**SCDT61 CW2- Jacque Scriven**

Contents:

[Testing: 3](#_Toc99092460)

[Issue Fixes: 16](#_Toc99092461)

[Additional Feature: 18](#_Toc99092462)

[Publication Platform: 20](#_Toc99092463)

[Evaluation of testing, verification and validation methods: 20](#_Toc99092464)

[References: 22](#_Toc99092465)

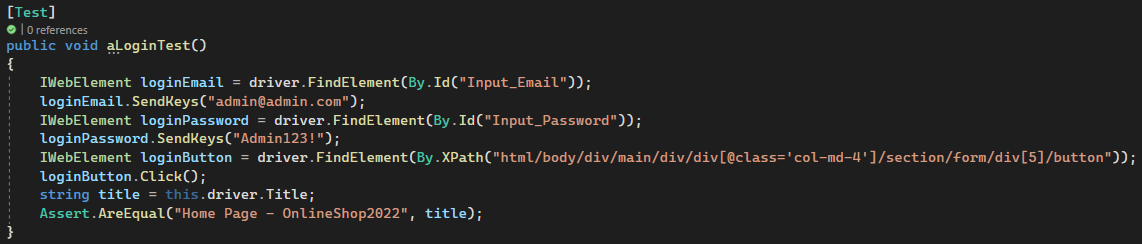
# Testing:

Id- 1

Name- LoginTest

Description- Tests if the user is able to login. The test is done by logging in with the admin account.

Code-



Outcome-



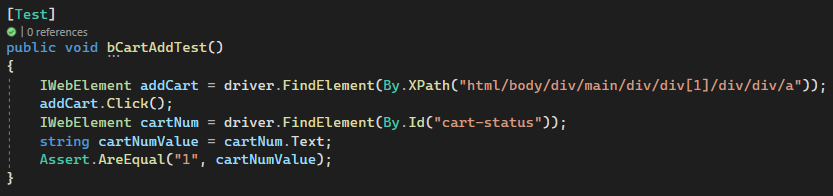
The test was successful.

Id-2

Name- CartAddTest

Description- Adds a product to the part and checks to see if the cart value has been changed to 1.

Code-



Outcome-



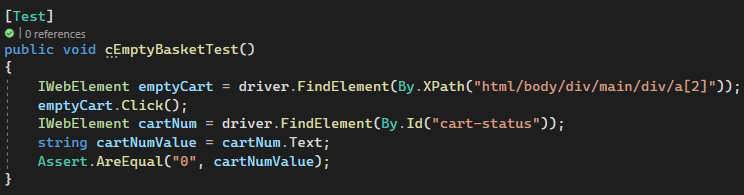
The test was a success.

Id- 3

Name- EmptyBasketTest

Description- Clicks on the empty cart button and checks to see if the cart value has been changed to 0.

Code-



Outcome-



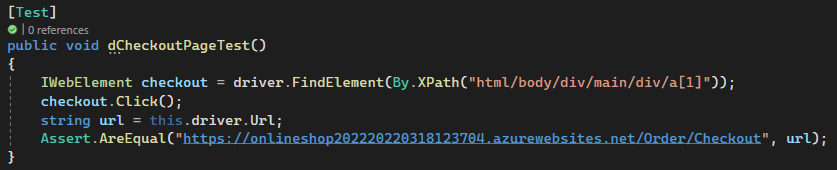
The test was successful.

Id- 4

Name- CheckoutPageTest

Description- Navigates to the checkout page and checks to see if the url matches the checkout page’s url.

Code-



Outcome-



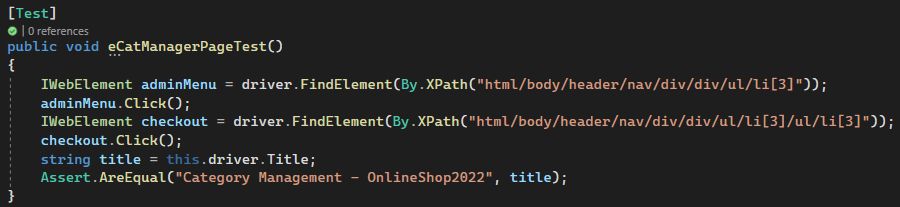
The test was successful.

Id- 5

Name- CatManagerPageTest

Description- Navigates to the category management page and checks to see if the title matches that of the category management page.

Code-



Outcome-



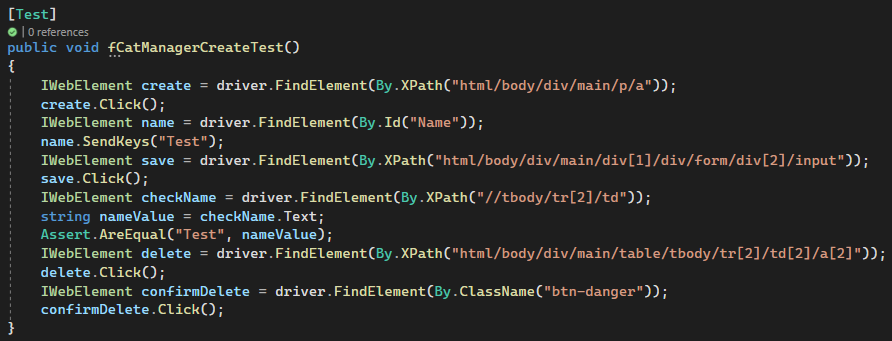
The test was successful.

Id- 6

Name- CatManagerCreateTest

Description- Creates a new category called tests and confirms that it has been successfully created and then deletes the new category.

