# CS 340 README Courtney Warner

## About the Project/Project Title

The Grazioso Salvare Dashboard is a full-stack web application designed to assist Grazioso Salvare in identifying and categorizing dogs suitable for search-and-rescue training. The dashboard interacts with a MongoDB database containing data from local animal shelters around Austin, Texas. Users can filter data based on rescue types, visualize the available dogs, and gain insights into the effectiveness of various breeds for different rescue scenarios.

## Motivation

The motivation for this project is to streamline the process of managing and analyzing animal shelter data. By offering an intuitive interface for CRUD operations (Create, Read, Update, Delete), the application makes it easier for Grazioso Salvare to manage dog records and identify search-and-rescue candidates, ultimately improving efficiency in training and decision-making.

## Getting Started

To get a local copy up and running simply go to (generic github for now):

git clone https://github.com/your-username/your-repository.git

cd your-repository

## Installation

To get a local copy up and running, you will need the following:

1. Python: The core programming language used to write the CRUD operations.
   1. Visit python.org for installation instructions.
2. MongoDB: The database used to store and manage animal shelter data.
3. Visit mongodb.com for installation instructions.
4. Ensure that MongoDB is running on your local machine or that you have access to a MongoDB server with the proper credentials (username, password, host, port, and database).
5. Jupyter Notebook: The environment for running and testing the code interactively.
6. To install the required libraries and launch Jupyter Notebook, enter the following commands in your terminal or command prompt:  
    pip install pymongo notebook.
   1. jupyter notebook  
      pymongo: The library used to interact with MongoDB from Python.
   2. This library will be installed with the above commands (pip install pymongo is included).

## Usage

Libraries are essential for implementing the CRUD operation and interacting with MongoDB.  
Additionally, libraries can have many purposes and uses.   
Libraries Used:

* Dash
  + Purpose: Web application framework for creating interactive dashboards.
* Pandas
  + Purpose: Data manipulation and analysis library for handling structured data.
* Matplotlib
  + Purpose: Plotting library for creating static and interactive visualizations.
* NumPy
  + Purpose: Fundamental package for numerical computing, supporting multi-dimensional arrays and matrices.
* Plotly
  + Purpose: Graphing library for creating interactive plots and visualizations.
* pymongo
  + Purpose: Library for interacting with MongoDB from Python, enabling database operations.

### Code ExampleA close up of a text Description automatically generated

### A screenshot of a computer Description automatically generated Incorporating the logo and unique identifier:

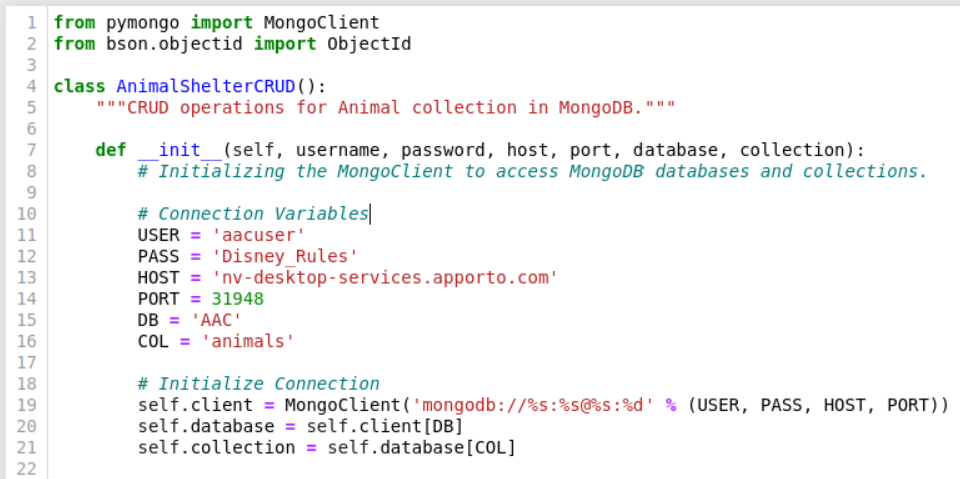
### Incorporating the logo with an anchor tag enhances the visual appeal of the dashboard and reinforces the connection between the application and Grazioso Salvare's mission. It serves as a direct gateway for users to learn more about the organization, fostering a stronger relationship between the application and its stakeholders. This is a screenshot of the implementation:

  
This short video shows the anchor tag in function:  

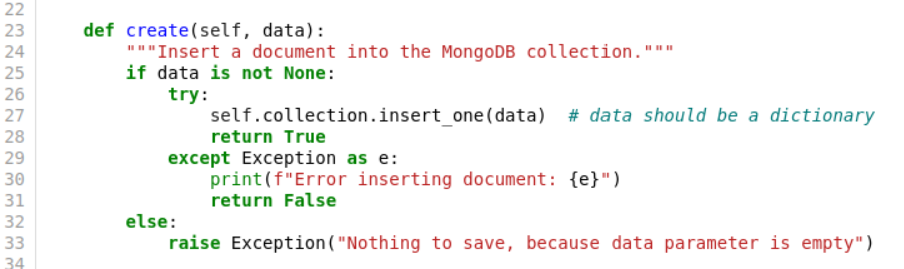

### Key functionalities:

### CRUD Operations: Users can create, read, update, and delete dog records.

1. Initialization: Connects to the MongoDB database using specified credentials.



1. Create: Inserts a new animal record into the collection.



1. Read: Retrieves records based on a specified query.



**CRUD Tests:**

Test setup with assertions to check the functionality of the CRUD methods.

To ensure the Create and Read operations of the AnimalShelterCRUD class work correctly.

