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

In Version 3.0 of my code I will create my product page which will display all the food items I have for sale.

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.*

Task 3: Identify information to be displayed

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

I need to display what each of my pieces of food look like, how much each of them cost and the amount of stock they have left.

Task 4: Identify user inputs

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

None in this version

Task 5: Identify any constants or existing data if required

Task 6: Identify indexed data structures

I will create a dictionary containing the information about each of my food items.

Task 7: Determine what calculations are necessary

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

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 have a function called product page. In this I will have a dictionary of all my food items.

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.*

PROGRAM product\_page

Make data a dictionary of all my food items and the data associated

Return data

END

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

My webpage contains cards on each item of food. These contain the name of each item, a picture of the item, the price and the stock. This page is the colour of other school websites. It is appeasing on the eye and is easy to navigate as every label and link is easy to understand and stick out. This makes it easier for the user to understand the purpose of my webpage and does not confuse whoever is viewing my page. There are not many relevant implications because this is a relatively small version which only displays items on sale. I do have to make sure it is obvious where the information on each item is though.

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.*

Error 1: my cards did not work correctly. I could not find out why my cards were not working properly. I decided I could make my code simpler and also fix it by deleting it and rewriting it.

Error 2: pictures not appearing. The issue was I had *@route("/picture/<filename>")* instead of *@route("/images/<filename>").*

I had an issue with every card being indented more than the previous one.

Example

I

I

I

Task 12: Refine the plan

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

I added a picture to each of my food items. I did this by adding more to my class and entering the below at the end of my code.

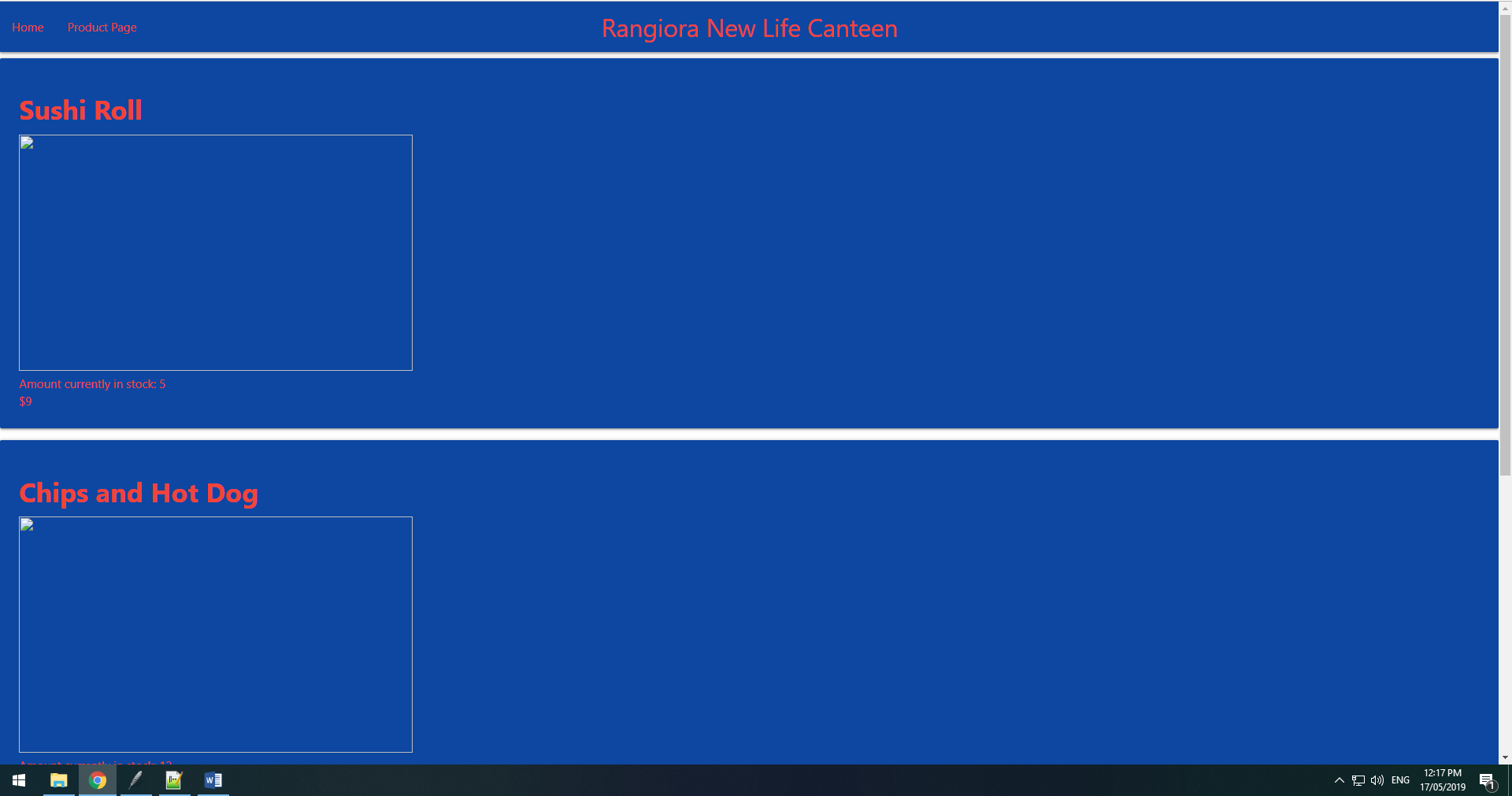
*@route("/images/<filename>")*

*def serve\_picture(filename): #need this for images to work on my website*

*return static\_file(filename, root ="./images")*

Task 13: Document testing

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



This is what I mean when I say pictures not appearing. The issue was I had *@route("/picture/<filename>")* instead of *@route("/images/<filename>")* as I said before.

Task 14: Evaluation

*How did your version turn out*

This version worked out as planned. I had a few errors as stated above but overall it came out as intended. I also addressed functionality and usability above. Overall there was not much testing and implications or anything because it is just displaying item names and images and choosing colours for my page.