**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

A blue book with a black ring around it

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

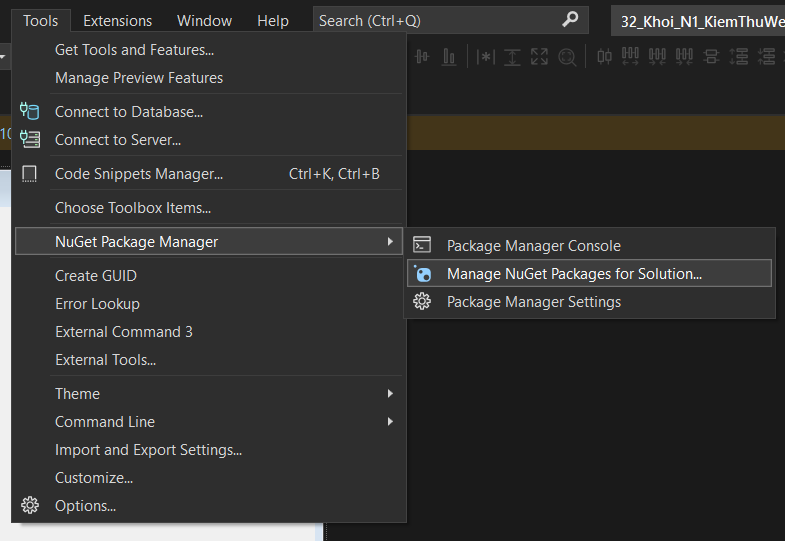
**ĐỀ TÀI**: **KIỂM THỬ CHỨC NĂNG WEBSITE AKKO**

|  |  |
| --- | --- |
| **LỚP:** | **DH21IT02** |
| **GIẢNG VIÊN BỘ MÔN:** | **NGUYỄN THỊ NGỌC THANH** |
| **SINH VIÊN THỰC HIỆN:** | **VÕ DUY KHÔI** |

1. **CÔNG CỤ KIỂM THỬ TỰ ĐỘNG**
2. **Selenium**

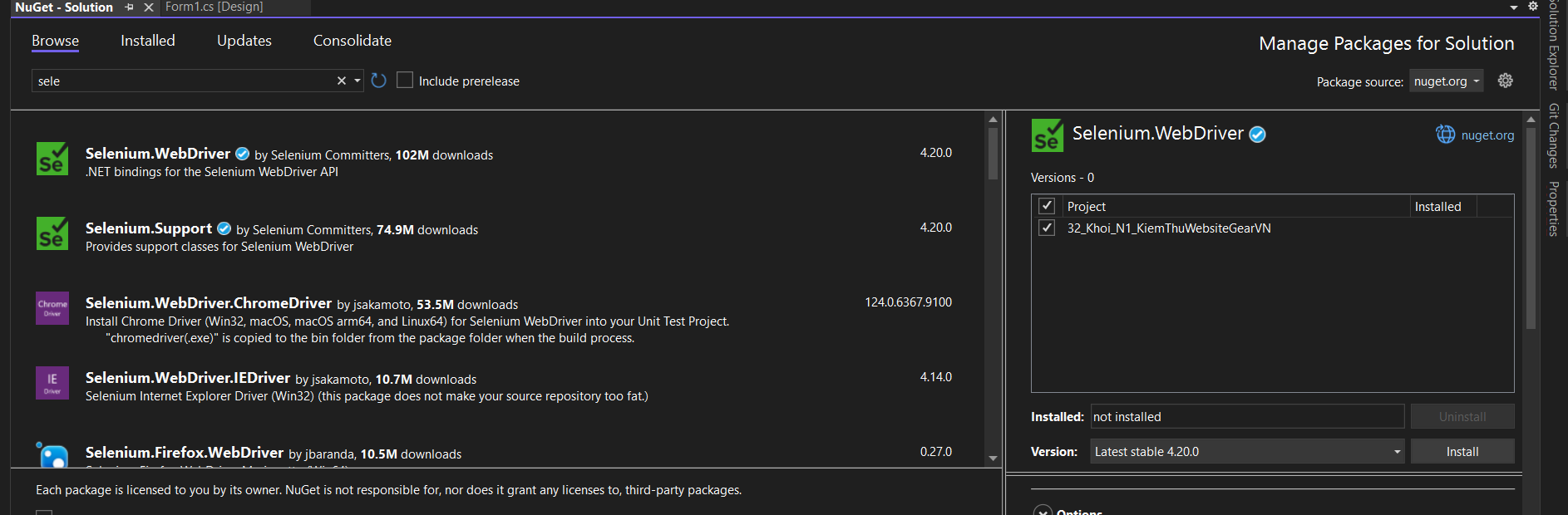
**Hướng dẫn cài đặt Selenium**

* Tại project C#, chọn menu Tools > NuGet Package Manager > Manage NuGet Package for Solution…



1. *Màn hình truy cập nơi cài đặt*

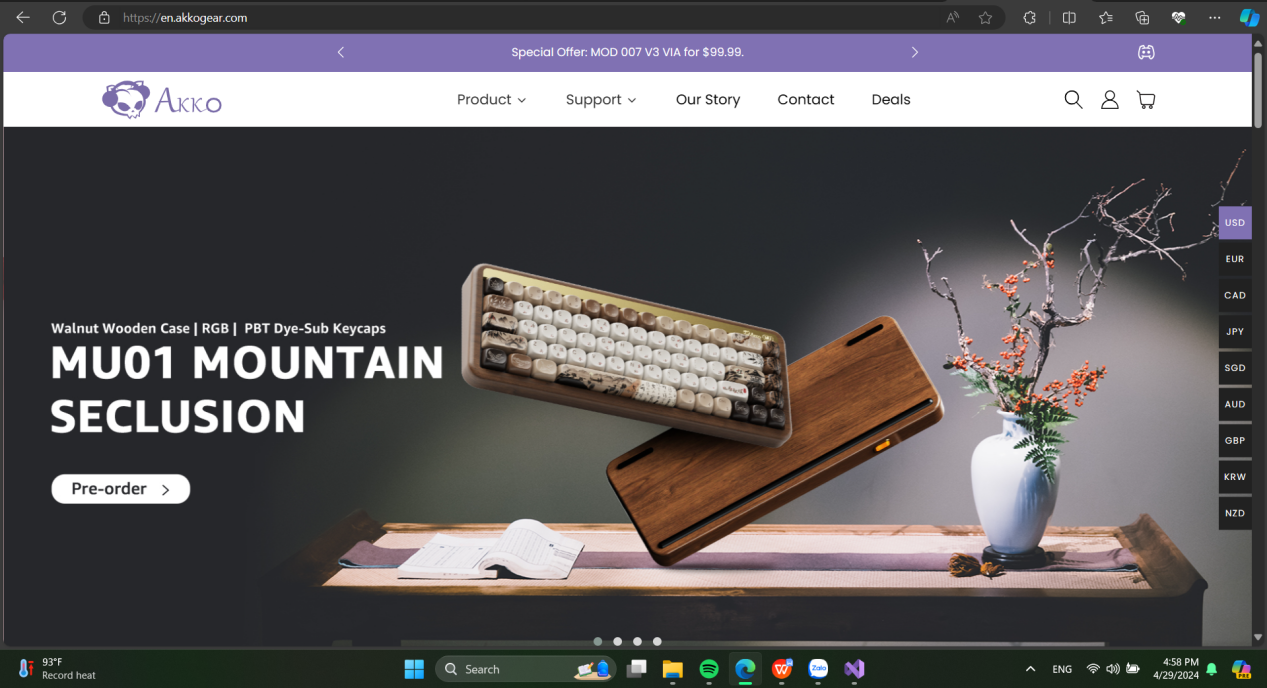
* Click tab Browse và tìm kiếm từ khóa Selenium và các WebDriver muốn kiểm thử để cài đặt.



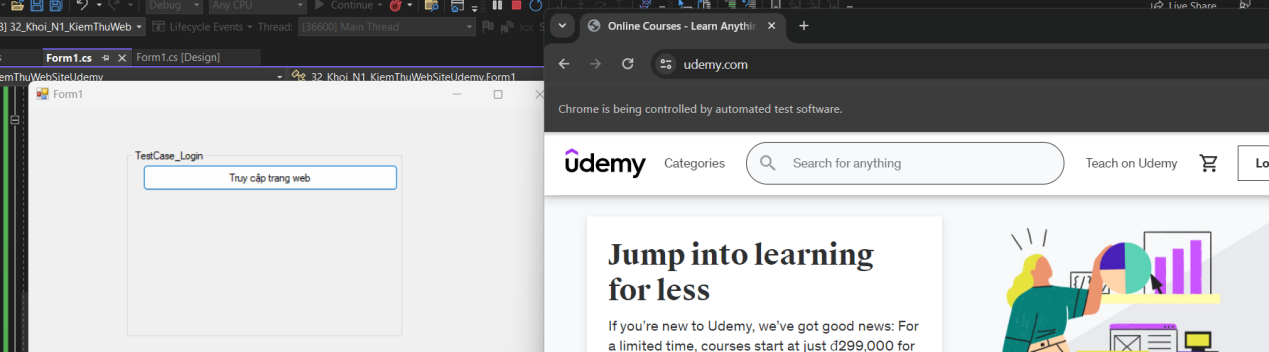
*Hình 2. Màn hình cài đặt*

* Khi chạy chương trình nếu bị lỗi hãy đọc thông báo lỗi để tải đúng theo phiên bản tương thích được yêu cầu

1. **KẾ HOẠCH TEST**
2. **Truy cập trang web cần test**



*Hình 3. màn hình giao diện trang web khi truy cập*



*Hình 4. kết quả chạy trang web bằng code*

**Code:**

|  |
| --- |
| private void btn\_32\_Khoi\_TruyCapTrangWeb\_Click(object sender, EventArgs e)  {  //32\_Khoi  ChromeDriverService chrome\_khoi\_32 = ChromeDriverService.CreateDefaultService();  chrome\_khoi\_32.HideCommandPromptWindow = true;  IWebDriver driver = new ChromeDriver(chrome\_khoi\_32);  //gọi trang web  driver.Navigate().GoToUrl("https://en.akkogear.com/");  } |

1. **Các chức năng cần test**
2. **Chức năng đăng nhập (login)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE** | **THUỘC TÍNH** | **DỮ LIỆU** | **Kết quả dự kiến** | **Kết quả thực tế** | **KẾT QUẢ** |
| 1 | Đăng nhập thành công | Tk:"voduykhoi32"  Mk:"khoi123456" | Đăng nhập thành công | Như dự kiến | PASS |
| 2 | Tên đăng nhập không hợp lệ | Tk:"voduykhoi12"  MK:"khoi123456" | Đăng nhập thất bại | - | FAIL |
| 3 | Mật khẩu không hợp lệ | TK"voduykhoi32"  MK:"khoi123" | Đăng nhập thất bại | - | FAIL |
| 4 | Tên đăng nhập để trống | TK:" "  MK:"khoi123456" | Đăng nhập thất bại | - | FAIL |
| 5 | Mật khẩu để trống | TK:"voduykhoi32"  MK;"" | Đăng nhập thất bại | - | FAIL |
| 6 | Tài khoản có chứa kí tự đặc biệt | Tk:"VoDuyKhoi32@"  Mk:"khoi123456" | Đăng nhập thất bại | - | FAIL |
| 7 | Đăng nhập bằng tài khoản chưa được kích hoạt | TK:"duykhoi32"  MK:"khoi123456" | Đăng nhập thất bại | - | FAIL |

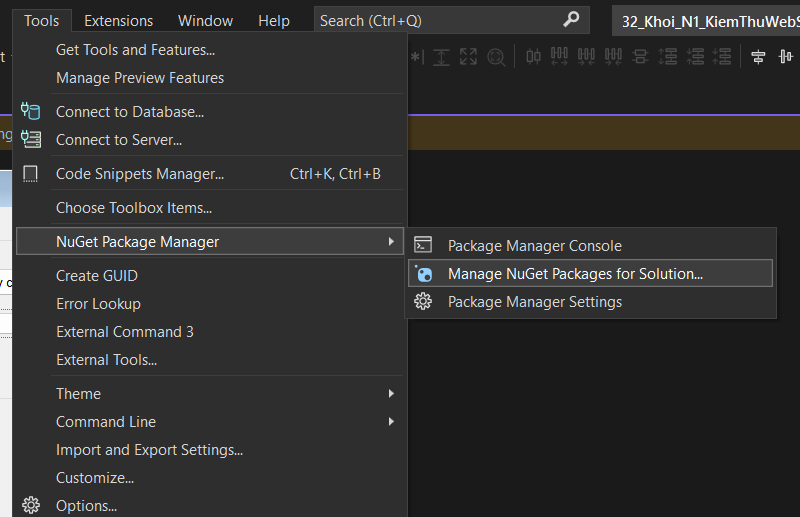
**1.1.đặc tả chức năng**

|  |  |
| --- | --- |
| **Tên chức năng** | Đăng nhập |
| **Mô tả** | Chức năng Đăng nhập cho phép người dùng xác thực và truy cập vào hệ thống bằng cách cung cấp tên đăng nhập và mật khẩu hợp lệ. |
| **Tác nhân** | Người dùng- người sử dụng hệ thống |
| **Điều kiện trước** | Giao diện đắng nhập đã hiển thị cho người dùng |
| **Điều kiện sau** | Tồn tại tài khoản người dùng trong hệ thống |
| **Ngoại lệ** |  |
| **Yêu cầu đặc biệt** |  |

**1.2.Thực hiện**

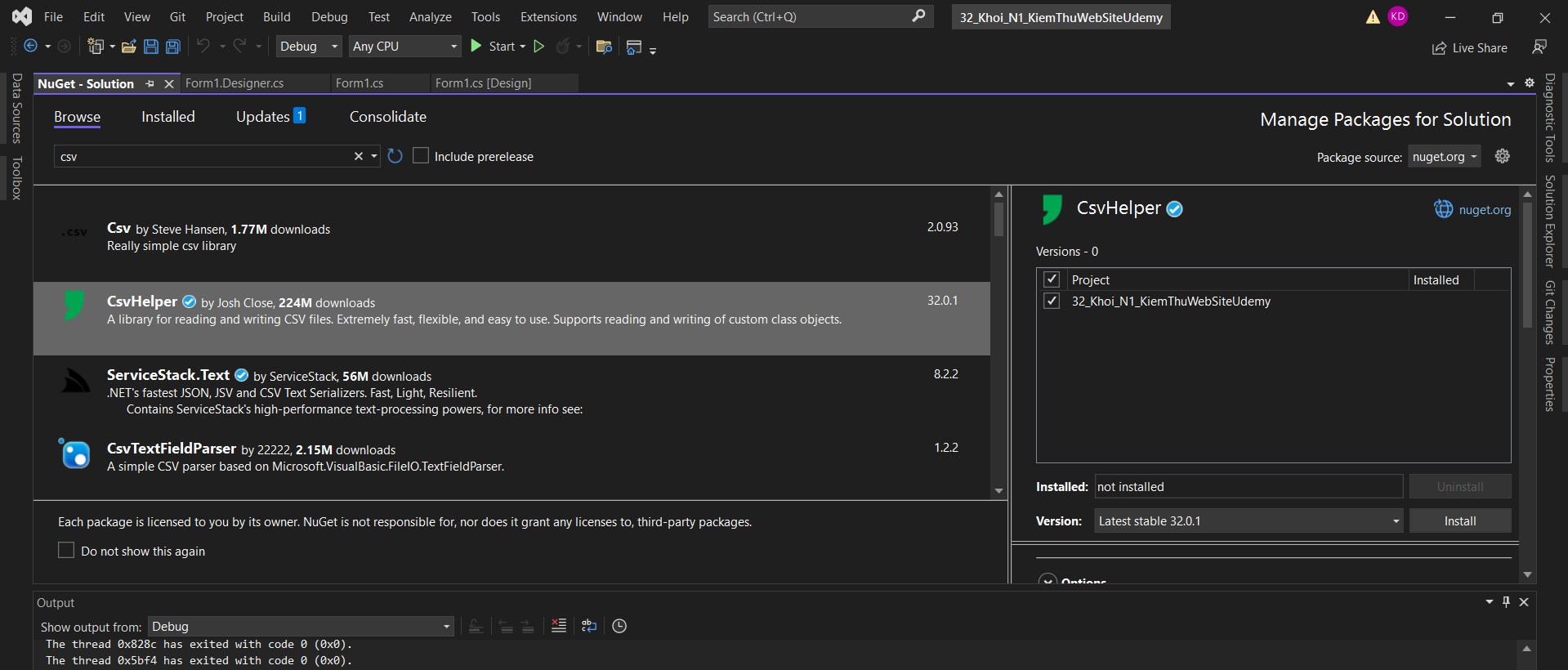
**Bước 1.Cài đặt CSV**

* Truy cập Manage NuGet Packages for Solution…



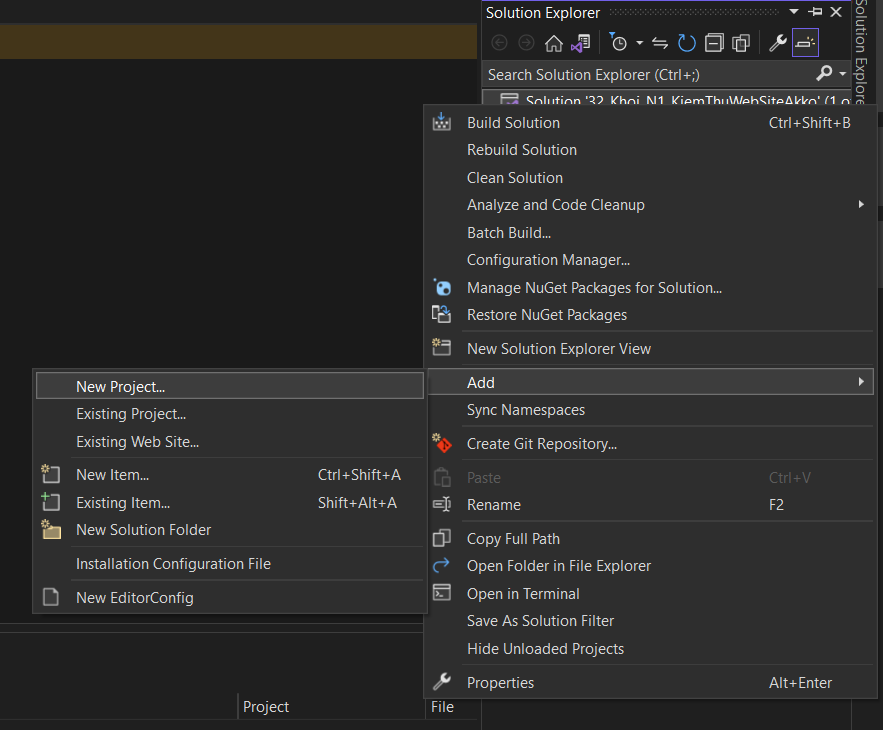
*Hình 5. Giao diện tuy cập Manage Nuget Packages…*

