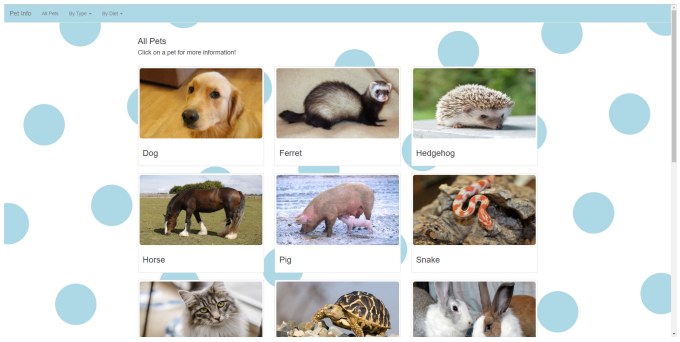
Coursework Report - Pet Information Web Application

Christopher Johnson
40275286@live.napier.ac.uk
Edinburgh Napier University - Module Title (SET09103)

1 Introduction

The aim of this coursework was to design a prototype web application for an online directory about a given subject. This would be completed using the Python Flask micro-framework.

For my website I chose to use information about pet care as my subject. This was because pets can easily be split into categories and sub categories which allows me to make a simple, yet effective, URL hierarchy. By the end of the project my main page for the website is displayed as follows:



On the main page all of the pets in the collection are shown as clickable thumbnails. Users can click on a pet to read more information about it such as diet info, care information and lifespan. The navigation bar provides users a way of searching for pets by their type (e.g. Reptile) and their diet (e.g. Omnivore) and general navigation.

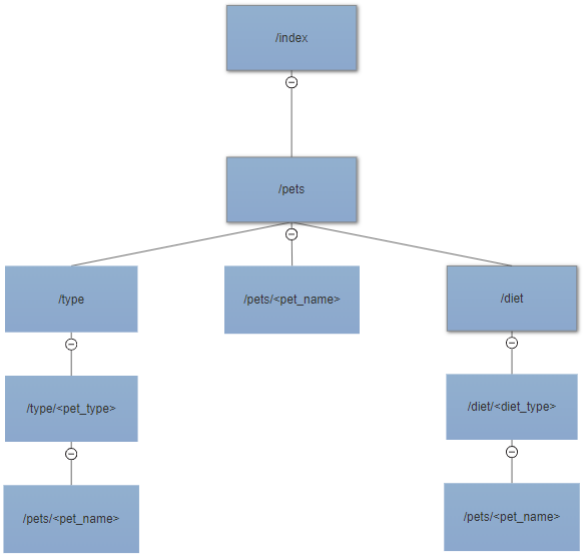
2 Design

The general design of my website is very simple. I wanted it to be obvious to the user immediately how to use the site. The background, though simplistic, is hard to read black text off of so most of the text on the web page is nested into divisions with white backgrounds. This makes reading easy for the user whilst keeping the website aesthetically pleasing.

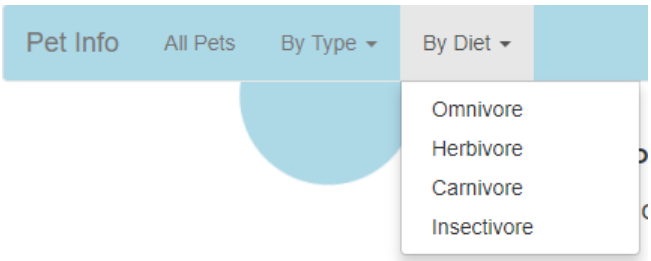
I wanted my website to reflect the positive nature of owning a pet so for my color scheme I went with blue (a color often associated with loyalty and trust), white and black. I knew that the pictures of animals that I'd use would be colourful and I didn't want to make the website look crowded so I kept the color's simple and any shapes/patterns used were clear and clean. I used friends for feedback on site visuals as they can provide quick and honest feedback eliminating the effect of any personal preference.

The navigation bar is consistent throughout the site. This was made easier by putting all of the navigation bar html into it's own file called navbar.html and using 'include "navbar.html"' in each page's template. This meant that any changes to the navbar was applied to every page simultaneously.

