Code Review

Group 17

Using what we learned in the lecture on Software Verification, we are going to do a walkthrough for the nutrition code segment. We are going to conduct this walkthrough using the document driven method.

* Fault 1: The food tracker display stopped working.
* In the nutrition log, there is an add button that should display the food that was entered below the text box. When we are pressing enter, it is reloading the page which is not what is expected. This is preventing us from saving our input to the local storage, since our save function is tied to the click event of the save button being pressed.
* Regan identified this error. Since she is was working on the logs, we assigned her to fix the fault since she understood her written code best.
* Regan fixed the fault when she removed the </form> element. This seemed quite an easy fix but it took some time to understand what was exactly needed to be changed in order to fix the error. After identifying this error, we went on to test the other elements that were also formed based. We found another error belonging to the original submit button displayed below the soreness text box. We removed this button in order to prevent further unintended refreshing of our page.
* Fault 2: The nutrition log stopped working.
* The nutrition tracker’s save function no longer worked correctly and did not save to the intended variable.
* Regan also identified this error. We assigned Natasha to fix this fault since she has a solid understanding of JavaScript and we all agreed she would be able to find a solution.
* Natasha was able to fix the fault. The tracker was working properly after she removed the link from index to JavaScript and removed the if statements that prevented null. Those if statements were originally placed to prevent a null error from occurring on the index page, which did not have the variables that needed to be accessed. She moved the index JavaScript to its own JavaScript page, removing the need for the if statements. After these changes were made, the tracker was working fine.

These were the two main faults we found within our nutrition log code segment. Other than that, we all agreed that the code was well written. We condensed and cleaned up the code to ensure we do not sound repetitive. We tried to use what we had learned in class and applied it in our code structure.

These are the features for which we had faults and we were able to go and fix them. However, there are a few other faults which we identified, but plan to fix for the next steps we discussed throughout our project of building an application for the website. Our goal in future would be to get the food log to be a part of the weekly summary as well. What we currently have does not include it and we wish to advance our website with this functionality. Additionally, in the nutrition log, where we have the blank for the description, we should only be able to input letters/word(s). However, currently we are able to input anything such as other characters and it does not give an error message saying that it should only be words. This would be something else we will fix when we go further with the project. Adding functionality for separate calorie, macro, and nutritional tracking would also be beneficial to our application.

Another area that causes faults is in the local storage. Local storage has limited capacity and capabilities so we weren’t able to implement everything exactly as we had planned. We had trouble with reloading the data on the day.html pages but had no issue when doing this for the index.html page. We believe that this is because there a many different types of values needing to be stored on the day pages vs the index page that only requires the reload of text areas. However, despite the local storage limitations we were still able to provide the weekly summary as we had planned. If we were to take this project a step further, we would look into different ways to store the data that would allow for more security and flexibility for the users.