Code-



Outcome-



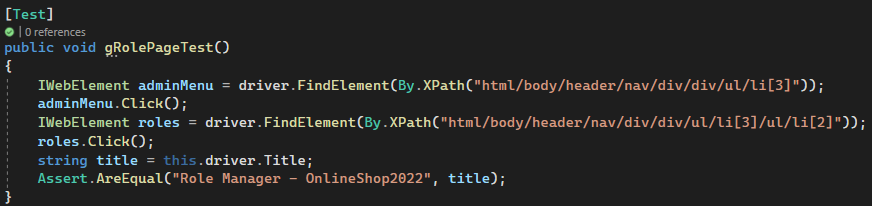
The test was successful.

Id- 7

Name- RolePageTest

Description- Navigates to the role management page and tests that the page name is equal to that of the role management page.

Code-



Outcome-



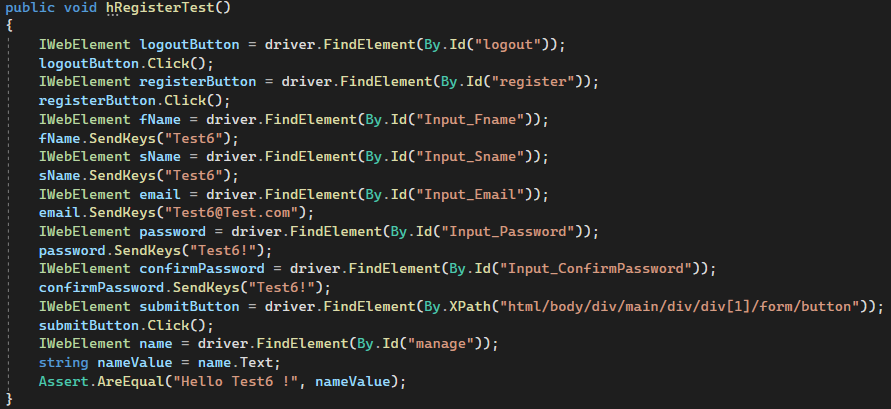
The test was successful.

Id- 8

Name- RegisterTest

Description- Logs out of the admin account navigate to the registration screen and enter details for a new test user then submits the new account. It then checks to see if the account was successfully created by seeing if its name is on the left of the page header.

Code-



Outcome-



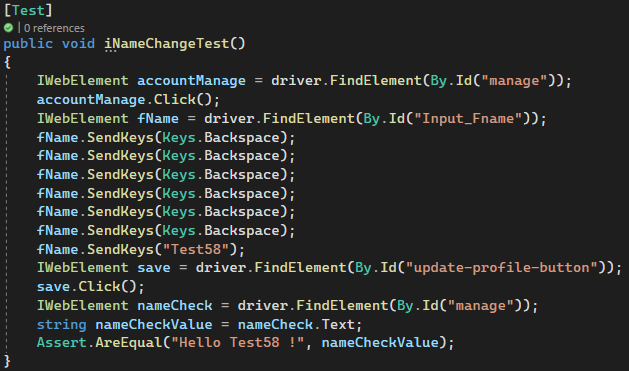
The test was successful.

Id- 9

Name- NameChangeTest

Description- Navigates to account management and replacing the current firstname of the test user with Test58 and then checks to see if their name display on the page header has been changed to Test58.

Code-



Outcome-



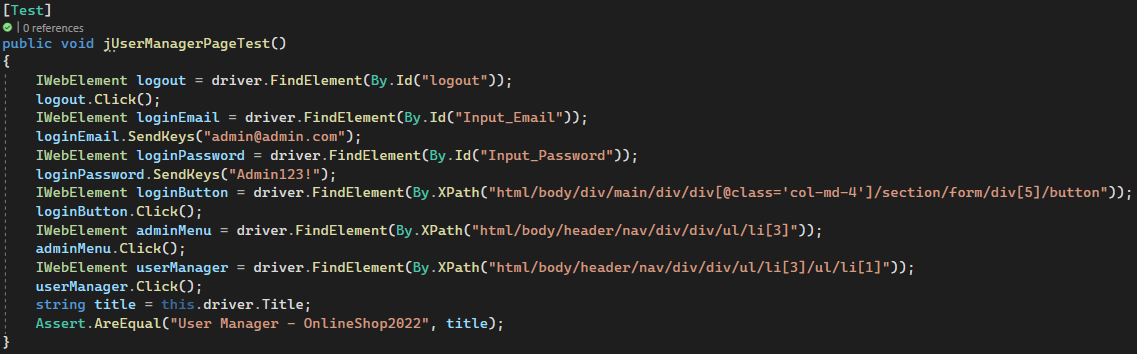
Test was successful.

Id- 10

Name- UserManagerPageTest

Description- Logs out of the test user account and logs back in with the admin account and then navigates to the user manager page and checks if the page title equals that of the user manager page title.

Code-



Outcome-



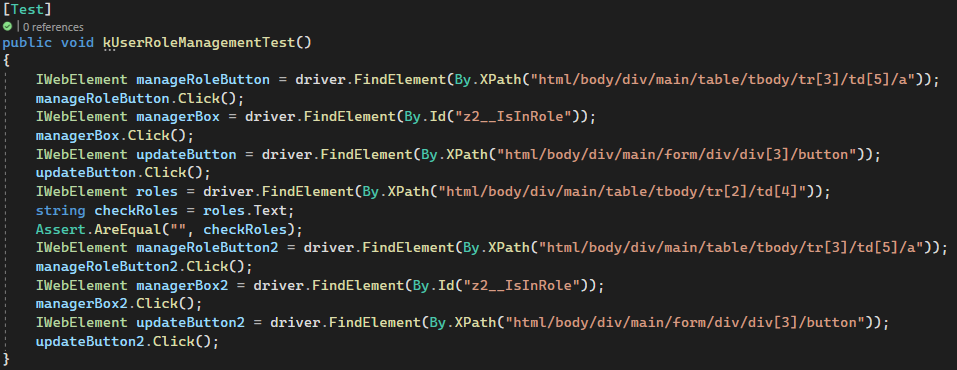
The test was successful.

