**Appendix 1: Planning Guide**

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

In this version I am going to create a purchase button and purchase success page on the success page there will be a function that automatically sends them back to the menu after a few seconds

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.*

N/A

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.*

Will edit the existing class to -1 from the stock

Task 3: Identify information to be displayed

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

I will need a button to be displayed to the user when there is more then 0 items left in stock for them to buy

Task 4: Identify user inputs

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

Users can choose to purchase an item of the menu

Task 5: Identify any constants or existing data if required

N/A

Task 6: Identify indexed data structures

N/A

Task 7: Determine what calculations are necessary

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

Program will need to minus one from stock

Task 8: Develop a modular structure for your program

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

Def purchase\_success: This function will -1 from stock when user clicks to purchase something

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 ‘/purchase-succcess/ <food.id>

AT VIEW ‘purchase-success’

PROGRAM purchase\_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 -= 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 am going to test this version by running it and seeing if the success page pops up and then reroutes

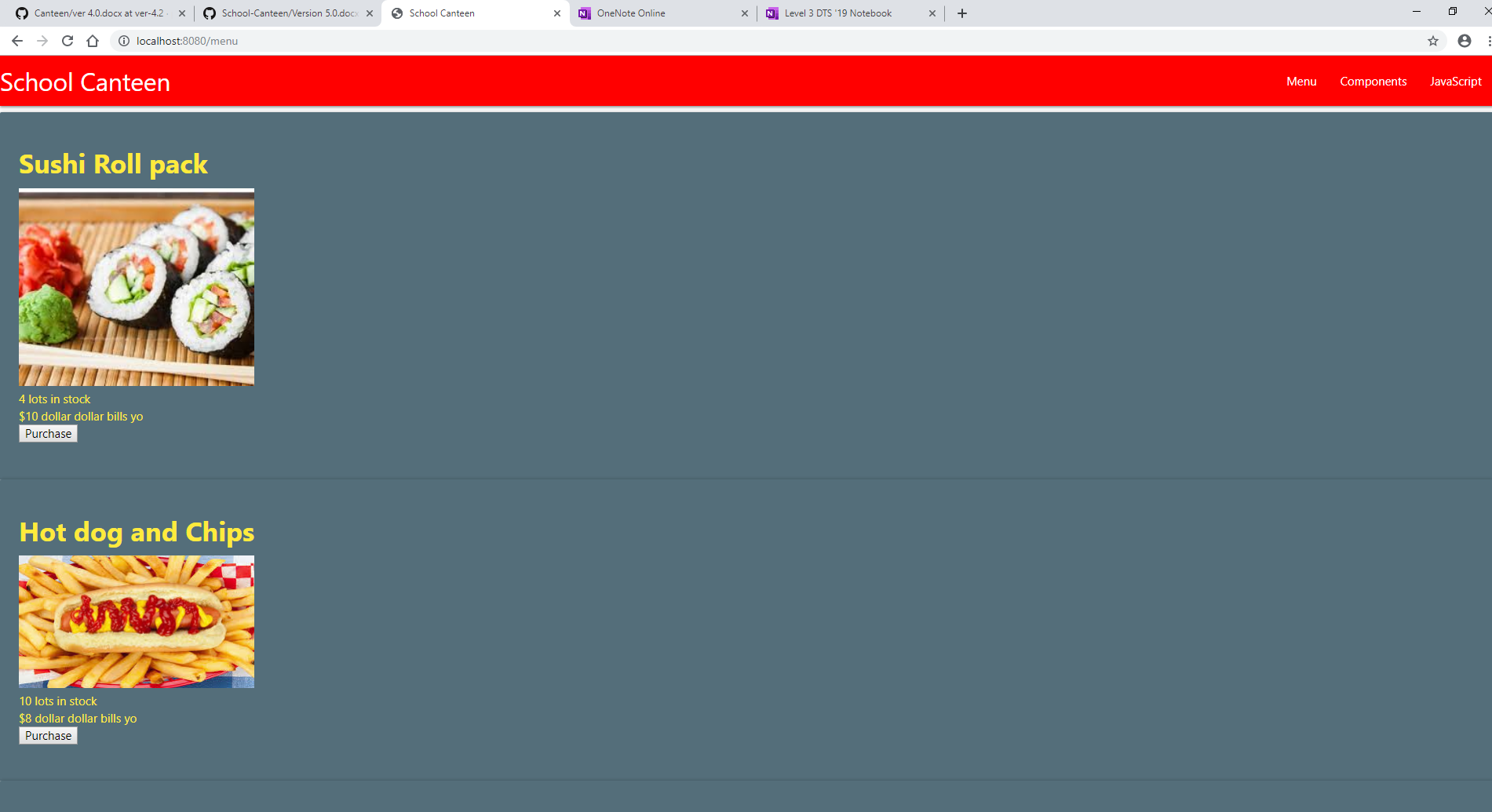
Task 12: Refine the plan

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

I had to change some variable names as the website didn’t load up properly at first because it was getting confused with the variable names.

Task 13: Document testing

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



Website works fine after I changed the variable names. The purchase success page also works and reroutes back to the homepage a few seconds after you click.

Task 14 : Evaluation

*How did your version turn out*

This version worked well after I changed some of my variable names. It successfully goes to a success page after purchase is selected and then it rerouted to my menu again after the purchase.