Version 2.0

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

In this version I will create a “showcase” page for all of the dogs to show up on. This is where users will be able to browse the dog options

Task 1: Sketch interface design

Task 2: Identify any classes required

Class Dog

Will need the variables, Name, Age, Description, Breed, Available, Gender

Task 3: Identify information to be displayed

This page will display the Dog class and all of its variables in the dog\_list

Task 4: Identify user inputs

No user inputs for this version

Task 5: Identify any constants or existing data if required

No constants for this version

Task 6: Identify indexed data structures

Data dictionary called dog\_list is used to get the dogs.

Task 7: Determine what calculations are necessary

No calculations needed in this version

Task 8: Develop a modular structure for your program

Set route to (‘showcase-page’)

Set view to (‘showcase’)

Set showcase function to:

Set var data to be dictionary of dog\_list

Return data to page

Task 9: Define the functions identified

Index function passes as there is nothing to send to the index page in this version.

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

Within this version, I will need to create a website interface that is easy to read and simple to understand as many of the users may be older/unfamiliar with technology. I also need to follow the general rules of design when it comes to websites with colours layout etc. Buttons need to be clear and laid out, everything should make sense.

It need to be functional, it should first fufill its purpose and secondly look aesthetically pleasing,

No copyrighted images. No illegal or explicit images etc.

Task 11: Document test cases for testing the program

Load, “localhost:8080”, and see if the page show up to how I imagined it within the interface design sketch.

Task 12: Refine the plan

VER 1.1

I realised I needed to be able to add my own CSS as well as images to my website using bottle so I added code to be able to do that. This code I copied from my previous bottle projects as it it flexible over any bottle framework.

# Images

@route('/image/<filename>')

def server\_static(filename):

return static\_file(filename, root='./assets')

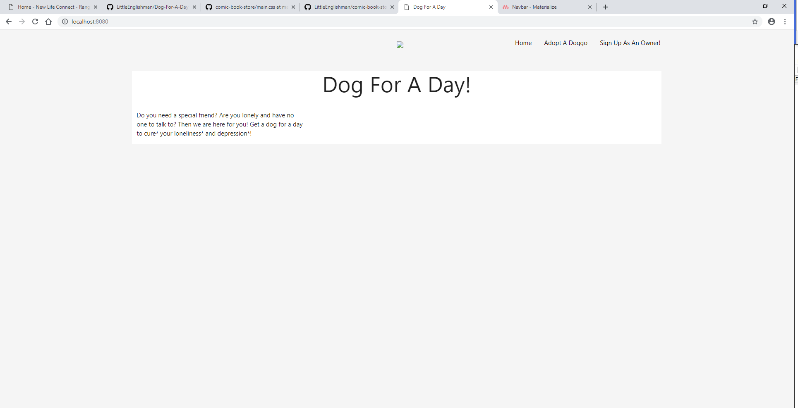
#Code to be able to link custom css Ver1.1

@route('/<filename>.css')

def stylesheets(filename):

return static\_file('{}.css'.format(filename), root='./assets')

Task 13: Document testing



Task 14: Evaluation

This version was successful in fulfilling its functionality of a working web page. I have decided to leave the aesthetics till a later update so that I may first focus on the functionality of the webpage.