name": "BlueWave Running Shoes",
    "category": "Sports",
    "sizes": "7, 8, 9",
    "price": 2000
  },
  {
    "id": 201,
    "image": "2.png",
    "name": "Elegant Leather Loafers",
    "category": "Formal",
    "sizes": "7, 8, 9",
    "price": 3000
  },
  {
    "id": 301,
    "image": "3.png",
    "name": "NeoFlex Athletic Shoes",
    "category": "Sports",
    "sizes": "7, 8, 9, 10",
    "price": 4500
  },
  {
    "id": 401,
    "image": "4.png",
    "name": "PowerStride Training Shoes",
    "category": "Sports",
    "sizes": "6, 7, 8, 9",
    "price": 6000
  },
  {
    "id": 501,
    "image": "5.png",
    "name": "StreetRunner Urban Sneakers",
    "category": "Sports",
    "sizes": "7, 9",
    "price": 4000
  },
  {
    "id": 601,
    "image": "6.png",
    "name": "VentureHike Trail Shoes",
    "category": "Sports",
    "sizes": "5, 8, 9",
    "price": 2500
  },
  {

```

```
    "id": 701,
    "image": "7.png",
    "name": "EnduraGrip Sports Sneakers",
    "category": "Sports",
    "sizes": "4, 6, 9",
    "price": 5000
  },
  {
    "id": 801,
    "image": "8.png",
    "name": "LightStride Performance Shoes",
    "category": "Sports",
    "sizes": "7, 8",
    "price": 1200
  },
  {
    "id": 901,
    "image": "9.png",
    "name": "MaxFit Pro Sports Shoes",
    "category": "Sports",
    "sizes": "7, 8, 9, 10",
    "price": 4700
  },
  {
    "id": 111,
    "image": "10.png",
    "name": "RapidFlex Running Shoes",
    "category": "Sports",
    "sizes": "4, 5, 6",
    "price": 2500
  },
  {
    "id": 211,
    "image": "11.png",
    "name": "ZenFlex Sports Sneakers",
    "category": "Sports",
    "sizes": "7, 8, 9",
    "price": 6000
  },
  {
    "id": 311,
    "image": "12.png",
    "name": "TrailBlaze Adventure Shoes",
    "category": "Sports",
    "sizes": "5, 6, 9",
    "price": 7500
  }
]
}
```

GetUsers

```
{
  "code": 101,
  "message": "3 Users Fetched Successfully.",
  "users": [
    {
      "name": "John Watson",
      "email": "john@example.com",
      "password": "john123"
    },
    {
      "name": "Fionna Flynn",
      "email": "fionna@example.com",
      "password": "fionna123"
    },
    {
      "name": "Sia Sen",
      "email": "sia@example.com",
      "password": "sia123"
    }
  ]
}
```

Add the product

```
{
  "code": 101,
  "message": "SampleShoe Added Successfully.",
  "shoe": {
    "id": 101,
    "image": "image_url",
    "name": "SampleShoe",
    "category": "Running",
    "sizes": "9",
    "price": 1000
  }
}
```

Delete the product

```
{
  "code": 101,
  "message": "Shoe with ID 101 Deleted Successfully."
}
```

Update the product

```
{  
  "code": 101,  
  "message": "saprx Updated Successfully.",  
  "shoe": {  
    "id": 201,  
    "image": "sadiq.png",  
    "name": "saprx",  
    "category": "running",  
    "sizes": "9",  
    "price": 2000  
  }  
}
```