Лабораторная работа 6

Цель работы: Основываясь на предыдущей тестовой документации ( тест-кейсы ), написать автоматизированную проверку этих тест кейсов ( автотесты ). Также создать Unit тесты к этой лабе

В данной работе я использовал инструмент «Web Selenium» для тестирования сайта «Ozon»

Программный код автотестов:

private readonly By \_button\_addItem = By.XPath("//button[@class = 'Button\_sklv-button\_\_kGBfm Button\_sklv-button\_blue\_\_TOjSO ProductListItem\_product-hover-btn\_\_wwPKC']");

private readonly By \_button\_orderItem = By.Id("btn-create-order");

private readonly By \_container\_ring = By.XPath("//img[@class='ProductListItem\_product-image\_\_mylAz']");

private readonly By \_button\_giftCart = By.XPath("//button[text()='Подарочная карта']");

private readonly By \_input\_cart = By.XPath("//input[@name='cardCode']");

private readonly By \_buttuon\_checkBalance = By.XPath("//button[@id='checkBalance']");

private readonly By \_button\_size = By.XPath("//button[text()='16']");

private IWebDriver driver;

Actions actions;

[SetUp]

public void Setup()

{

driver = new OpenQA.Selenium.Chrome.ChromeDriver();

driver.Manage().Window.Maximize();

}

// Открытие сайта

[Test]

public void Test1()

{

driver.Navigate().GoToUrl("https://sokolov.ru/");

}

// Добавление товара в корзину

[Test]

public void Test2()

{

driver.Navigate().GoToUrl("https://sokolov.ru/jewelry-catalog/rings/");

Thread.Sleep(1000);

actions = new Actions(driver);

actions.MoveToElement(driver.FindElement(\_container\_ring)).Perform();

var button = driver.FindElement(\_button\_addItem);

button.Click();

Thread.Sleep(1000);

var button\_size = driver.FindElement(\_button\_size);

button\_size.Click();

}

// Покупка товара

[Test]

public void Test3()

{

driver.Navigate().GoToUrl("https://sokolov.ru/jewelry-catalog/rings/");

Thread.Sleep(1000);

actions = new Actions(driver);

actions.MoveToElement(driver.FindElement(\_container\_ring)).Perform();

var button = driver.FindElement(\_button\_addItem);

button.Click();

Thread.Sleep(1000);

var button\_size = driver.FindElement(\_button\_size);

button\_size.Click();

driver.Navigate().GoToUrl("https://sokolov.ru/basket/");

Thread.Sleep(1000);

var button\_order = driver.FindElement(\_button\_orderItem);

button\_order.Click();

}

// Ввести существующие данные в поле "Номер карты"

[Test]

public void Test4()

{

driver.Navigate().GoToUrl("https://sokolov.ru/");

Thread.Sleep(500);

var button\_giftCart = driver.FindElement(\_button\_giftCart);

button\_giftCart.Click();

Thread.Sleep(1000);

}

// Переход на страницу "Подарочная карта"

[Test]

public void Test5()

{

driver.Navigate().GoToUrl("https://sokolov.ru/gift-card/");

var input\_cart = driver.FindElement(\_input\_cart);

input\_cart.SendKeys("131 313 131 313 1");

var button\_checkBalance = driver.FindElement(\_buttuon\_checkBalance);

button\_checkBalance.Click();

}

}

Программный код Unit тестов:

namespace UnitTests

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestAddition()

{

int a = 3;

int b = 4;

int result = a + b;

Assert.AreEqual(7, result);

}

[TestMethod]

public void TestSubtraction()

{

int a = 8;

int b = 3;

int result = a - b;

Assert.AreEqual(5, result);

}

[TestMethod]

public void TestMultiplication()

{

int a = 5;

int b = 6;

int result = a \* b;

Assert.AreEqual(30, result);

}

[TestMethod]

public void TestDivision()

{

int a = 15;

int b = 3;

int result = a / b;

Assert.AreEqual(5, result);

}

[TestMethod]

public void TestStringConcatenation()

{

string str1 = "Hello";

string str2 = "World";

string result = str1 + " " + str2;

Assert.AreEqual("Hello World", result);

}

}

}