Testing – Sinead – to include

Website testing was performed on the deployed site. It included testing of the following:

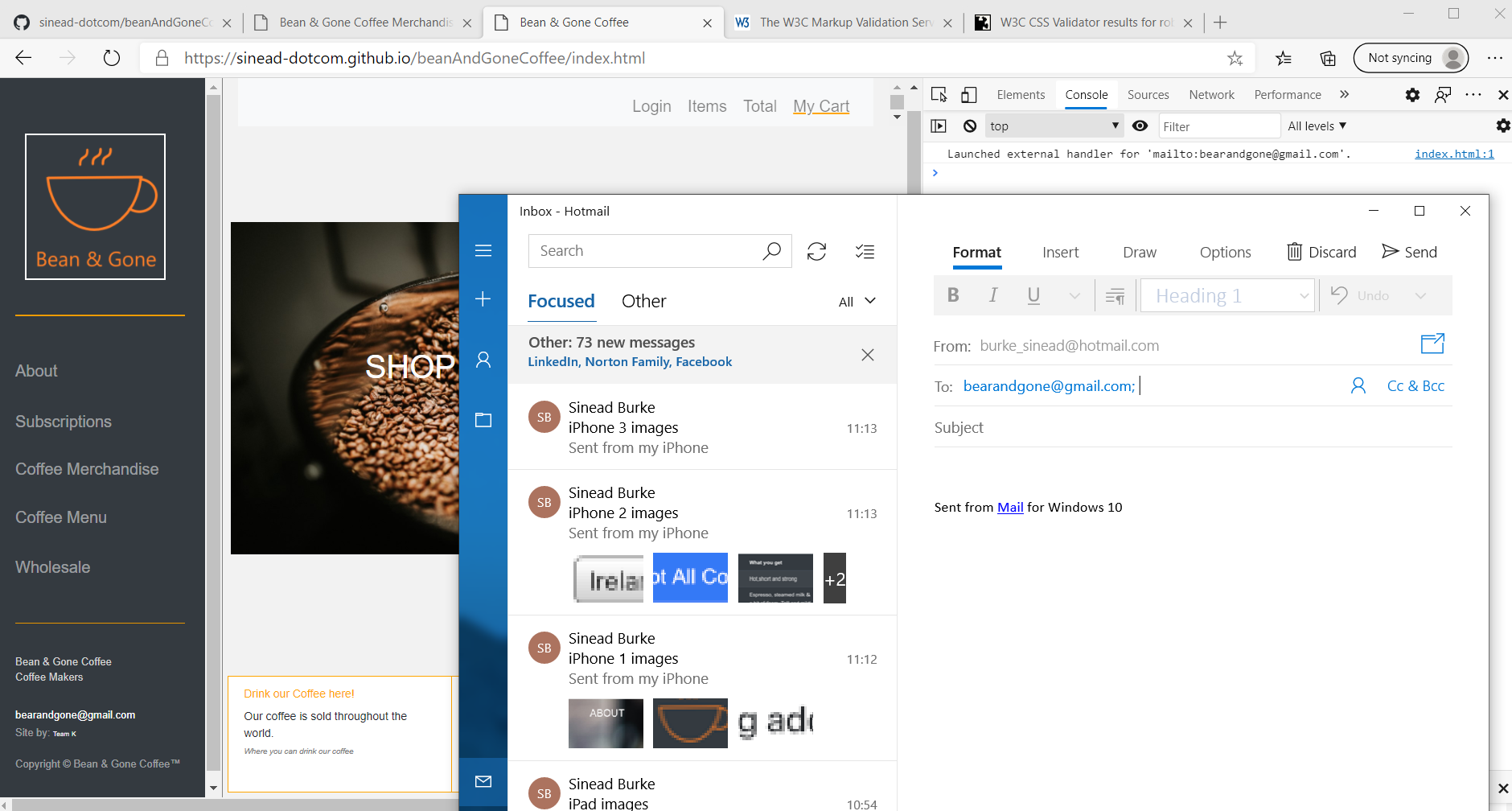
* Outgoing links
* Internal links
* Anchor links
* MailTo links
* Testing of forms – confirm default values work. Confirm submit works.
* Testing of the cookies – to check that 2 places where cookies are used.

Testing on different platforms:

* PC laptop
* iPhone8
* iPhone7
* iPAD
* Apple MACPro

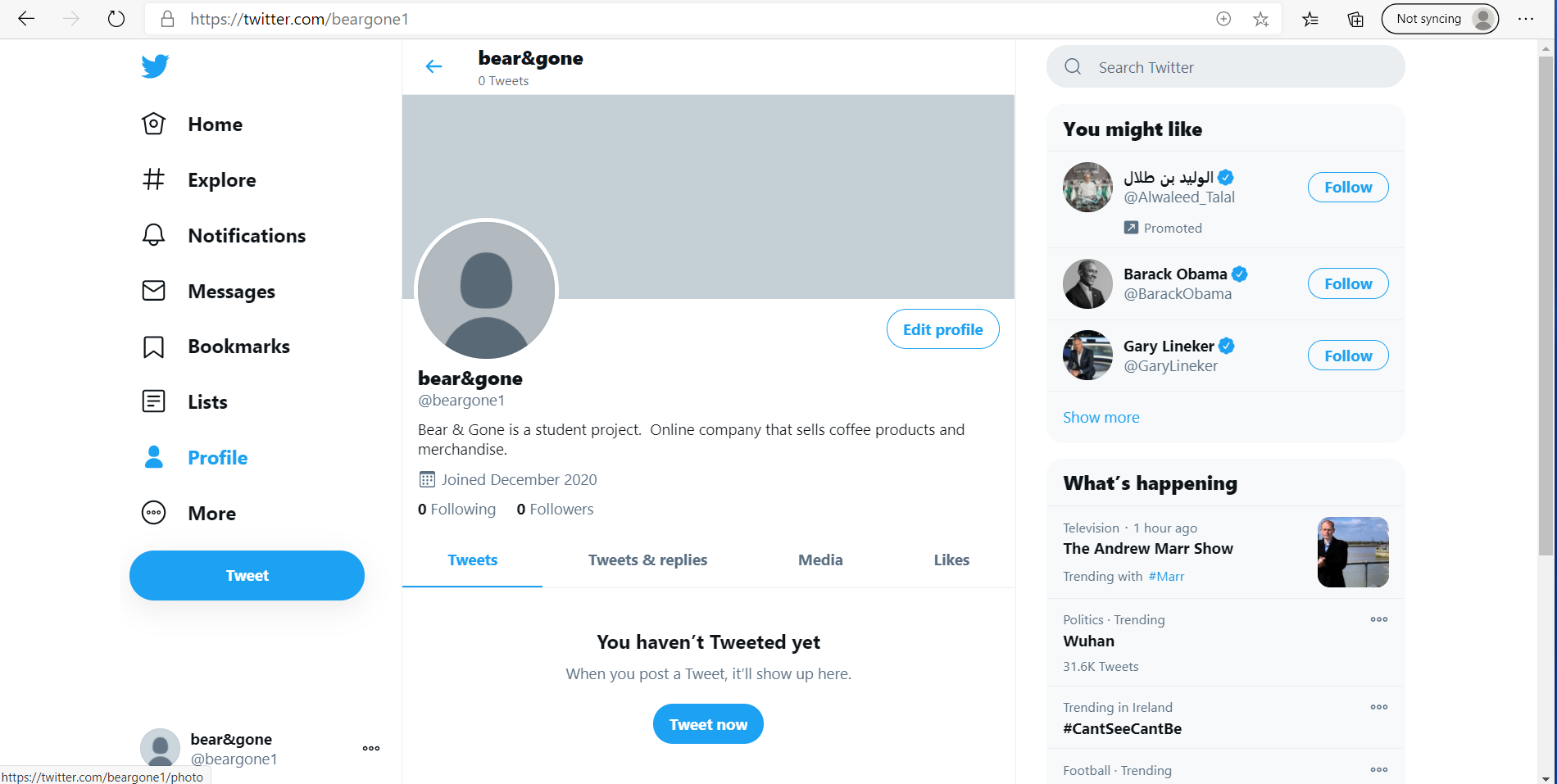
There were a number of issues to clear up with outgoing links – we had to go back and change the teamK website to “#” as we had no operating website to link to.

The following shows the mailto testing:

