**Grazioso Salvare Animal Rescue Dashboard**

## About the Project/Web Dash Dashboard

I developed this project for a company called Grazioso Salvare which is an organization for Animal Rescue. For this project, I focused on building a dashboard which is web-based focusing on data-driven so that they can help the company identify different types of dogs that can conduct search and rescue training. The search which was conducted manually can now be conducted with filters visualization and locate dogs that meet different type of specific criteria. The dashboard helps save time

**Key features:**

* Dynamic filtering
* Interactive Data table
* Visualization
* Identifiers

**Tools Used:**

* Python
* CRUD Module (Python)
* MongoDB
* Dash (Plotly)
* Plotly Express
* dash-leaflet
* pandas
* base64
* Jupyter Notebook

One of the first things this project requires is to be able to select between different rescue categories. These categories are established at water mountain disaster tracking and lastly the option to reset. The option to reset was important because we wanted to make sure that the data can instantly be refined. What this means is that if you click or select something everything else should update the second you click on that thing. This way the visualizations are dynamic.

Apart from dynamic filtering and making the data table interactive, something I focused on was making sure that there are distribution charts and the geolocation map which can show the location of the selected dogs. Lastly the project requires that there is some sort of unique identifier. For this project the unique identifier is my name which goes all the way to the top along with the logo for the company. This way the application is clearly branded and can easily be tied back to me.

Before we get into the motivation for this project it is important to talk about the design pattern for this dashboard. The main design pattern for this project can be understood given the model view controller pattern. The model for this database comes from the mongo DB database which gives data storage. We can view this dashboard because of the dash-based user-friendly interface. Lastly, we can use controller features because earlier in one of the assignments I developed python's create read update delete module which is handling all the data operations for the model.

## Motivation

The main motivation for this project comes from the fact that we need to streamline the selection process for training candidate dogs. One of the things we should not be doing is manually sorting through large datasets trying to understand which one’s matches were. This process is highly inefficient. Therefore, a better way to go about it is to make a data dashboard which can automatically filter by rescue type and give location along with breed distribution.

## Getting Started

**Requirements:**

* Python
* MongoDB (connecting to the data)
* Jupyter Notebook
* Python Packages: dash, dash\_leaflet, plotly, pandas, PyMongo, base64

## Installation

**Tools to be installed:**

* **Python** (Jupyter Notebook)
* CRUD Module (Python, needs to be coded/updated/**can copy mine**)
* **MongoDB** (database)
* **Dash** (Plotly)
* **Plotly** **Express** (another lib)
* **dash**-**leaflet** (another lib to see)
* **pandas** (more lib)
* **base64** (lib)
* **Jupyter** Notebook (tool to **run python project** file

## Details

This section provides more details about the code such as the examples from the code and details for that along with different testing with screenshots.

### Code Example

In the code below different libraries are helping achieve different things. For example, dash app to run insight Jupyter notebooks, providing the components for interactive maps, using leaflet, filtering data, output, callback, plotting, manipulation, CRUD, and encoding images.

### *A white background with black and white clouds Description automatically generated with medium confidence*

### Tests

In the screenshot below you can see how the unique identifier was coded using my name, the course name and the logo which was fitted using the UX design PDF report.

*A red and black dashboard with a dog head

Description automatically generated*

Starting the logo images, and establishing unique identifier.

A computer screen shot of a computer code

Description automatically generated

Starting the options to have radio buttons as widgets

A computer code on a white background

Description automatically generatedEstablishing how different data will be updated using different categories and names

A computer code with red and white text

Description automatically generated

Establishing Graphing system for the pie graph to show values

A screen shot of a computer

Description automatically generatedEstablishing the mapping system to be updated dynamically

A screen shot of a computer code

Description automatically generated

Establishing the longitude latitude id, showing last screenshot which starts the dash app successfully.

A screenshot of a computer

Description automatically generated

Proof of widgets working

*A screenshot of a computer

Description automatically generated*

*A pie chart with numbers and a few words

Description automatically generated with medium confidence*

Proof of the pie and mapping charts working.

*A map with a blue pin on it

Description automatically generated*

## Contact

Your name: Noah Khomer

Email: [noah.khomer@snhu.edu](mailto:noah.khomer@snhu.edu)

Linkedin: <www.linkedin.com/in/noah-khomer>

Course: CS 340 – Client Server Development

Project: 2

Thank you for reading my Readme file.