**Technical report**

**PA3 Ajax + Personalization**

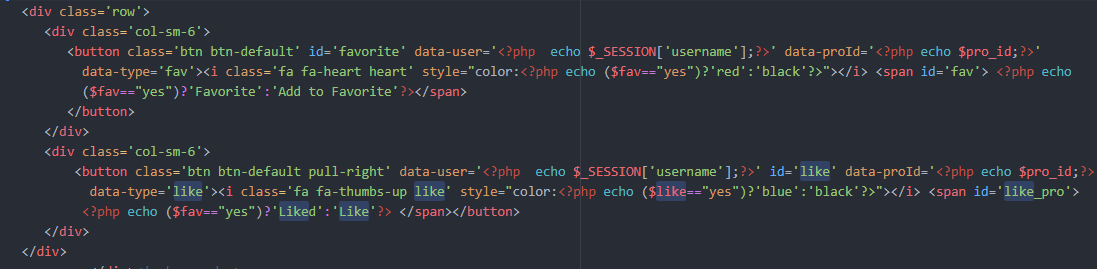
Being my first PHP and Ajax project, I learnt a lot in this course. A lot of new type of coding languages were introduced to us , which was daunting at first, but now after practice looks decent enough to work with.

Ajax- Asynchronous Javascript and XML works like magic without refreshing pages and fits really well into websites without any delay or lag in showing results.

My Project is called HealthSupps, a supplement store, with various categories of supplements, which also allows signed up members to choose and filter their results by selecting different characteristics of a particular category.

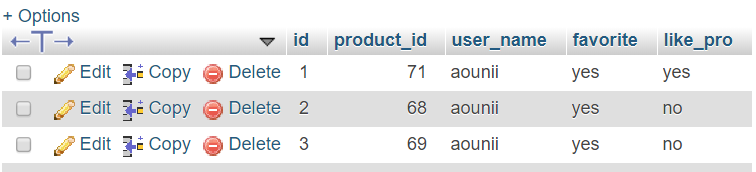
**PA3 Milestones**

1. **Personalization**

In order to add personalization in my code, I made some changes, in order to add a ‘like’ and ‘add to favourites’ button on each product description page. The HTML code of the two buttons is as follows: 

Both buttons use bootstraps calsses btn-default , with different IDs and have sessions starting in php code in them. In the php code, it echos that is ($fav==”yes”)?:Favourite:’Add to Favourite’. This checks if the add to favourite button is clicked, and if yes, then it adds the whole product description/details to the pro\_fav\_like table.

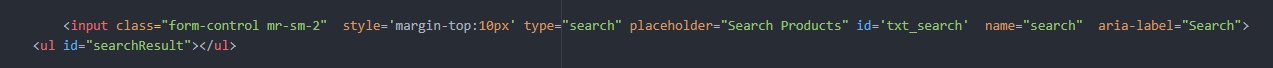
This pro\_fav\_like table has two additional fields which account for both the above buttons and if they are selected or not.



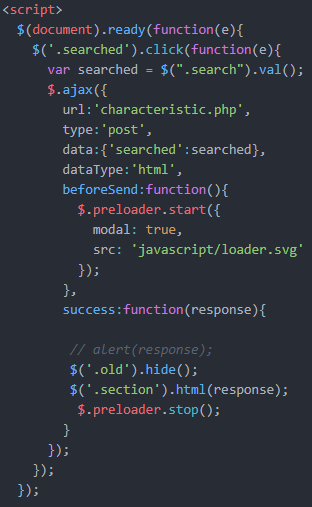
1. **Ajax Live search**

To implement this live search, which basically looks for products throughout the products table and sends result back to this input , has a number of steps which I used .

1. I call the header.php file in the main index.php file , in which the header contains an input with name=”search” and also has a <ul> under it which shows the query results as follows.



1. After this, I call the <script> tags in which I use jquery and run an ajax block which further refers to characteristics.php to get a response, and on further success, takes the response from the .php file and shows it in the search bar as specified.



Here, we use the famous (document).ready function, in which we have ajax, and url is characteristics.php. The type is post to retrieve the data and searched is the variable we make.

BeforeSend is the loading part we show whenever there is a small delay while loading.

We use DOM, so that the whole website loads first, and then the jquery part comes on.

$(old).hide part is to hide the old content on screen before the ajax call and immediately show new results using the $(section).html part.

**Learning Experience**

This was a great course and a great learning experience for me. The number one thing I learnt apart from all the new coding ways and tricks, is not to procrastinate and leave code for last minute, because implementing an efficient code block needs good time spent. I struggled in the course because of a silly mistake of missing the first project 1.

This was due to me being overwhelmed by the names and jargons of php when all I had to do was make a front end html code. This teaches me that it is important to not get overwhelmed by new code and work on it step by step.

**Challenge**

One of the major challnges for me in this term at the start was to do a simple GET and POST methods to retrieve data from a form, although looks very simple, but I struggled a lot with this and also the register and login forms. Another struggle was to implement the Live search, but after practice, it all seemed possible and looks decent in the implementation.