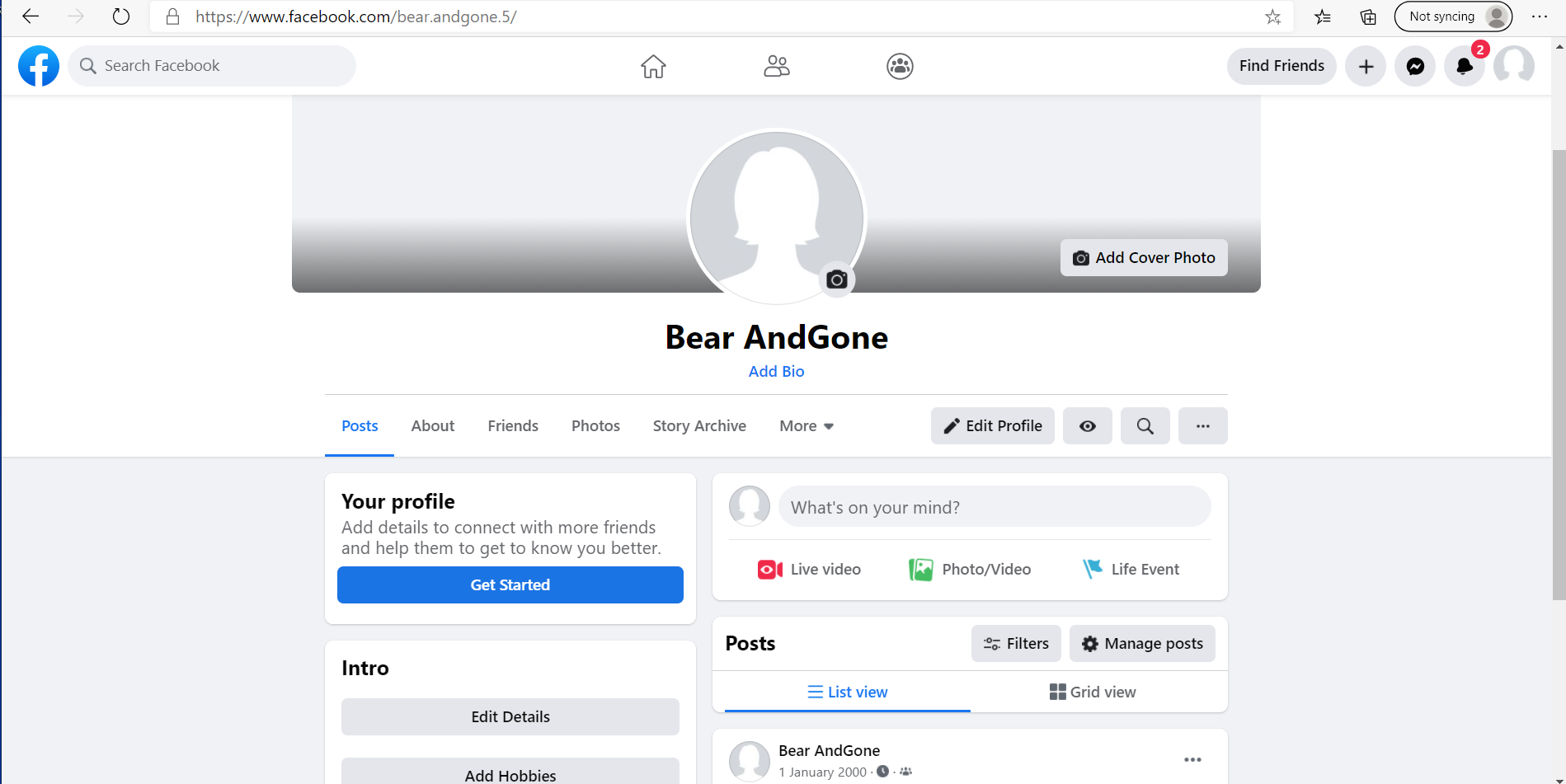


There are 2 places on each page where this can be called. Tested all.

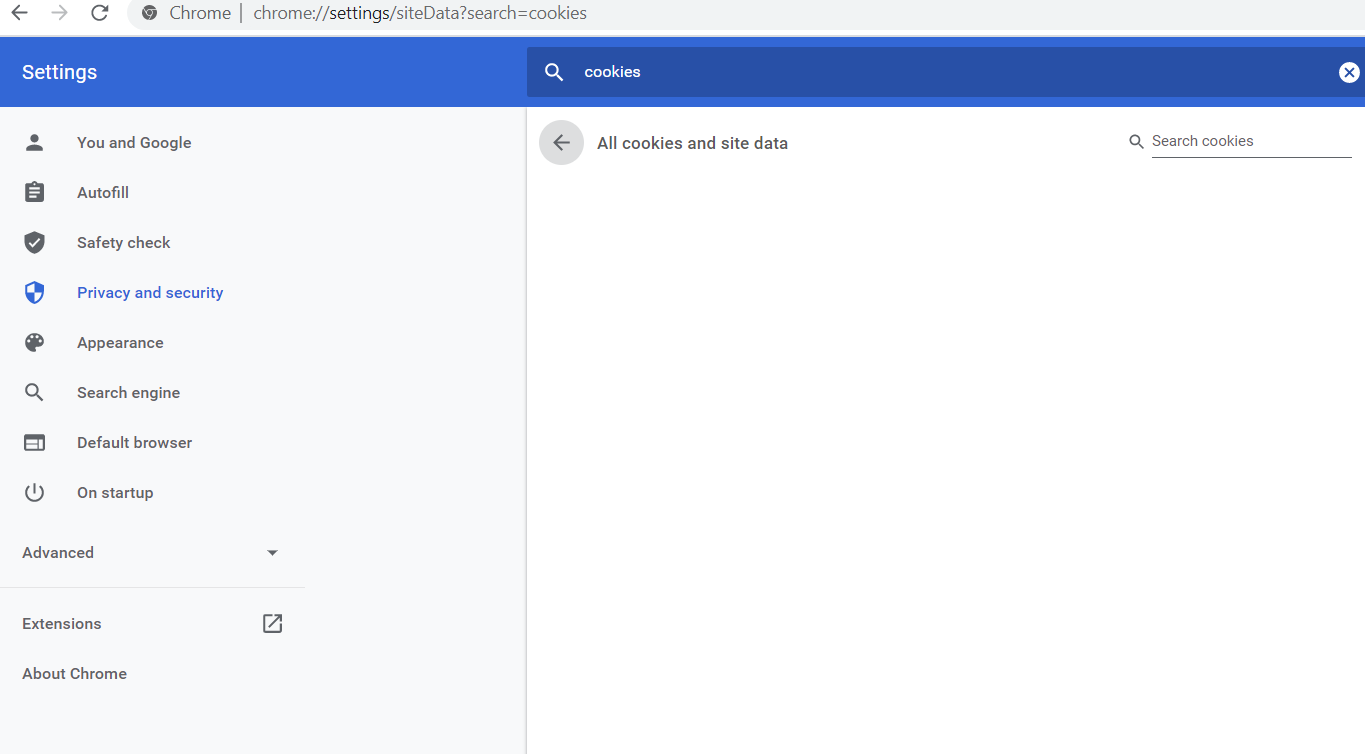
Tested link to Twitter:

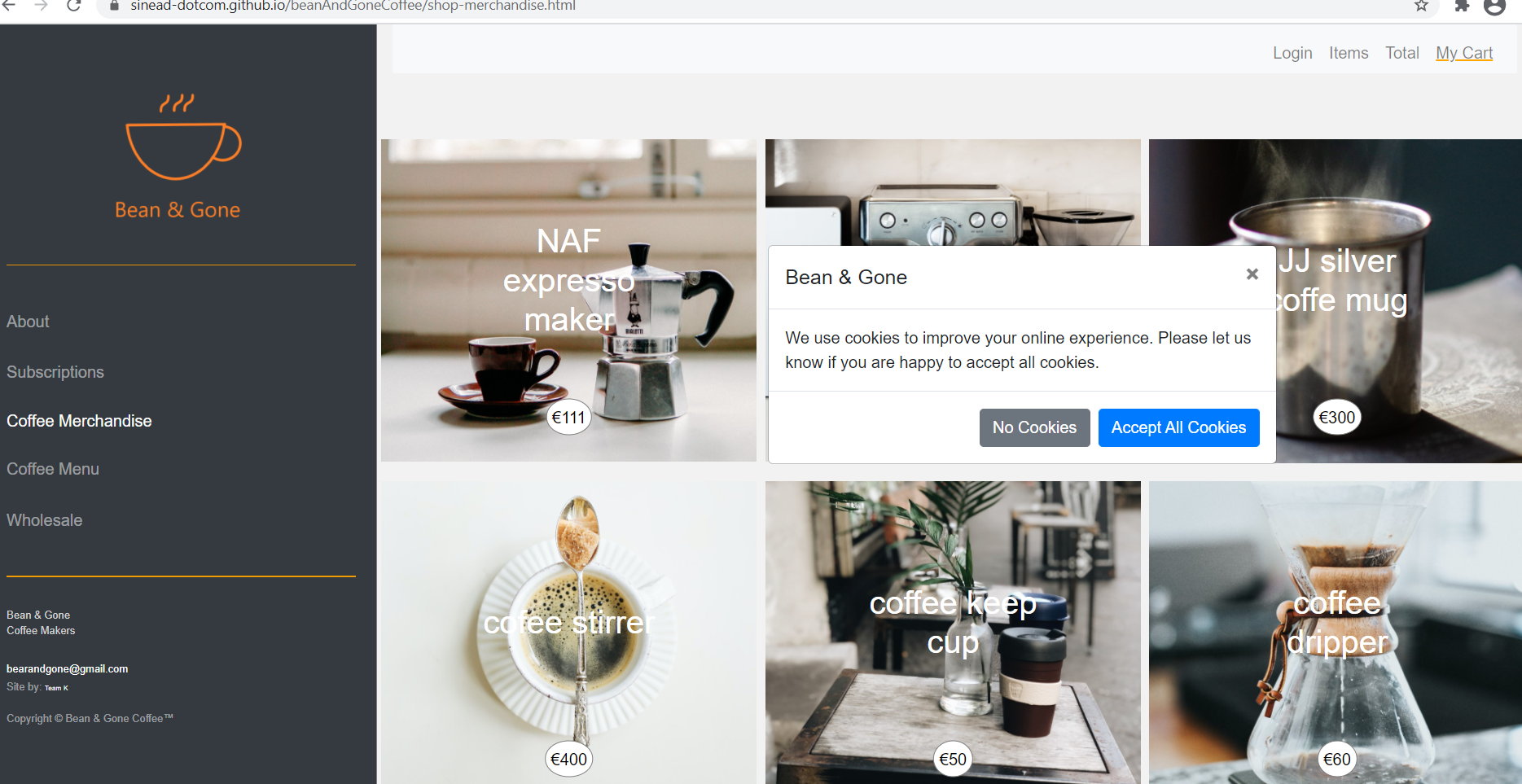


And tested the link to Facebook:

