SNHU

7-2 Project Two: README

Jonathan Sussan

### Satish Penmatsa, Ph.D.

### CS 340 Client/Server Development

02/18/2022

**CS 340 README Project Two**

**About the Project/Project Title**

The following README file contains information and guides on how to use the Amimal Finder tool, developed for Grazioso Salvare. The purpose of this tool is to give a user the ability to search existing animal shelter databases for animals that would be candidates for search and rescue training. The search results are based on the animal traits and provide the current location of the animals.

**Installation**

The tools we need to use the software are as follows.

* MongoDB, all editions will work but the community edition is adequate for our needs.
* Jupyter Notebook or similar compiler, as well as the Pymongo driver which is used to facilitate communications between Python and the Mongo database.

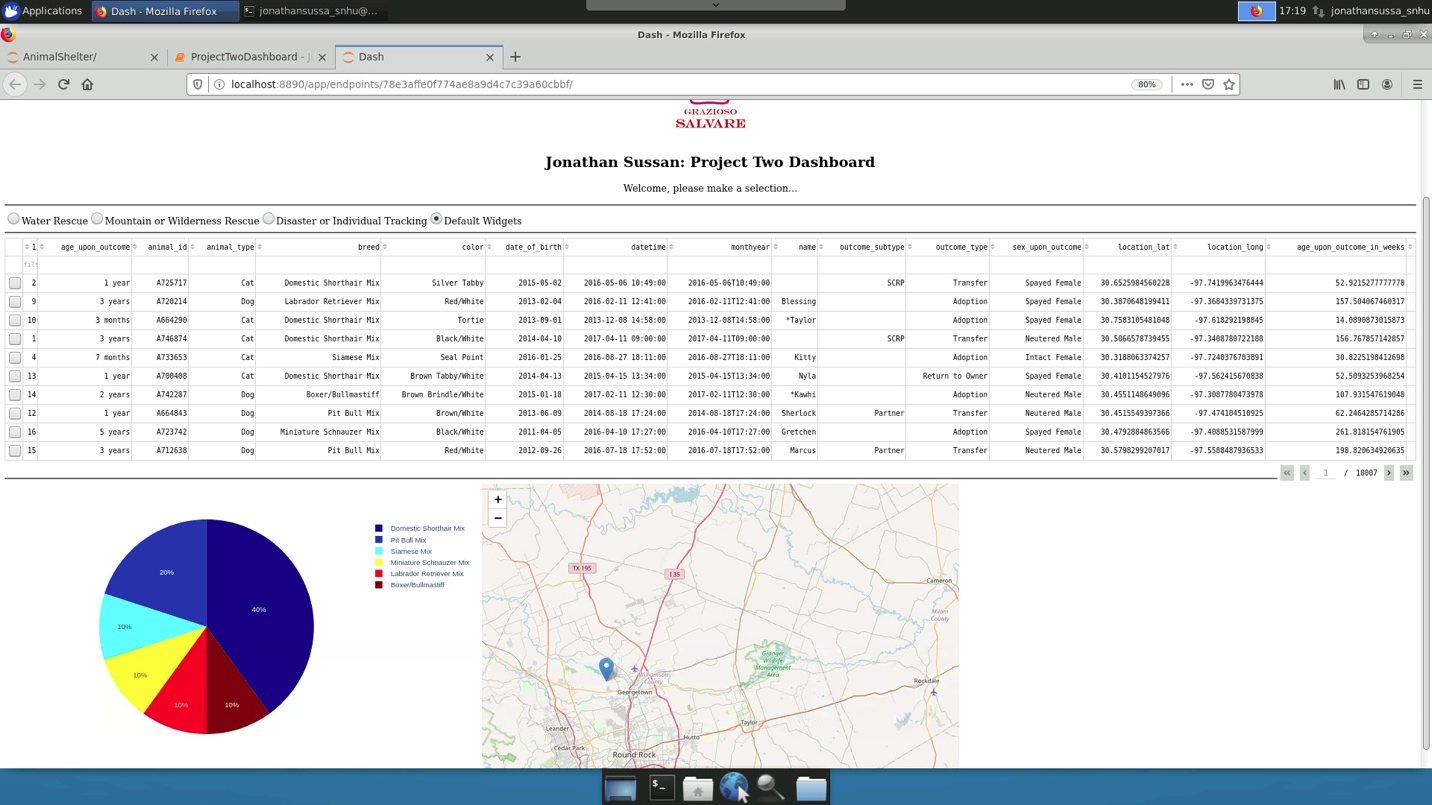
**Usage, Including Screenshots.**

**Screenshot 1:** Grazioso Salvare logo, my name, pie chart, and map in their default state.

**Graphical user interface, application

Description automatically generated**

**Screenshot 2:** I couldn’t fit all the widgets on one screen shot so the below one is to show the pie chart and map in their entirety.

****

**Screenshot 3:** Table with Water Rescue radio button selected and results. The pie chart and map update automatically based on selection.

**Graphical user interface, application

Description automatically generated**

**Screenshot 4:** Table with Mountain or Wilderness Rescue radio button selected and results. The pie chart and map update automatically based on selection.

**Graphical user interface, application

Description automatically generated**

**Screenshot 5:** Table with Disaster or Individual Tracking radio button selected and results. The pie chart and map update automatically based on selection.

**Graphical user interface, application

Description automatically generated**

**Screenshot 6:** Table with Default Widgets radio button selected and results. When selected, the table results will reset to their original state.

**Graphical user interface, application

Description automatically generated**

**Screenshot 7:** Highlighting the options and tooltip buttons for the pie chart.

**Graphical user interface, application

Description automatically generated**

**Screenshot 8:** Highlighting the options and tooltip buttons for the map.

**Graphical user interface, application

Description automatically generated**

