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

I am going to create a server in a python page which will contain the data for my food items.

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.*

Not applicable in this phase of planning

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.*

Class:

Food\_Items

Task 3: Identify information to be displayed

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

None in this phase of my code

Task 4: Identify user inputs

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

None in this phase of planning

Task 5: Identify any constants or existing data if required

Class: Food\_Items –

Sushi Roll pack – 5 in stock

Hot dog and Chips – 12 in stock

Ham and Cheese Sandwiches - 4 in stock

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.*

\_\_init\_\_ holds the data for the class: 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.*

Class Food\_Items

Set ids count to 0

DEFINE \_\_init\_\_ PASS self, name, image, stock

Increase ids count by 1

Set name to self.name

Set image to self.image

Set stock to self.stock

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

N/A. 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.*

N/A. I can not test this beyond if it doesn’t throw an error. It didn’t.

Task 12: Refine the plan

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

I added run(host ='0.0.0.0', port = 8080, reloader = True, debug = True) to the end of my code. This runs my page server.

Added bottle functions:

run, route, view, get, post, request, static\_file

And count from itertools

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*

This version worked just as designed it. There were no floors or errors. It successfully runs my server. I added more functions that I will use later and added a line of code at the end which runs my server. The thing this version accomplished was creating a foundation for the rest of my code so there are not many implications as such.