Jay Zinzuwadia

CS340 – T3703

February 27th, 2021

## SNHU CS340 Animal Rescue Dashboard Readme

**About the Project**

This project is essentially a database connected to a server to create a dashboard. After learning the functionality of MongoDB, and creating a CRUD file for the animal rescue .csv, Jupyter notebook is used to create a dashboard to output the information including a table with all the entries, some filters required by the client and also a GeoMap to show where the animals were rescued.

**MongoDB Usage**

MongoDB was used as the “backend” or database for this program. It is a NoSQL based database that can be used with the python language for quick access. The syntax to run queries in the database are JSON type formats.

**Dash Usage for Dashboard**

Dash is a tool used and imported into the Jupyter notebook to create the dashboard. It is based on React JavaScript. Dash includes html code that can be implemented within and recognizes its syntax. In the experience using Dash, it is quick and responsive to changes made when selecting and deselecting filters. The update of the geo map and pie chart within the code are also relatively quick.

**Installation: Tools Used**

Below are the tools required to run a local copy of this project. Tools include MongoDB, Python, Jupyter Notebook.

**Main Programs:**

Python: <https://www.python.org/downloads/>

MongoDB: <https://docs.mongodb.com/manual/installation/>

Jupyter Notebook: <https://jupyter.org/install>

* Jupyter is also automatically installed with Anaconda. You can use the integrated Jupyter Notebook that installs. For conda installation, please visit anaconda website.

**Tools:**

* Plotly – used for the pie chart
* Pandas – used for the data frame feature
* Numpy – used for calculations within Python
* Dash – It is the main framework for the Web Dashboard.

**Setup for Program**

1. Create mongo Database as AAC and impor the animals.csv file
2. Create an admin username and password and then create a “aacuser” with read and write privileges on the AAC database.
3. On the ProjectTwoDashboard.ipynb file, make sure that your import of the filename and class name is concurrent with your CRUD file.
4. Make sure that the username and password is correct in the project file for it to be able to access MongoDB
5. Once ensured that above steps are completed, **MongoDB needs to be running** with aac user as. Log in with authentication as shown below

Text

Description automatically generated

1. Following that, you are able to run the program and interact with the dashboard.

**CRUD File Screenshot for all MongoDB Interactions**

**A picture containing graphical user interface

Description automatically generated**

**Server Connection for Project File (Must be like this and include your unique port number)**

**A picture containing text

Description automatically generated**

**Using the Program and Demo Pictures**

This program has a few functions. Initially the dashboard will include all entries from the animals.csv file. You can filter the data according to the type of rescue. There is also a “RESET” button to return the raw values you saw initially. Once a rescue type is selected, you can select up to 5 rows of animals. Selecting these animals will generate markers of where the animals were picked up. On the left of the map, you will see a pie chart that shows the breed of the animal along with the percentage on the pie chart itself. Below are some sample pictures of the full display, along with Filters implemented.

**FULL DISPLAY**

**Logo

Description automatically generatedMap

Description automatically generatedGraphical user interface, application, table, Excel

Description automatically generated**

**Examples of Filters used and what it returns.**

**Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated**

**Shows the RESET button used below**

**Graphical user interface, application

Description automatically generated**

**Code Screenshots (Full Project 2 Code)**

**Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated**

**Resources Used:**

<https://dash.plotly.com/>

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>

<https://docs.python.org/3/>