

Coursework Report

Bruce Morel-Barnes

40202779@live.napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

1 Introduction

Sharkopedia is the name of this web application, this is simply because it is a collection of hyperlinks that lead to the Wikipedia pages of all sharks that have not yet gone extinct. Sharkopedia allows the user to navigate through the scientific Order, Family, and Genus of shark. The user may then select from either a specific shark (by delving into each of the previously stated options) or a list of each shark species.

2 Design

This web application uses a very simplistic and self explanatory UI, where the user must simply click the buttons to find the shark that they want to find. Initially, they are greeted with a choice of eight Orders and another choice which gives them the ability to see **all** of the sharks' links. If the user chooses an Order then they are given another set of buttons that either let them see each shark from that Order or a specific Family of that Order, the same is true for choosing a Genus. Once the user selects a Genus, they will be shown hyperlinks that lead to the Wikipedia webpages for each of the sharks of that Genus.

For a more in-depth perspective, the application is split into multiple important segments, one of which is the Python/Flask file which is the driving force behind the file system-like shark filtering available to the user (which updates the selection of buttons and/or URLs that are visible at any one time so that they are appropriate for the options that have been chosen).

Another significant portion is the Jinja2 template which, although is closely related to the Python/Flask file, it helps to send data pertaining to what the user chooses back to the Python/Flask file to let it update the path after every selection.

The final section is the hierarchy of the Order, Family, Genus, and Scientific Name of sharks which looks much like a file system and is used as much for the purposes of this project.

3 Enhancements

I believe that the web application could be improved with the use of a database in place of the awkward directory system that is currently in place, the quantity of items may, however, not warrant the implementation of such a database. In a future version of this I feel that instead of simply sending the user to Wikipedia, the web-app could hold data on each of the sharks (if only what can be found on their individual Wikipedia pages).

I think that the UI is perhaps **too** simplistic and just is not very aesthetically pleasing because of said lack of complexity. In the same vein, I would want to add "back" and "home" functions which would permit the user to reverse some of the actions that they have taken or just start over respectively. Although it does use bootstrap, I am of the opinion that more of bootstrap's features could be explored in the future to improve the overall aesthetic.

Adding mobile compatibility could prove very beneficial to attract a greater number of users than the web-app does already, this could even be branched out into its own separate but similar mobile application. Perhaps integrating a greater number of creatures, perhaps expanding from just sharks to adding other water based animals, this too would garner more interest and potentially increase the interest for such a service.

A feature that I regret not implementing is a "standard" search function, that would have returned any shark Scientific Names that contain the search parameter. This would have been a great addition because while the current system is simple and obvious, the addition of the search function would allow for more pinpointed searches.

4 Critical Evaluation

I think that a feature that works well is the "all" button that shows the user all of the sharks from any specific point in the file system, it works well because it is fully flexible and not locked to a specific characteristic of the shark e.g. Family or Genus. This is great since the user may not know the exact scientific terminology to identify the shark that they are trying to find.

I think that the file system like architecture, while simple and improvable was a downside, or at least not as interesting as it could have been as there was no overlap between sections. This means that there very little data involved with only the Latin and English names required to create the hyperlinks dynamically. One major bonus to the file system structure is that it is completely dynamic and the code would not need any manual updating as it doesn't interact with specific shark types, and only deals with the nebulous sounding variables "sharkorder", "sharkfamily", "sharkgenus" and "sharkname" which are obviously not directly linked to any one item in the collection.

5 Personal Evaluation

Overall, I had to learn Vim, Git, and Jinja2. Vim was particularly useful to learn as the shortcuts were quite easy and even often useful, saving me an undefined amount of time. Learning the Git syntax was obviously very useful since Github is a powerful versioning tool that is also a cloud-based storage system. Although I hadn't used Jinja2 previously, I would undoubtedly use it again if I had to create another web-app as many of its features made it very simple to use and it adds in several more features over just plain HTML.

I faced several challenges, one of the largest being when I was making the user redirect to the correct location (i.e. to each stage of the filter and to Wikipedia).

My single, largest problem, however, was the data. I had to use multiple Python scripts to obtain and sort the data, initially putting it into a single large .csv file and then moving them into multiple .txt files. Eventually I thought of creating a file system-like scenario for the data which would allow me to navigate through each "stage" of the filtration for the user of the web-app. I then implemented this, learning about the "os" import and how it could be used to detect files and folders within specific directories, and used it to show the user a different view depending on where they were trying to view the files from (in the filesystem).

I believe that even though I succeeded in moving past these issues, I severely underestimated the task, and left working on it until too close to the deadline. In hindsight I would have started at least a week prior to when I did as this would have meant that I could have implemented several of the features that I would have wanted to (e.g. the search function). That being said, I feel that with the time I gave myself, I performed adequately to create a functional tool.

6 Conclusion

In conclusion I learned a great deal from this coursework, even if I didn't use all the time given. The main points of success were my data analysis, where I spent several hours working through how I wanted to sort them, and then coming up with the filesystem solution, which let me mostly ignore the potential need for a database, making a dynamically updateable system that could store data within the text files for each individual shark.

7 References

I obtained the list of shark names, genera, families, and orders from:
http://www.supportoursharks.com/en/education/biology/Shark_Database/List_of_Sharks.htm