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

In this version of my code I am going to create my index, navbar and header pages. This is the base layer of my website, the fundamentals. I will also draft the layout of my page.

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.*

None added in this version

Task 3: Identify information to be displayed

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

In this version I will display the header and navbar on my page. This will contain information like the name of my canteen and a link to the purchase page (eventually).

Task 4: Identify user inputs

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

At this point the user can not interact with my page.

Task 5: Identify any constants or existing data if required

Task 6: Identify indexed data structures

Task 7: Determine what calculations are necessary

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

N/A

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 index. The only thing I will be using this function for is to attach a route and view function to it. This allows me to make a webpage.

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.*

Make route /

Make view index

PROGRAM index

pass

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

With my webpage I need it to be appealing on the eye and not confusing to navigate. This is because it makes it easy to understand the purpose of my webpage and does not confuse whoever is viewing my page. The thing this version accomplished was creating a foundation for the rest of my code so there are not many implications as such.

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.*

There is no real testing I had for this version of my code. It worked as intended first time and there is not much to test other than if the page loads.

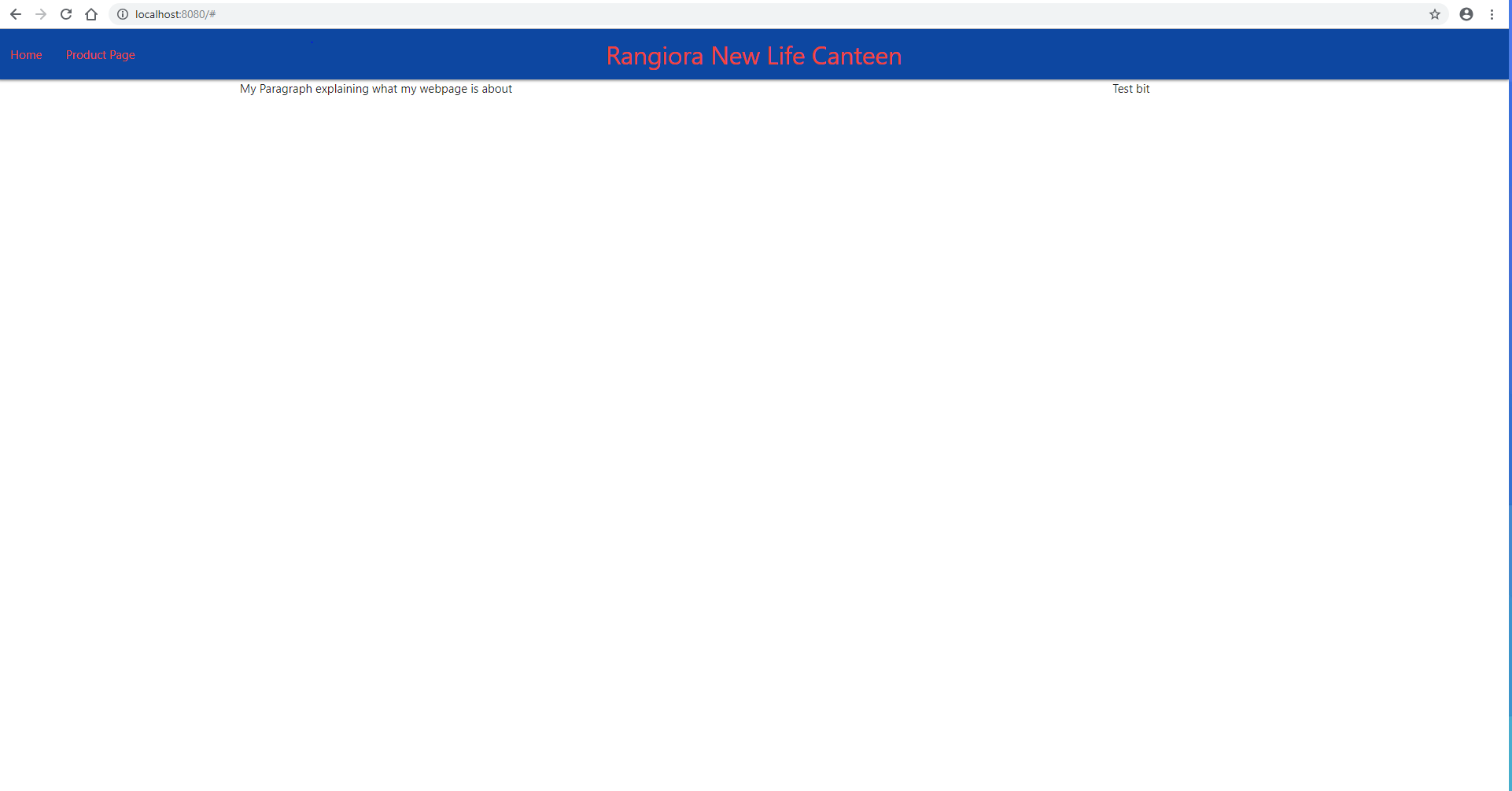
Task 12: Refine the plan

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

N/A

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*

My version works as intended. My index, navbar and header code all perform their intended purpose. I struggled to change the text colour to red in my navbar code. I solved this by putting red-text inside a tag property called class. Other than this I had no problems.