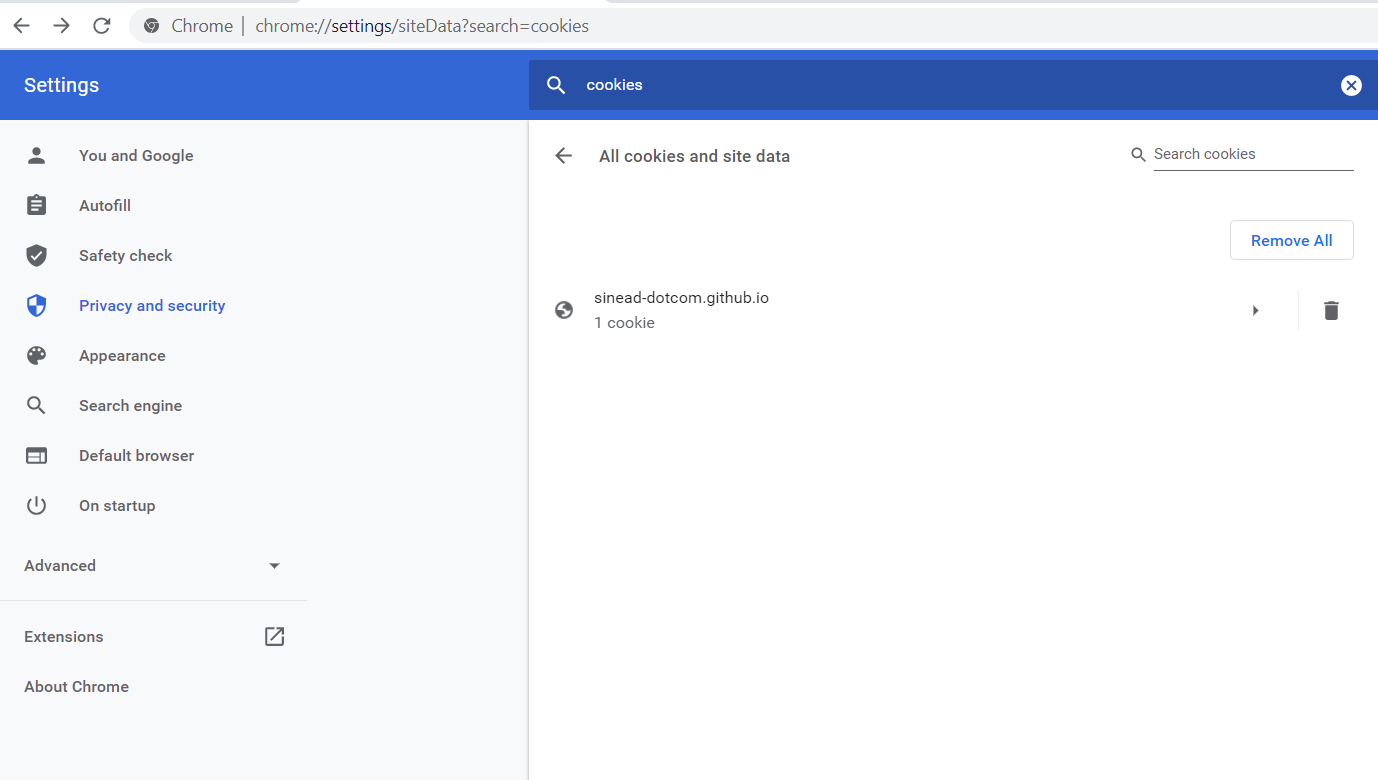


Testing of Cookies:

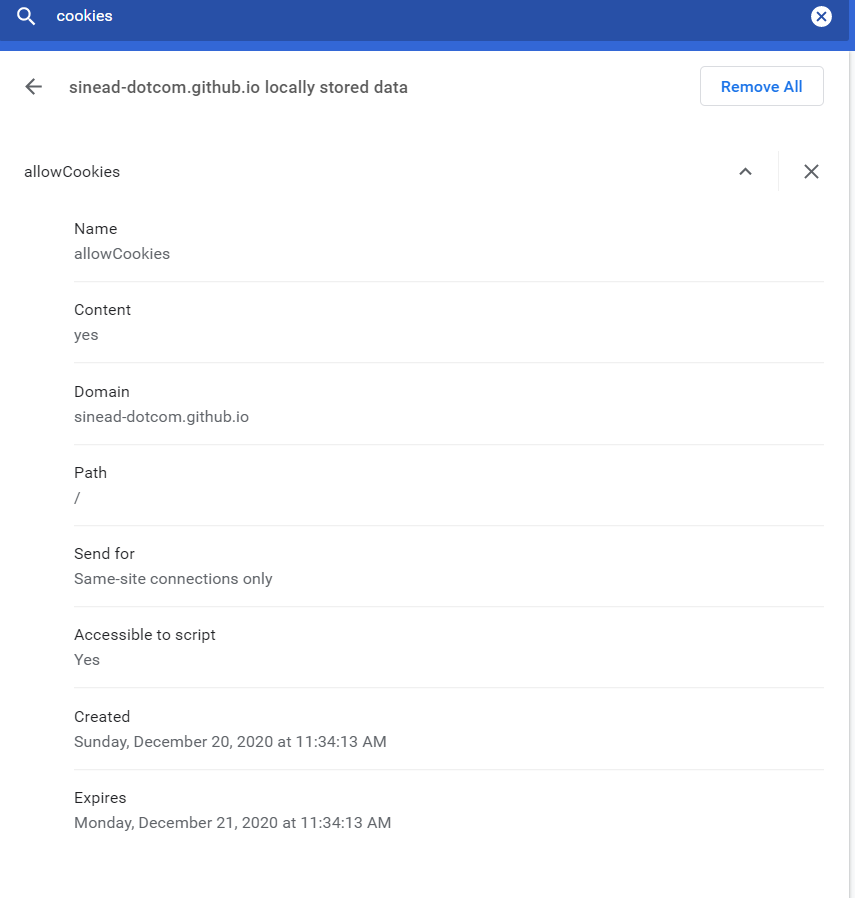
* The previous days cookies had disappeared – would like to retest this again as only did this test once
* On a browser with a LIVE site we tested the cookies. Starting from zero cookies.
* 



Firstly, we accept all cookies in this modal popup. We see that the cookie is seen and stored.

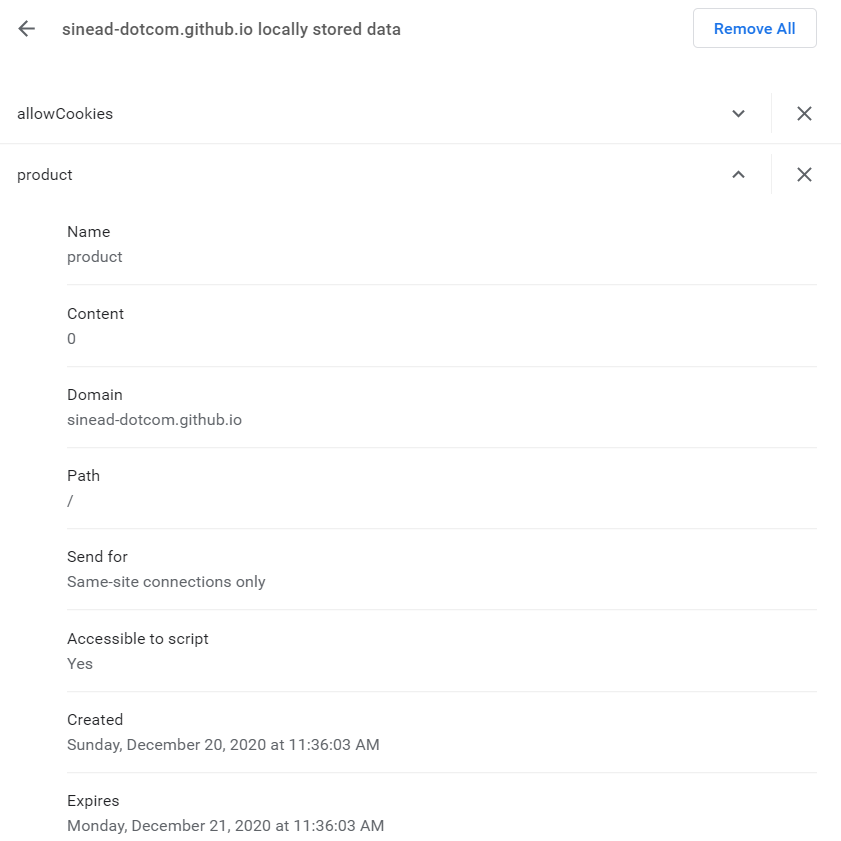


And the details of the cookie are:



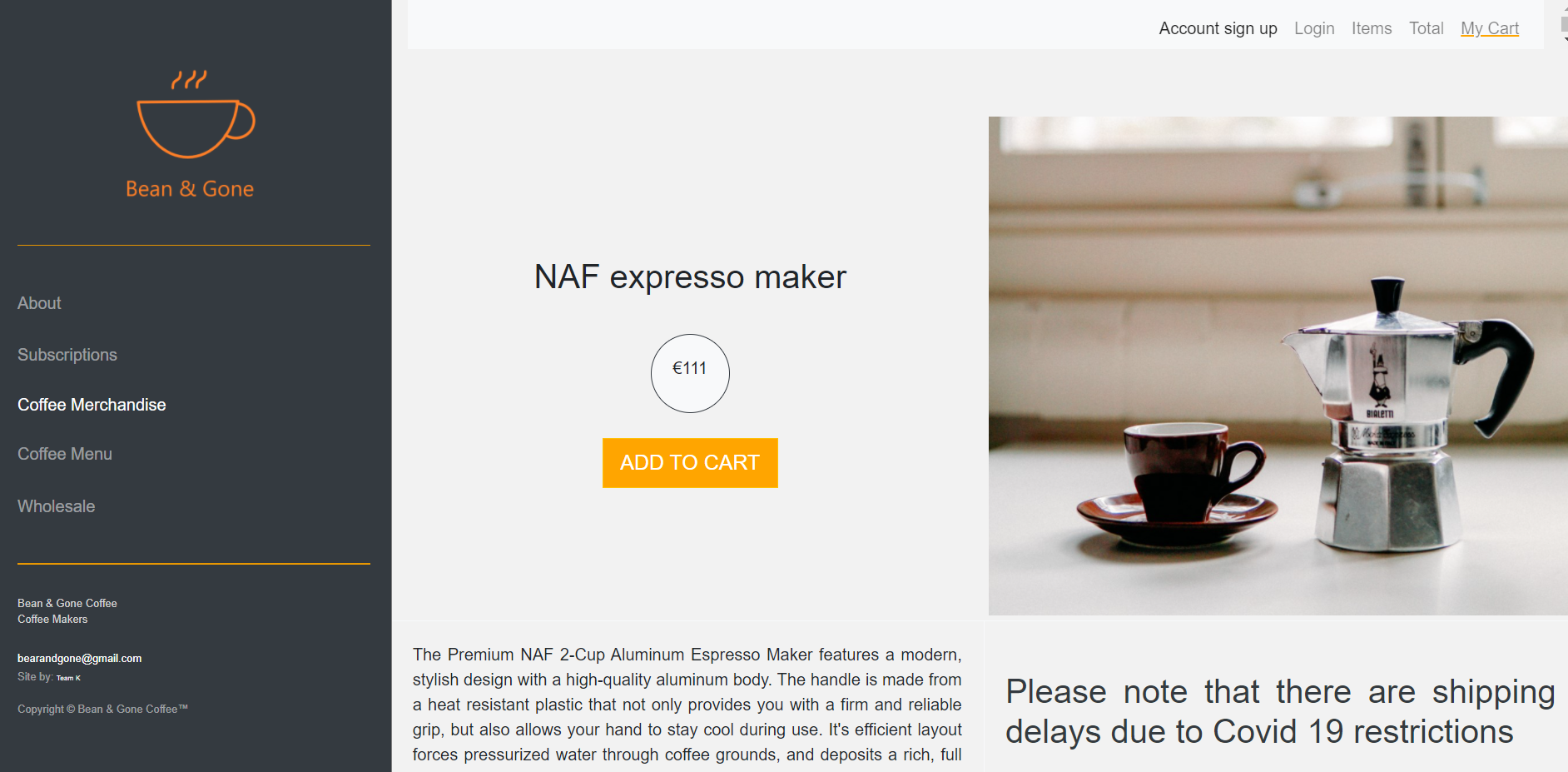
As you can see this is to identify that the user has agreed for cookies to be used/stored. Also, the expiry date is set correctly.

