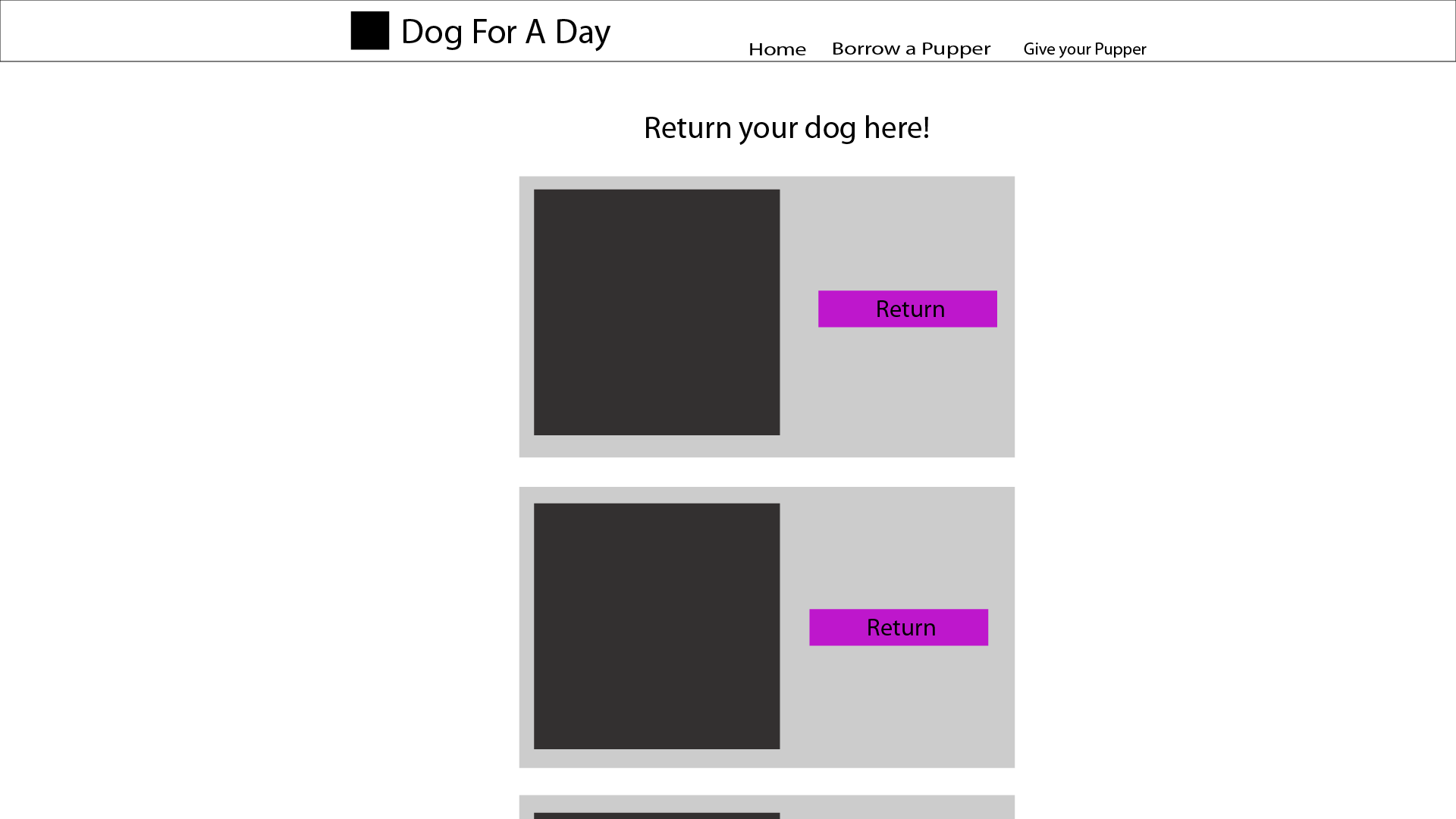
Version 7.0

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

In this version I will make a way to return the dog. As well as a success page to show the dog has been returned.

Task 1: Sketch interface design



This page follows a very similar design to the showcase page. However with less information as the user is not looking for info about the dog, only a name and pic to recognise them.

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

This version accesses the dog\_list as well as person\_list

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 the return pages for each dog. Return dog shows a list of the dogs so you can select which one to return. Return success does the action of returning the dog.

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, this should come from the pages earlier so it should be fine.

It should be as easy as possible to return a dog as I can make it. That is why I made the page similar to the showcase page, so it has repeated design features which make the user feel “comfortable” returning a dog. This allows for the program to be efficient to use and give the user freedom and control over the program.