* Tìm CSV Helper, thực hiện cài đặt

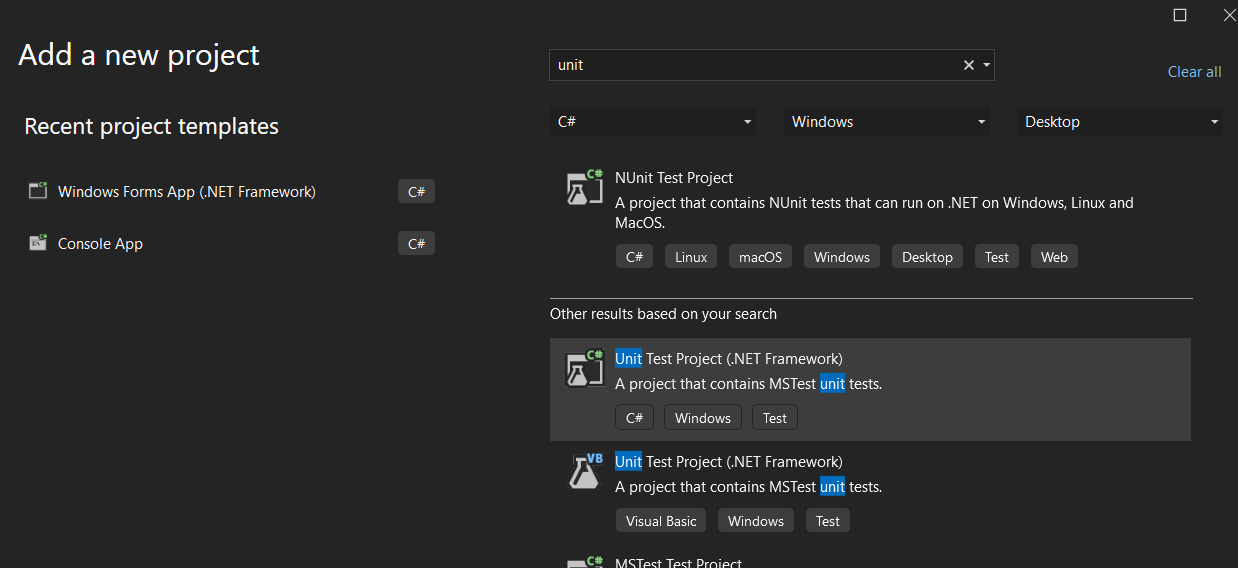


*Hình 6. Tìm kiếm CSV Helper*

* Tạo một project mới từ Solution: chuột phải vào solution->Add->new Project

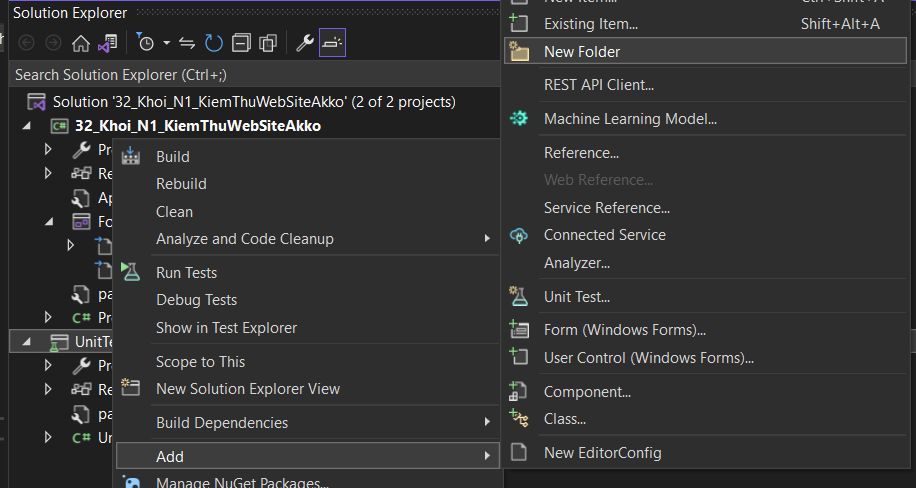


*Hình 7. Tạo Project*

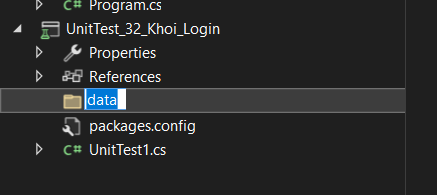


*Hình 8. Tạo Project Unit Test*

* Sau đó hãy cài lại webdriver và cho project mới rồi mới làm bước tiếp theo
* Thêm Folder data vào project mới



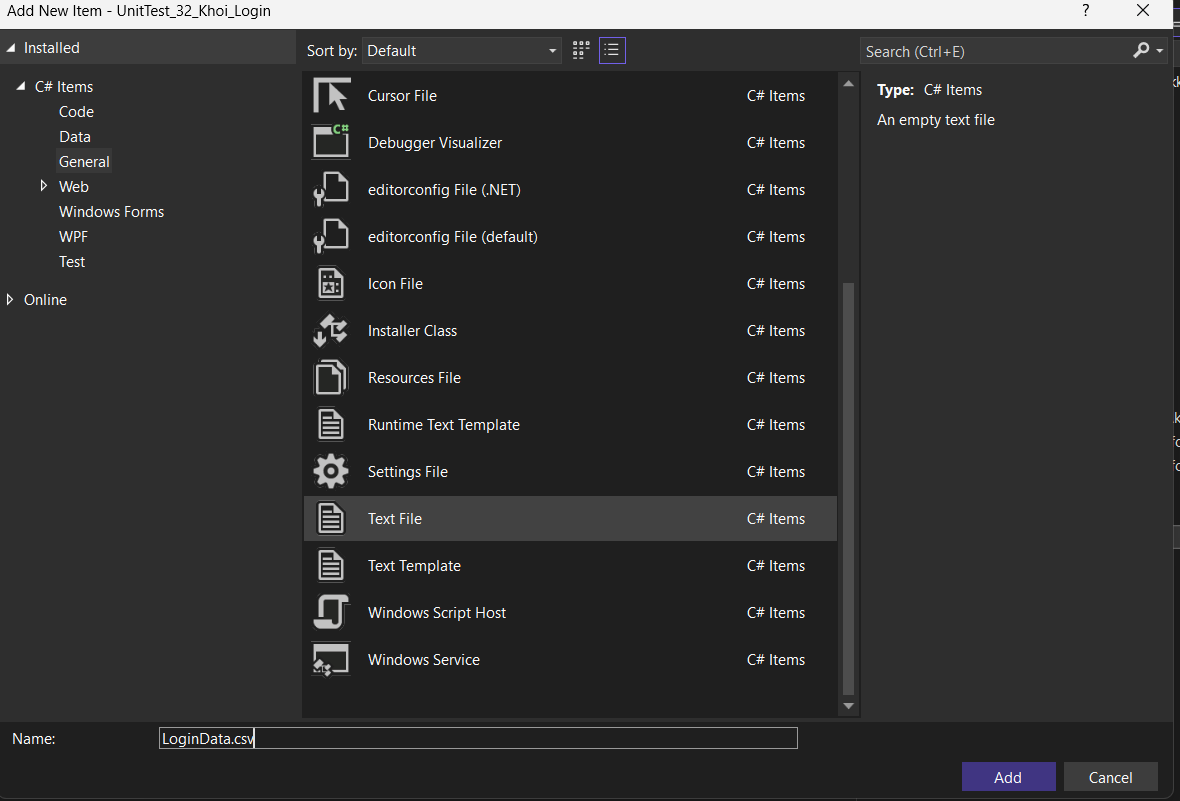
*Hình 9. giao diện thêm folder*



*Hình 10. folder sau khi thêm*

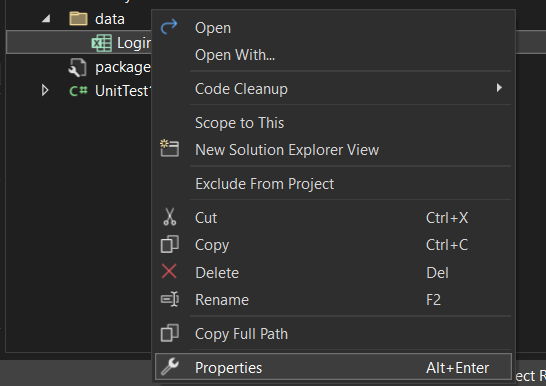


*Hình 11. giao diện thêm CSV cho data (1)*



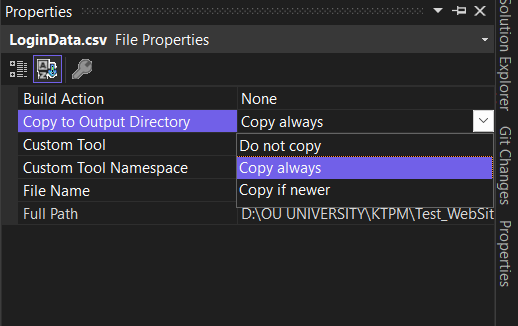
*Hình 12. giao diện thêm CSV cho data (2)*

Chuột phải tập tin LoginData.csv chon Propeties



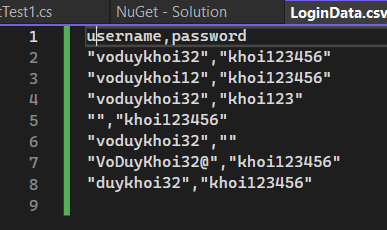
*Hình 13. giao diện thêm CSV cho data*

Thiết lập thuôc tính “Copy to Output Directory” thành “copy always” để tập tin này sẽ được sao chép vào thư mục bin khi build prokect



*Hình 14. giao diện đổi properties của LoginData*

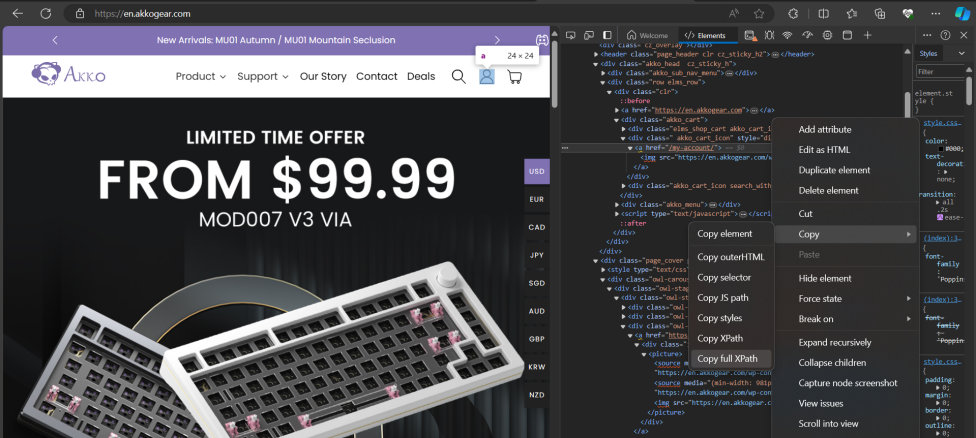
Thêm dữ liệu ứng với các TestCase đã phân tích



*Hình 15.Chuẩn bị dữ liệu trong file LoginData.csv*

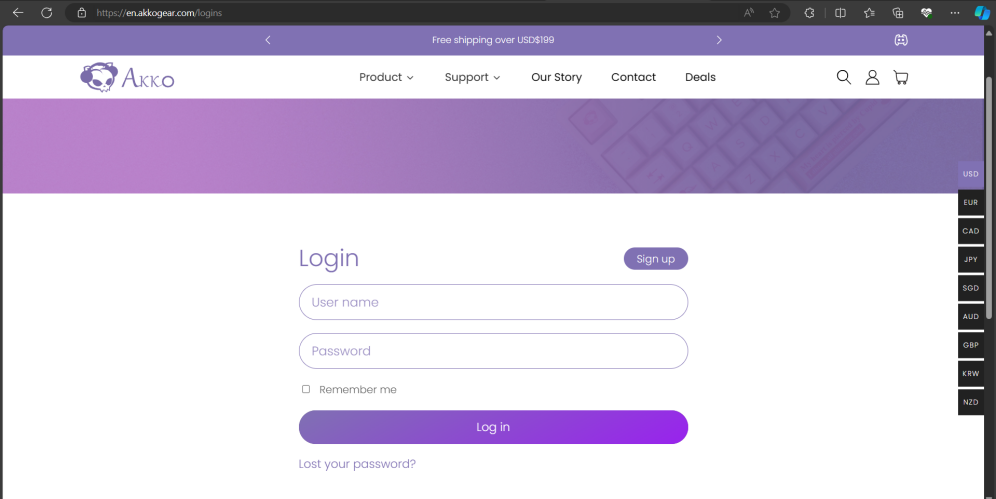
**Bước 2. Tự động truy cập trang web bằng cách nêu trên ( I )**

