# CS 340 README

## About the Project/Project Title

For project two, I was contracted by Grazioso Software to develop a web application for an international rescue-animal training company. This project allows the user to view a data table of all available rescue dogs within the database showing data such as name, breed, age etc. Along with the data table, there is a geolocational chart that shows the location of the rescue-dog and a pie chart that shows the breakdown of breeds depending on the rescue filter chosen.

## Motivation

This application gives its users the ability to quickly identify rescue dogs in training or graduated, depending on their geographic location and what types of rescues they perform. This application is still a work in progress so at the moment it is very rough but future iterations will see better organization of the data table along with a more stylistic, eye appealing features for users. Right now, I am more concerned about functionality than presentation.

## Getting Started

This project will be made available on my GitHub, and it should work with any CSV data set. To upload your CSV, you will need to use the mongoimport tool within the mongo shell. Depending on the data set being used for this program, adjustments will have to be made to the update\_map function. This function locates the column that contains the map coordinates that makes the geolocational chart possible so any dataset must have coordinates on every row and the function must be adjusted to the column that contains those coordinates for each entry. The project uses the dash framework along with Pymongo to connect the front end to our MongoDB database and a foundational knowledge of these technologies are needed to modify this application. To connect your database to this application, replace the highlighted syntax with your own MongoDB database link in the CRUD module:

Graphical user interface, text

Description automatically generated

## Installation

The first tool you would need to work with this application would be the Python programming language. To install Python, use the link <https://www.python.org/downloads/> and follow the instructions suited to your OS. You will need Pymongo as well. To install Pymongo, follow the documentation on <https://pymongo.readthedocs.io/en/stable/> and find the instructions suited to your OS. The base of this application is built in the dash framework. Use the link, <https://dash.plotly.com/installation> to install dash onto your local machine. Finally, you will need my custom CRUD module that I built for this application. It is your basic CRUD functionality module and will be included in this project on GitHub.