Id- 11

Name- UserRoleManagementTest

Description- The test clicks on the manage role button for the test user and removes its customer role so it no longer has any roles and then checks that it no longer has any roles and then gives it back its customer role.

Code-



Outcome-



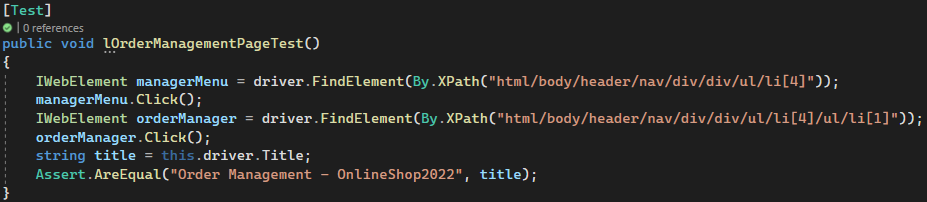
The test was successful.

Id- 12

Name- OrderManagementPageTest

Description- The test navigates to the order management page and then confirms the title of the page matches the order management page title.

Code-



Outcome-



The test was successful.

Id- 13

Name- OrderManagementTest

Description- The test edits the first order changing the first name to test and then checks to see if the first name was successfully changed to test and then changes it back to the previous name.

Code-



Outcome-



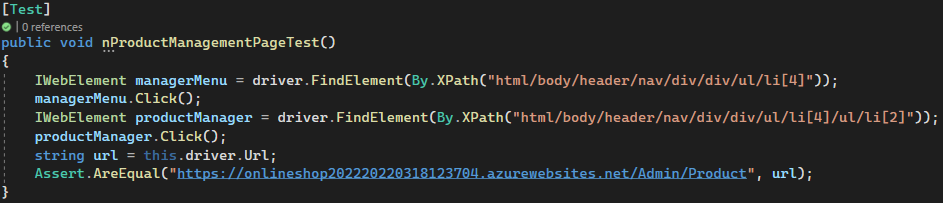
The test was successful.

Id- 14

Name- ProductManagementPageTest

Description- The test navigates to the product management page and confirms that the url of the page matches that of the product management page.

Code-



Outcome-



The test was successful.

Id- 15

Name- PasswordChangeTest

Description- The test goes to the account management page, clicks on the password section and changes the password to a new password and then confirms that the password update message has appeared and then changes the password back to its previous value.

Code-



Outcome-



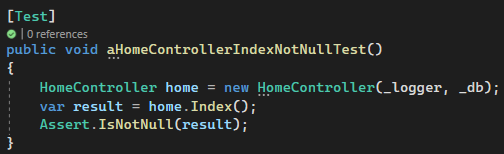
The test was successful.

Id- 16

Name- HomeControllerIndexNotNullTest

Description- Checks to see if the home controller’s index is null.

Code-



Outcome-



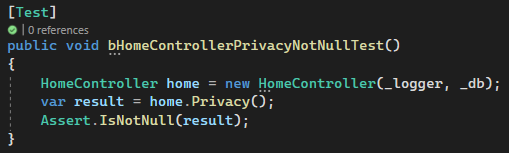
Test was successful the index is not null.

Id- 17

Name- HomeControllerPrivacyNotNullTest

Description- Checks to see if the home controller’s privacy function is null.

Code-



Outcome-



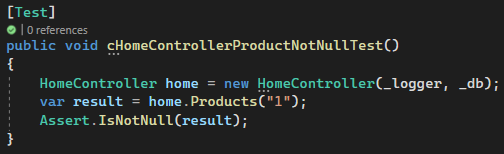
Test was successful.

Id- 18

Name- HomeControllerProductNotNullTest

Description- Check to see if the home controller’s product function is null.

Code-



Outcome-



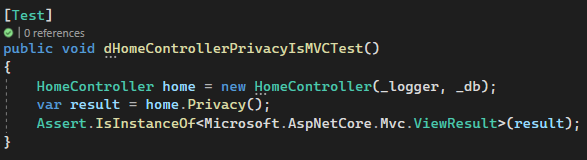
Test was successful.

Id- 19

Name- HomeControllerPrivacyIsMVCTest

Description- Checks to see if the home controller’s privacy function is an mvc controller.

Code-



Outcome-



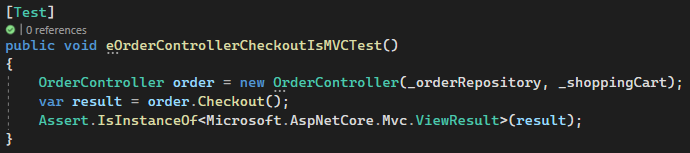
Test was successful.

Id- 20

Name- OrderControllerCheckoutIsMVCTest

Description- Tests to see if the order controller’s checkout function is an mvc controller.

