**Appendix 1: Planning Guide**

Task 0 : Explain what you are doing/ going to accomplish

***In this version I will be creating a function that allows me to add 1 stock at a time***

Task 1: Sketch interface design

*Draft a rough design for the interface that allows the user to trigger functionality in task 1, while also annotating where the information in task 2 will be displayed. Create another sketch listing the interface widgets used to create the interface.*

Task 2: Identify any classes required

*Explain what the class will represent, plus listing what information will be stored in the class and any functions the class will have.*

**N/A**

Task 3: Identify information to be displayed

*What information will the interface need to display to the user?*

***Picture of the food and the name of the food will be displayed to the user. There will also be an add stock button displayed and a restock success page***

Task 4: Identify user inputs

*What program functions can the user trigger through the interface?*

**User will be able to choose which product they want to be able to restock**

Task 5: Identify any constants or existing data if required

N/A

Task 6: Identify indexed data structures

Menu\_list – Will hold a list of all my food

Task 7: Determine what calculations are necessary

*Write out the calculations the program will have to compute.*

**Program will have to add one to stock to the food of the users choice**

Task 8: Develop a modular structure for your program

*Describe any functions that the computer program will have, identifying any sub-functions where required.*

I will need to have a function that adds one to the ‘found\_item.stock’ (the users food choice)

Task 9: Define the functions identified

*Describe the functions for both the main program and any classes in terms of input and/or output where required. You may choose to do this with flow charts or pseudo-code (not Python code!). Add in additional steps or explanations using sequential, conditional, iterative statements where required. Identify global and/or local variables.*

AT ROUTE ‘/restock/<food\_id>

AT VIEW ‘restock-success’

PROGRAM restock\_success (item\_id)

SET item\_id TO int(item\_id)

SET found\_item TO NONE

FOR item IN food:

IF item.id EQUALS item\_id

SET found\_item TO item

SET data TO dict(item EQUALS found\_item)

SET found\_item.\_stock TO plus 1

RETURN data

Task 10: Address any relevant implications such as usability, functionality, legal/ethical requirements.

Task 11: Document test cases for testing the program

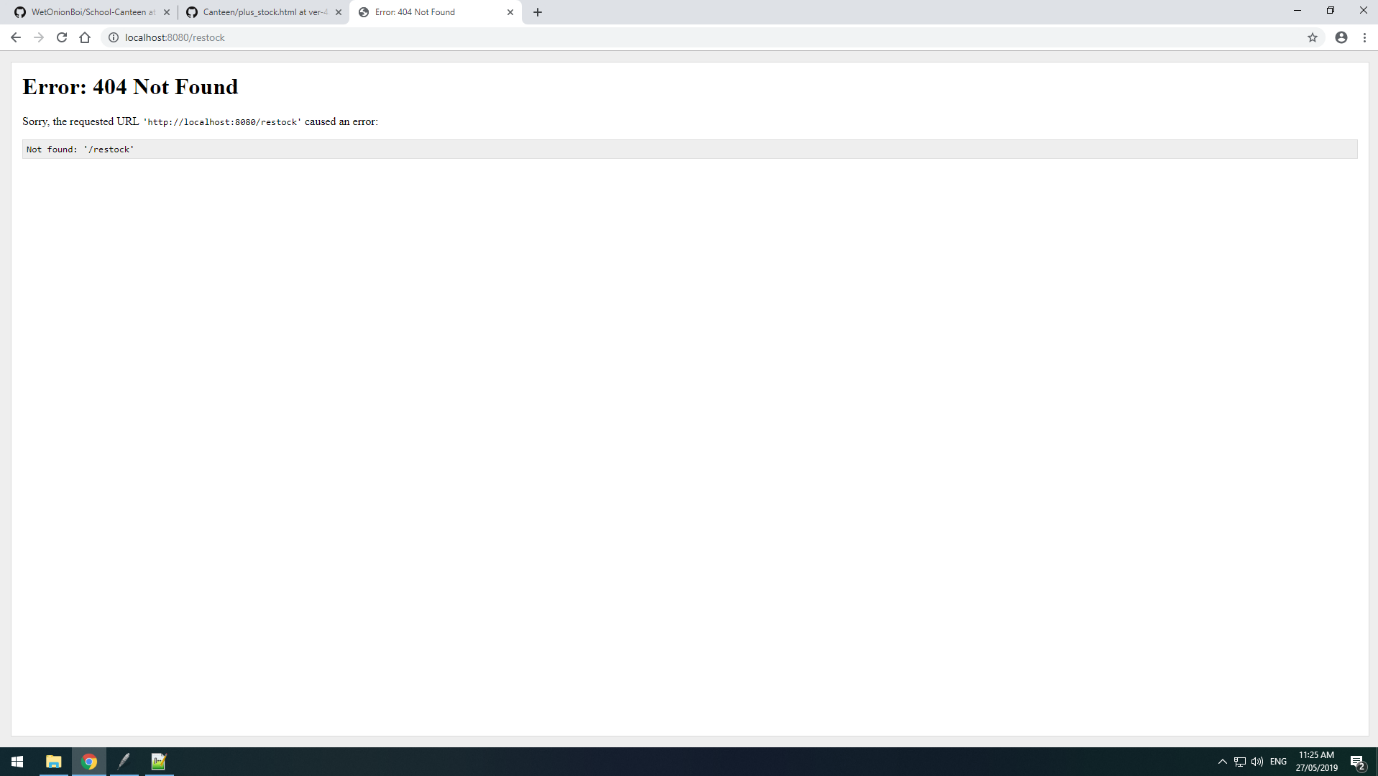
*Document any testing that can be used to test your program. If any input is inputted using the keyboard, describe the expected input, plus any exceptional, boundary or invalid cases.*