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

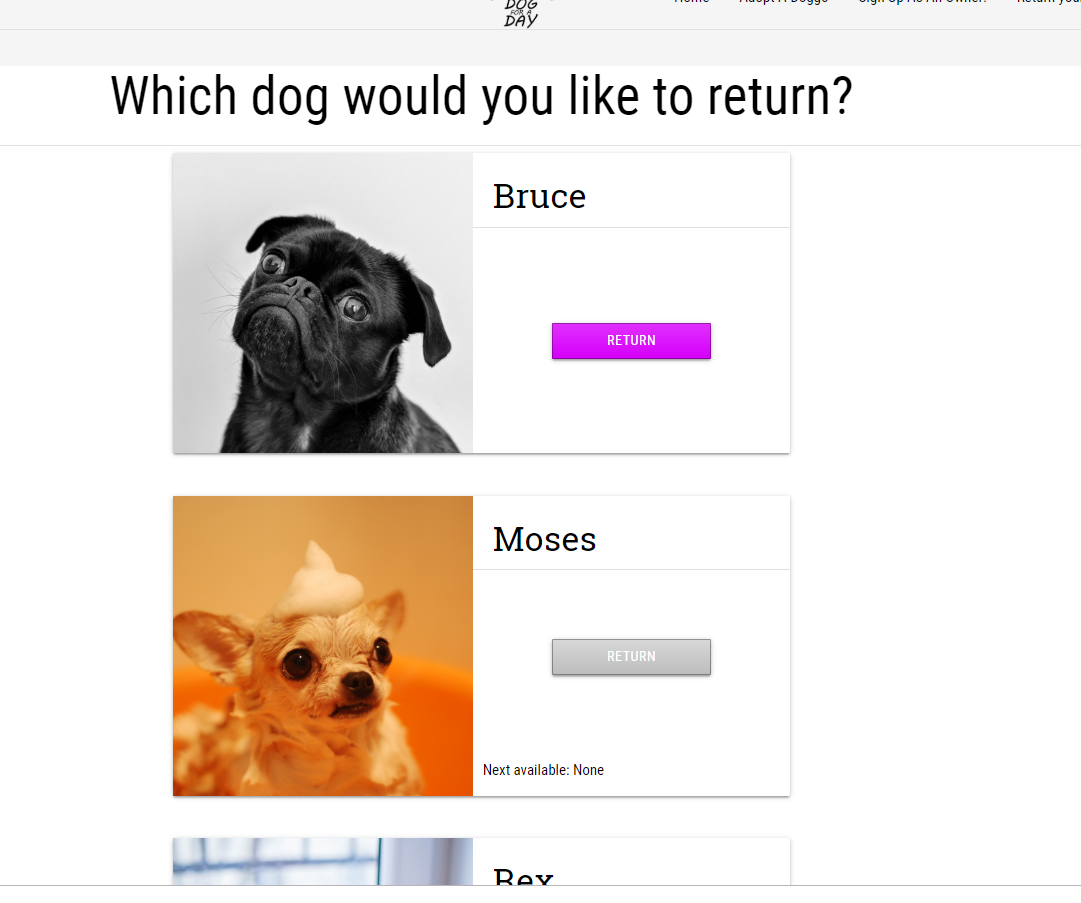
Set found\_dog.available to 0

This fixed the problem.