Next we choose a product, a cookie is used to store this chosen product. This means that when it clicks through to the product page that it knows which one to load.

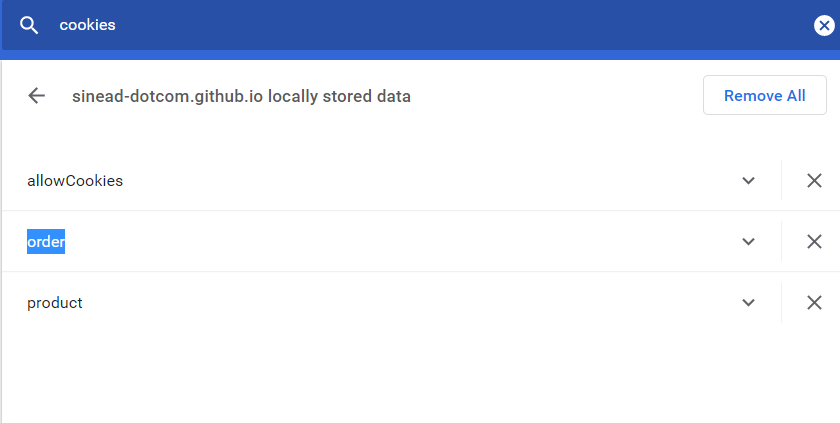


And, it also is stored for 1 day only.

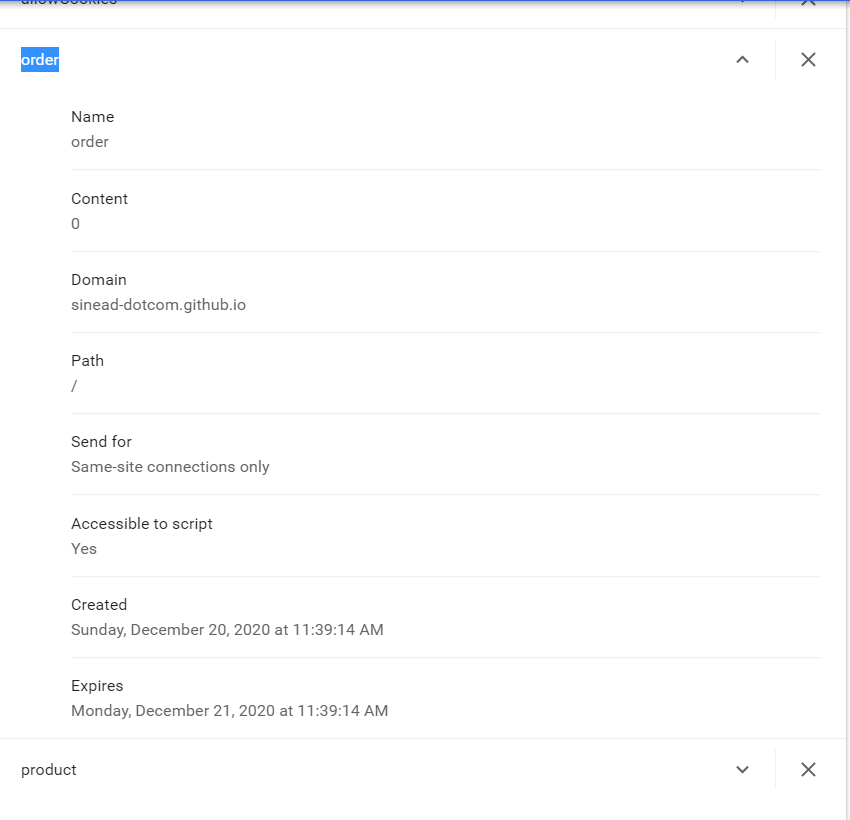
Now that we have the correct product loaded. Lets add this product to the cart. Again a cookie is successfully stored to record this. This will help when I get the basket working.



We now have 3 cookies.



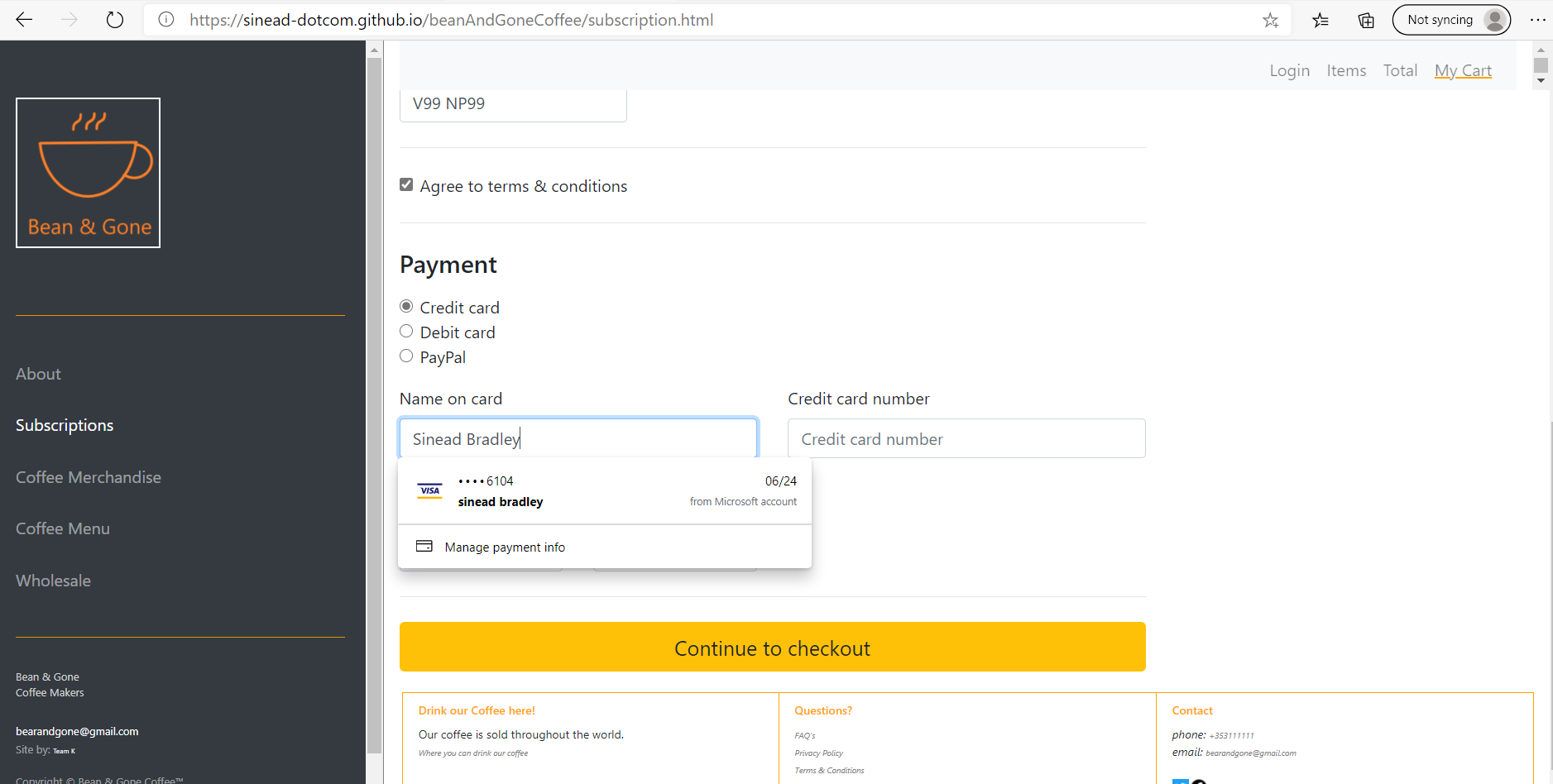
And this is the order cookie details – correctly set! Again this happened when the checkout button was clicked. I would like to show the addition of this item to the basket on the top NAV bar if there is enough time.