## Usage

### Code Example

Text

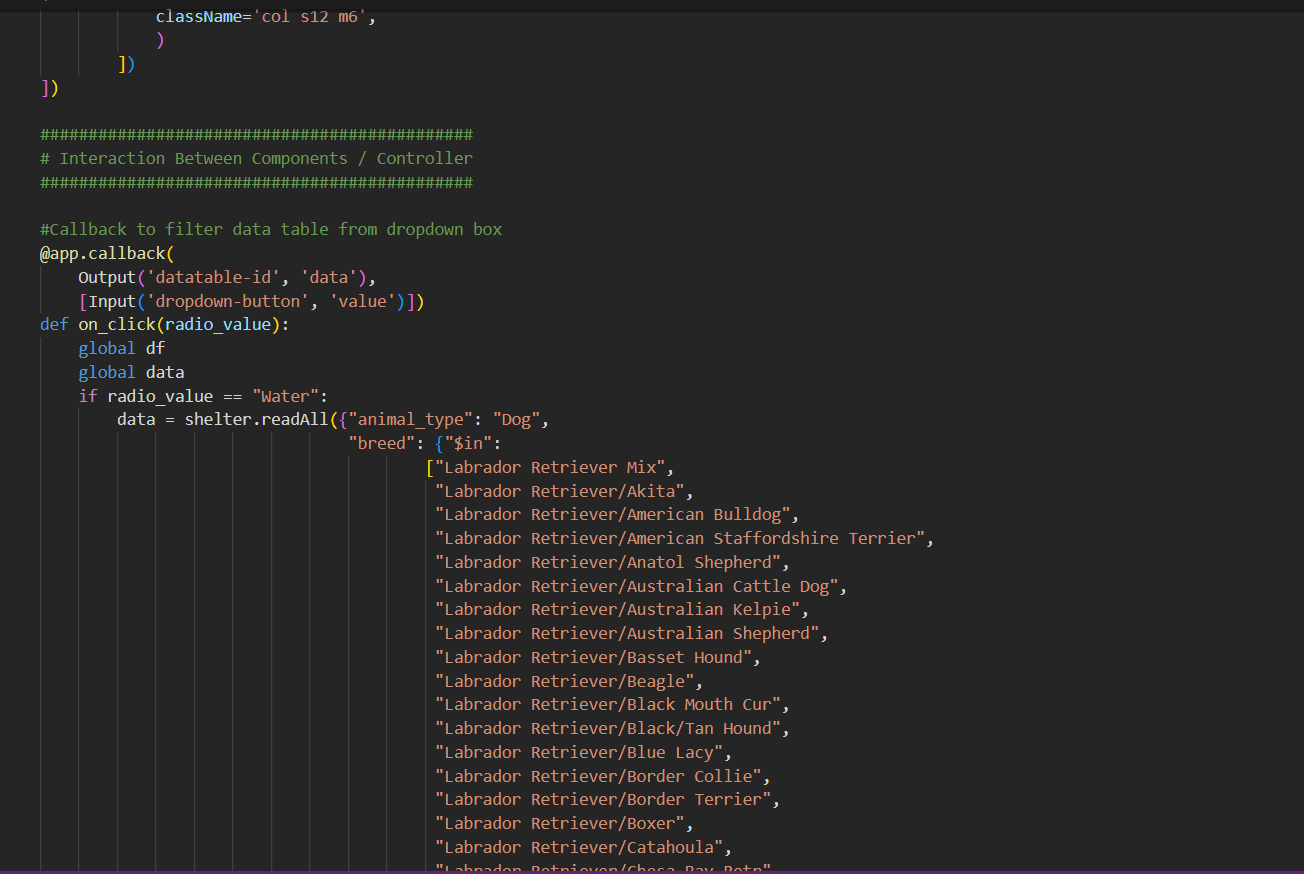
Description automatically generated

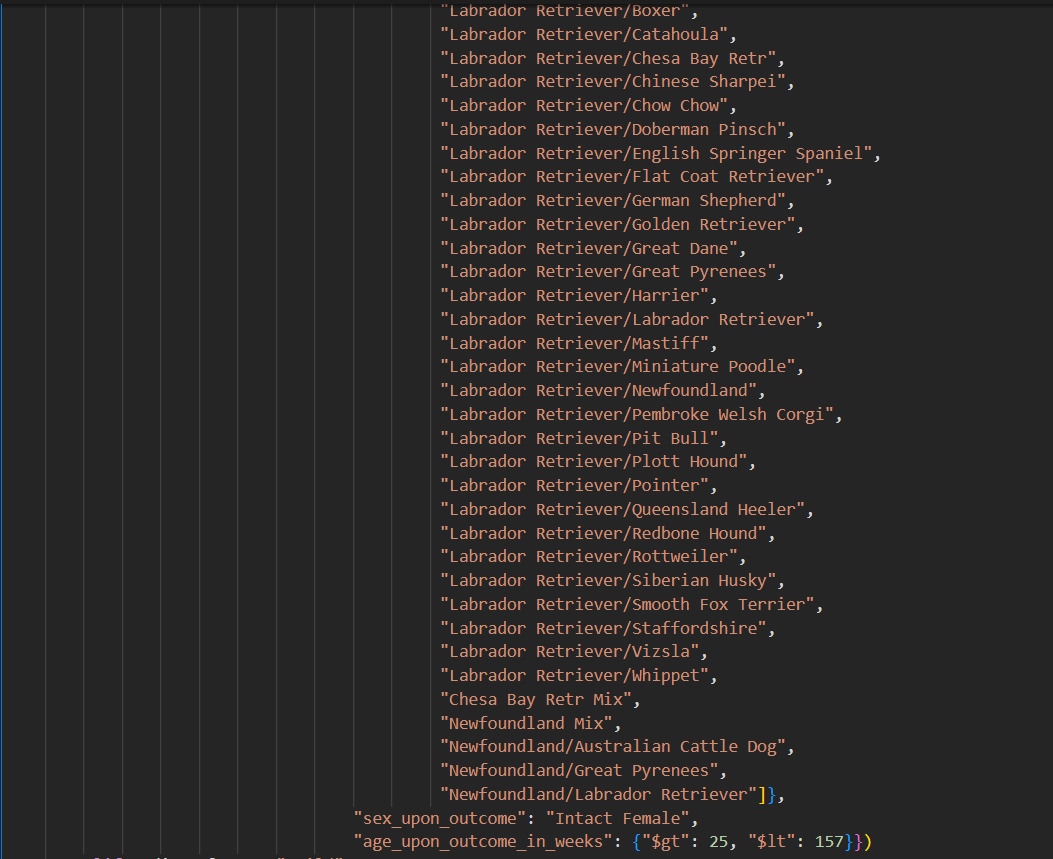
Text

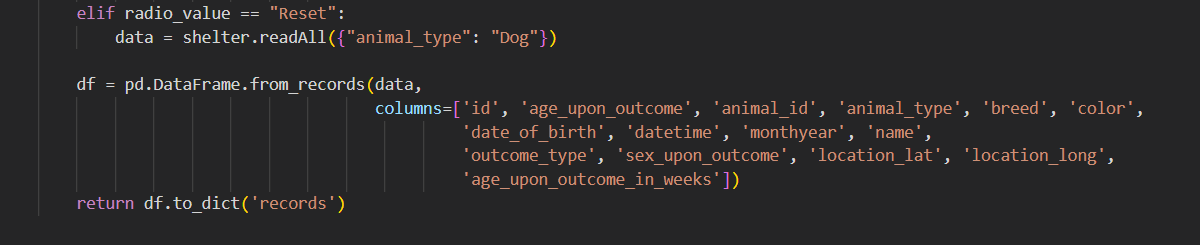
Description automatically generated

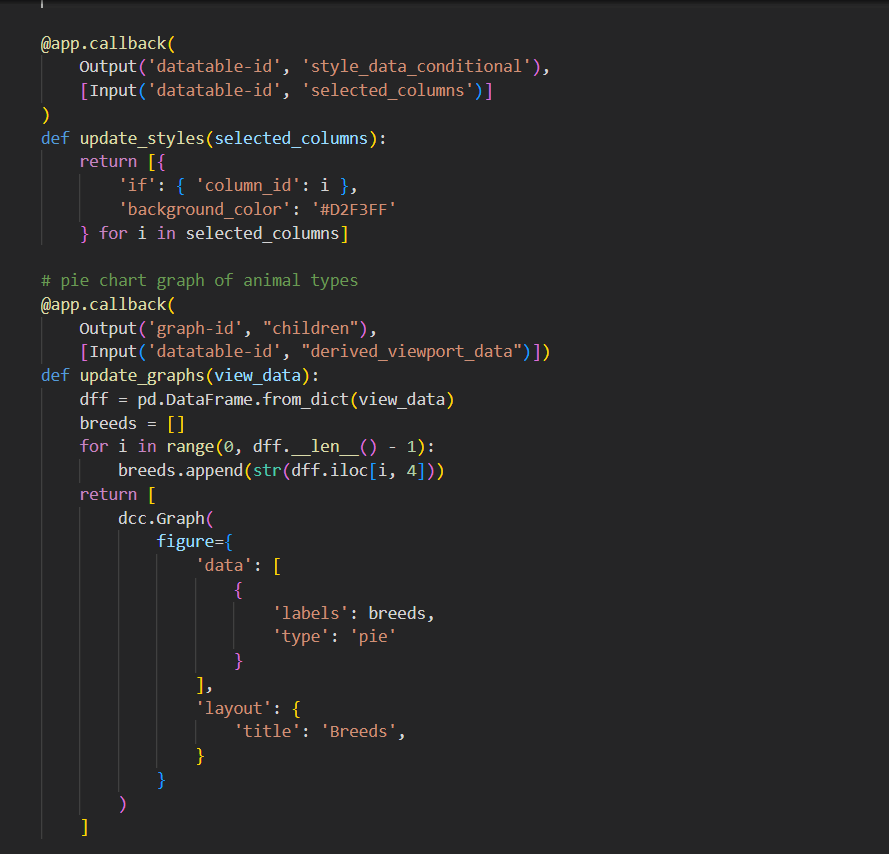
Text

Description automatically generated









Text

Description automatically generated

This is a snapshot of the code base for the application. For the sake of saving space on this document, I only included screenshots of how the filter for the water rescue was structured. To examine the other filters, view the file in its entirety on GitHub.

### Tests

To test this application, upload your CSV as instructed in the getting started section. Adjust the update\_map function to fit your data set then run the application. If your MongoDB is set up properly in the CRUD module, you should not have any problems running this application. Try changing the rescue filter types in the drop-down menu and observe the data table, pie chart, and geolocational chart to make sure they adjust to the filter. Scroll through different pages within each rescue filter to make sure the geolocational chart and pie chart are adjust to the information on each page.

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

### Screenshots

A picture containing logo

Description automatically generated

Table

Description automatically generated with low confidence

A picture containing map

Description automatically generated

## Contact

Jonathan Santiago

Jonathan.Santiago@SNHU.edu