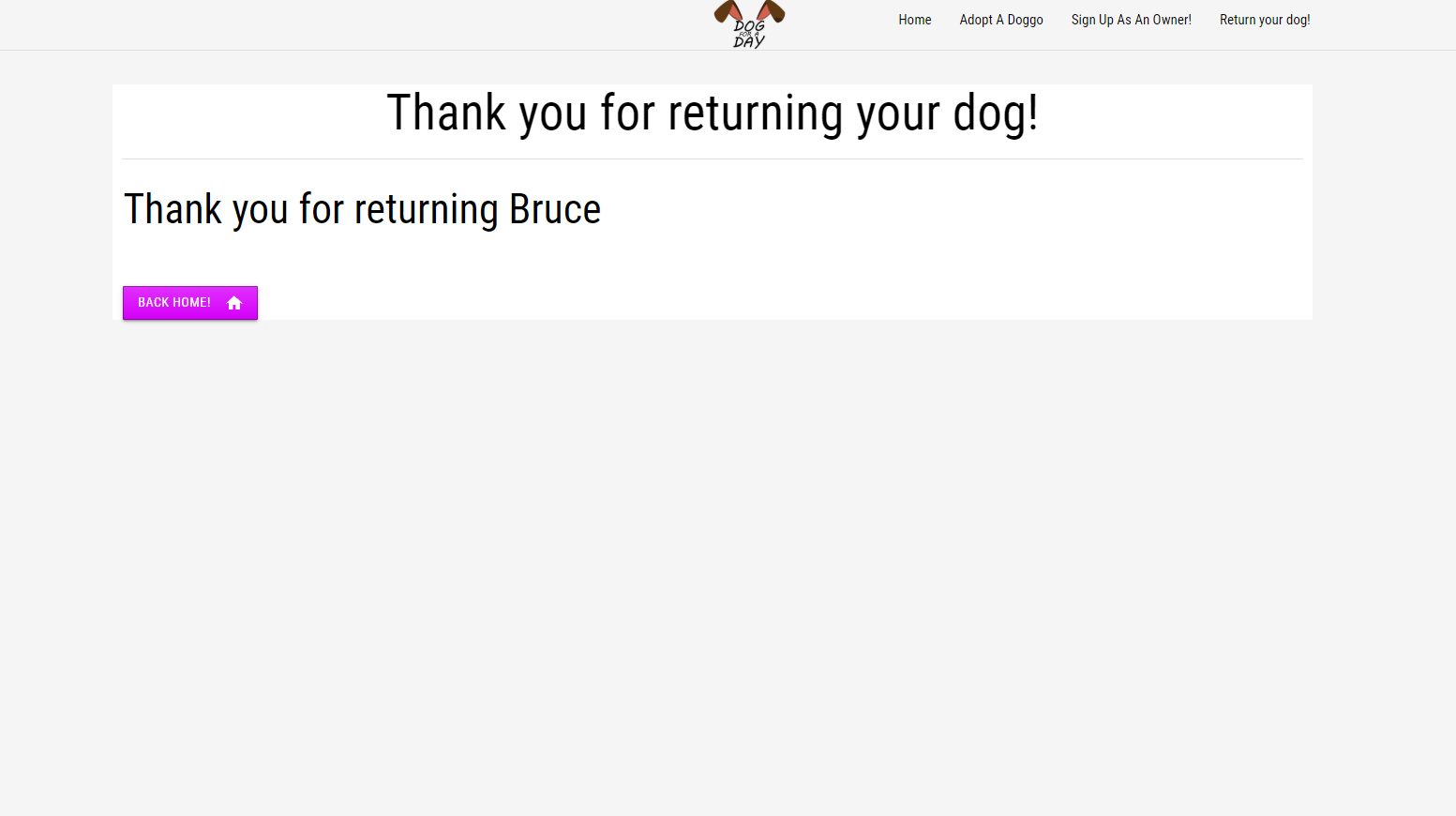
When testing, I found that I the date did not show up on the showcase page for when dogs are next available. This is because I changed the date to be set to the Person object but did not set it to the dog object. To fix this, I added in the line

“dog.date = date.strftime("%d/%m/%Y")” Which sets the date to the date var of the Dog object.

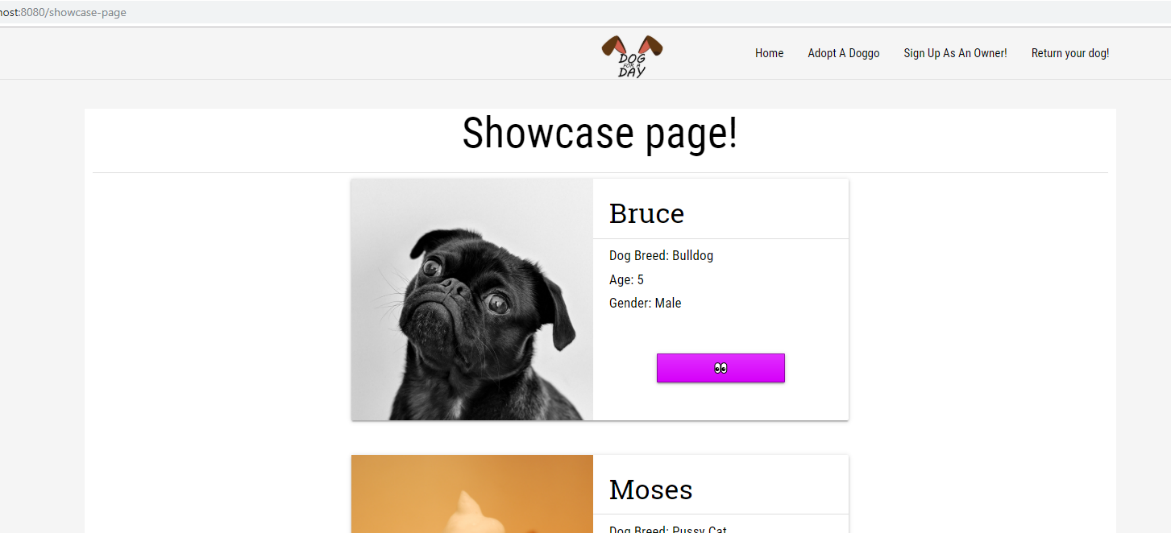
Task 13: Document testing



Here you can see the option to return the dog successfully shows up for those dogs which are already out.



Here you can see the return success message. Which is still to be formatted.



In this page the dog is now available. As it was previously taken out. Showing this was a success.

Task 14: Evaluation

This version worked very well and does as functioned to. There were a few small mishaps and bugs along the way. However, they were easy to fix. Usually only one line. The end result was a clear, easy to use system which enables the user to return a dog with little to no hassle as I was wanting. This makes the webpage easier to use and fulfils the need of the brief.