**Bước 3. Lấy element Đăng nhập**



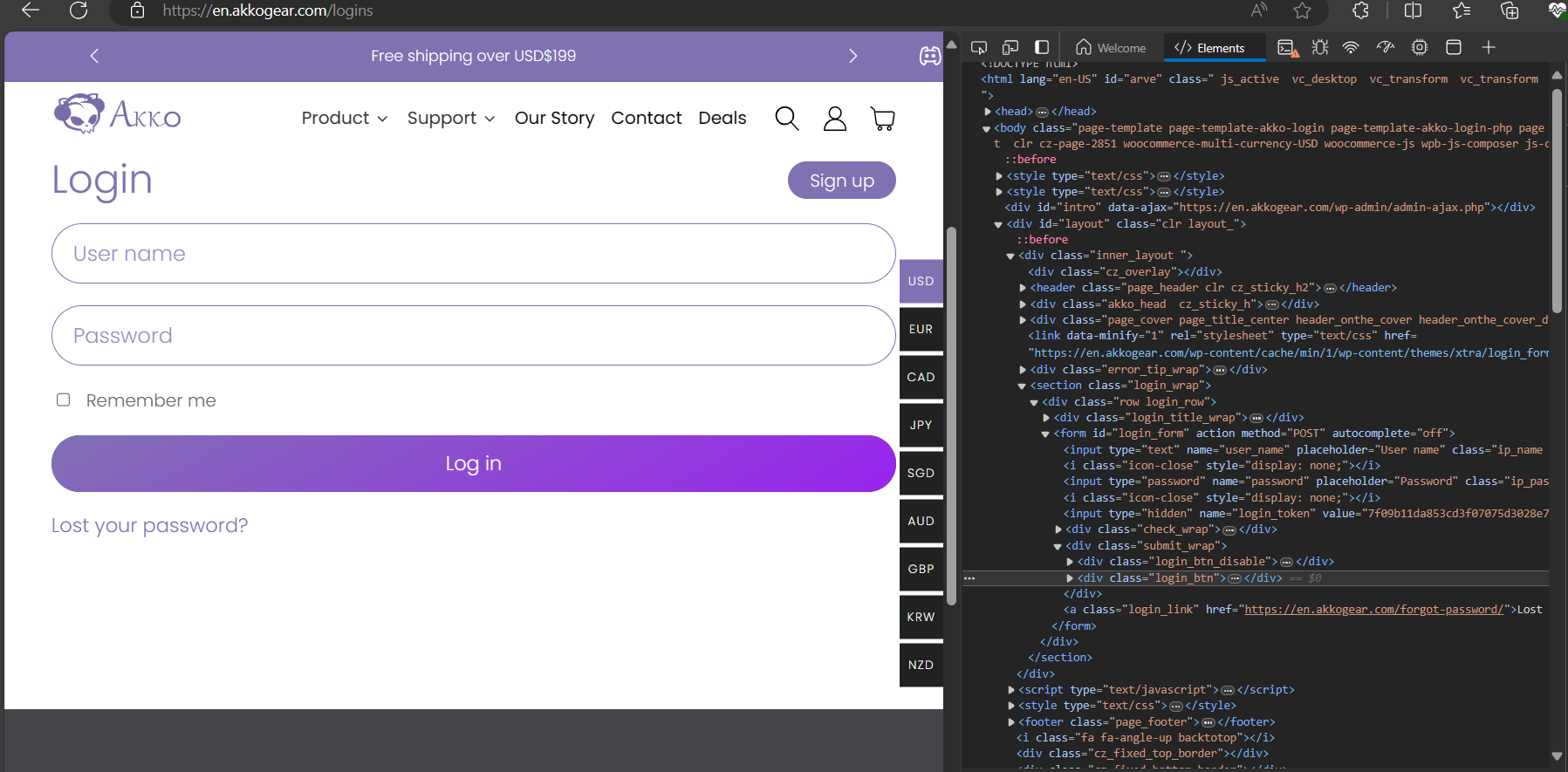
*Hình 16. lấy element của “ Đăng nhập”*

**Bước 4. Truy cập giao diện “Đăng nhập”**



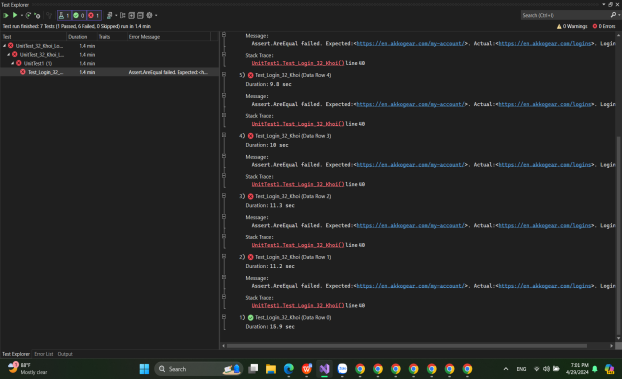
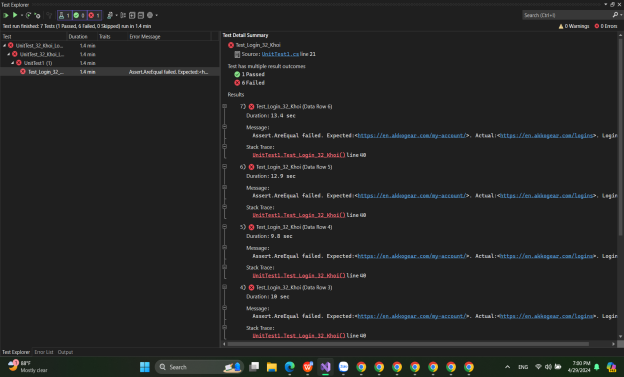
*Hình 17. giao diện khi nhấn vào “ Đăng nhập”*

**Bước 5. Lấy các Elenemts cần thiết**



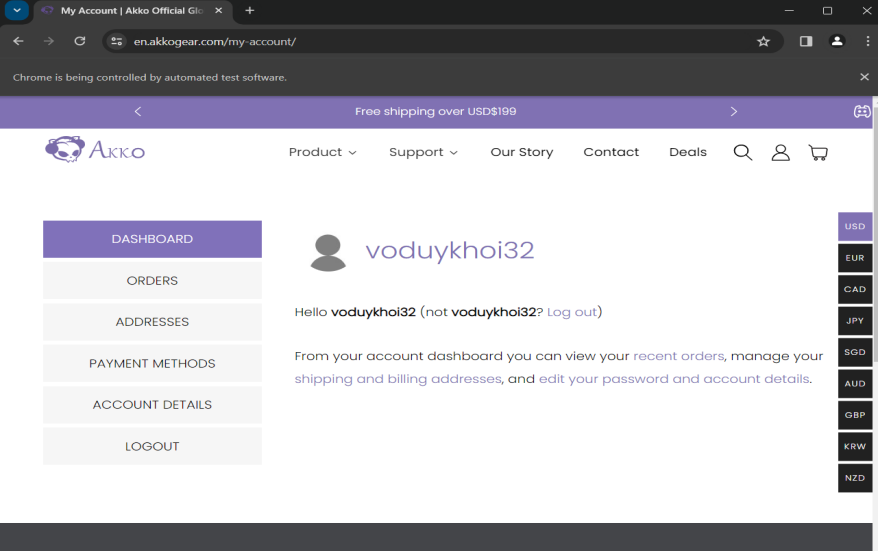
*Hình 18. giao diện lấy element của trang đăng nhập*

**1.3. Kết quả chạy TestCase**



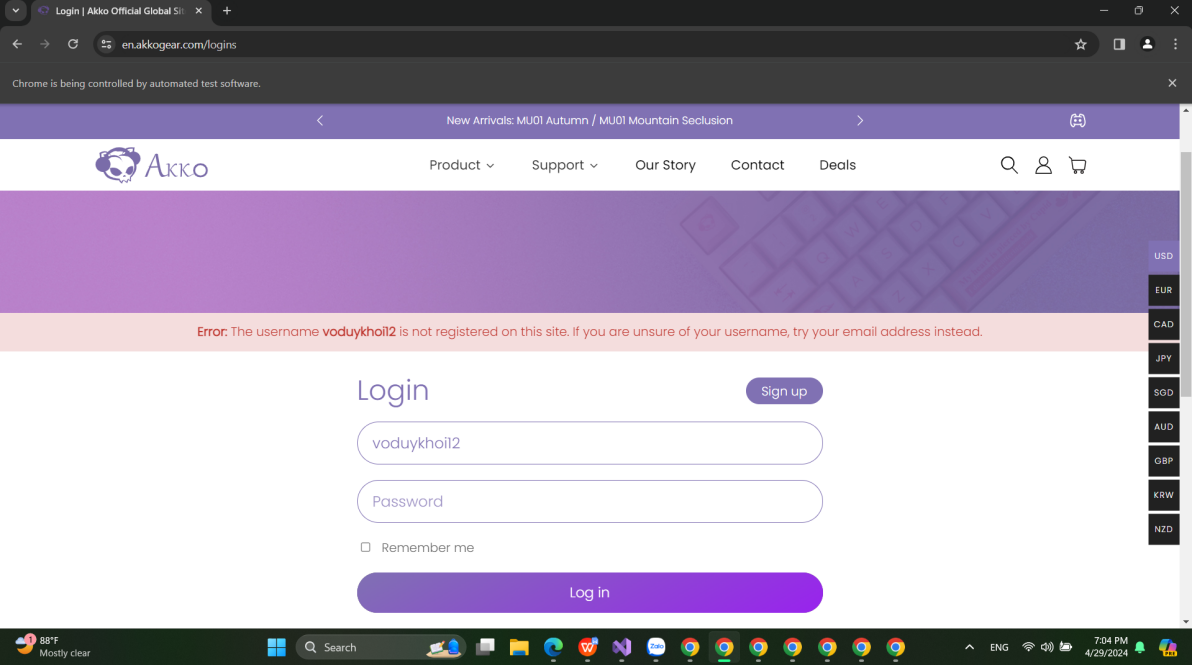
1 PASS - 6 FAIL

TestCase 1



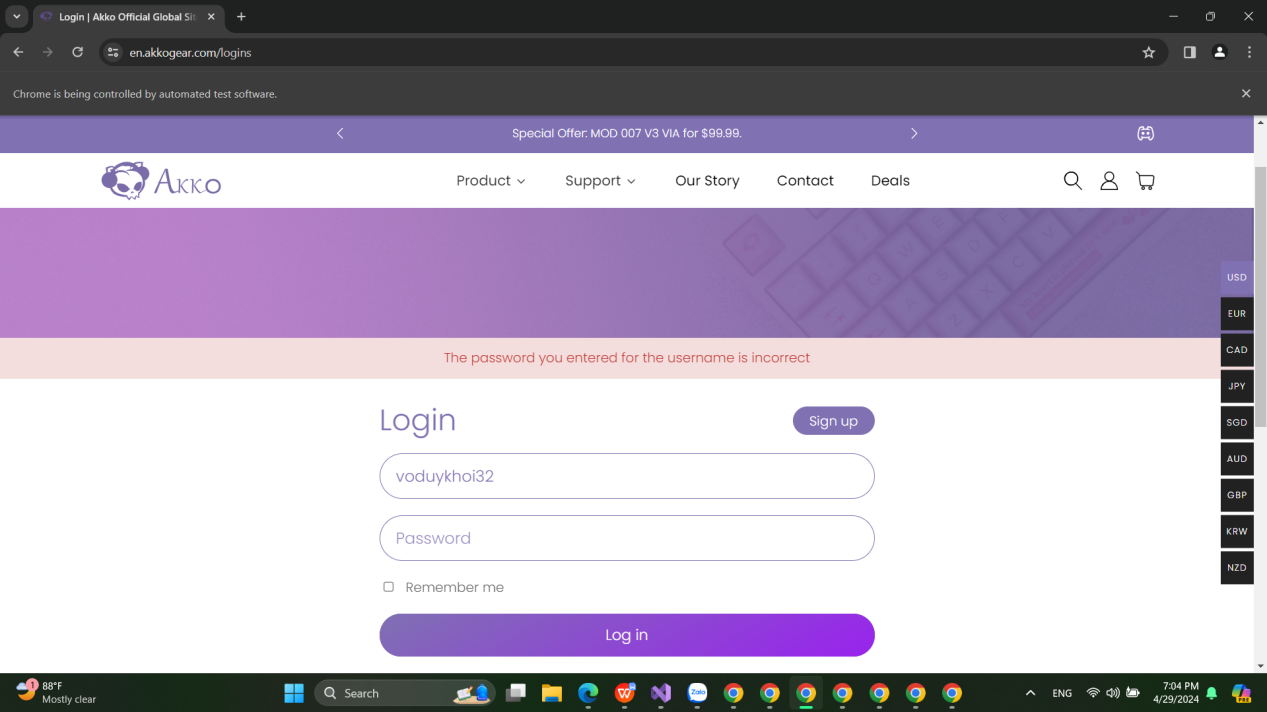
*Hình 19. TestCase Login 1 (đăng nhập thành công)*

TestCase 2



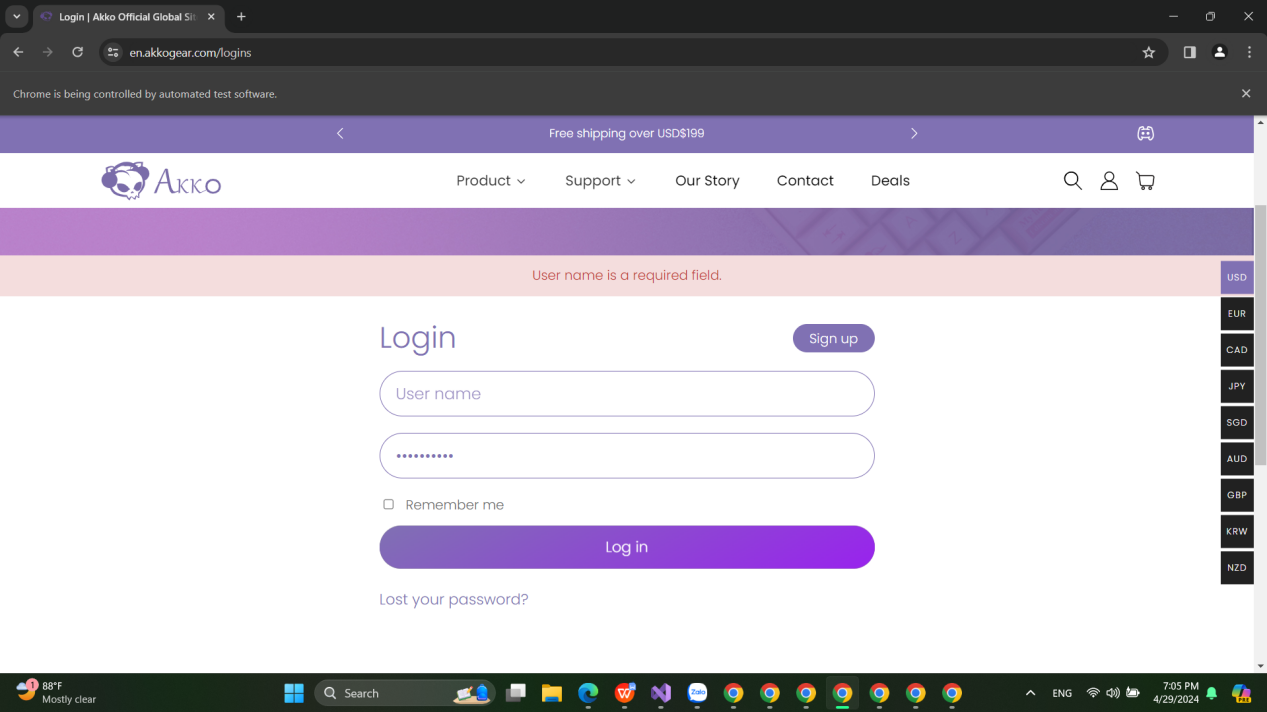
*Hình 20. TestCase Login 2 (đăng nhập sai tài khoản)*

TestCase 3



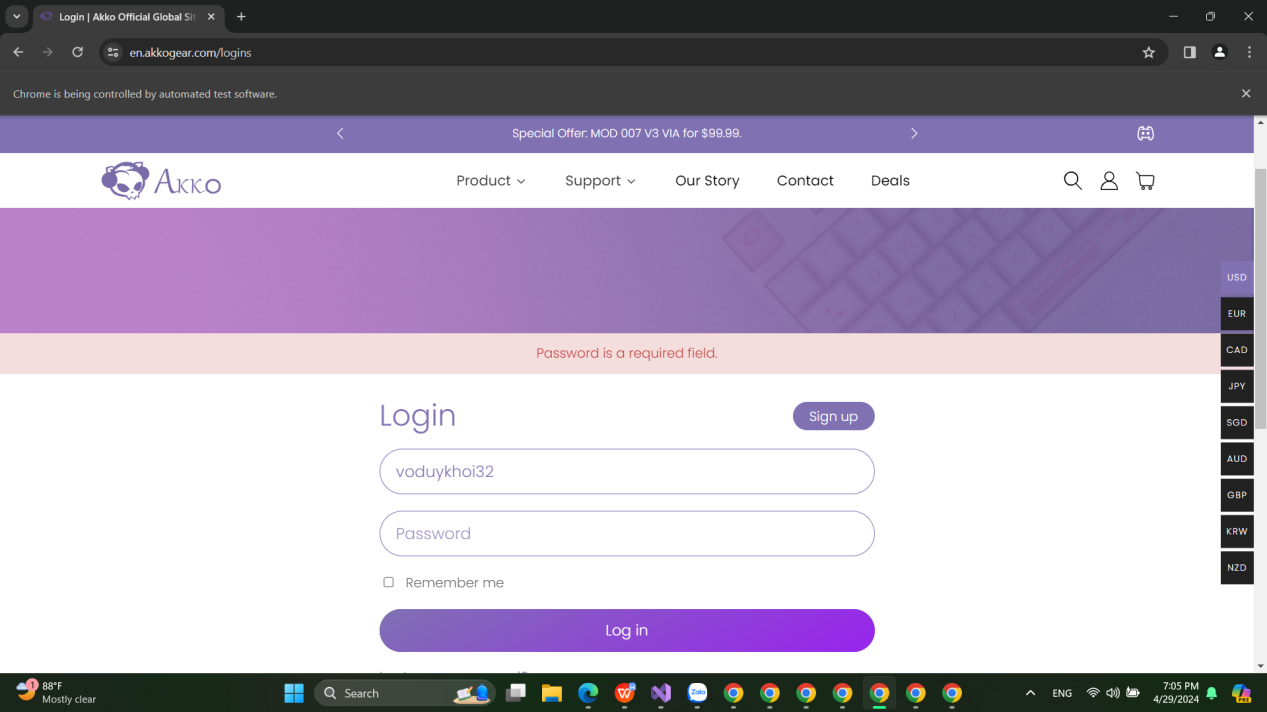
*Hình 21. TestCase Login 3 (đăng nhập sai mật khẩu)*

