**4-2 Milestone 3**

by Brandon Hollada

Southern New Hampshire University

CS-499-19649-M01

Submitted to Neil Kalinowski

For artifact two I chose to use the final from CS340. The goal of the original project was to have a program that would authenticate the user and interact with a MongoDB. The program would read in information from a database that stored animal shelter animals with their associated details. The information would be displayed in an application that could be used to filter results, see where the animal was found, and find potential rescue animals. The reason that I found this to be an ideal choice for showing my improvements in algorithms and data structures is because of the design revolving around sorting. While practicing for interviews, I found that within algorithms and data structures, bigO time complexity is huge focus of this area. So, with the main function of the program to be able to sort and search, this seemed like a great place to show my understanding of bigO time complexities and how to apply it to real solutions. Looking at CS340, I found that the search for common rescue animals could be optimized. It is one that is called on for a constant set of parameters. Because of this consistent set of parameters, the results could be precomputed and recalled for future searches. This means that instead of every search having a complexity of O(n), which means it has to search through every row each search, the complexity could be reduced. The first search through the database would require O(n) but repeat searches would only require O(1) as they are only pulling up the already known list of search results. I think this shows I have an understanding and ability to apply one of the most important concepts of algorithms and data structures. So, this made it an easy decision to include this artifact.

Outside of that main enhancement, I made some visual improvements to the program by formatting it in an easier to read manner. I also added the ability to search for an age range or name that would return close results. The results were based on a formula that would determine if a match was close to the user’s input. Other improvements I added were to clean up some areas of the code by removing unused code, removing potential errors with incorrect search fields, and adding descriptive comments to the new code. One of the areas I fell short in was being able to store search results to be able to go back to previous searches. While working on this, I felt it was an addition that did not add much into showing my abilities for this section. Some other modification that I made was to make the program run standalone instead of through Jupyter notebook. I did this by making it run through standard dash instead. This was one of the biggest challenges that I had, trying to get the program to run outside of the school virtual machines that were setup for this program. I tried to get it to work through the web version of Jupyter but decided it would be better if the user could run it without relying on a web version of a program. The other area that I modified was having the program directly access the .csv file instead of going through mongoDB. The reason for this is that the future modifications that I want to make to this is to have it run through an SQL database. So, it seemed like it would have been unproductive to get it to work with MongoDB and have to change it in the future. While I think that I have some areas I can still improve in this artifact, I think that this does a good job of showing my abilities in the area.

Below shows the program running off dash with direct access to the csv file. This shows a similar layout with so visual improvement and a search result for 100 week old giving both 1 year and 2 year old animals. This range is determined by the algorithm provided in the code.

A screenshot of a computer

AI-generated content may be incorrect.

Below is the original, similar in design but functioning differently in the background.

A computer screen shot of a computer

AI-generated content may be incorrect.