

Week 3 (Automation Testing)

→ Tools Used

- **Automation Tool:** Selenium WebDriver
- **Language:** C#
- **Test Framework:** NUnit
- **API Tool (Optional):** Postman
- **Test Application:** SauceDemo (<https://www.saucedemo.com>)

→ Automation Testing Overview

Automation testing is used to execute test cases automatically using scripts instead of manual effort. It helps save time, improve accuracy, and support regression testing.

In this project, Selenium WebDriver is used to automate browser actions such as login, navigation, and feature validation.

→ Automated Test Cases

Test Case 1: Login Validation

Objective:

Verify that a user can successfully log in using valid credentials.

Steps Automated:

1. Open SauceDemo website
2. Enter valid username
3. Enter valid password
4. Click Login button
5. Verify user is redirected to Products page

Expected Result:

User should log in successfully and see the Products page.

Status: Passed

Test Case 2: Feature Functionality Add Product to Cart

Objective:

Verify that a product can be added to the cart.

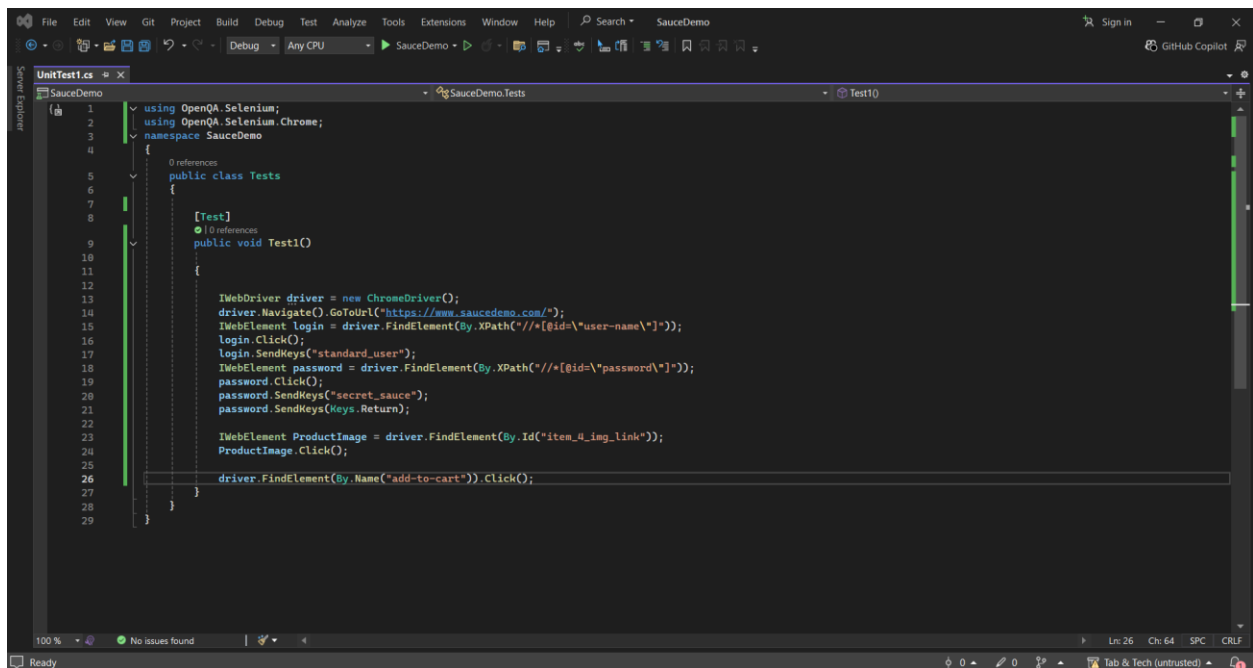
Steps Automated:

1. Login to the application
2. Click on a product
3. Click “Add to Cart” button
4. Verify cart icon shows item count

Expected Result:

Product should be added to cart successfully.