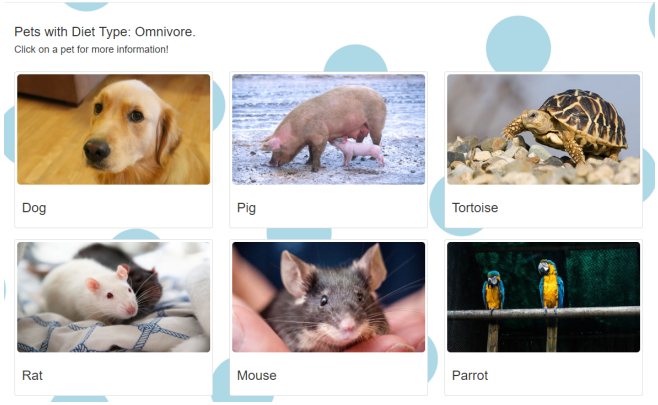
The two main ways I thought pets could be sorted is by "type" and "diet type". The website is designed to let the user easily see pets by the sub categories of each of these. The website navigation structure is as shown:



The user accesses these pages using the navigation bar. Clicking either "By Type" or "By Diet" brings up a drop down including the sub categories which can then be click to bring up all pets that fit the user's choice. For example if the user selects by diet type - omnivore:



The following page is displayed:



My URL hierarchy design means that it is easy for a user to take a guess as to how to get to a specific page by typing into the URL bar. The page template filtering is not case sensitive so if the user was to append "Omnivore" to /pets/diet/ they would still get the correct pet/diet/omnivore page. This applies for all pages.

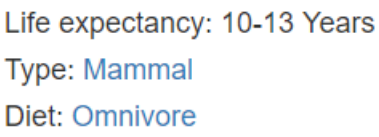
All of the pet data is stored in a static JSON file. All of the pet type and diet type pages use the same template and I used for loops and if statements to populate the templates with the correct "type" or "diet" in their JSON entry:

```
{% for pets in data.pets %}
        {% if pets.diet|lower == diet|lower %}
            <div class="col-sm-6 col-md-4">
                <div class="thumbnail"><a href="{{ url_for('single_pet', pet=pets.name) }}">
                    <img src="{{ pets.img }}" style="width: 100%; height: 200px"i class="img-rounded">
                    <div class="caption">
                        <h3>{{ pets.name }}</h3>
                    </div>
                </a></div>
            </div>
        {% endif %}
    {% endfor %}

```

This code from the Diet Type page template shows how I looped through the data and compared it against the users choice of Diet to display the pets with that diet type.

When a pet is clicked on any page the user is taken to pets/¡petname¿. This displays all of the information about the pet. Any values that can be used to sort the pets are linked to the appropriate page. For example on the dog page the words "Mammal" and "Omnivore" when clicked will take the user to the page:



Thanks to bootstrap's responsive nature and applications the website is fully mobile friendly. The number of pets, when shown as thumbnails, in each column changes depending on the screen size/orientation.

3 Enhancements

There are quite a few features I would add to the website to improve it if given more time. A simple users system could allow a user to log in, save their favourite pets and view all their favourites in one place. Each pet thumbnail could have a toggle-able favourite icon that when pressed will automatically add the pet to the users favourites.

Another feature I would include, which I attempted but ran out of time before it could be properly implemented, is a search box in the navigation bar. This would search for pets with names that include whatever the user inputs to the search box. This would just add another way for users to navigate the site however it is not essential as there aren't too many pets and the user can still narrow pets down by type and diet.

Further more I could improve navigation by including a way to navigate to the previous page without having to use the web browsers built-in back button. This is a relatively small change that wouldn't effect the functionality of the website too much but would have improved user experience.

4 Critical Evaluation

I think my website is a simple well functioning way of displaying and sorting a collection of data. All of the sorting features work and the site is easy to navigate thanks to the navigation bar and simple page structure.

One of the issues with my website is that there is no easy way to add/edit/remove pets from the site. This would have to be done manually via the pets.JSON file. Although a Create/Read/Update/Delete API wasn't required for this specification it would have been a nice addition to make changing the data populating the site easier.

I think that ,visually, the website is very appealing. I am happy with my choice of background images and colors. I feel that I could have present the information on each pet better but I was focused more on website function than aesthetics at the time.

Overall I am happy with my web application and all of the implemented features work fine giving an enjoyable user experience. Better time management could have perhaps allowed me to implement features discussed in section 3 "Enhancements", but all I can do is use that information to do better in the second coursework for the module.

5 Personal Evaluation

This was my first time using Python (and therefore my first time using Flask). I liked it's easy to read syntax and simplicity and I think it will definitely be my language of choice for any personal projects I undertake this year. This project has taught me how to create a simple Flask web app as well as more on website styling and functionality. These are skills that will prove useful for the rest of the module and for any personal projects.

I definitely could have added more features to the website as it feels very basic but I ran out of time and focused more on making the website look presentable. Any problems that I encountered were solved by Googling the problem and using web resources (such as Stack Overflow) to aid in the debugging process.

Overall I think I am well prepared for the next coursework and I'll be able to apply what I've learned effectively and efficiently.

6 References

Additional Libraries Used (Licenses are provided in the git repository for my work):

- bootstrap : https://getbootstrap.com/