TestCase 4



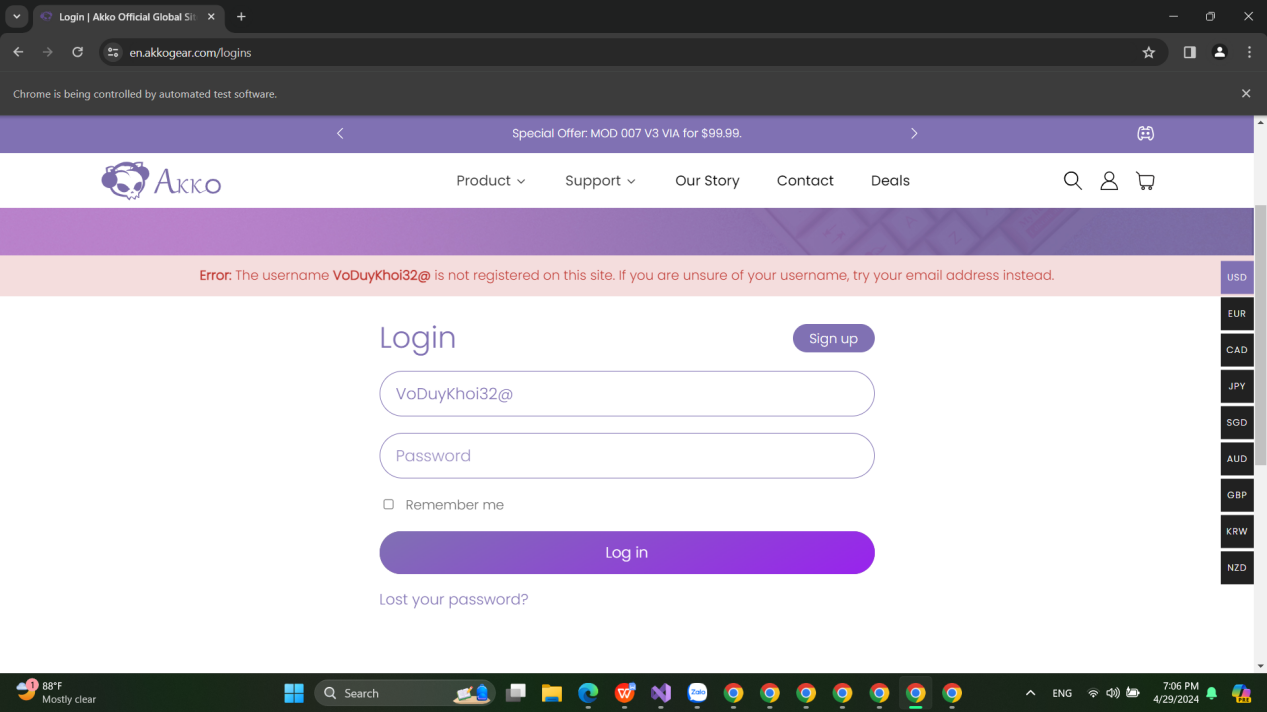
*Hình 22. TestCase Login 4 (đăng nhập thiếu tài khoản)*

TestCase 5



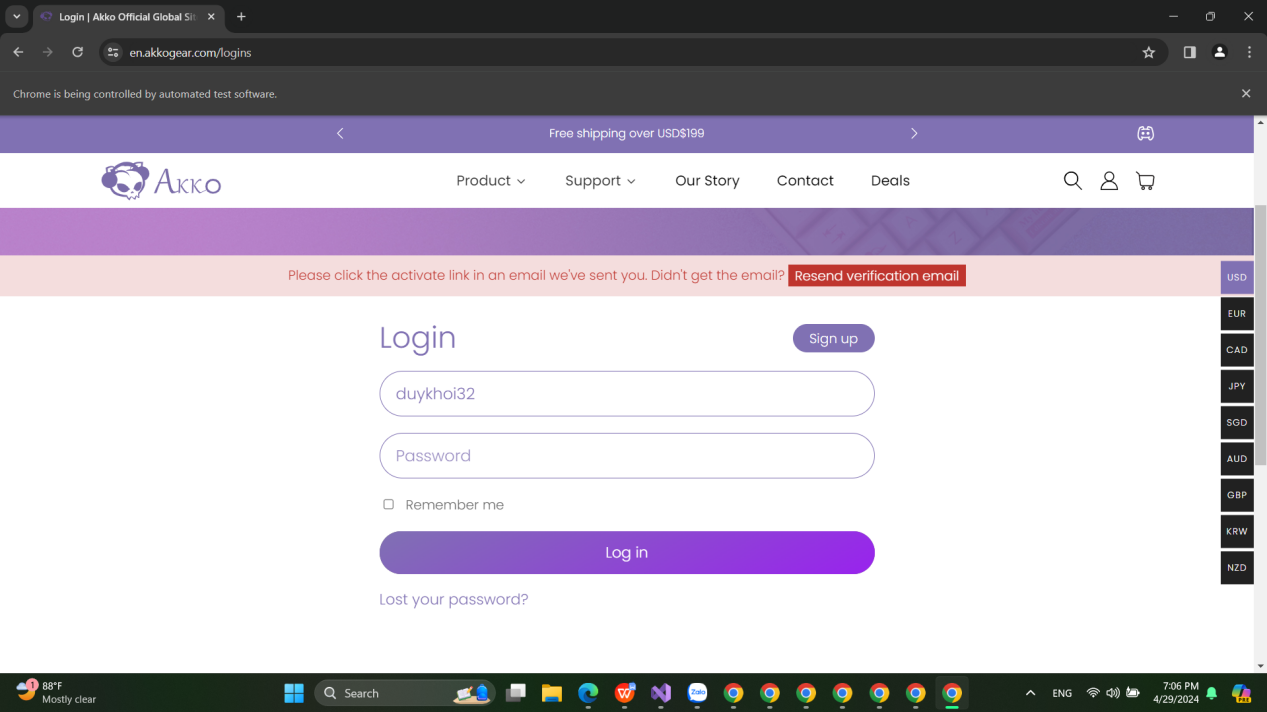
*Hình 23. TestCase Login 5 (đăng nhập thiếu mật khẩu)*

TestCase 6



*Hình 24. TestCase Login 6 (đăng nhập tài khoản có kí tự đặc biệt)*

TestCase 7



*Hình 25. TestCase Login 7 (đăng nhập tài khoản chưa được kích hoạt)*

**1.4. Code**

|  |
| --- |
| namespace UnitTest\_32\_Khoi\_Login  {  [TestClass]  public class UnitTest1  {  public TestContext TestContext { get; set; }  [DataSource("Microsoft.VisualStudio.TestTools.DataSource.CSV",  @".\data\LoginData.csv", "LoginData#csv", DataAccessMethod.Sequential)]  [TestMethod]  public void Test\_Login\_32\_Khoi()  {  /\*  Test case 1\_Đăng nhập thành công  Test case 2\_Đăng nhập sai tài khoản  Test case 3\_Đăng nhập sai mật khẩu  Test case 4\_Đăng nhập thiếu tài khoản  Test case 5\_Đăng nhập thiếu mật khẩu  Test case 6\_Đăng nhập với tài khoản có kí tự đặc biệt  Test case 7\_Đăng nhập với tài khoản được tạo nhưng chưa kích hoạt  \*/  //lưu dữ liệu ra biến  var username = TestContext.DataRow[0].ToString();  var password = TestContext.DataRow[1].ToString();  ChromeDriverService chrome\_khoi\_32 = ChromeDriverService.CreateDefaultService();  chrome\_khoi\_32.HideCommandPromptWindow = true;  IWebDriver driver = new ChromeDriver(chrome\_khoi\_32);  // Gọi trang web  driver.Navigate().GoToUrl("https://en.akkogear.com/");  Thread.Sleep(2000);  driver.FindElement(By.XPath("/html/body/div[3]/div/div[2]/div[2]/div/div[1]/div[2]/a")).Click();  driver.FindElement(By.Name("user\_name")).SendKeys(username);  driver.FindElement(By.CssSelector("input[type='password']")).SendKeys(password);  driver.FindElement(By.ClassName("login\_btn")).Click();  Assert.AreEqual("https://en.akkogear.com/my-account/", driver.Url, "Login Fail!");  }  }  } |

1. **Chức năng quản lý giỏ hàng**

| **TEST CASE** | **THUỘC TÍNH** | **DỮ LIỆU** | **Kết quả dự kiến** | **Kết quả thực tế** | **KẾT QUẢ** |
| --- | --- | --- | --- | --- | --- |
| 1 | Thêm sản phẩm vào giỏ hàng | Sản phẩm: Điện thoại A | Thêm sản phẩm thành công | Như dự kiến | PASS |
| 2 | Xóa sản phẩm khỏi giỏ hàng | Sản phẩm: Điện thoại B | Xóa sản phẩm thành công | Như dự kiến | PASS |
| 3 | Cập nhật số lượng sản phẩm | Sản phẩm: Điện thoại C | Số lượng được cập nhật đúng | Như dự kiến | PASS |
| 4 | Tính toán tổng giá trị đơn hàng | - | Hiển thị giá trị đơn hàng chính xác | Như dự kiến | PASS |
| 5 | Xóa toàn bộ giỏ hàng | - | Giỏ hàng trở thành trống, không có sản phẩm nào trong đó | Như dự kiến | PASS |
| 6 | Xem chi tiết đơn hàng trong giỏ | - | Hiển thị thông tin chi tiết đơn hàng đúng | Như dự kiến | PASS |

**2.1. Đặc tả chức năng**

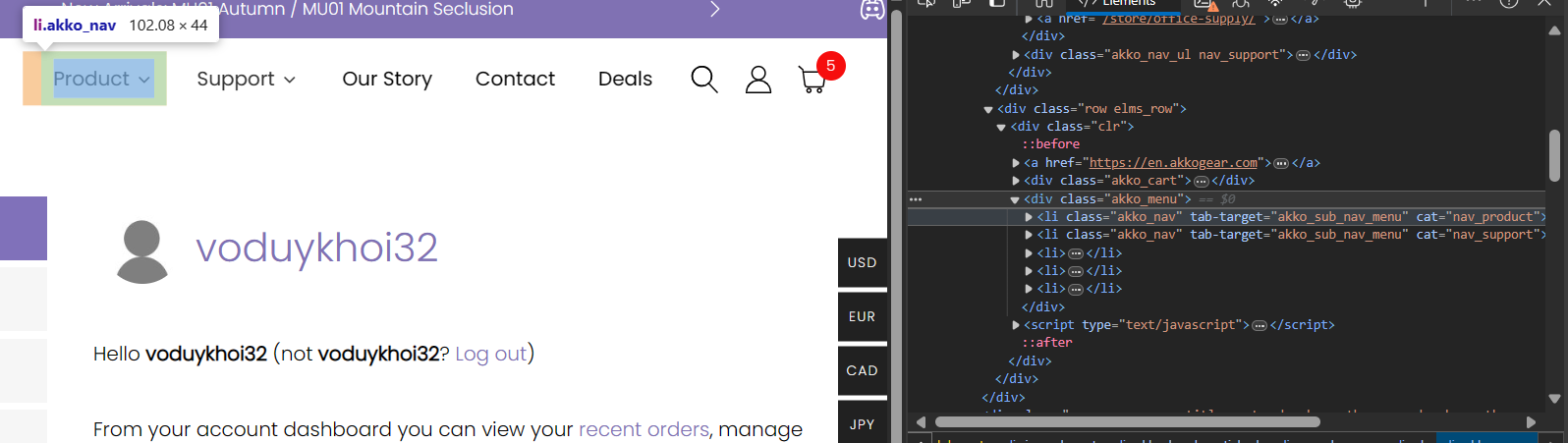
| **Tên chức năng** | Quản lý giỏ hàng |
| --- | --- |
| **Mô tả** | Chức năng Quản lý giỏ hàng cho phép người dùng tạo, xem, cập nhật và hủy đơn hàng trong hệ thống. |
| **Tác nhân** | Người dùng - người sử dụng hệ thống |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống và có quyền truy cập vào chức năng Quản lý giỏ hàng. |
| **Điều kiện sau** | Thông tin đơn hàng được người dùng cập nhật và hiển thị trong giỏ hàng ít nhất 1 sản phẩm. |
| **Ngoại lệ** |  |
| **Yêu cầu đặc biệt** |  |

