Project 2 - Animal Shelter - Read Me

CS-340 SNHU

Sarah Hodge

**Functionality:**

The required functionality of this project was to develop a dashboard which allows users to access the database of Austin Animal Center, giving them the ability to filter the listed animals meeting certain criteria for training as rescue animals. The program provides a complete list of available animals, as well as geolocation information, and pie charts showing distribution of specified animal criteria.

*Screenshot of successful execution of dashboard default/reset state:*

*A screenshot of a computer screen

Description automatically generated*

*Screenshot of successful execution of disaster rescue and individual tracking filter:*

*A screenshot of a computer screen

Description automatically generated*

*Screenshot of successful execution of Mountain/Wilderness rescue filter:*

*A screenshot of a computer screen

Description automatically generated*

*Screenshot of successful execution of water rescue filter:*

*A screenshot of a computer

Description automatically generated*

**Motivation:**

Motivation for development of this program was foremost the satisfaction of the client, Grazioso Salvare. In addition to providing a quality product for the customer, the development of this dashboard also allowed me to implement database development skills learned throughout CS-340, combining the use of MongoDB and Python, as well as Jupyter notebook, html, and a variety of other programming skills obtained throughout the CS program at SNHU.

**Tools, and execution:**

MongoDB is the primary driving component for this program. The data for the Austin Animal Center is imported to MongoDB via a .csv file. The primary functions for the manipulation of the database, create, read, update, and delete, are defined in the AnimalShelter.py file. The .ipynb file defines the processes that allow the user to interface with the database, including html, and additional python script that creates the dashboard. Current versions of Python, Jupyter Notebook, and MongoDB are required to operate all aspects of the program. Local access to the .csv file containing the AAC animal information is also required.

MongoDB was used as the database for this project because it interfaces easily with Python. The PyMongo driver, and a host of available library assets make the combination of Mongo and Python ideal for any flexible, scalable, and quickly executed program. The use of Jupyter Notebook allows for the simultaneous use of both .py and .ipynb files.

**Development:**

This program was developed in four main phases: the database creation in MongoDB, development of the core “CRUD” functions in Python, the development of the primary user interface and .ipynb file, and finally the development of the interactive dashboard which allows return of sorted data from the Austin Animal Shelter database. The perfection of each of these phases is critical for the overall success of the project, as each phase builds upon the previous completed phase. This process creates a built-in testing method that identified any deficiencies in the early phases of development as the later phases are used.

**Challenges:**

In the first phase, ensuring appropriate user access to databases was tedious, but a necessary process for ensuring appropriate allowances and security for the database. These challenges were overcome by reviewing MongoDB database tools [here](https://www.mongodb.com/docs/database-tools/) to ensure proper use and implementation of database aspects. In phase two and three, the challenge was simply writing the most efficient code that would return usable information to the user. The final phase, the dashboard development, did pose some challenge: predominately, the appropriate use of “df.drop(columns=['\_id'],inplace=True)” any time information is being pulled again from the database was needed to actually allow the dashboard to function as intended. Without this line, the radio buttons did nothing. This challenge was overcome by old fashioned trial and error, mostly in the form of print statements to find what was breaking the code.