Code-



Outcome-



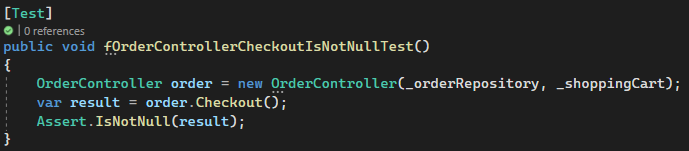
The test was successful.

Id- 21

Name- OrderControllerCheckoutIsNotNullTest

Description- Tests to see if the order controller’s checkout function is null.

Code-



Outcome-



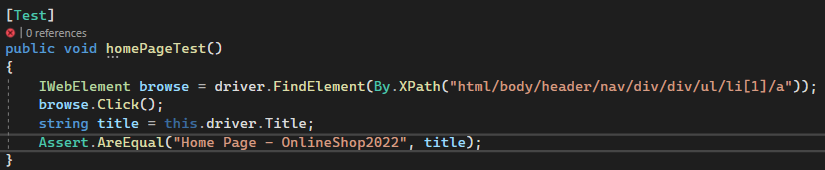
The test was successful.

Id- 22

Name- homePageTest

Description- Clicks on the browse button to see if it navigates to the home/products page by checking if the title of the navigated to page equals Home Page.

Code-



Outcome-



The test failed due to the page name not been equal to Home Page as the browse button does not successfully navigate to the home/products page and instead navigates to a unused privacy page.

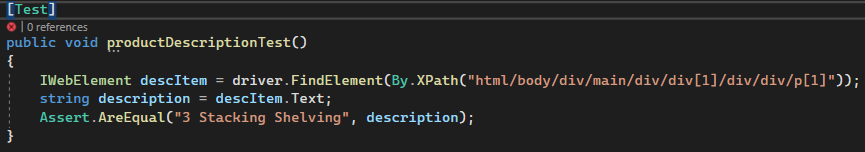
Resulting action(s)- As a result of this failed test the privacy page will be deleted and the browse button will be updated to navigate to the home/products page instead.

Id- 23

Name- productDescriptionTest

Description- The tests locates the product description displayed on the home page and sees if it matches up with the products actual description.

Code-



Outcome-



The test failed as the description displayed out the page did not match the products actual description.

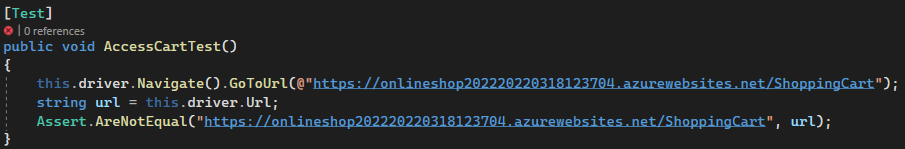
Resulting action(s)- The page will be fixed to make sure it displays the correct product description.

Id- 24

Name- AccessCartTest

Description- Tests whether the user is able to access the cart page without logging in,

Code-



Outcome-



The test failed as the user is able to access the cart page without logging in.

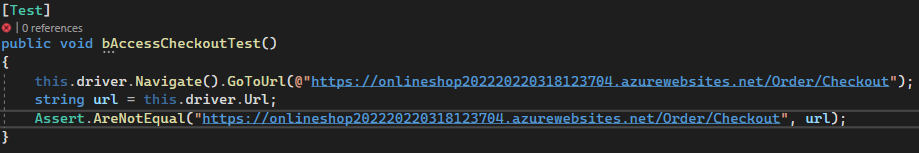
Resulting actions(s)- The shopping cart controller will be updated to make sure that only a user with the customer role may access any pages associated with that controller.

Id- 25

Name- AccessCheckoutTest

Description- Tests whether the user is able to access the checkout page without logging in.