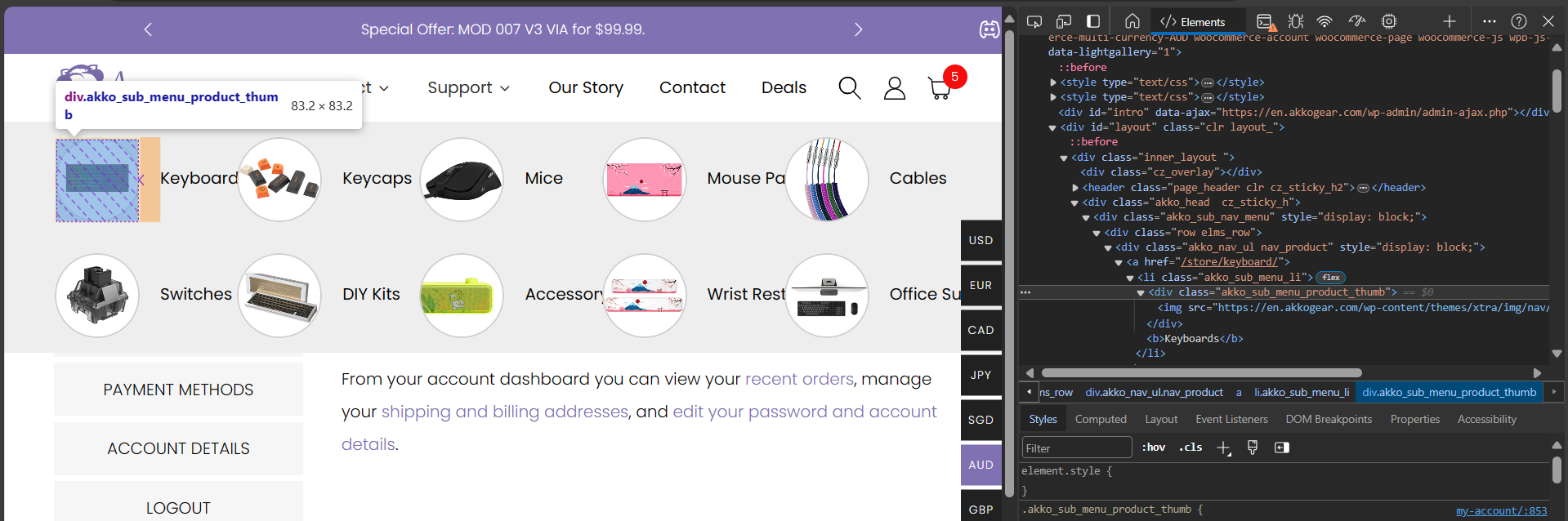
**2.2. Các bước Thực hiện**

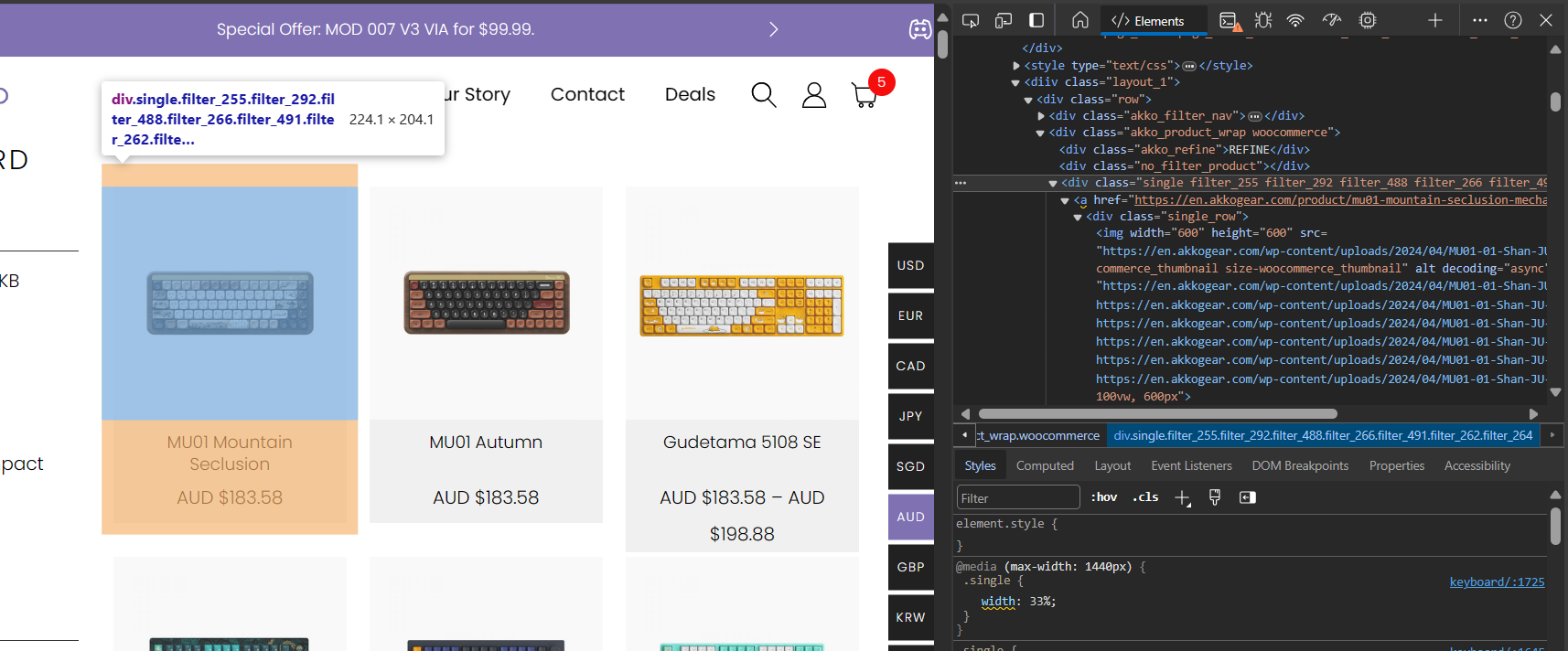
**Bước 1: Tạo project và cài đặt các**

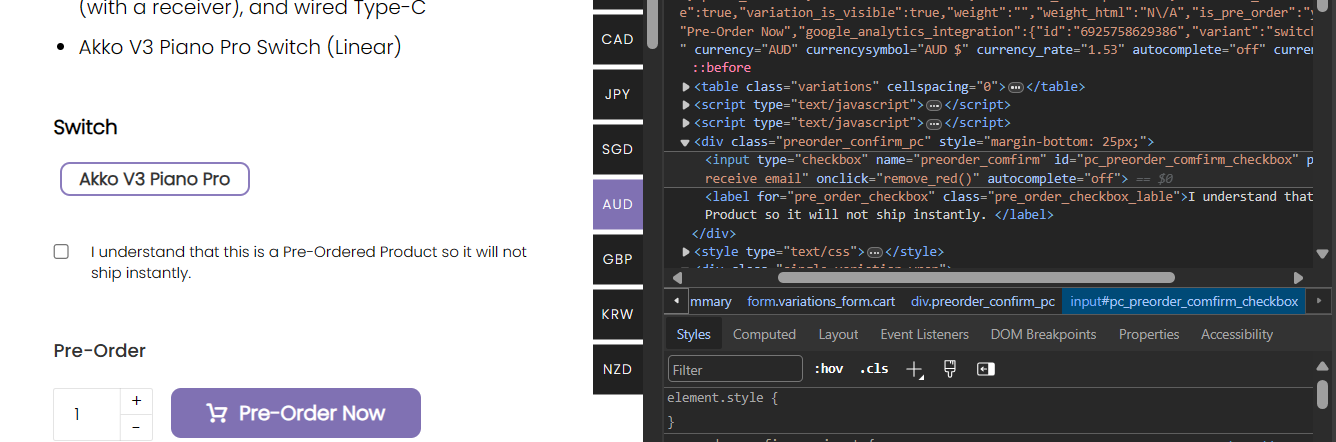
Tạo project mới đặt tên là **”UnitTest\_32\_Khoi\_QuanLyGioHang**” cài đặt webdriver cho project mới như hướng daanc ở trên