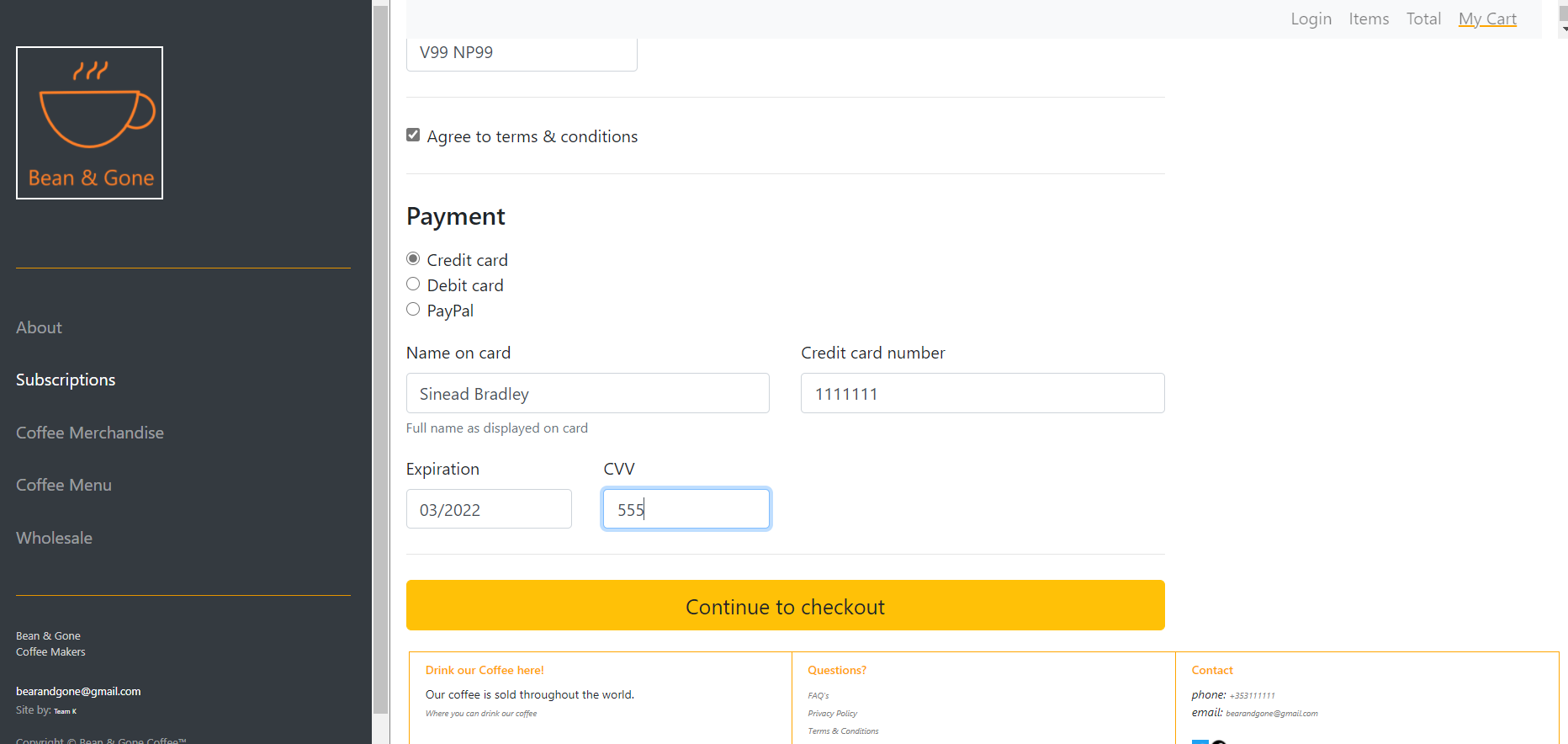


So, next test was to delete the cached cookies from the browser. Then reload the page and see what happens. The user was prompted to consent to cookies again – using the lovely bootstrap modal.

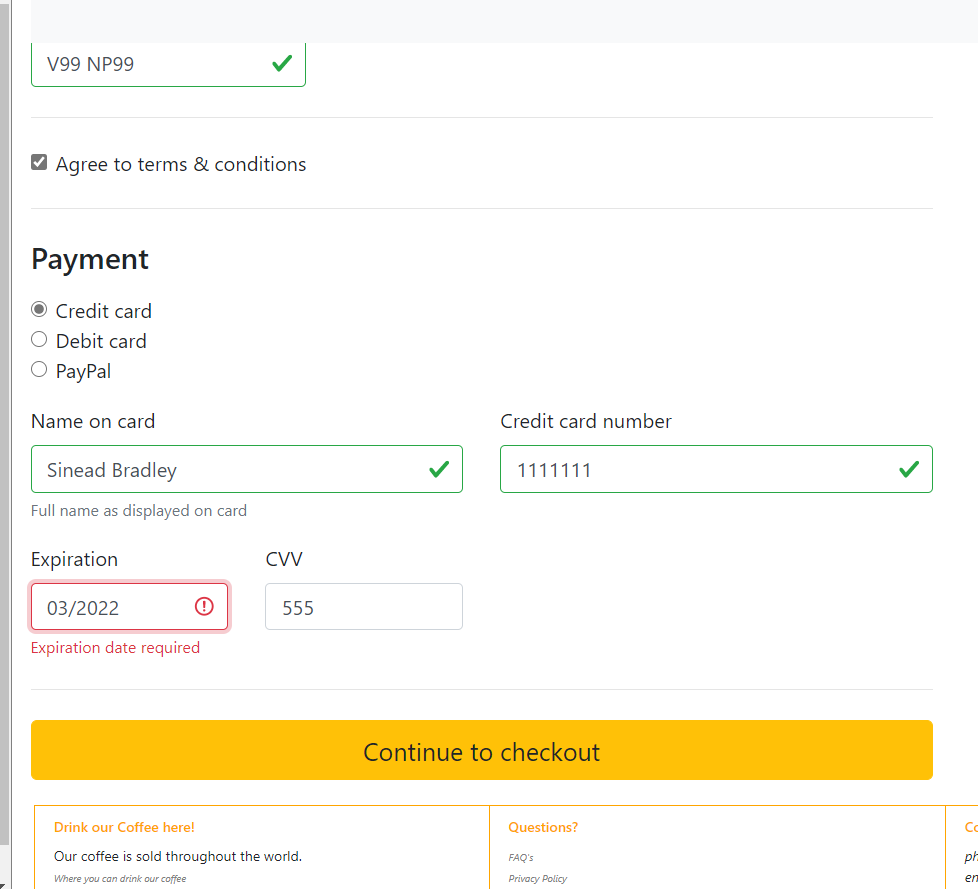
Form testing for the subscription page.

It was interesting to see that the autofill function was working.

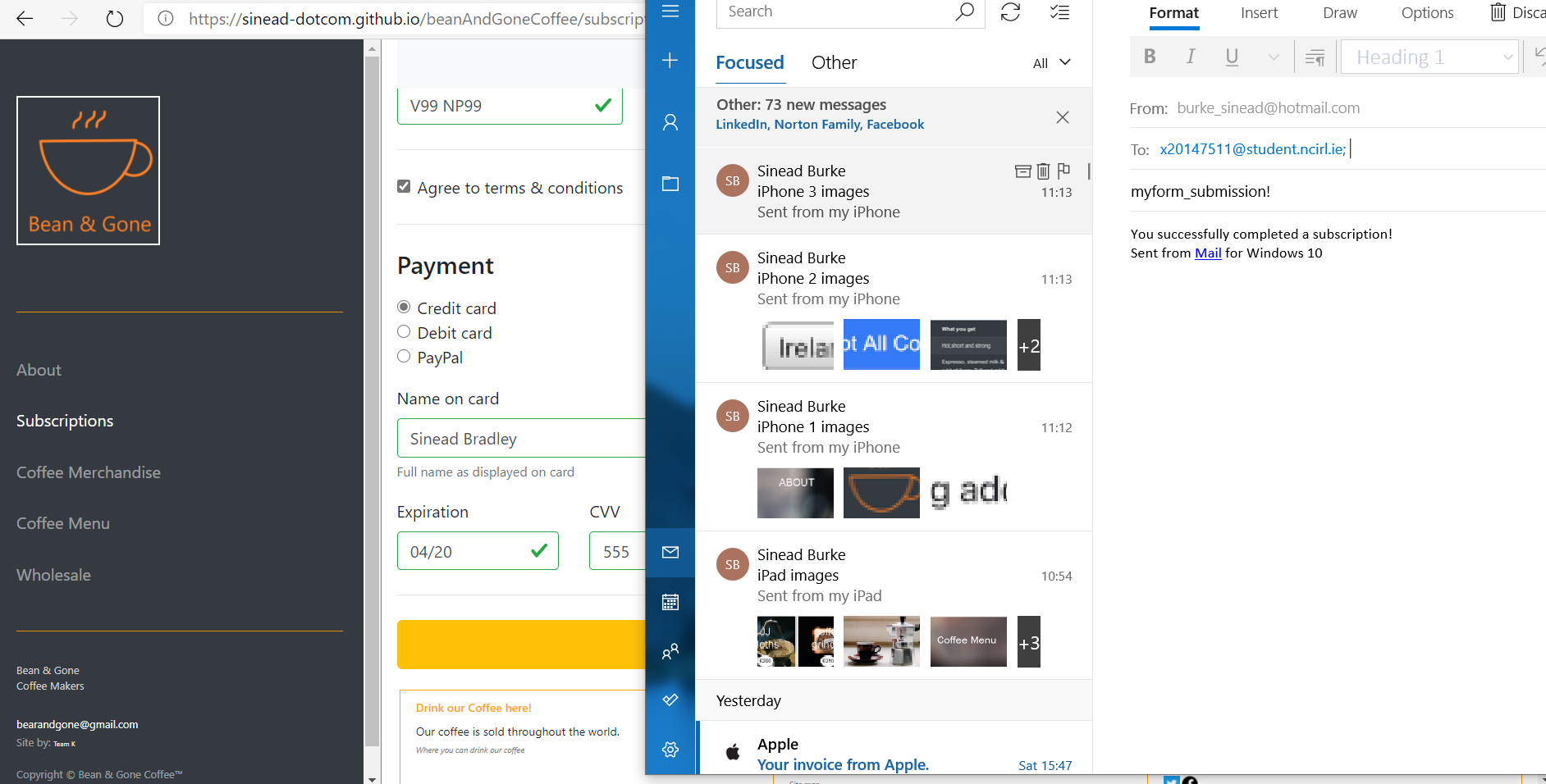




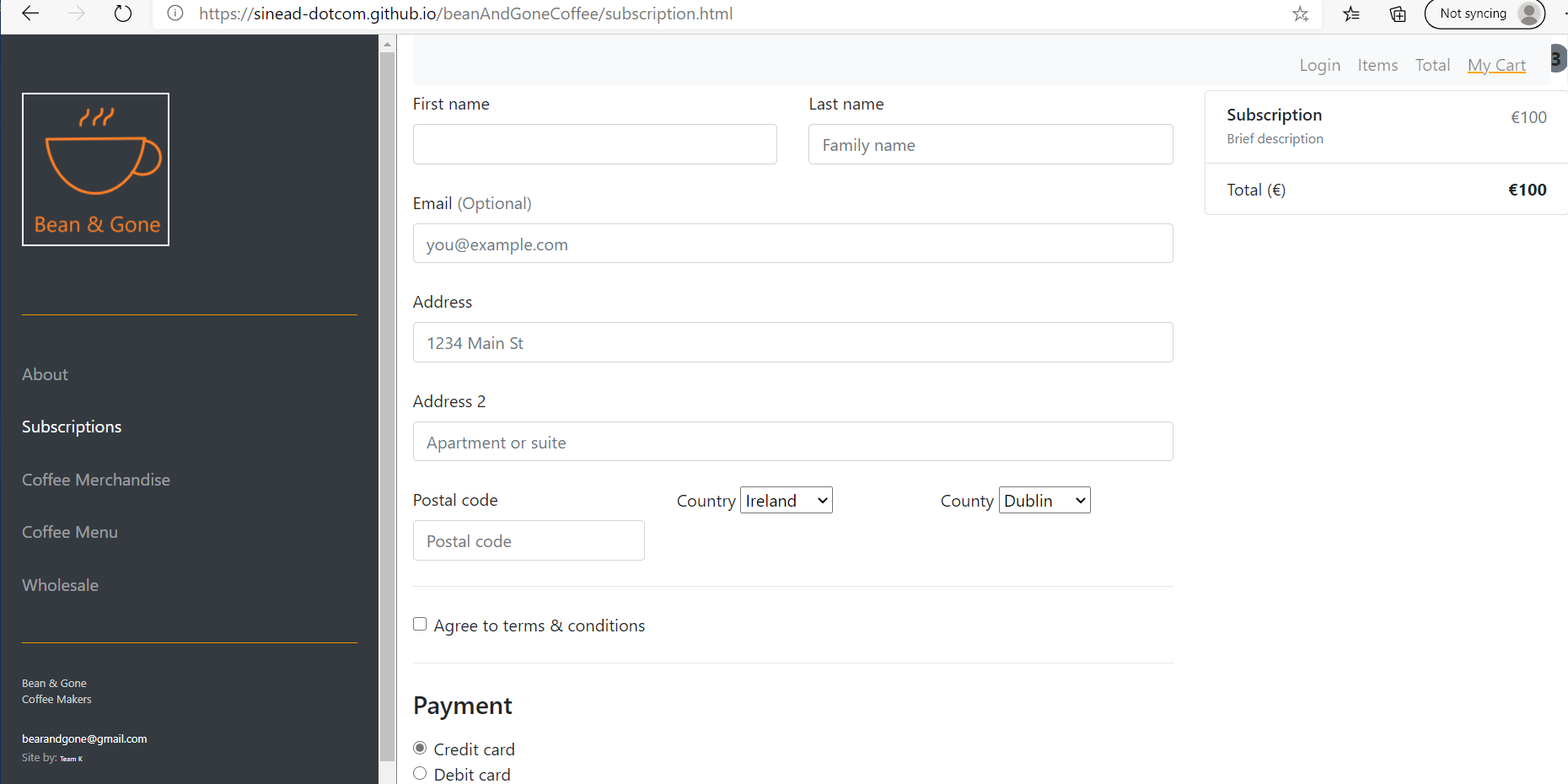
The error checking and validating worked on the subscriber page.