Code-



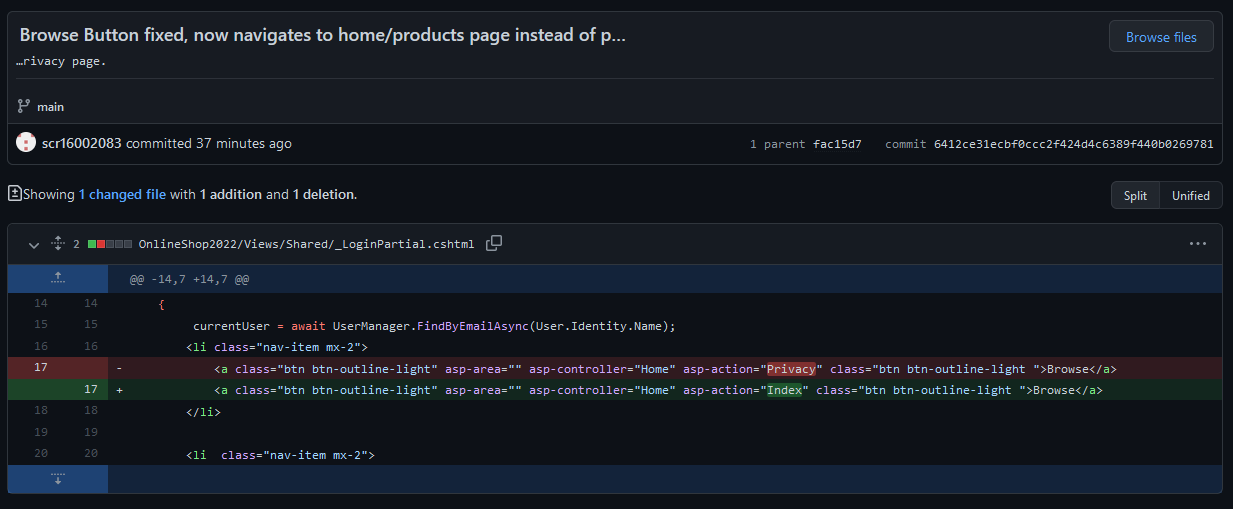
Outcome-



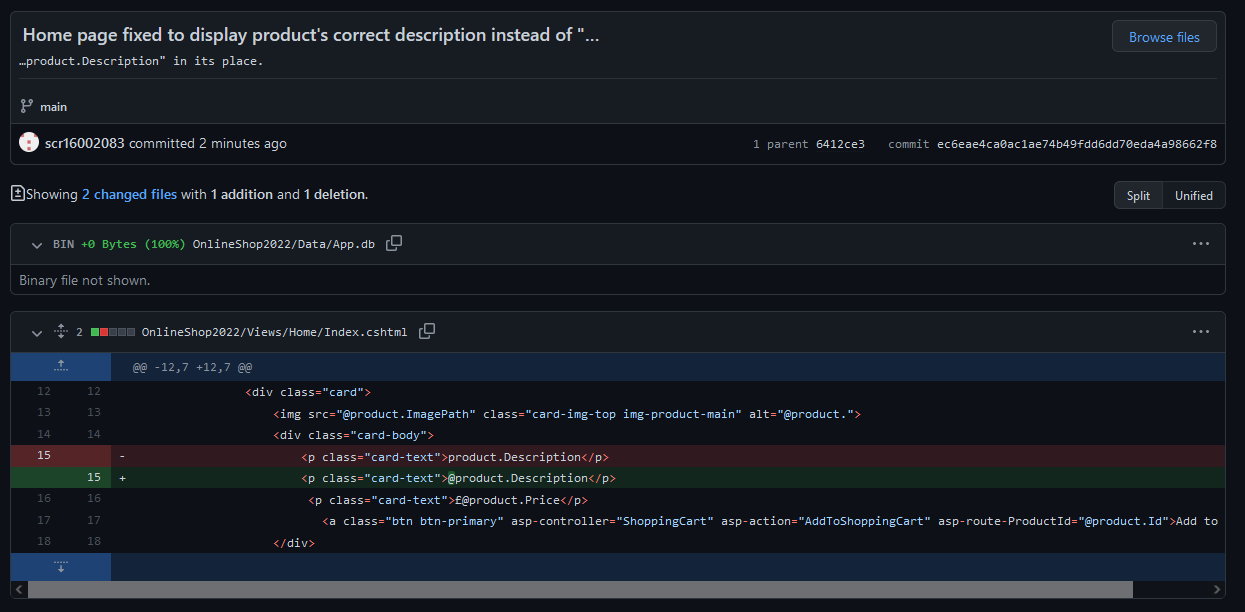
The test failed as the user is able to access the checkout page without logging in.

Resulting actions(s)- The order controller will be updated to make sure that only a user with the customer role may access any pages associated with that controller.

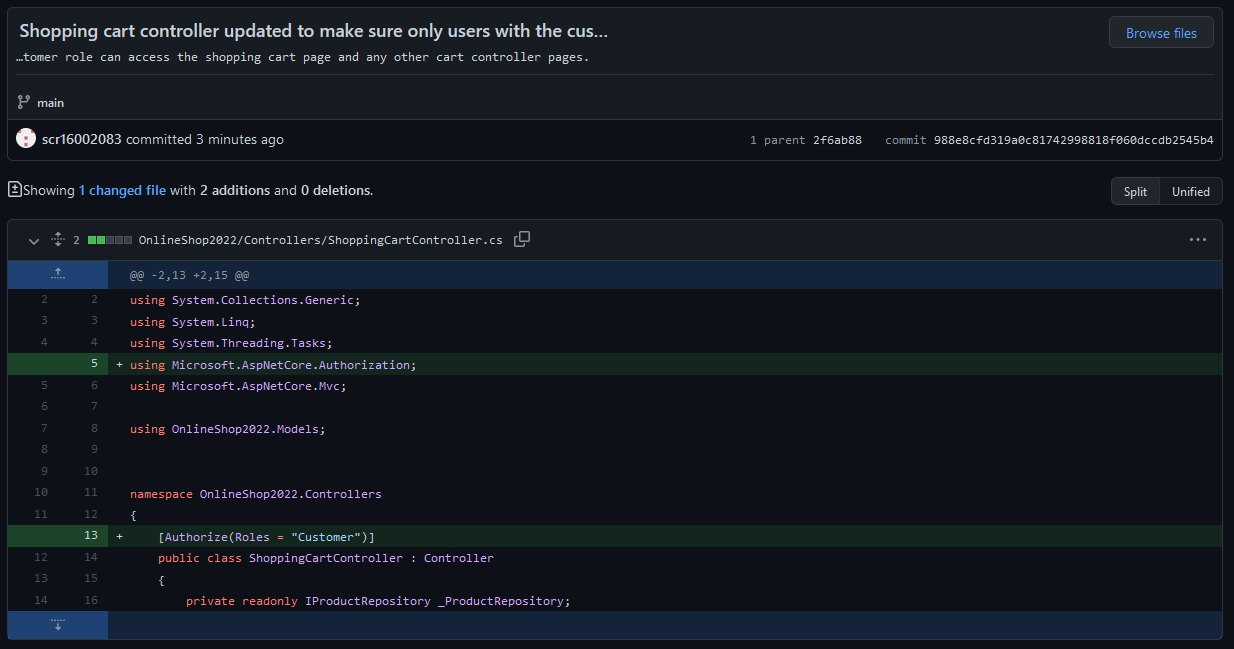
# Issue Fixes:



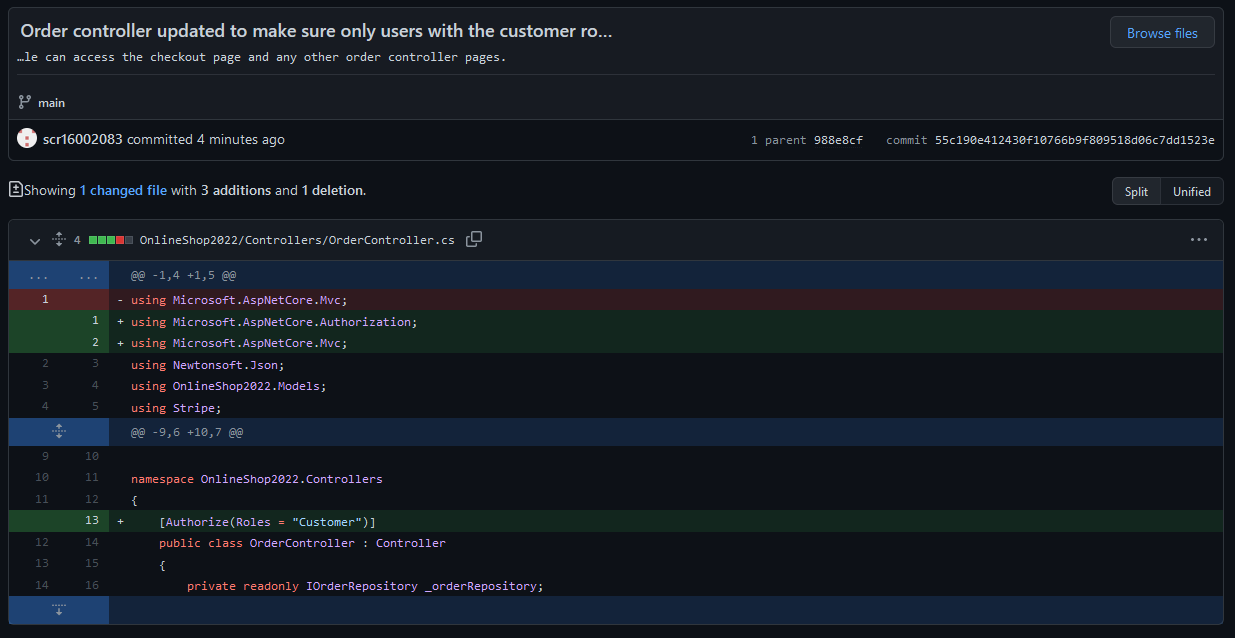
This was a fix for the issue found in test Id 22 in which the browse button would navigate to the privacy page instead of the home page. As you can see in the GitHub commit the button has been changed to load the asp action index of the home controller which is the home page instead of the privacy asp action of the home controller.



This fix was for the issue found in test Id 23 in which the product’s description displayed on the home page would be displayed as “product.Description” as opposed to the product’s actual description. As you can see in the GitHub commit this was due to the “@” missing from the product.Description call.



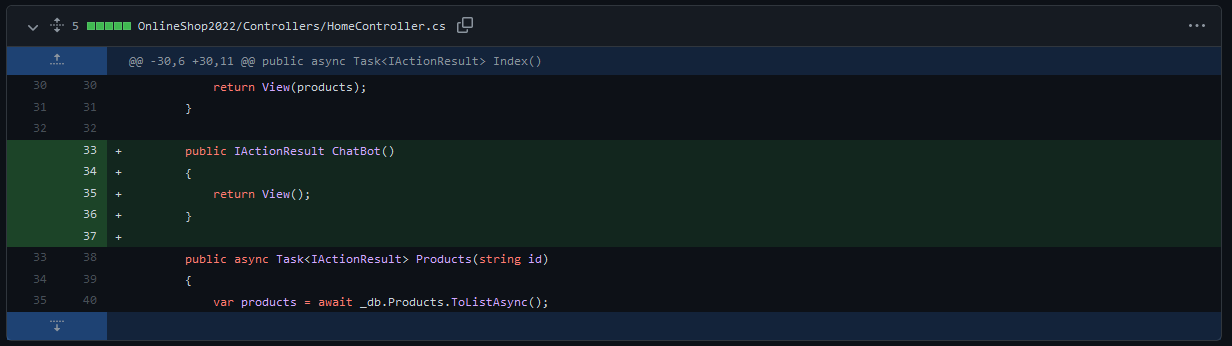
This fix was for the issue found in test Id 24 in which the user was able to access the cart page without first logging in. This issue was fixed by adding AspNetCore.Authorisation to the Shopping Cart Controller and adding an authorizer that made it so only users with the customer role could access the cart page and any other pages associated with the shopping cart controller.