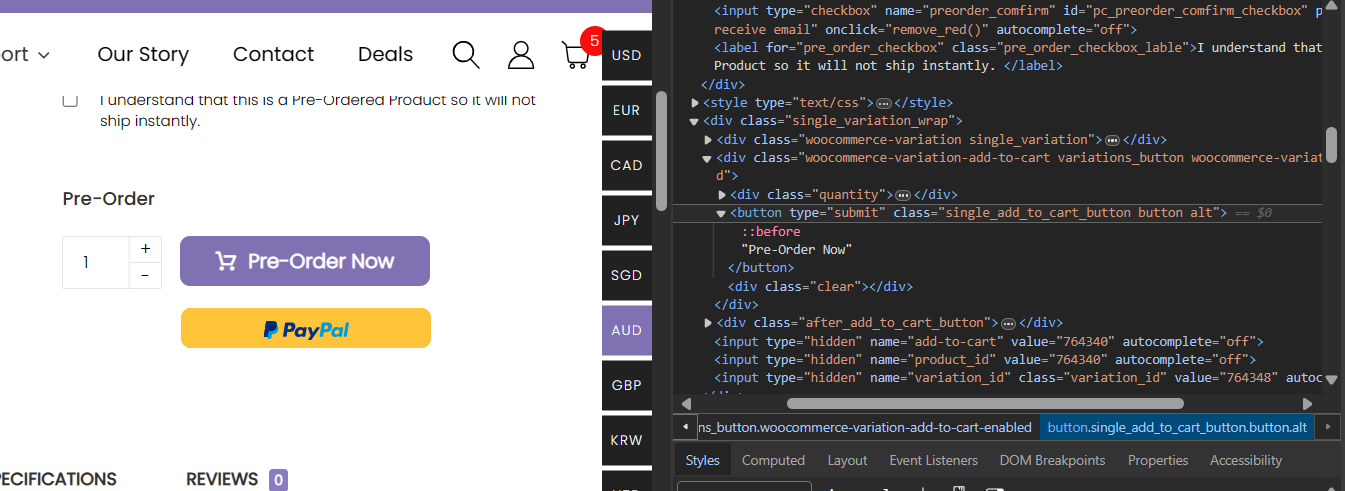
**Bước 2: Lấy các Elenemts hợp lý**

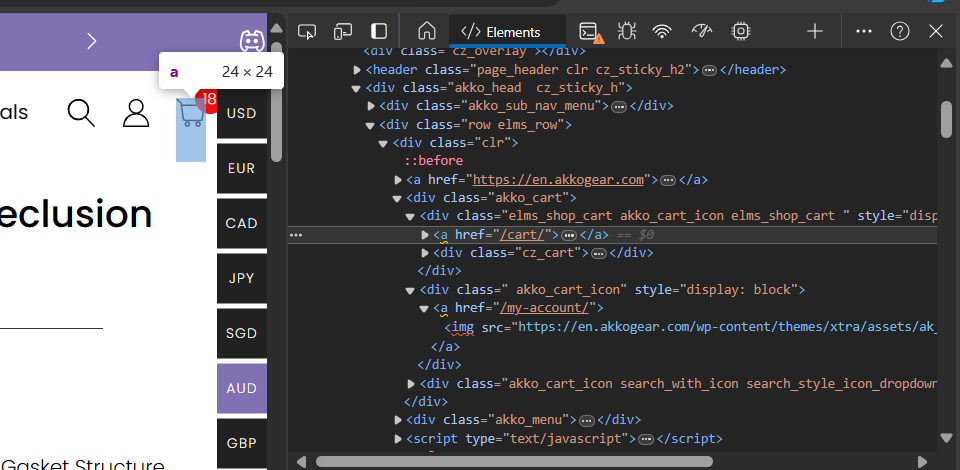


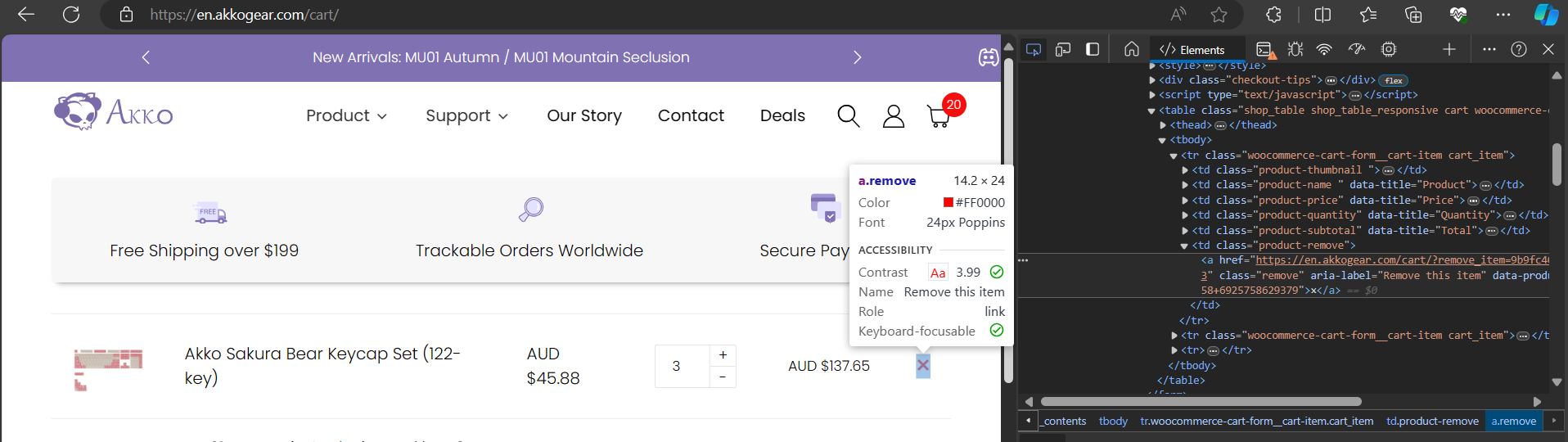


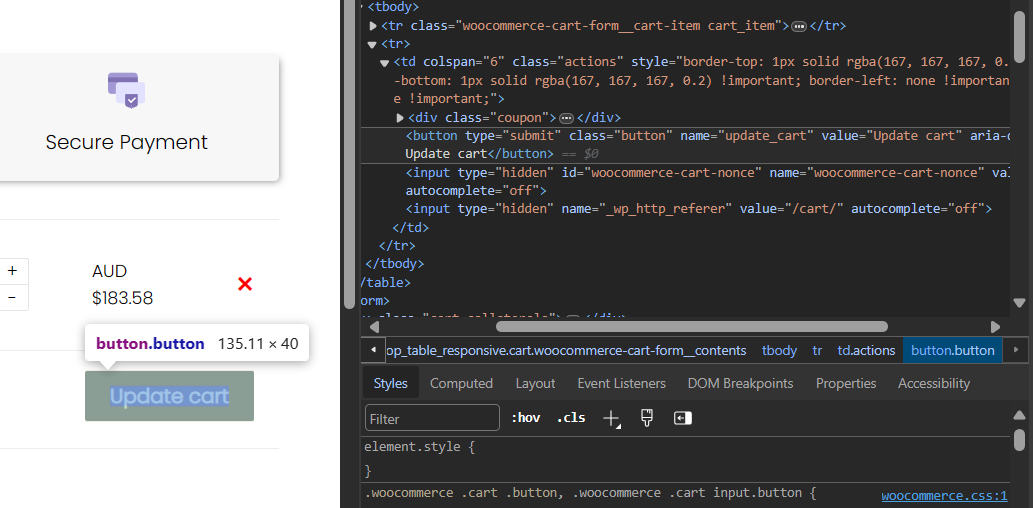


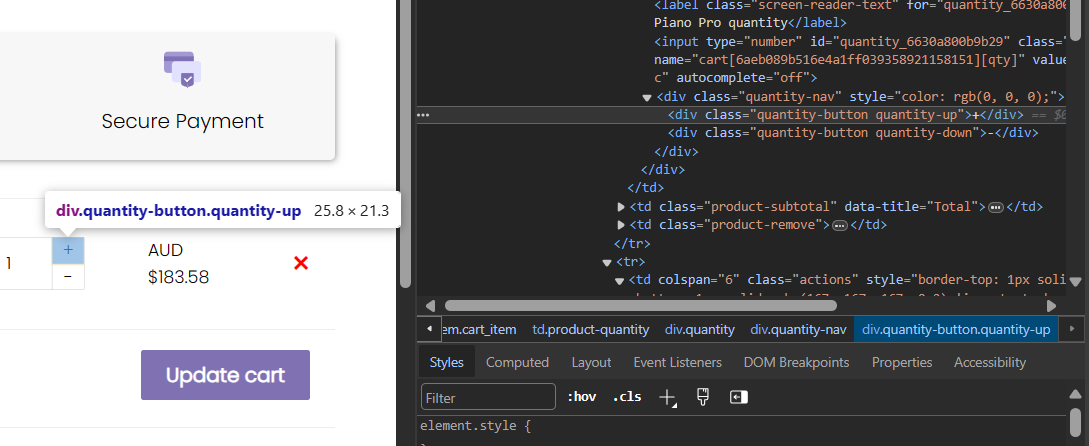












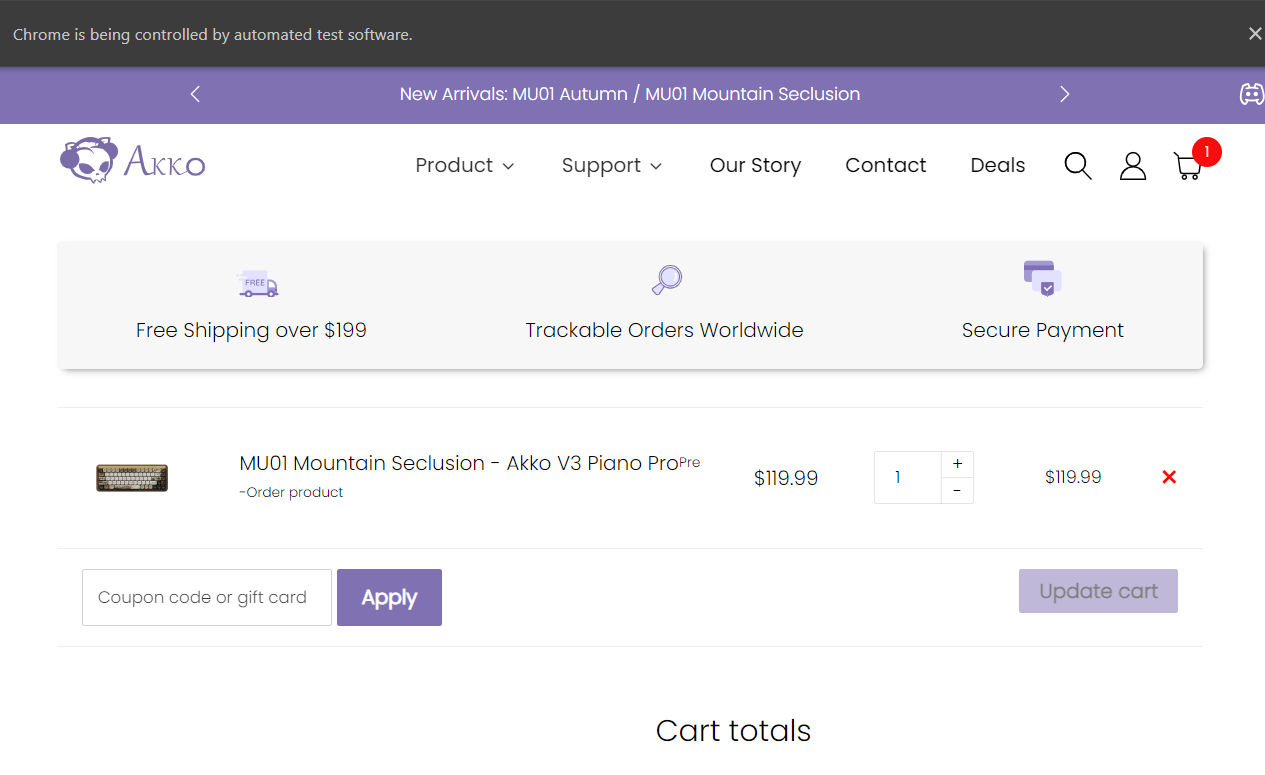
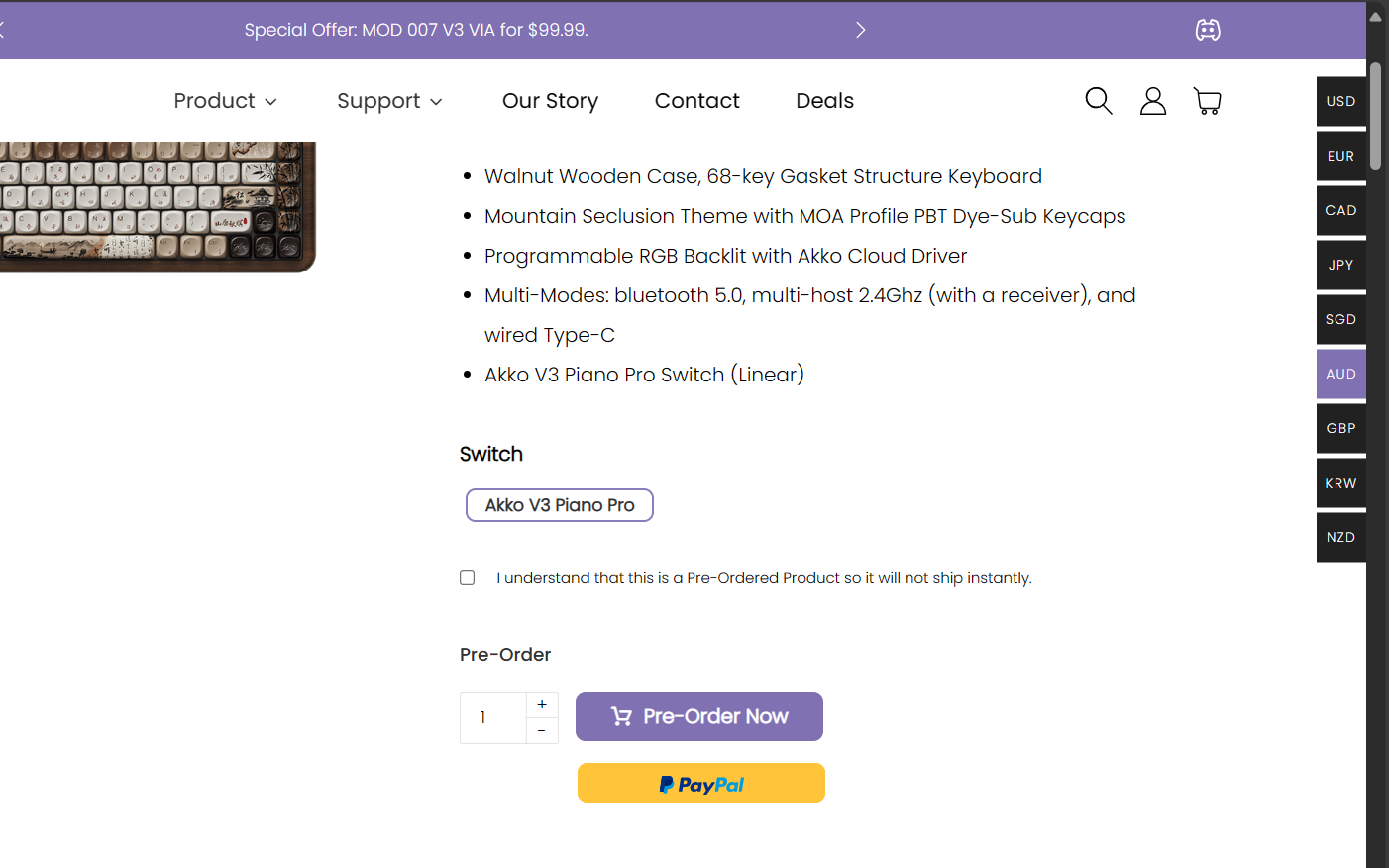
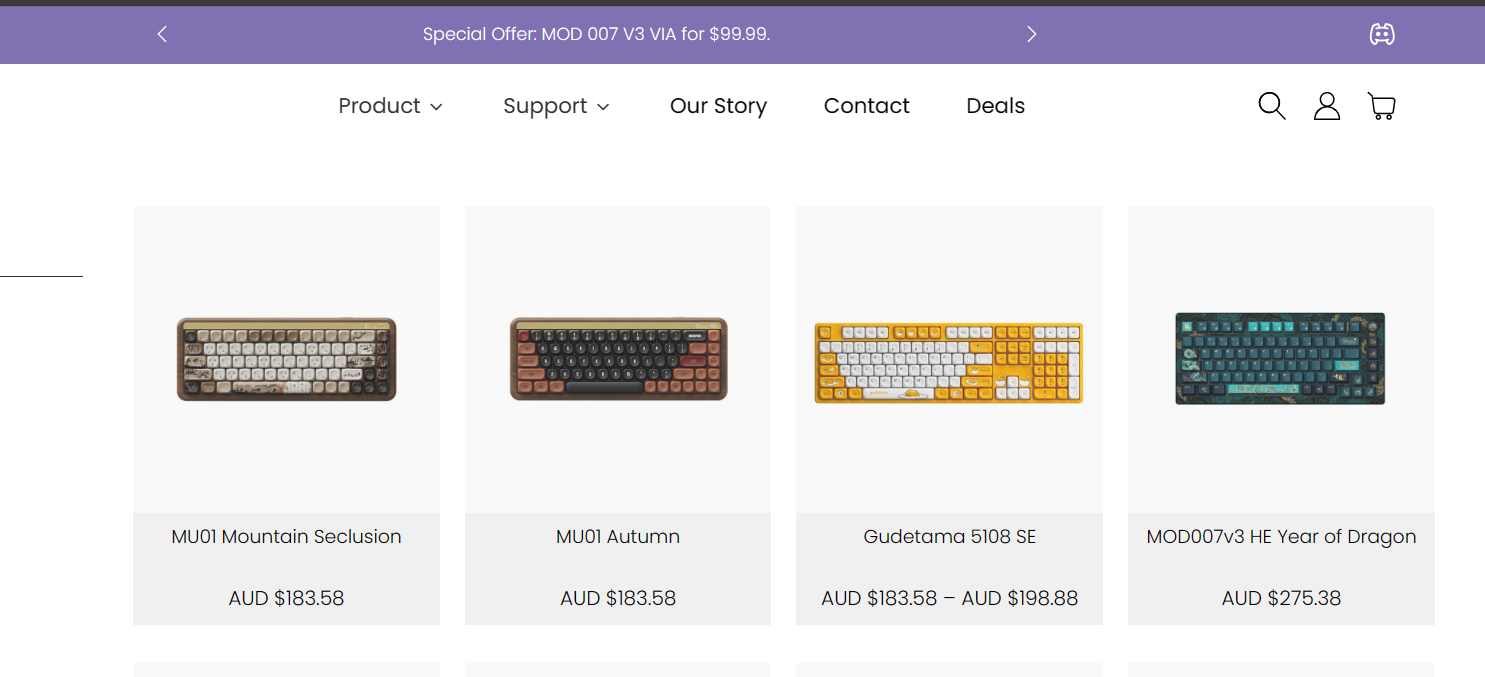
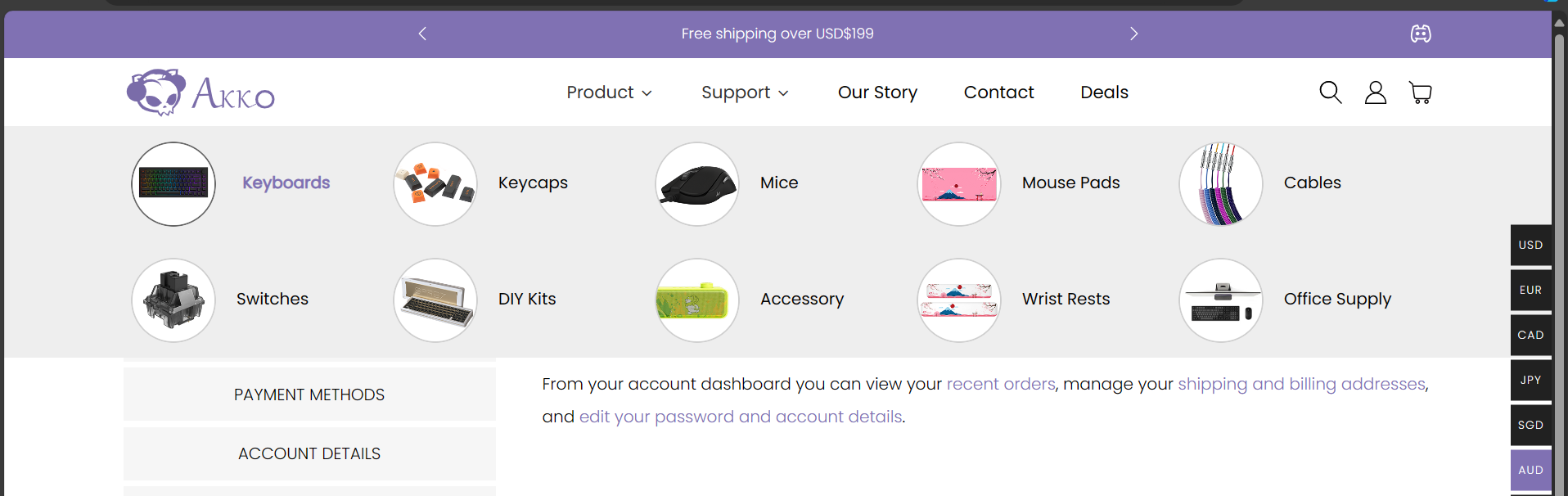
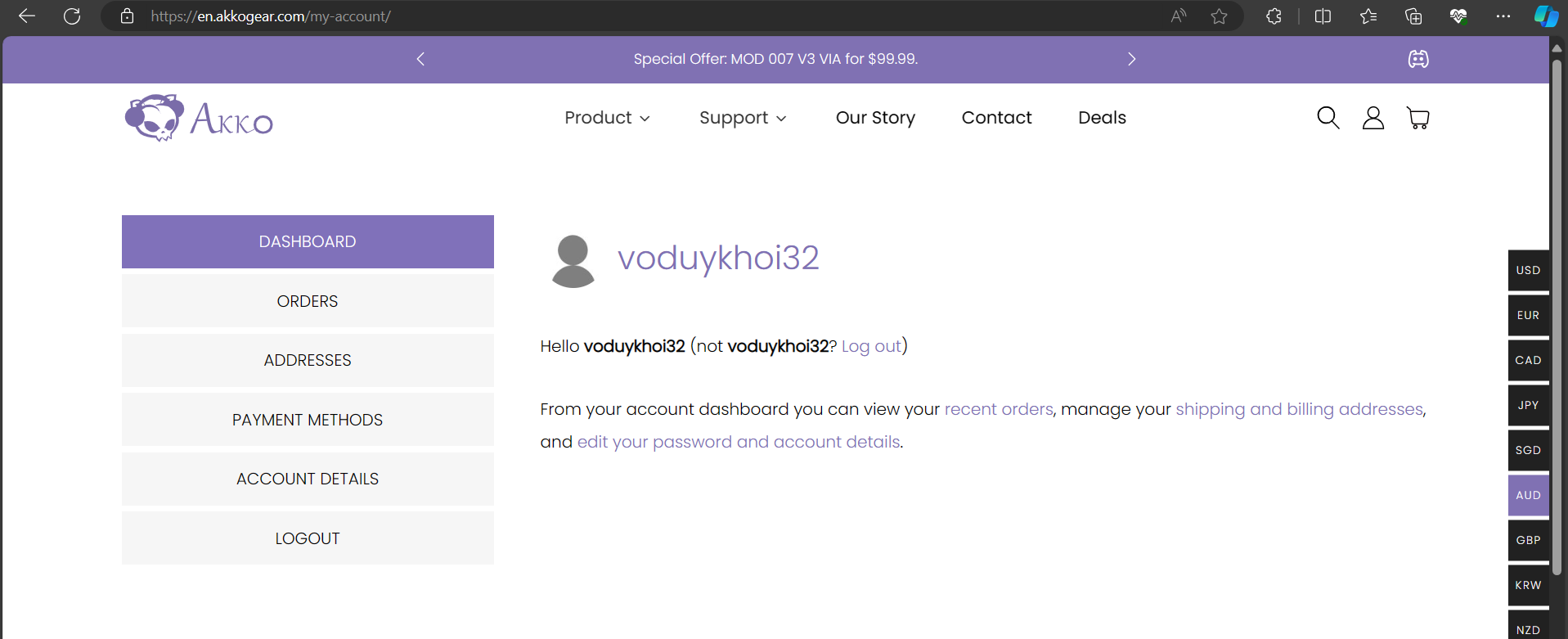


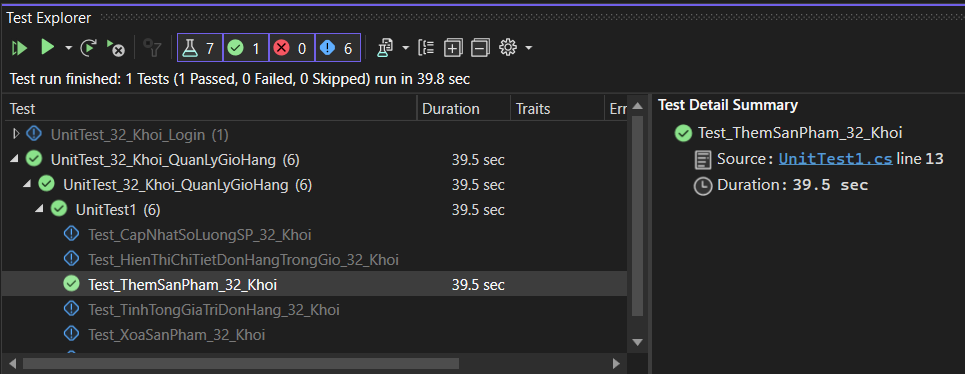
**2.3. kết quả chạy TestCase**

**[IMAGE]**

**TestCase 1 : Thêm Sản Phẩm**

|  |
| --- |
| /\*  chuẩn bị trong giỏ hàng không có sản phẩm(emty)  Thực hiện: thêm 1 sản phẩm và nhấp vào giỏ xem kết quả  kết quả mong muốn: sản phẩm mới dược thêm  \*/ |





