**Animal Shelter Dashboard README**

**Required Functionality**

The animal shelter dashboard was made to give easy access to animal shelter information regarding dogs sought after for search and rescue situations. The dashboard gives access to the database as a whole as well as predefined filter options for the types of rescue dogs required. It then shows a pie chart of the breeds of animals in the current filter and a map of where the selected animal can be found.

A screenshot of a map

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A map with a location pin on it

AI-generated content may be incorrect.

A map with many different colored lines

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Tools Included**

Pymongo is used to interact with the database using python. MongoDB was used because of the compatibility with Python as well as the flexibility it provides. The database used will not need an overhaul as animal shelters change how they store data because of its flexibility (MongoDB, n.d.). And MongoDB allows the use of its functions in Python to easily create, read, update, and delete data. Dash, plotly, and matplotlib were used to set up the dashboard including the pie chart and map. Dash provides a way to structure the website using HTML and ways to handle events like button presses using callbacks.

Pymongo: <https://www.mongodb.com/docs/languages/python/pymongo-driver/current/>

Dash: (<https://dash.plotly.com/>)

**Project Steps:**

First the python code to interact with the provided database was created. This added the create, read, update, and delete functionality for the dashboard. It also authenticates and logs the user into the database. Then the structure of the dashboard was coded in with HTML. This included the filter dropdown, data table, pie chart, map, logo, and developer credit. Finally, the callbacks were coded in to handle changes in the database like filters and data selection.

**Challenges:**

The most challenging thing for me was understanding the callbacks. For example, I was not returning the data frame when handling the filter callback, so the filter was not actually changing the data table. After changing it to return the data frame, the data table filtered, and the visualizations changed with the filter. Another issue I had was the username and password authentication since my password has the @ symbol in it. This one took some asking around, but I figured out that special symbols need their ASCII codes for the database connection to work such as the @ symbol to %40.

References

MongoDB. (n.d.). *Why use MongoDB and when to use it?*

https://www.mongodb.com/resources/products/fundamentals/why-use-mongodb