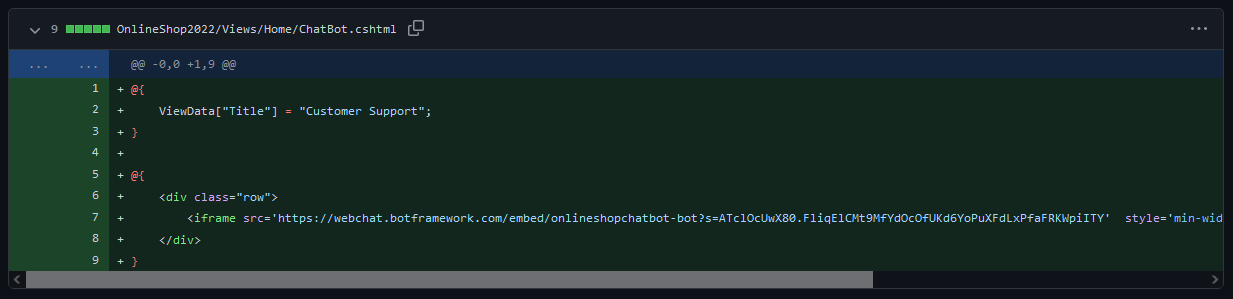
This fix was for the issue found in test Id 25 in which the user was able to access the checkout page without first logging in. This issue was fixed by adding AspNetCore.Authorisation to the Order Controller and adding an authorizer that made it so only users with the customer role could access the checkout page and any other pages associated with the order controller.

# Additional Feature:

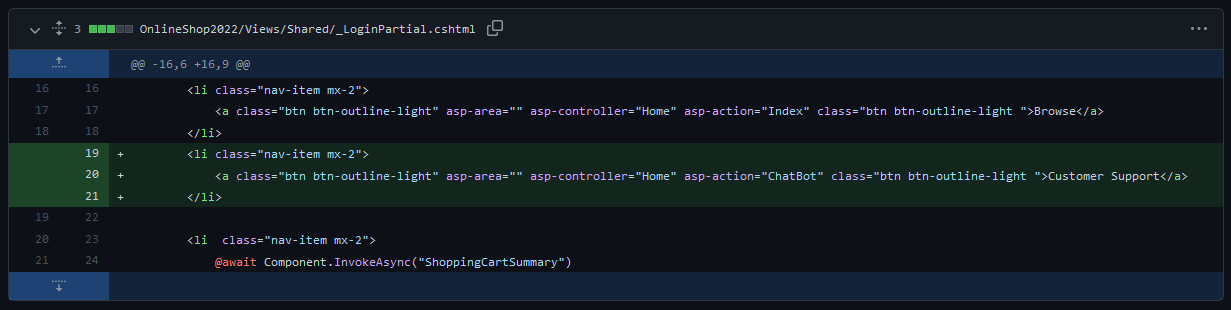
The additional requirement that is going to be added is Live Customer Support. This is going to be achieved using a chatbot to answer FAQs such as how to log out of the website. The requirements for this feature are that it must be fully integrated into the website, be easy to access through the website and be able to answer FAQs for the customer.

The chatbot was created using Azure DevOps QNA services which allowed for the creation of Q and A chatbot which can be fully integrated into the website. To integrate it the following had to be done:

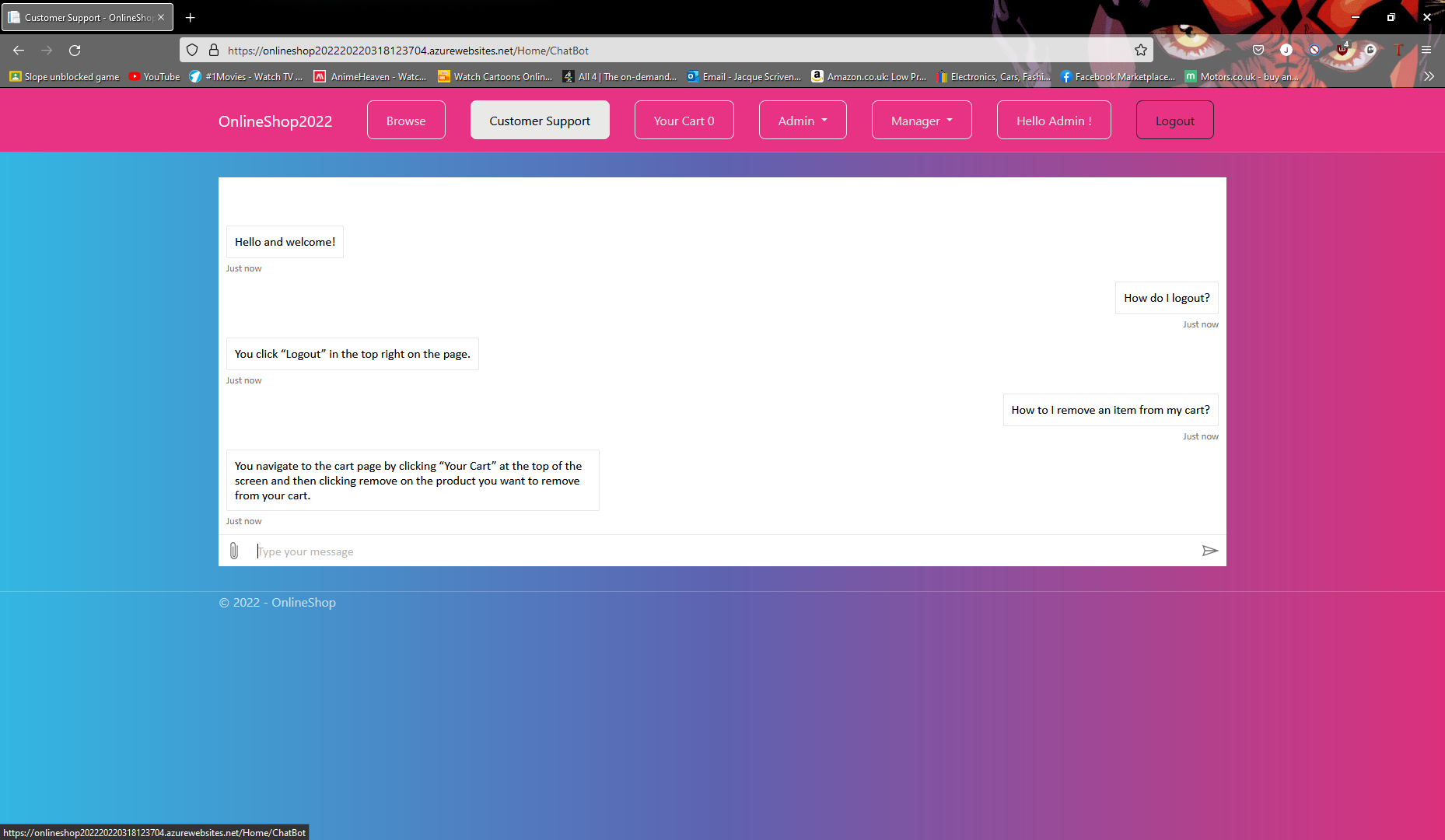
Firstly a ChatBot action had to be added to the home controller so the chatbot page could be navigated to.