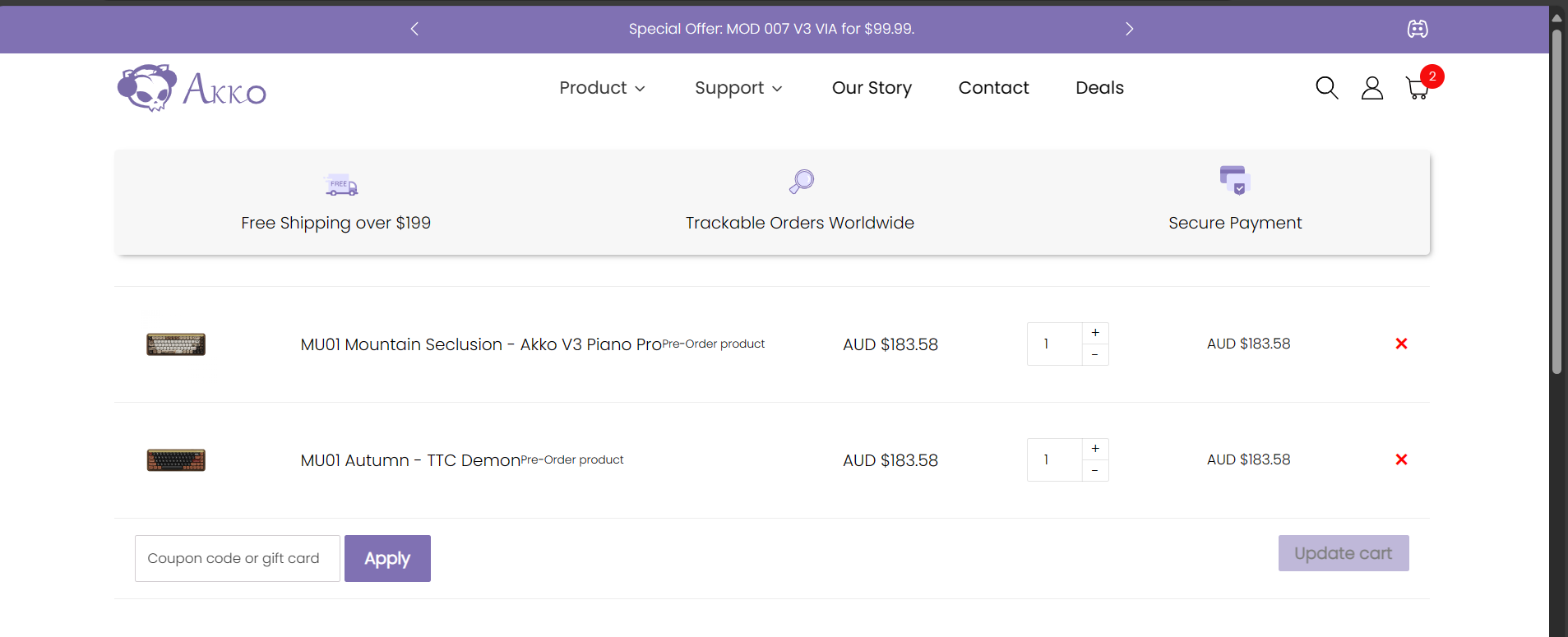
**Code**

|  |
| --- |
| [TestMethod]  public void Test\_ThemSanPham\_32\_Khoi()  {  //32\_Khoi  //tat man den  ChromeDriverService chrome\_khoi\_32 = ChromeDriverService.CreateDefaultService();  chrome\_khoi\_32.HideCommandPromptWindow = true;  IWebDriver driver = new ChromeDriver(chrome\_khoi\_32);  //gọi trang web  driver.Navigate().GoToUrl("https://en.akkogear.com/");  Thread.Sleep(2000);  //Đăng nhập vào trang web  driver.FindElement(By.XPath("/html/body/div[3]/div/div[2]/div[2]/div/div[1]/div[2]/a")).Click();  driver.FindElement(By.Name("user\_name")).SendKeys("voduykhoi32");  driver.FindElement(By.CssSelector("input[type='password']")).SendKeys("khoi123456");  driver.FindElement(By.ClassName("login\_btn")).Click();  Thread.Sleep(1000);  //Nhấn vào product trên thanh menu  driver.FindElement(By.CssSelector(".akko\_menu li:first-child")).Click();  //nhấn vào phân loại sản phẩm keyBoards  driver.FindElement(By.XPath("/html/body/div[2]/div/div[2]/div[1]/div/div[1]/a[1]/li/div")).Click();  Thread.Sleep(2000);  //nhấn vào sản phẩm đầu tiên  driver.FindElement(By.XPath("/html/body/div[2]/div/diiv/div/div[2]/div[3]")).Click();  Thread.Sleep(2000);  //nhấn vào check box đồng ý  driver.FindElement(By.Id("pc\_preorder\_comfirm\_checkbox")).Click();  // nhấn vào đặt hàng  driver.FindElement(By.CssSelector("button[type='submit'].single\_add\_to\_cart\_button.button.alt")).Click();  Thread.Sleep(2000);  //nhấn vào icon gioie hàng  driver.FindElement(By.CssSelector(".elms\_shop\_cart.akko\_cart\_icon.elms\_shop\_cart > a")).Click();  Thread.Sleep(2000);  //Gán giá trị cho actual và expected  IWebElement countCartAfter = driver.FindElement(By.XPath("/html/body/div[2]/div/div[2]/div[2]/div/div[1]/div[1]/div/span"));  int actual = int.Parse(countCartAfter.Text);  int expected = 1;  // Kiểm tra  /\*  chuẩn bị trong giỏ hàng không có sản phẩm(emty)  Thực hiện: thêm 1 sản phẩm và nhấp vào giỏ xem kết quả  kết quả mong muốn: sản phẩm mới dược thêm  \*/  Assert.AreEqual(expected, actual, "Failed to add product from cart.");  } |

**TestCase 2 : Xóa 1 sản phẩm**

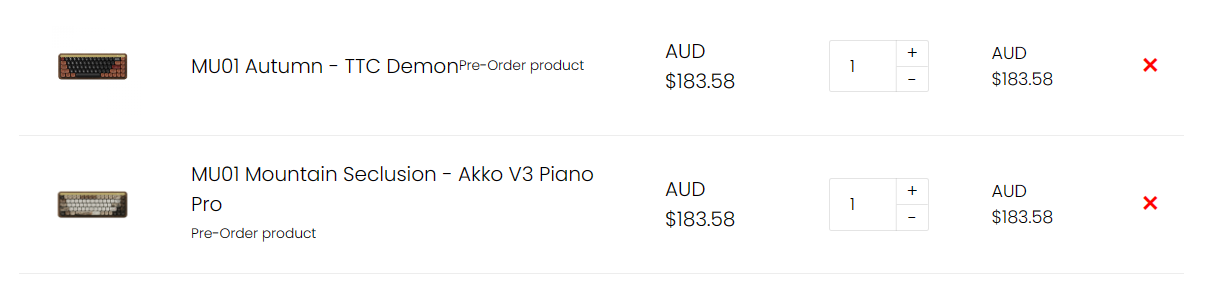
**Chuẩn bị:**

|  |
| --- |
| /\*  Chuẩn bị:  trong giỏ hàng ít nhất 2 sản phẩm khác nhau  sp1:1 - sp2:1  Thực hiện: xóa 1 sp;  kết quả:Số lượng sp:1  \*/ |

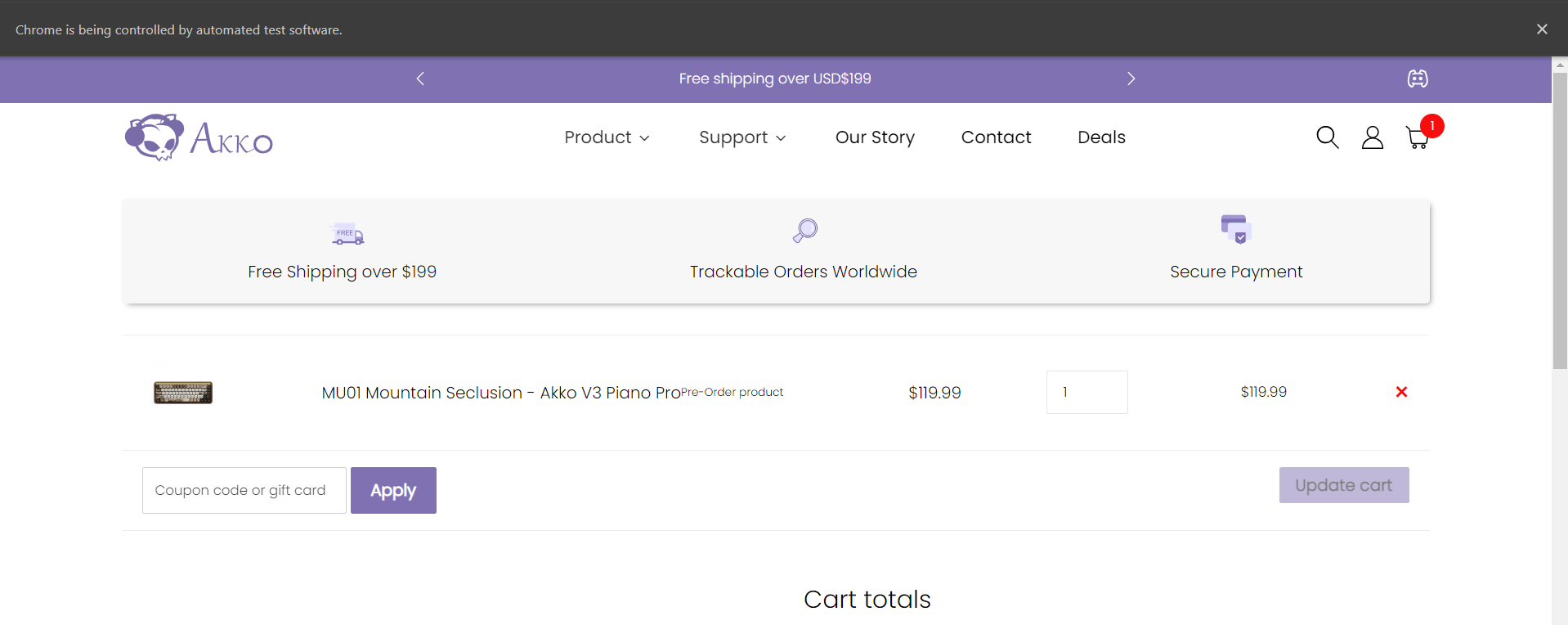


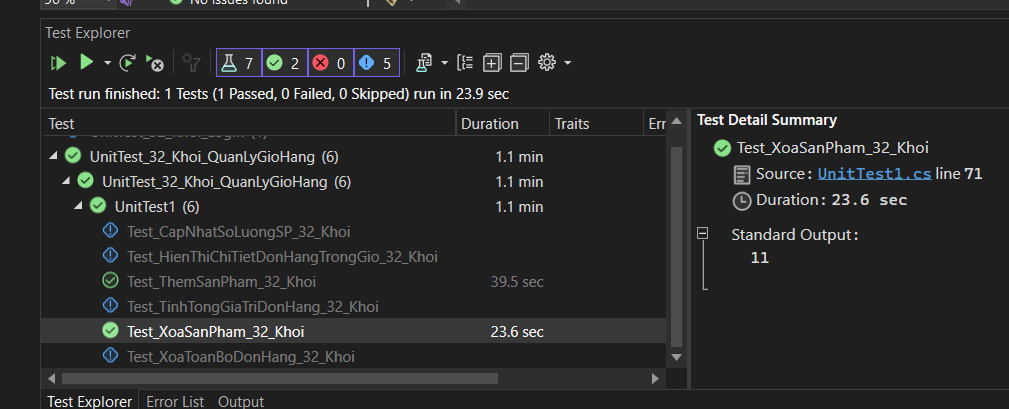
**Thực Hiện**

Đăng Nhập -> nhấp vào Icon Giỏ hàng -> chọn dấu X ở mỗi sản phẩn để xóa



**Kết Quả:**





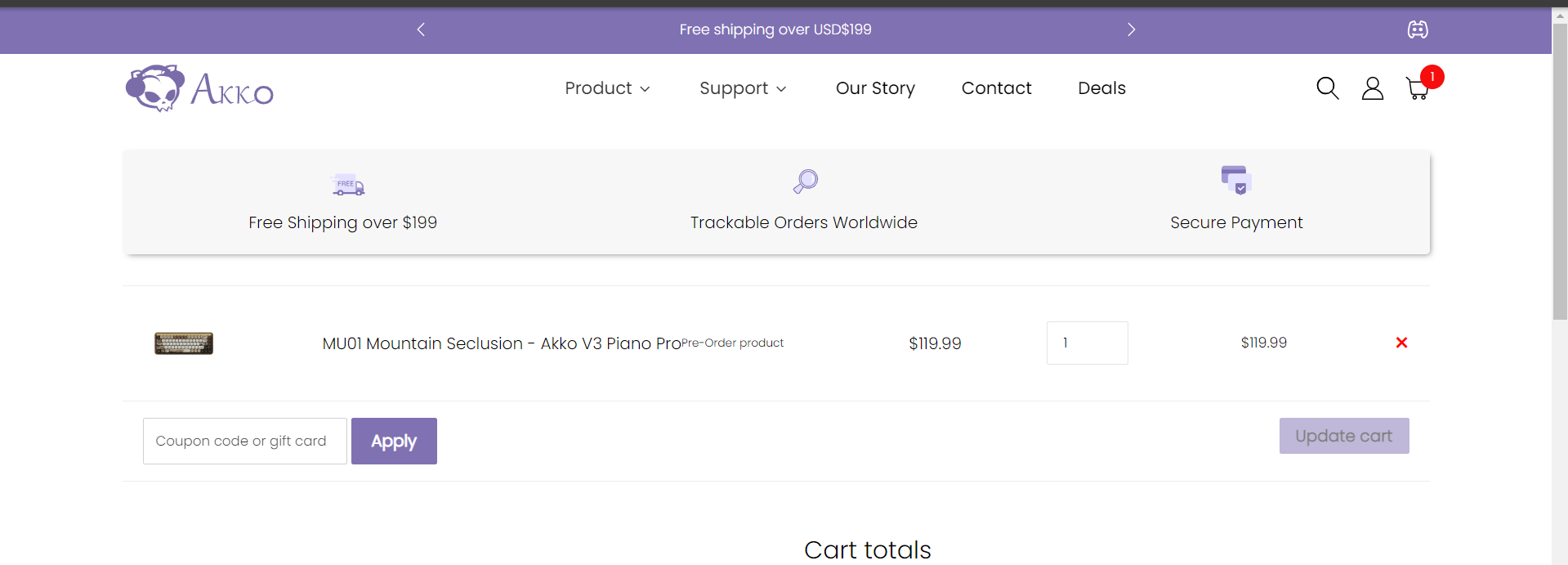
Code:

|  |
| --- |
| [TestMethod]  public void Test\_XoaSanPham\_32\_Khoi()  {  //32\_Khoi  //tat man den  ChromeDriverService chrome\_khoi\_32 = ChromeDriverService.CreateDefaultService();  chrome\_khoi\_32.HideCommandPromptWindow = true;  IWebDriver driver = new ChromeDriver(chrome\_khoi\_32);  //gọi trang web  driver.Navigate().GoToUrl("https://en.akkogear.com/");  Thread.Sleep(2000);  //Đăng nhập vào trang web  driver.FindElement(By.XPath("/html/body/div[3]/div/div[2]/div[2]/div/div[1]/div[2]/a")).Click();  driver.FindElement(By.Name("user\_name")).SendKeys("voduykhoi32");  driver.FindElement(By.CssSelector("input[type='password']")).SendKeys("khoi123456");  driver.FindElement(By.ClassName("login\_btn")).Click();  Thread.Sleep(1000);  //tiến hành test giỏ hàng  //nhấn vào icon giỏ hàng  driver.FindElement(By.CssSelector(".elms\_shop\_cart.akko\_cart\_icon.elms\_shop\_cart > a")).Click();  Thread.Sleep(2000);  //xóa dơn hàng  driver.FindElement(By.LinkText("×")).Click();  Thread.Sleep(3000);  //Gán giá trị tổng sản phẩm  IWebElement countCartAfter = driver.FindElement(By.XPath("/html/body/div[2]/div/div[2]/div[2]/div/div[1]/div[1]/div/span"));  int actual = int.Parse(countCartAfter.Text);  int expected = 1;  Console.WriteLine(actual + " and " + expected);  /\*  Chuẩn bị:  trong giỏ hàng ít nhất 2 sản phẩm khác nhau  sp1:1  sp2:1  Thực hiện: xóa 1 sp;  kết quả:Số lượng sp:1  \*/  Assert.AreEqual(expected, actual, "Failed to remove product from cart.");  } |

**TestCase 3: Cập nhật số lượng sản phẩm**

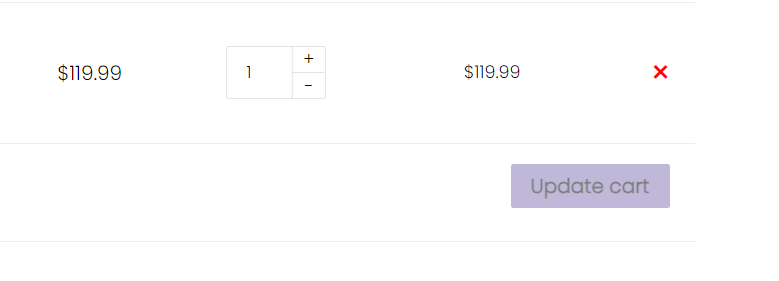
**Chuẩn bị:**

|  |
| --- |
| /\*  Kiểm tra  chuẩn bị trong giỏ hàng có 1 sp số lượng = 1  tổng sp ban đầu=1;  kết quả:  tổng sp sau khi thêm :2  \*/ |

