Version 7.0

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

In this version I will make a way to return the dog.

Task 1: Sketch interface design

Task 2: Identify any classes required

Class Dog

Will need the variables, Name, Age, Description, Breed, Available, and Gender etc.

Class Owner. To make the owner return the dog.

Task 3: Identify information to be displayed

This will need to display the dog’s information as well as whether they are in use or not.

Task 4: Identify user inputs

Clicking on the “return” button will set the dog as “returned”

Task 5: Identify any constants or existing data if required

No constants for this version

Task 6: Identify indexed data structures

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 (/return-page)

Set view to (“return-page”)

Define return page as:

Data dictionary called dog\_list

Data dictionary called person\_list

Set data to dictionary of dogs

Return data

Set route to /return-success<dog\_id>

Set view to /return success

Def return-success as:

Set integer of dog.id to dog\_id

Found\_ dog set to noe

For dog in dog\_list:

If dog\_id matches dog\_id

Set found\_dog to dog

Set found\_dog.availabile to 1.

Return found\_dog

Task 9: Define the functions identified

Return-dog and return-success defines each personal dogs pages

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 fulfil 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

Also try returning a dog and see that it becomes “available” after it has been returned.

Task 12: Refine the plan

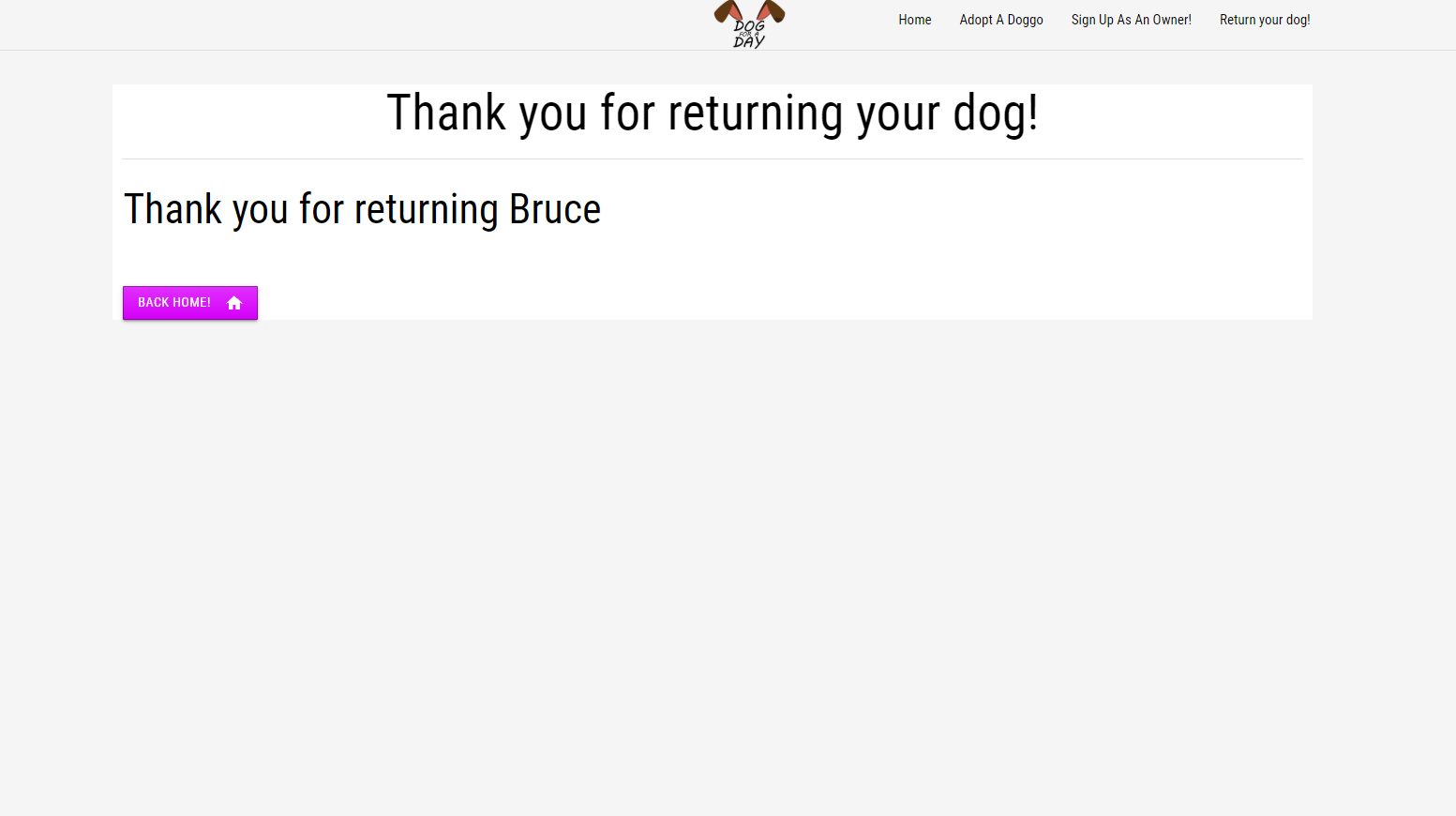
Within testing this version, I realised that I had accidentally taken out the code that makes the dogs unavailable when you rent them out.