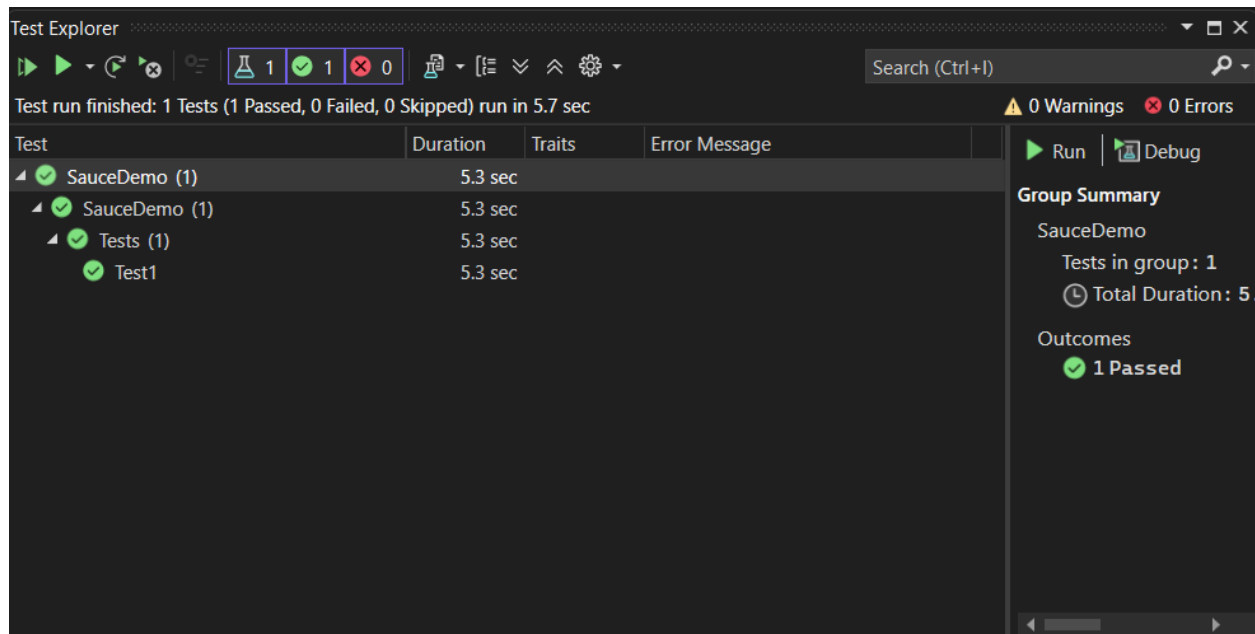
Status: Passed



The screenshot shows the Visual Studio IDE with a C# Selenium test file named `UnitTests.cs` open. The code is for a test class `Tests` in the `SauceDemo` namespace. It includes the following steps:

```
1 using OpenQA.Selenium;
2 using OpenQA.Selenium.Chrome;
3 namespace SauceDemo
4 {
5     // 0 references
6     public class Tests
7     {
8         // 1 0 references
9         [Test]
10        public void Test1()
11        {
12            IWebDriver driver = new ChromeDriver();
13            driver.Navigate().GoToUrl("https://www.saucedemo.com/");
14            IWebElement login = driver.FindElement(By.XPath("//*[@id='user-name']"));
15            login.Click();
16            login.SendKeys("standard_user");
17            IWebElement password = driver.FindElement(By.XPath("//*[@id='password']"));
18            password.Click();
19            password.SendKeys("secret_sauce");
20            password.SendKeys(Keys.Return);
21
22            IWebElement ProductImage = driver.FindElement(By.Id("item_4_img_link"));
23            ProductImage.Click();
24
25            driver.FindElement(By.Name("add-to-cart")).Click();
26        }
27    }
28 }
29
```

The status bar at the bottom indicates "100 %", "No issues found", and the file is located at "Ln: 26 Ch: 64 SPC: CRLF".



➔ SCRIPT:

```
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
namespace SauceDemo
{
    public class Tests
    {
        [Test]
        public void Test1()
        {
            IWebDriver driver = new ChromeDriver();
            driver.Navigate().GoToUrl("https://www.saucedemo.com/");
            IWebElement login = driver.FindElement(By.XPath("//*[@id=\"user-name\"]"));
            login.Click();
            login.SendKeys("standard_user");
            IWebElement password = driver.FindElement(By.XPath("//*[@id=\"password\"]"));
            password.Click();
            password.SendKeys("secret_sauce");
            password.SendKeys(Keys.Return);

            IWebElement ProductImage = driver.FindElement(By.Id("item_4_img_link"));
            ProductImage.Click();

            driver.FindElement(By.Name("add-to-cart")).Click();
        }
    }
}
```

→ API Testing Using Postman (Simple Books API)