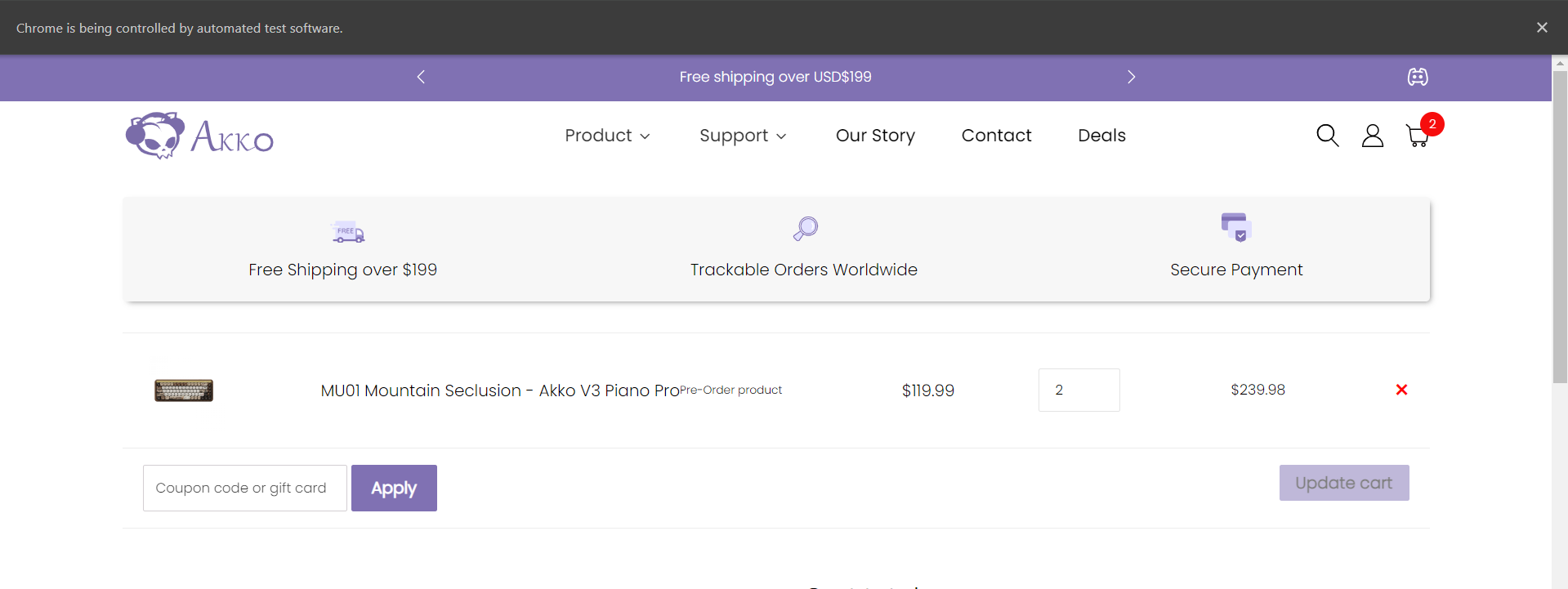


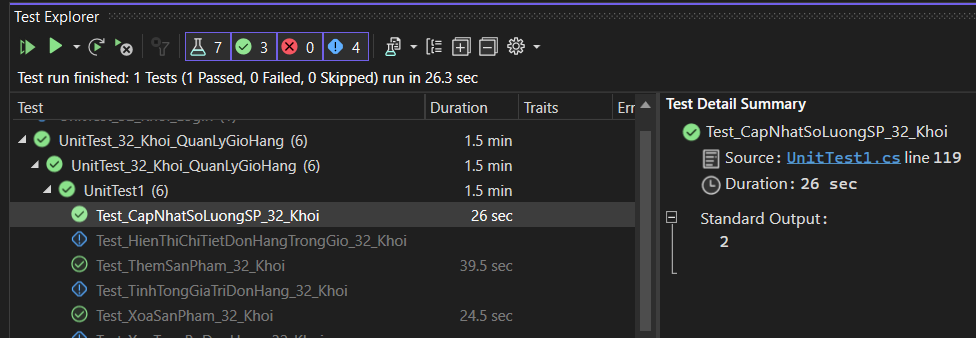
**Thực hiện**

Đăng Nhập -> nhấp vào Icon Giỏ hàng -> chọn dấu button “+” ở sản phẩn để thêm số lượng và sau đó nhấn vào button “Update Cart”



**Kết quả**





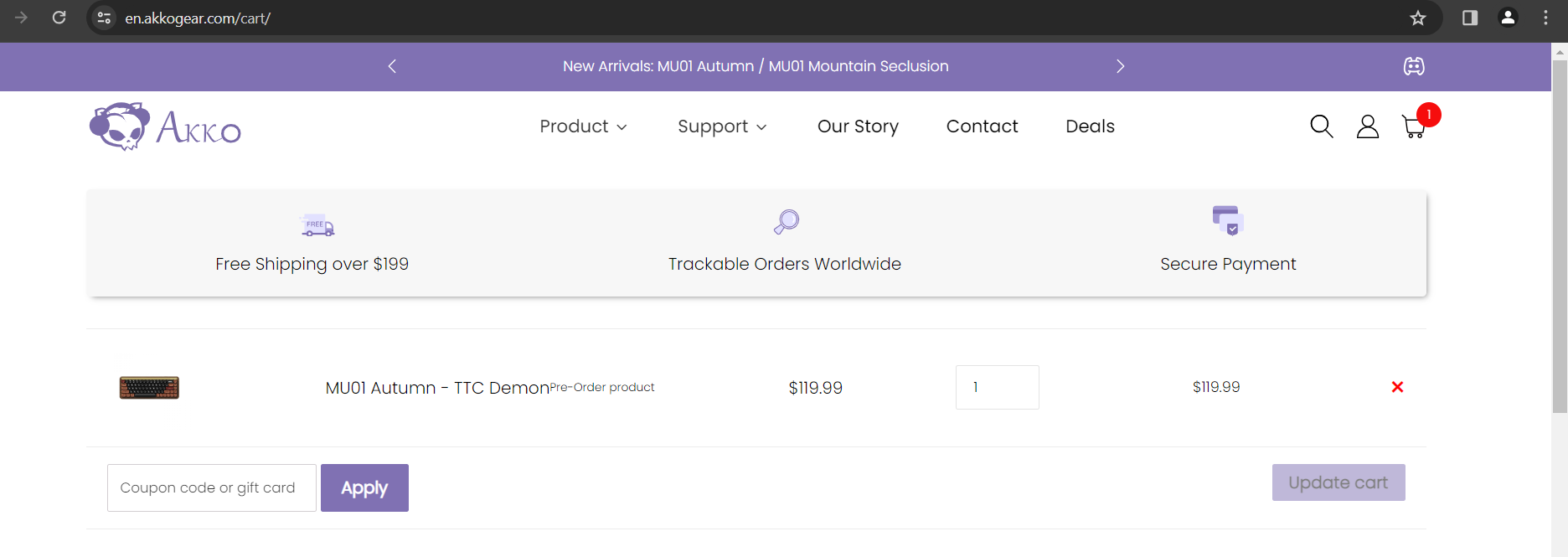
**Code**

|  |
| --- |
| [TestMethod]  public void Test\_CapNhatSoLuongSP\_32\_Khoi()  {  //32\_Khoi  //tat man den  ChromeDriverService chrome\_khoi\_32 = ChromeDriverService.CreateDefaultService();  chrome\_khoi\_32.HideCommandPromptWindow = true;  IWebDriver driver = new ChromeDriver(chrome\_khoi\_32);  //gọi trang web  driver.Navigate().GoToUrl("https://en.akkogear.com/");  Thread.Sleep(2000);  //Đăng nhập vào trang web  driver.FindElement(By.XPath("/html/body/div[3]/div/div[2]/div[2]/div/div[1]/div[2]/a")).Click();  driver.FindElement(By.Name("user\_name")).SendKeys("voduykhoi32");  driver.FindElement(By.CssSelector("input[type='password']")).SendKeys("khoi123456");  driver.FindElement(By.ClassName("login\_btn")).Click();  Thread.Sleep(1000);  //nhấn vào icon giỏ hàng  driver.FindElement(By.CssSelector(".elms\_shop\_cart.akko\_cart\_icon.elms\_shop\_cart > a")).Click();  Thread.Sleep(2000);  //nhấn vào + tăng số lượng sản phẩm  driver.FindElement(By.CssSelector(".quantity-button.quantity-up")).Click();  Thread.Sleep(2000);  //nhấn vào button "UpdateCart"  driver.FindElement(By.Name("update\_cart")).Click();  Thread.Sleep(3000);  //Gán giá trị tổng sản phẩm  IWebElement countCartAfter = driver.FindElement(By.XPath("/html/body/div[2]/div/div[2]/div[2]/div/div[1]/div[1]/div/span"));  int actual = int.Parse(countCartAfter.Text);  Console.WriteLine(actual);  int expected = 2;  /\*  Kiểm tra  chuẩn bị trong giỏ hàng có 1 sp số lượng = 1  tổng sp ban đầu=1;  kết quả:  tổng sp sau khi thêm :2  \*/  Assert.AreEqual(expected, actual, "Failed to update product Count from cart.");  } |

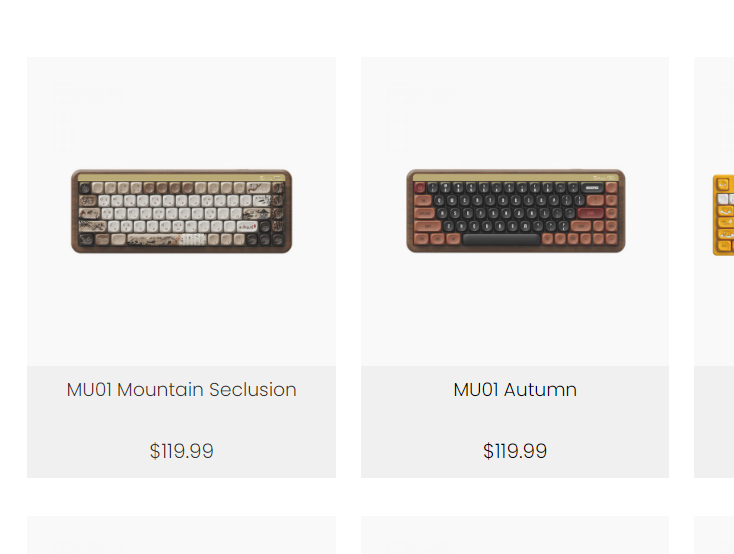
TestCase 4: Tính tổng giá trị đơn hàng

Chuẩn bị

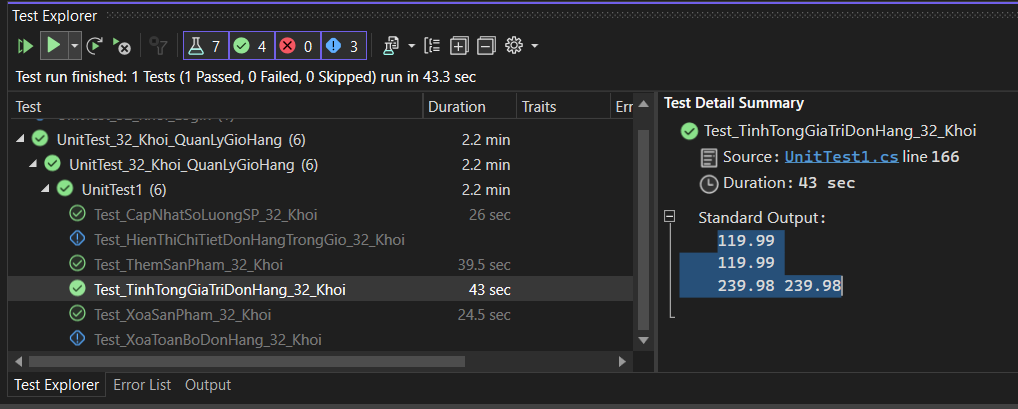
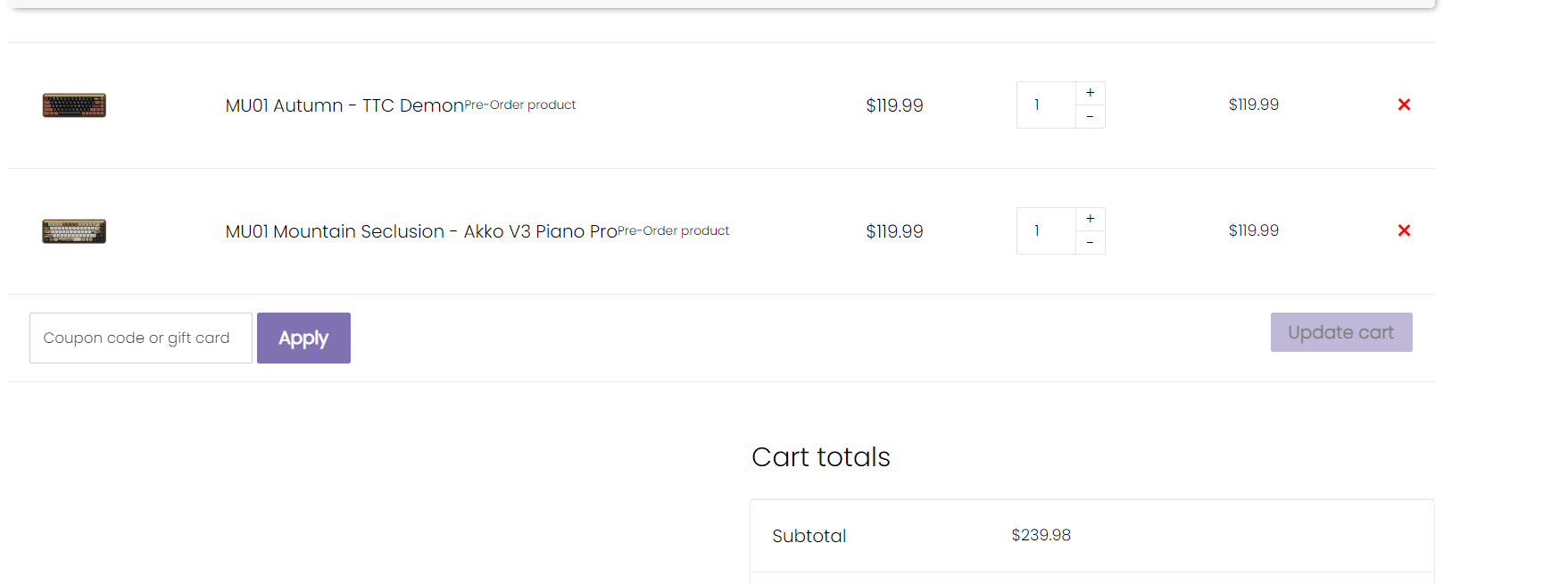
|  |
| --- |
| //Kiểm tra kết quả  /\*  chuẩn bị:  MU01 Autumn - so luongl: 1 - GiaTien:$119.99  thực hiện:  người dùng thêm sp: MU01 Mountain Seclusion - so luong: 1 - GiaTien:$119.99  kết quả dự kiến:239.98  \*/ |



Thực hiện:



Kết quả:



Code:

|  |
| --- |
| [TestMethod]  public void Test\_TinhTongGiaTriDonHang\_32\_Khoi()  {  //32\_Khoi  //tat man den  ChromeDriverService chrome\_khoi\_32 = ChromeDriverService.CreateDefaultService();  chrome\_khoi\_32.HideCommandPromptWindow = true;  IWebDriver driver = new ChromeDriver(chrome\_khoi\_32);  //gọi trang web  driver.Navigate().GoToUrl("https://en.akkogear.com/");  Thread.Sleep(2000);  //Đăng nhập vào trang web  driver.FindElement(By.XPath("/html/body/div[3]/div/div[2]/div[2]/div/div[1]/div[2]/a")).Click();  driver.FindElement(By.Name("user\_name")).SendKeys("voduykhoi32");  driver.FindElement(By.CssSelector("input[type='password']")).SendKeys("khoi123456");  driver.FindElement(By.ClassName("login\_btn")).Click();  Thread.Sleep(1000);  //Nhấn vào product trên thanh menu  driver.FindElement(By.CssSelector(".akko\_menu li:first-child")).Click();  //nhấn vào phân loại sản phẩm keyBoards  driver.FindElement(By.XPath("/html/body/div[2]/div/div[2]/div[1]/div/div[1]/a[1]/li/div")).Click();  Thread.Sleep(2000);  //nhấn vào sản phẩm đầu tiên  driver.FindElement(By.XPath("/html/body/div[2]/div/diiv/div/div[2]/div[3]")).Click();  Thread.Sleep(2000);  //nhấn vào check box đồng ý  driver.FindElement(By.Id("pc\_preorder\_comfirm\_checkbox")).Click();  // nhấn vào đặt hàng  driver.FindElement(By.CssSelector("button[type='submit'].single\_add\_to\_cart\_button.button.alt")).Click();  Thread.Sleep(2000);  //nhấn vào icon gioie hàng  driver.FindElement(By.CssSelector(".elms\_shop\_cart.akko\_cart\_icon.elms\_shop\_cart > a")).Click();  Thread.Sleep(2000);  // Lấy danh sách các phần tử có class "product-subtotal"  var subtotalElements = driver.FindElements(By.ClassName("product-subtotal"));  //gán tổng tiền = 0  float total = 0;  // Duyệt qua từng phần tử và tính tổng giá tiền  foreach (var element in subtotalElements)  {  string priceText = element.Text;  int startIndex = priceText.IndexOf('$') + 1;  string numberPart = priceText.Substring(startIndex).Trim();  if (float.TryParse(numberPart, out float countItem))  {  Console.WriteLine(countItem);  total += countItem;  }  }    float expected = 239.98f;  float actual = total;  //Kiểm tra kết quả  /\*  chuẩn bị:  MU01 Autumn - so luongl: 1 - GiaTien:$119.99  thực hiện:  người dùng thêm sp: MU01 Mountain Seclusion - so luong: 1 - GiaTien:$119.99  kết quả dự kiến:239.98  \*/  Console.WriteLine(expected+ " " + total);  Assert.AreEqual(expected, actual, "toal price FAIL");  } |