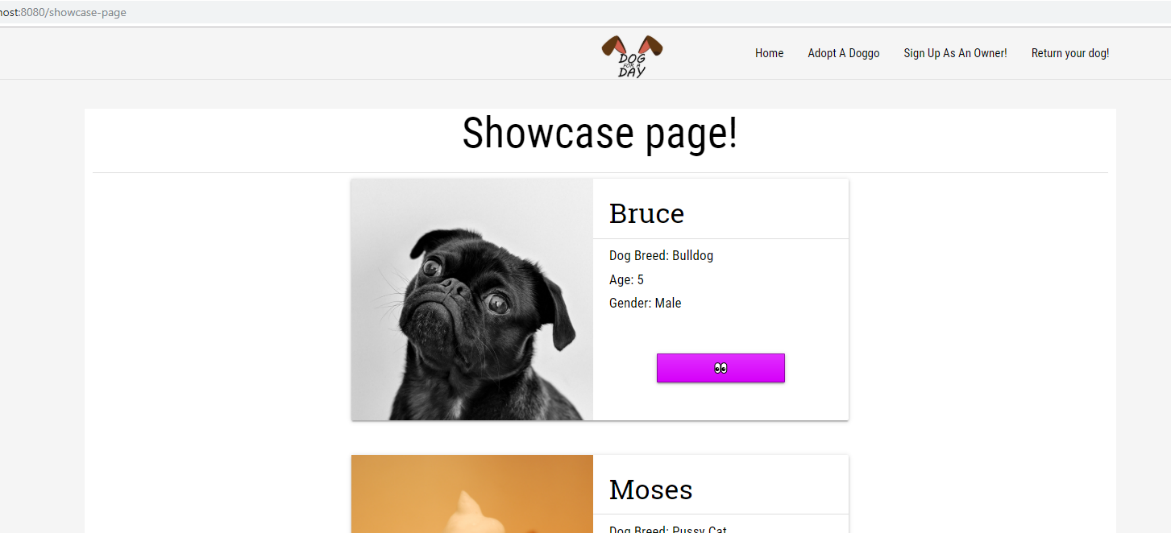
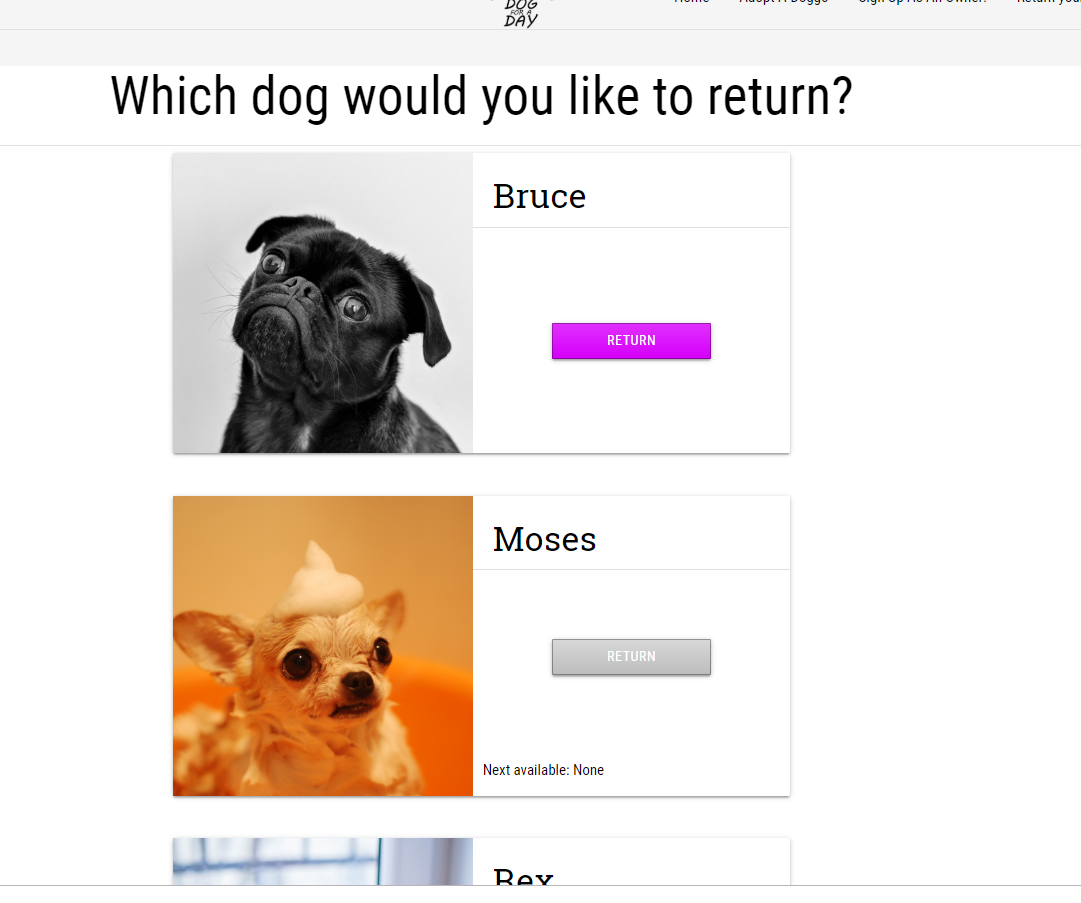
So I added in this code

Set found\_dog.available to 0

This fixed the problem.

Task 13: Document testing





Task 14: Evaluation