Secondly the Customer Support page had to be created and the bot embedded into the page. This was achieved using the iframe tag which is provided by Azure and it contains a link to the bot along with a key to access it.



Lastly there had to be a button in the websites navbar to be able to navigate to the Customer Support page.



As you can see the Customer Support button has been added to the navigation and the user is able to access the support page and the chatbot as well as ask it questions on how to use the site and have accurate answers given.

# Publication Platform:

The application was deployed to Azure DevOps it can be found here: <https://onlineshop202220220318123704.azurewebsites.net>

Azure DevOps is an online platform with multiple applications. It was created by Microsoft and enables developers to quickly deploy their applications to azure where the application will be hosted. If the application is made within the Visual Studio IDE it can be easily uploaded to Azure using a quick and simple publishing menu built into the Visual Studio IDE. It also has many other additional features such as application development tracking and easy collaboration within development groups. (Rossberg, 2019) Azure was the obvious deployment choice due to its easy and quick publishing wizard within the Visual Studio IDE making it the most efficient and time-saving deployment method.

# Evaluation of testing, verification and validation methods:

Selenium is one of the chosen testing methods for this application. Selenium allows for the automated testing of web apps by selecting HTML tags via id’s, class names, etc or XPath which is the HTML path of a specific element (example: html/body/div/a/p[1]), and then interacting with the element by clicking on it or entering text. (Bruns, Kornstadt, Wichmann, 2009) You can then use this functionality to test the application by for example having selenium load up the web app and try to navigate to the registration page from the home page and register a new account. Selenium can then verify this by selecting the username displayed on the website header and seeing if it matches up with the username selenium gave the newly registered account and if it does match Selenium will pass the test if not it will display an error either saying it couldn’t find the element or that they did not match.

Selenium also works with multiple languages and works with multiple IDEs making it very flexible to work with. The web application that has been tested was a .NET Web Application made within the Visual Studio IDE, Selenium is very well integrated within Visual Studio and can work with C#. (Razal, Fahrurazi, 2011)

This combination of flexibility and functionality made Selenium a very good testing tool, it allowed for a wide range of frontend testing within the same work environment making the whole process very efficient.

NUnit is a testing framework made for C# and the .NET framework. It contains multiple tags that can be applied to public voids and classes. (Hamilton, 2004) When applied to a class it is used to define a group of tests for different parts of the application. When applied to a public void it is usually to indicate a test but it can also be used for setups that run before every test or Tear Downs that run after every test.

NUnit tests and test classes are displayed in a testing window in the Visual Studio IDE where they can be run and monitored making for easy debugging of the application and tests. Additionally, Selenium can be used within NUnit to make testing with Selenium even easier.

NUnit can also be used for the testing of MVC controllers allowing the tester to check if a controller function is null or not and if a controller function is an MVC function or not. This additional functionality allows for the backend of .NET applications to be tested increasing the amount of application testing that can be automated.

# References:

Bruns, A. Kornstadt, A. Wichmann, D. (2009) ‘Web Application Tests with Selenium’, IEEE Software, vol. 26, no. 5 [Online]. Available at <https://ieeexplore.ieee.org/abstract/document/5222802?casa_token=H86WirFtRUgAAAAA:f-n8SppQWQh3xD9Sfo99h_Dfpw7YOAg2EU2Kuf48b4B5p4pOh9wHetFQxD-JMi3oFC3vV7EugTs> (Accessed 21/3/2022)

Hamilton, B. (2004) NUnit Pocket Reference [Online], 1005 Gravenstein Highway North, Sebastopol, CA, United States, O’Reilly. Available at <https://www.google.co.uk/books/edition/NUnit_Pocket_Reference/kSS5RgyXbGcC?hl=en&gbpv=1&dq=NUnit&pg=PA1&printsec=frontcover> (Accessed 21/3/2022)

Razak, R. Fahrurazi, F. (2011) ‘Agile testing with Selenium’, 2011 Malaysian Conference in Software Engineering. Johor Bahru, Malaysia, 13-14 December. IEEE [Online]. Available at <https://ieeexplore.ieee.org/abstract/document/6140672> (Accessed 21/3/2022)

Rossberg, J. (2019) Agile Project Management with Azure DevOps Concepts, Templates and Metrics [Online], Apress. Available at <https://4p8h5xe1ne.pdcdn.xyz/dl2.php?id=188629757&h=968028d559796dbd76eaf3a29c730ea1&u=cache&ext=pdf&n=Agile%20project%20management%20with%20azure%20devops%20concepts%20templates%20and%20metrics> (Accessed 19/3/2022)