Michael Antoniazzi

CS 340 Client/Server Development

**Project Two README File**

**Grazioso Salvare Dashboard**

**Project Description:**

Grazioso Salvare has asked us to create an interface they can interact with to view and filter data to identify animals well suited to various search and rescue situations. The data being accessed is provided to Grazioso Salvare by a non-profit agency that operates animal shelters in the Austin, Texas region. The required functionality of this project include:

1. Data Viewing: The interface will display the animal data in a tabular view.
2. Data Filtering: The dashboard needs a means of filtering the data by animal type, breed, sex, age, etc. to help identify ideal candidates for rescue operations.
3. Graphical Representation: The client has requested we use a pie chart to show the distribution of the filtered data.
4. Map Integration: The dashboard will need to be able to display the individually selected animals’ location on a map.

**Screenshots**: A screenshot of a computer

Description automatically generatedA map of the city

Description automatically generated with medium confidence

A red and white logo with text

Description automatically generatedA map with a couple of images

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Tools:**

**MongoDB**

The database used for this dashboard interface project was MongoDB. MongoDB was used because it is ideal for handling large amounts of data. It natively supports JSON documents which helps it to interface with Python. If Grazioso Salvare would like to expand outside of the Austin area and include data from other shelters MongoDB is perfect for further expansion due to its ability to scale horizontally.

**Dash Framework**:

The Dash framework is used to develop this web application because it is one of the most simple and effective tools for building an interactive application using the Python programming language as well as allowing for the MVC pattern. The View was accompanied by using Dash HTML components to help create the layout of the dashboard while the Controller used Dash callbacks to handle how the user would interact with the dashboard.

**Plotly Express:**

Plotly Express is a high-level interface for drawing informative statistical graphics. It provides simple syntax to help generate plots with very little code. It was used in this project to help display the data into a pie chart.

**Dash Leaflet:**

Dash Leaflet is a library used to render Leaflet maps within applications using the Dash Framework. It is one of the most popular open-source libraries for interactive maps and was used in this dashboard to show the map location of individual animals as they were selected in the dashboard.

Links to Resources

<https://www.mongodb.com/>

https://dash.plotly.com/

<https://plotly.com/python/plotly-express/>

<https://dash-leaflet.herokuapp.com/>

**Steps to Complete the Project:**

1. The first step was to import the data from the Animal Rescue shelter into a MongoDB database that included a username and password to enable access to Grazioso Salvare. Queries needed to be written to retrieve the data needed to transform it into the proper format.
2. Next the layout was created using Dash HTML components to add elements like radio buttons to filter data, tables to display everything in an easy-to-read format, and charts and maps to help visualize the data from the database.
3. Now callbacks needed to be created to enable the actual filtering of the data in the dashboard.
4. Finally, everything needed to be tested to make sure all the interactions and functionality of the dashboard were working properly so it could be deployed for use by Grazioso Salvare.

**Challenges & Solutions:**

When we first began this project, I didn’t have much of an issue following the instructions to set up and import the data into a MongoDB database. Even interacting with the data wasn’t too difficult because I felt the instructions were easy to follow along with. When it came to integrating the Dash Framework to set up the layout, I felt like I struggled to get everything to display correctly and the only way I was able to resolve the issues was through trial and error. I also made small mistakes setting up the filtering options by not using the proper syntax for the queries which I also figured out after a bit of trial and error as well as Googling the error codes I was receiving when I ran the program. I feel like the more I use MongoDB and the Dash Framework the more practiced I would be which should help to not make the syntax errors I made during the later parts of the project.