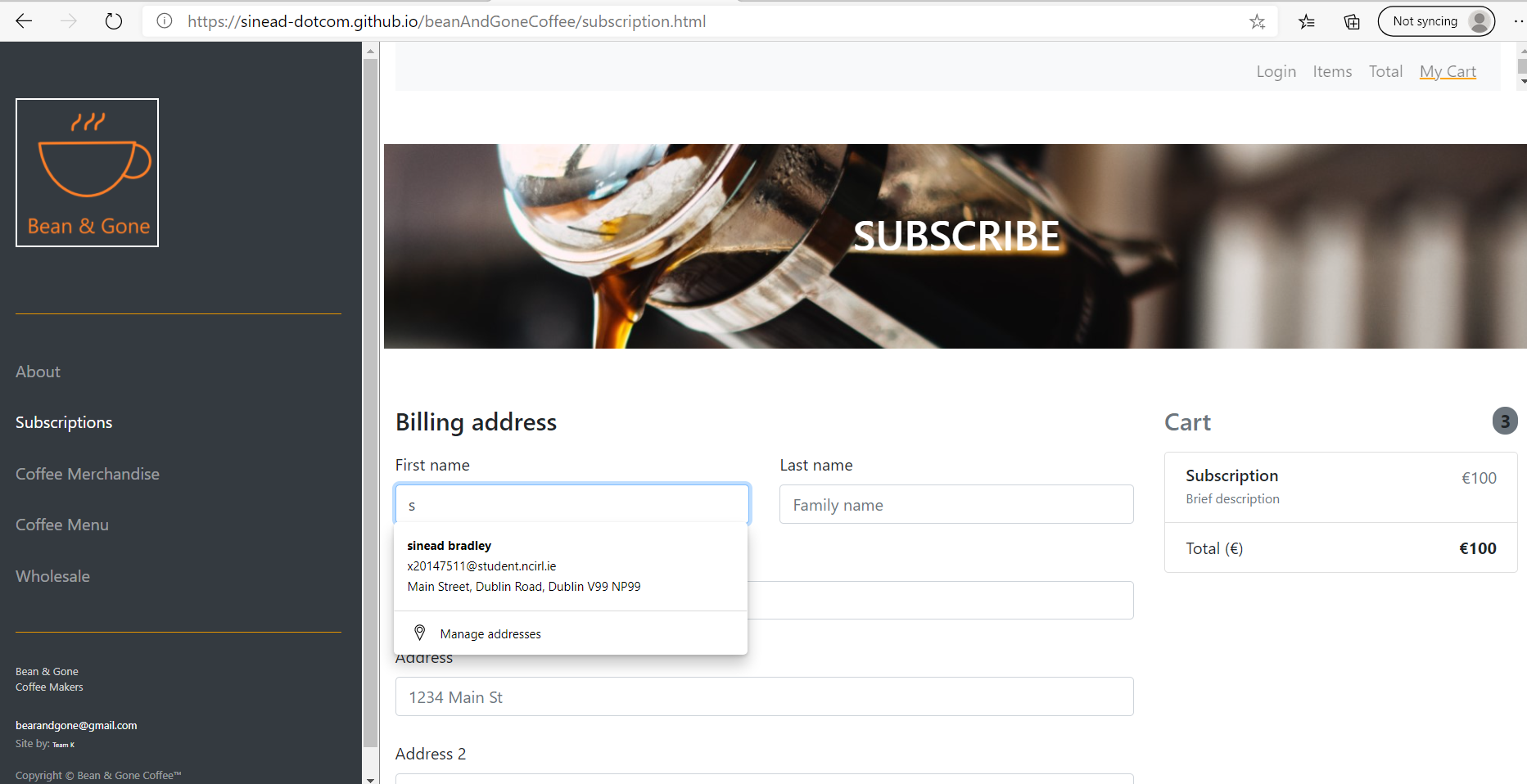
And once all fields were valid, and I clicked “Continue to checkout”, an email was launched to show the action.



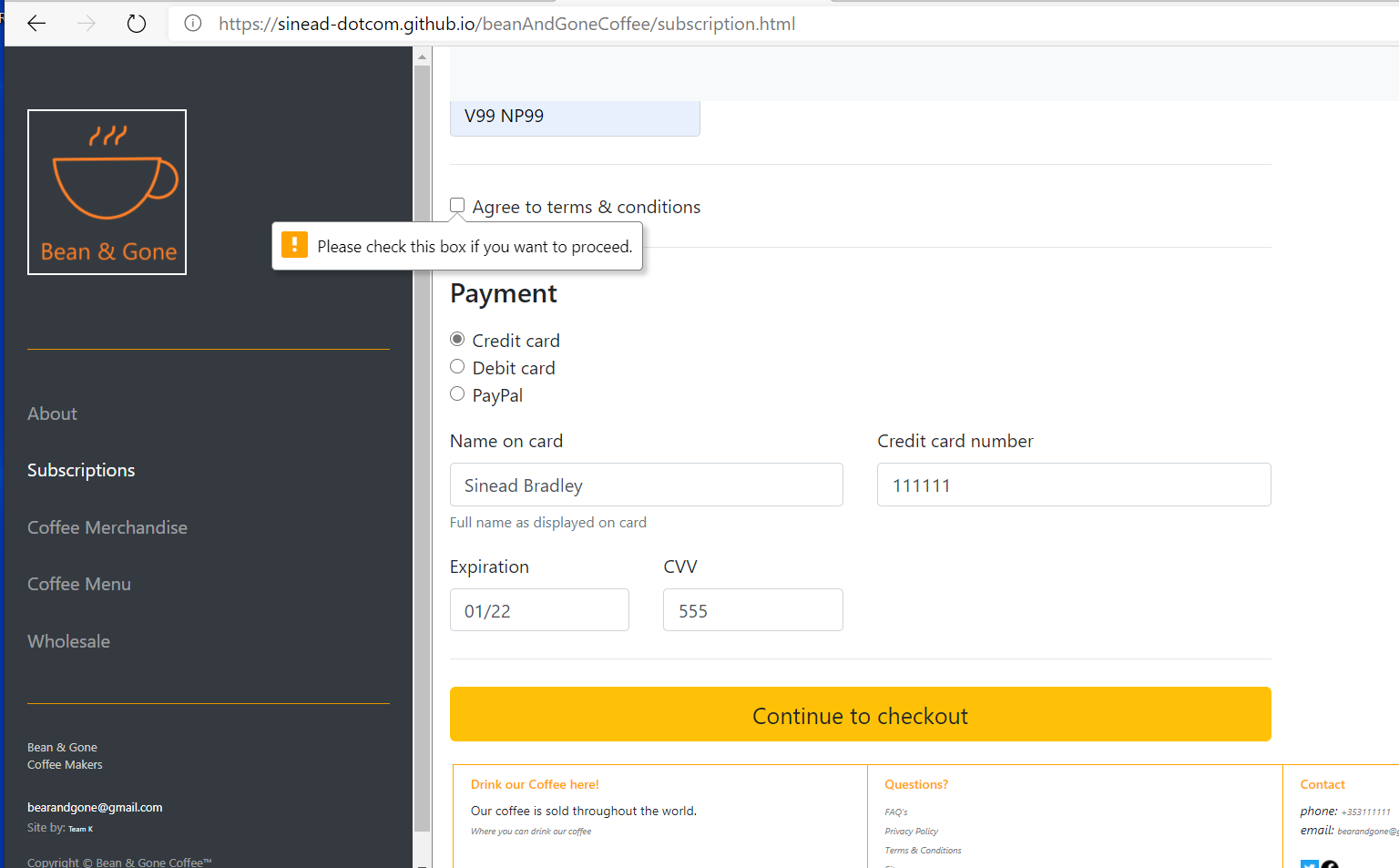
During the testing I noticed that the firstname field did not have a placeholder value:



It was lovely to see how the autofill worked:

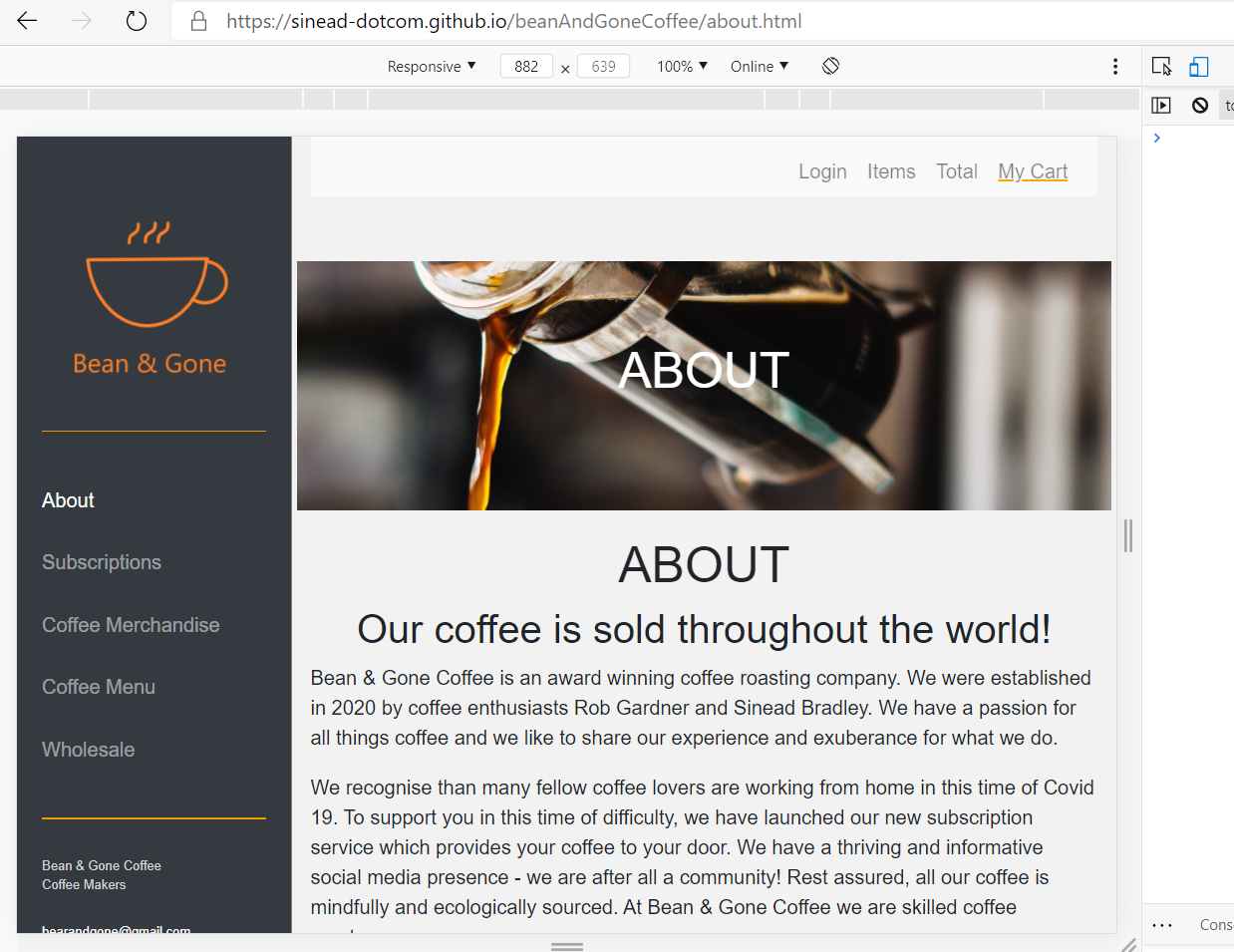


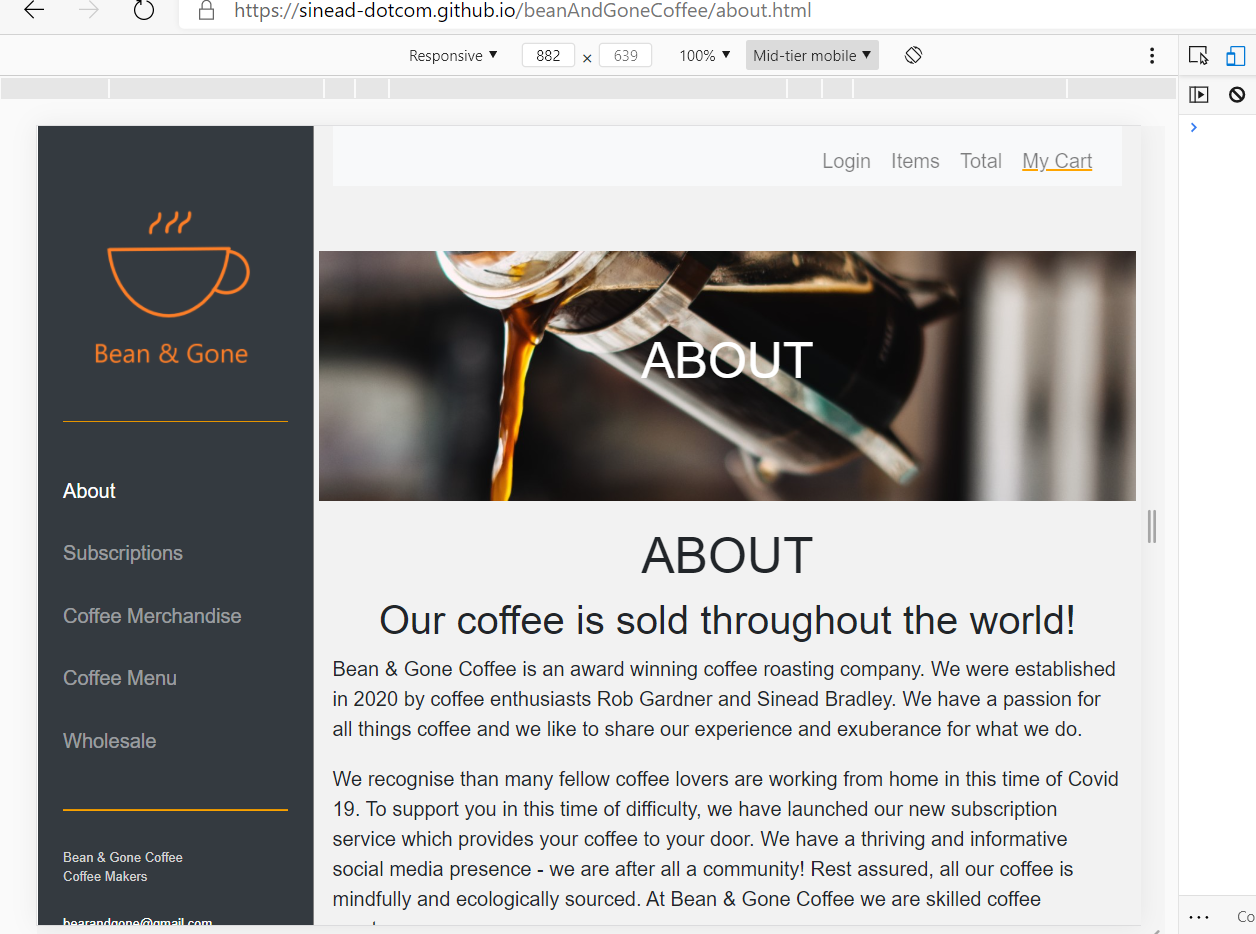
And the form made the user complete all fields before it would proceed:



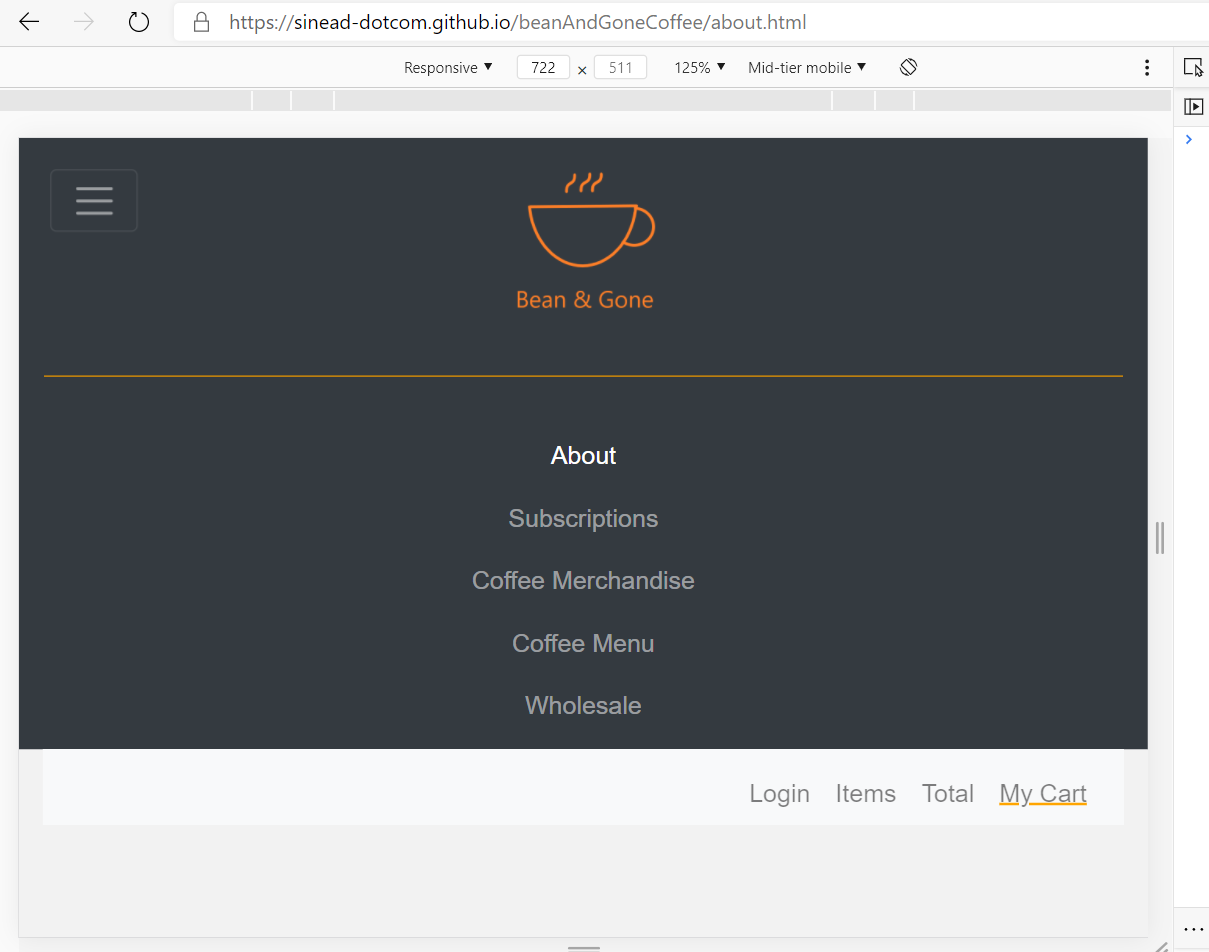
Device Emulation

Next we used the inbuilt browser developer tool to test the live sites using device emulation.