I will test this version by running my website and python code and try to access the page. After this I will click restock and see if it sends me to a success page and then adds one to the chosen items stock. I will try to input letters or symbols into the box although, this should not work as the input field only accept integers.

Task 12: Refine the plan

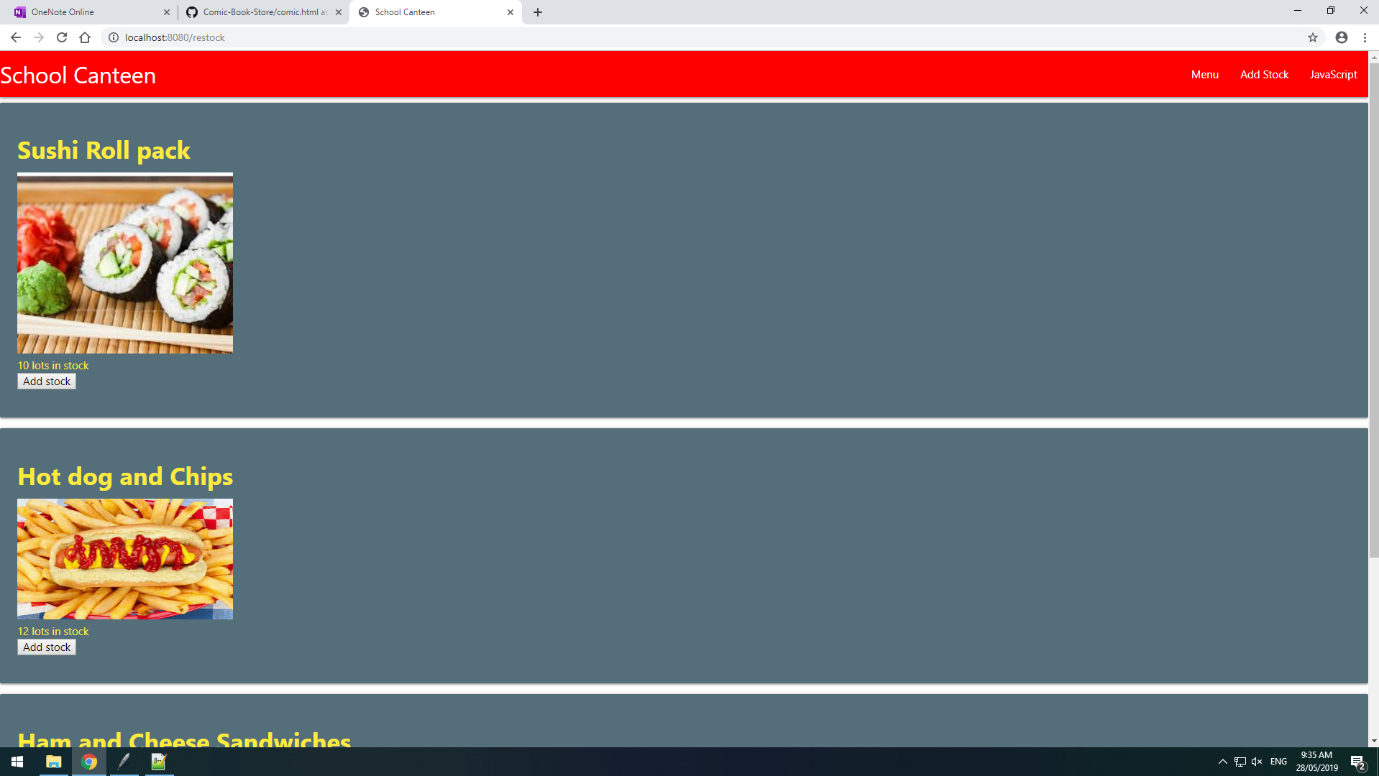
*Note any modifications here when iterating through the development cycles.*