API Reference Used

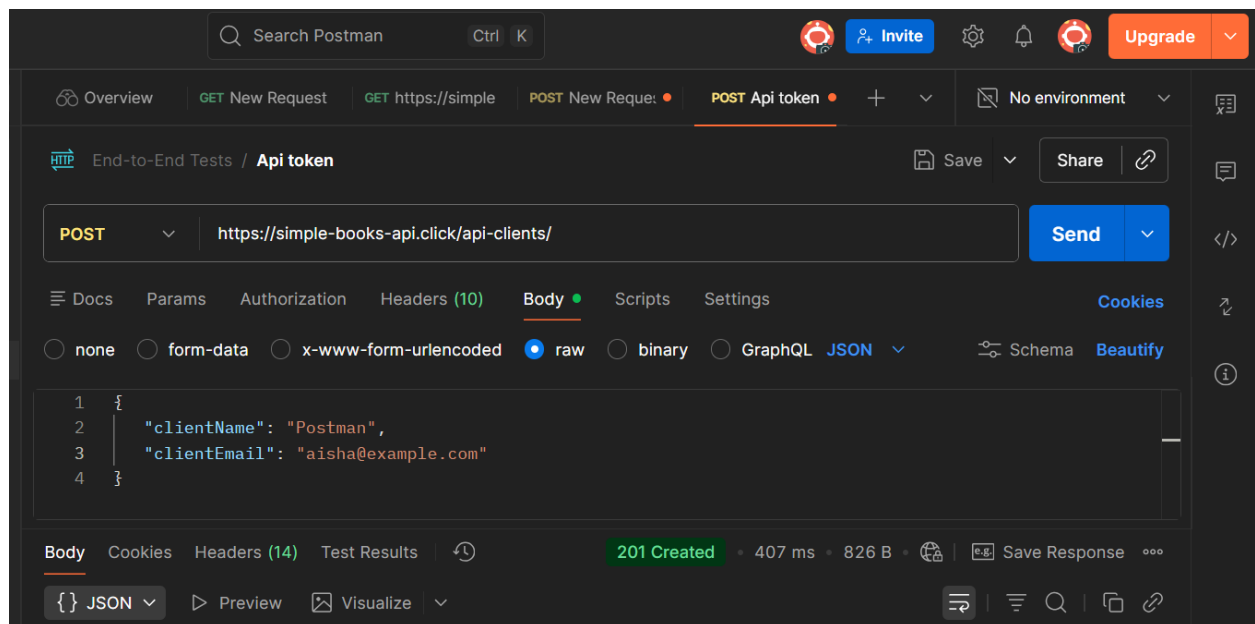
For API testing, the **Simple Books API** was used.

This API provides endpoints for practicing common HTTP methods such as **GET, POST, PUT, and DELETE**.

API Documentation Source:

Simple Books API , Introduction to Postman Course

→ Post (Token Generation):



→ GET (List Of Books):

Search Postman Ctrl K

Overview GET New Reql GET https://sin POST Submit t POST Api toke GET List of bor + No environment

End-to-End Tests / List of books Save Share

GET https://simple-books-api.gltch.me/books Send

Docs Params Authorization Headers (8) Body Scripts Settings Cookies

Body Cookies Headers (14) Test Results 200 OK 939 ms 1.13 KB Save Response

{ } JSON Preview Visualize

	id	name	type	available
0	1	The Russian	fiction	true
1	3	The Vanishing Half	fiction	true
2	4	The Midnight Library	fiction	true
3	6	Viscount Who Loved Me	fiction	true
4	2	Just as I Am	non-fiction	false
5	5	Untamed	non-fiction	true

Postbot Runner Capture requests Cloud Agent Cookies Vault Trash ?

➔ POST (Submit an Order):

Search Postman Ctrl K

Overv x GET New Reql GET https://sin POST Submit t POST Api toke POST New Req + No environment

End-to-End Tests / Submit an Order Save Share

POST https://simple-books-api.click/orders Send

Docs Params Authorization Headers (10) Body Scripts Settings Cookies

Auth Type Bearer Token Token

The authorization header will be automatically generated when

Body Cookies Headers (14) Test Results 201 Created 370 ms 794 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "created": true,
3   "orderId": "gydYtys08L6UeXp9Li9hh"
4 }
```

➔ DELETE (Delete an order):

OverviewGET New RGET https://POST SubmPOST Api tcGET List of IDEL New R+No environment

HTTP End-to-End Tests / New RequestSaveShare

DELETEhttps://simple-books-api.click/orders/gydYtys08L6UeXp9Li9hhSend

DocsParamsAuthorizationHeaders (8)BodyScriptsSettingsCookies

Headers8 hidden

	Key	Value	Description		Bulk Edit	Presets
	Key	Value	Description			

BodyCookiesHeaders (14)Test Results204 No Content517 ms746 BSave Response

{ } JSONPreviewVisualize

1