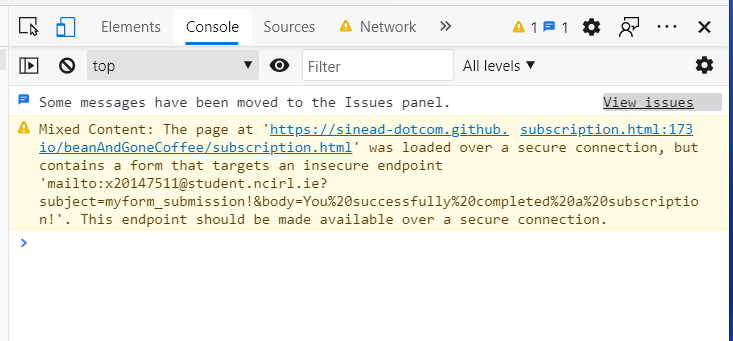


Also, checking what happened when Zoom was used on each device.

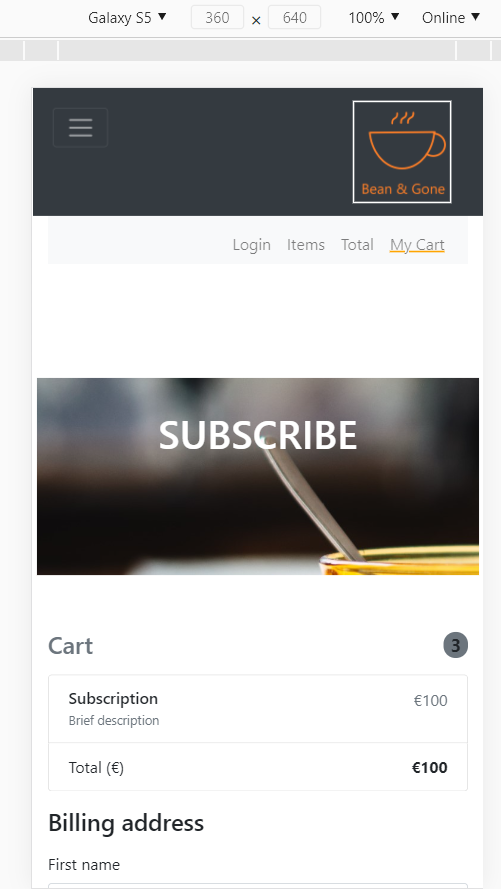


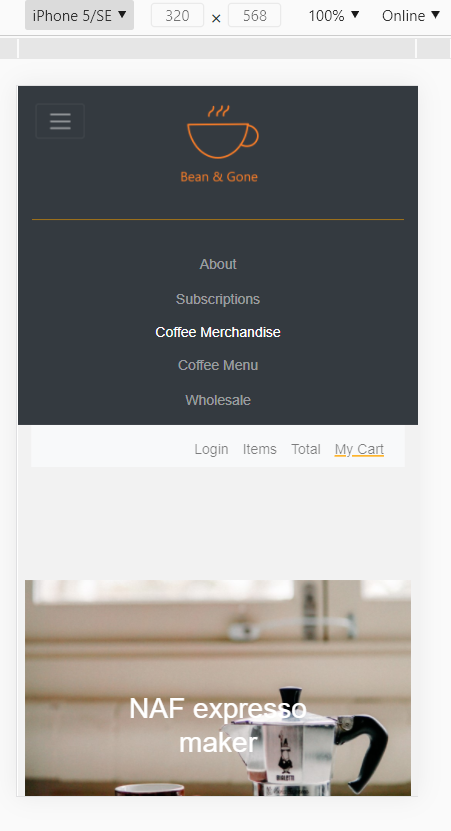
It nicely switched to the mobile – smaller setup! Perfect!

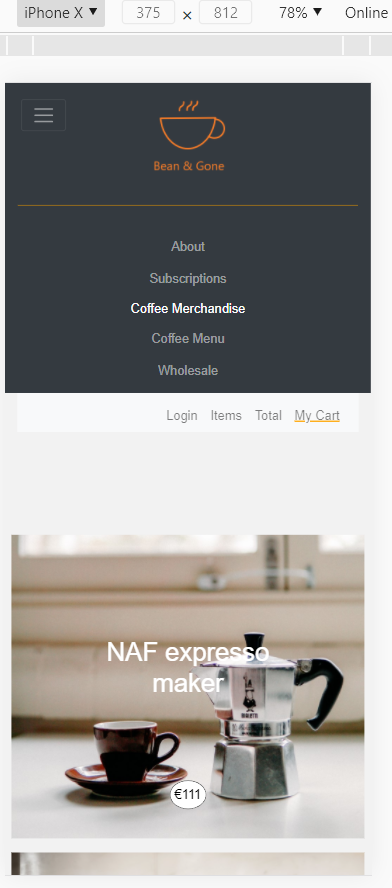
While doing this testing an interest warning came up:



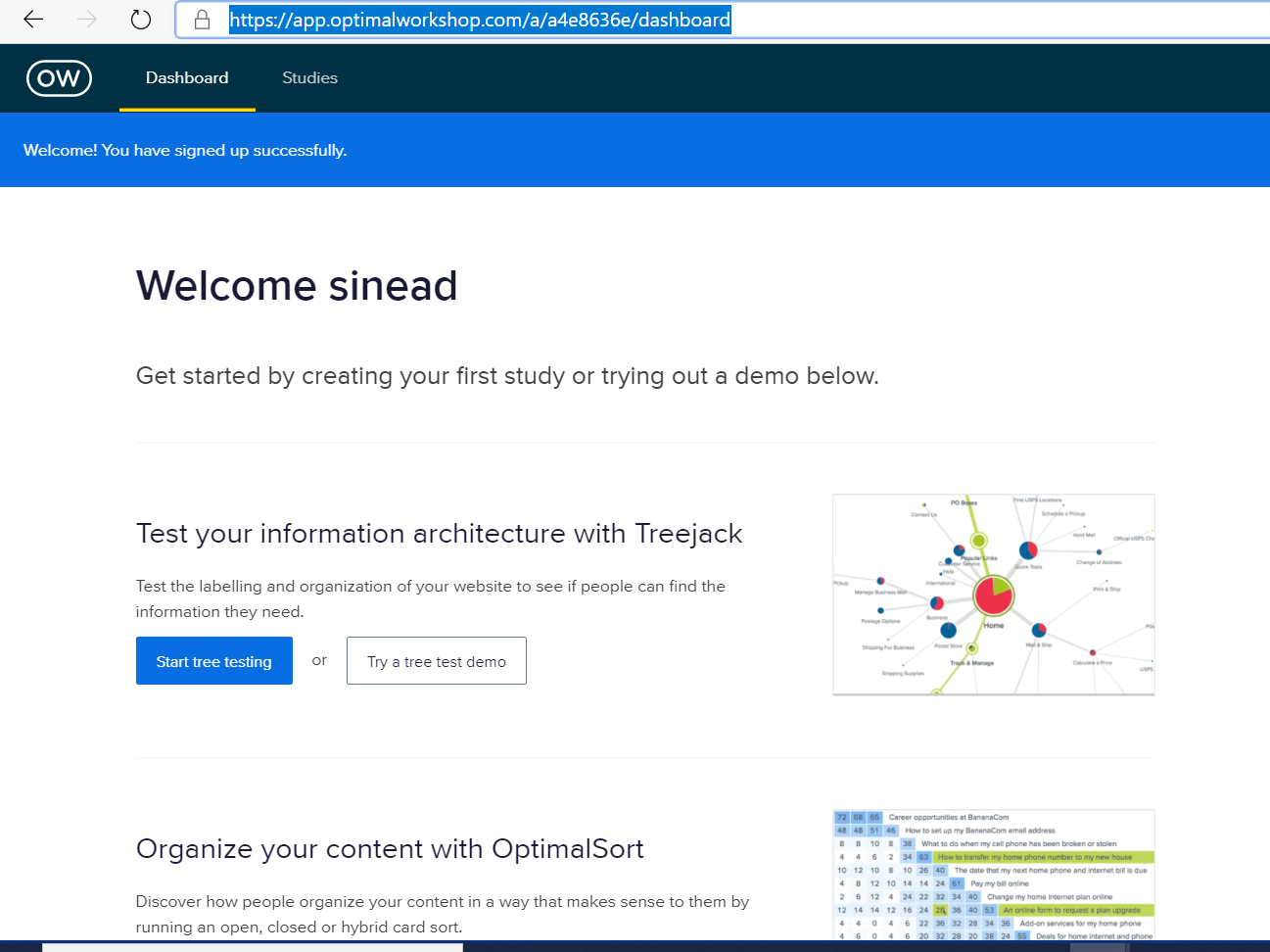
As, we do not use backend server functionality in this project, we are using email to demonstrate each step and show that the subscription checkout is working. Hence, the browser is warning about the use of email in this way! Definitely, something to be revisited next term when we do server side development.







Usabiliyt Testing



Unfortunately we didn’t know about this site until after we had started the development stage. Otherwise, it would have more useful to use Questions instead of Google forms and to use TreeJack and Chalkmark to automate the usability testing. We have already completed a survey phase and an interview phase where test users were presented with the wireframes, overall design, questionnaires and draft personas.

Standards:

Project was W3C CSS & HTML compliant.

Interface Testing

As we have no server backend in this project the interface testing consisted of testing the simulated backend. This was done by checking that cookies worked as designed (see above) and that emails were sent upon checkout/submit (see above).

Compatibility testing:

The web site was tested across:

Safari on MAC

Safari on iPhone devices (iPhone8…)

Safari on iPAD

MS Edge

Chrome

Firefox

NetMechanic deployment.

This seems to be now part of Dynatracecorp. The older download sites for netmechanic.com were not secured with HTTPS – so, there were not used.

I had problems deploying dyntracecorp agentless monitoring. This is definitely an area that will have to be revisited after this project is finished.

So, instead downloaded Loadrunner by Microfocus for load testing. Unfortunately, there were a load of errors when windows was trying to install the package! 30 minutes later and the windows installer is still struggling.

So, instead went to the <https://webpagetest.org/> site where I was easily able to do performance and image review testing on the websites pages! Please see the attached PDF’s which are included in the /doc directory of the project. Of enormous interest was the potential for compression of the images on the shop-merchandise.html page. Also, the web page performance test for index.html was of interest. There are multiple areas of improvement. It also highlighted that we did not have a favicon. We will try to address that today.