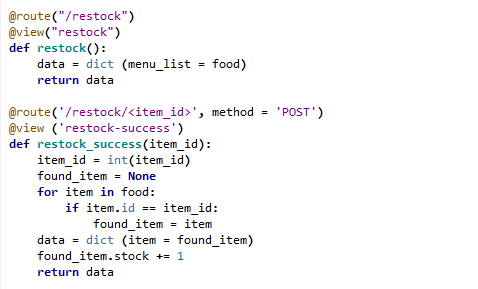
**When creating this version I forgot to add a function in python to show my menu and only created a success function in python.**

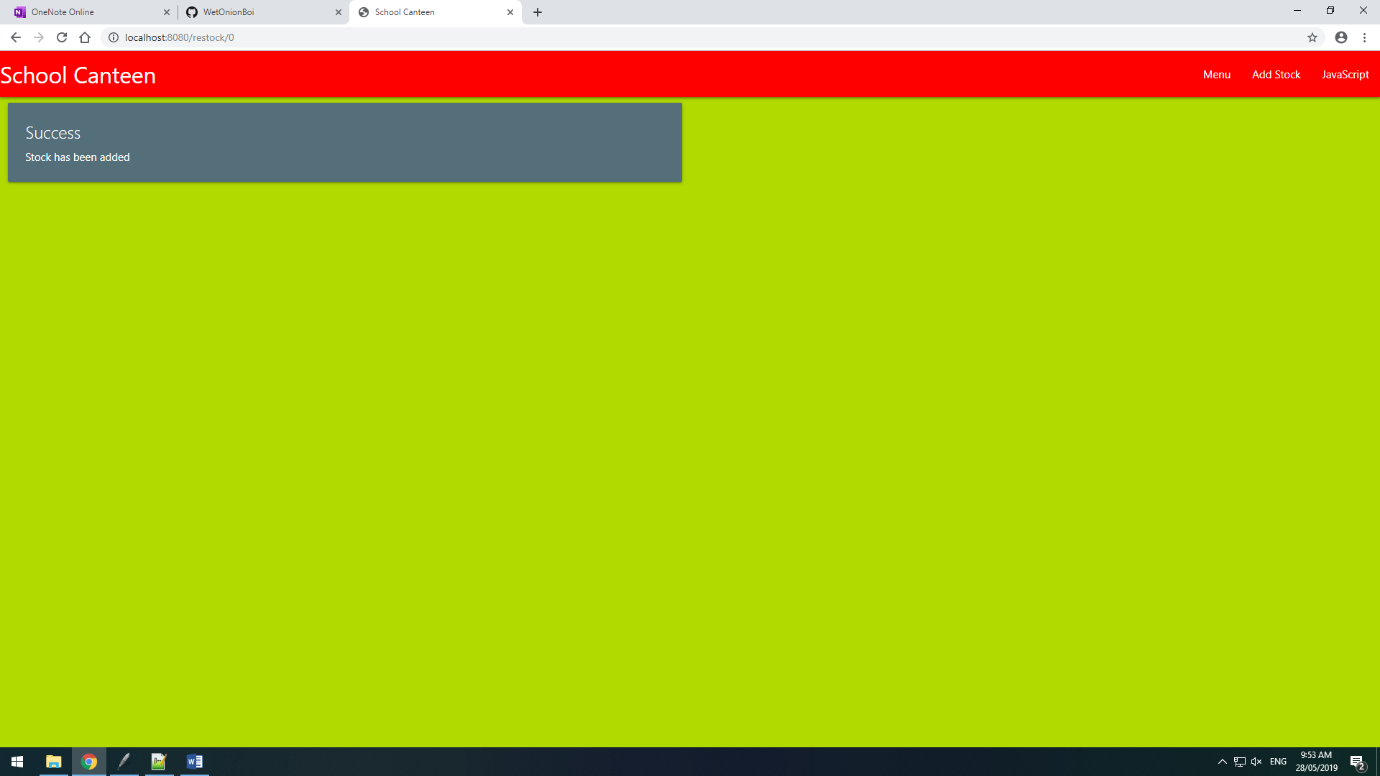
****

Task 13: Document testing

*Show screenshots of your program working with descriptions of each image. These images should test the tests cases listed above.*







Task 14: Evaluation

*How did your version turn out?*

After fixing the problem where I didn’t have a function for my actual restock page the code works exactly as planned in Task 0. In the next version I will be looking to add a way to have more than 1 stock being added at once.