**Describe the tools used to achieve this functionality and a rationale for why these tools were used.**

The Animal Shelter tool require two components to operate which are a Mongo database with all the records and information for the animals, and PyMongo to communicate with our database. MongoDB was selected for our database due to its deep query ability; it can handle large volumes of data at high speed, its simplicity, scalability, and availability. MongoDB as a NoSQL, reads and writes much faster than SQL.

PyMongo was chosen because it is a natural selection when choosing a library for MongoDB. For the user interface (UI), we incorporated the Dash framework to create a simple and user-friendly dashboard. To create the pie chart, we used Dataframes and Plotly Express. The file contains the Dash framework that provides the view and controller structure for the application.

<https://pymongo.readthedocs.io/en/stable/#>

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

<https://plotly.com/python/builtin-colorscales/>

**Explain the steps that were taken to complete the project.**

The steps that were taken to complete this project occurred throughout this course. We learned and experimented with all these steps within each module. There was a lot of trial and error, including frustrations with the virtual lab and finding all the invalid syntax errors. To begin:

* I needed to review the specs that were given to us.
* Next, I got started on the dashboard with an unfiltered data table of shelter animals.
* Then I needed to develop queries that allow users to filter the database.
* Build interactive options through which users can activate filters.
* Add widgets for dynamic presentations of data that is retrieved.
* Test the dashboard to verify functionality.
* Create a README file to document my work.

**Identify any challenges that were encountered and explain how those challenges were overcome.**

As mentioned above, there were many challenges that I encountered. The first was with the virtual lab and accessing MongoDb. Once I got used to understanding how to stop and start Mongo, along with all the various keywords to use, I was able to overcome this challenge. My next challenge was in creating the script to run the CRUD commands. There was a lot of trial and error here, along with starting over multiple times. The main issue I was having with my read command was answered in a class discussion post which helped a lot.

The last challenge I needed to overcome was primarily due to syntax errors when creating the dashboard. This took the longest to overcome. I ended up working on the python code on the Pyhcharm IDE I have installed on my Macbook instead of the virtual lab which helped me find and fix these syntax errors.

**Contact**

**Jonathan Sussan**