Import the Class and Initialize the ConnectionA screen shot of a computer

Description automatically generated

**Test Insert/Create Operation:**A screenshot of a computer code

Description automatically generated

**Test Read Operation**:

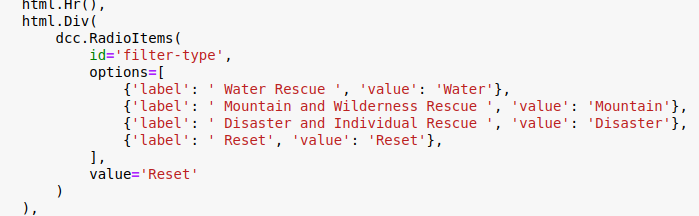
A close up of text

Description automatically generated

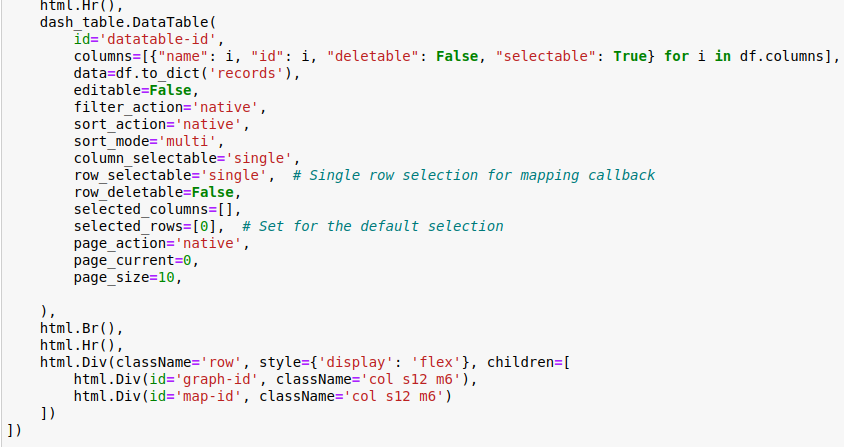
**Expected Output**:

A close-up of a computer screen

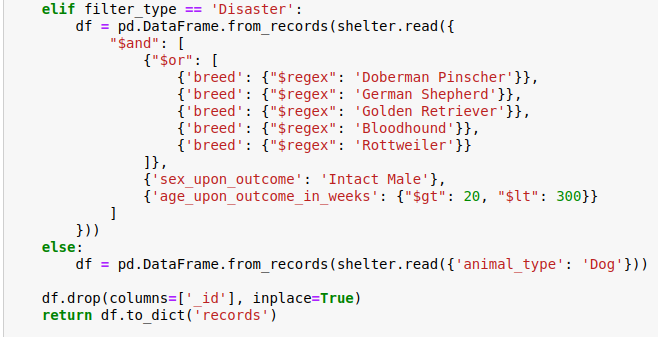
Description automatically generated  
2. Data Filtering: Users can filter dogs based on rescue types.

Screenshot of Filter with radio button labels:  


The screenshot shows a data table for displaying and interacting with dog records.

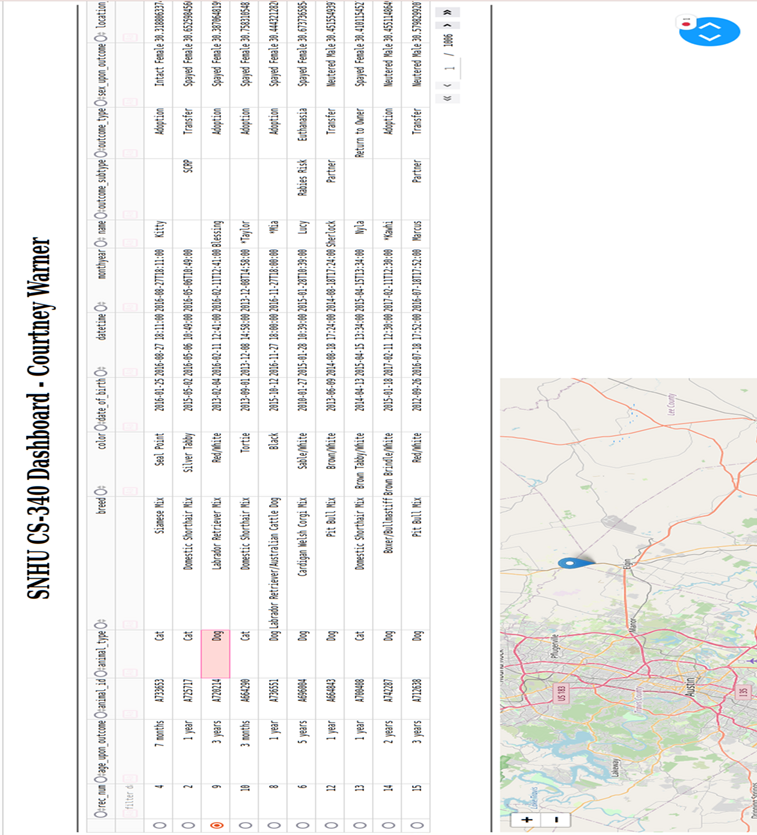
  
These screenshots define a callback function that dynamically updates the data displayed in a Dash data table based on the type of rescue selected by the user from the radio buttons. It retrieves specific records from the shelter based on the selection criteria and returns them for display. If no filter is selected, it returns an empty list to clear the table.





1. Visualization: The dashboard visualizes the available dogs and provides insights into the effectiveness of various breeds for different rescue scenarios. With interactive filter options users can filter dog records by rescue type (Water, Mountain, Disaster, or Individual Tracking), facilitating quick access to relevant data for training decisions. The dashboard includes a responsive table that displays detailed information about available dogs, allowing Grazioso Salvare to easily identify suitable candidates based on age and breed.

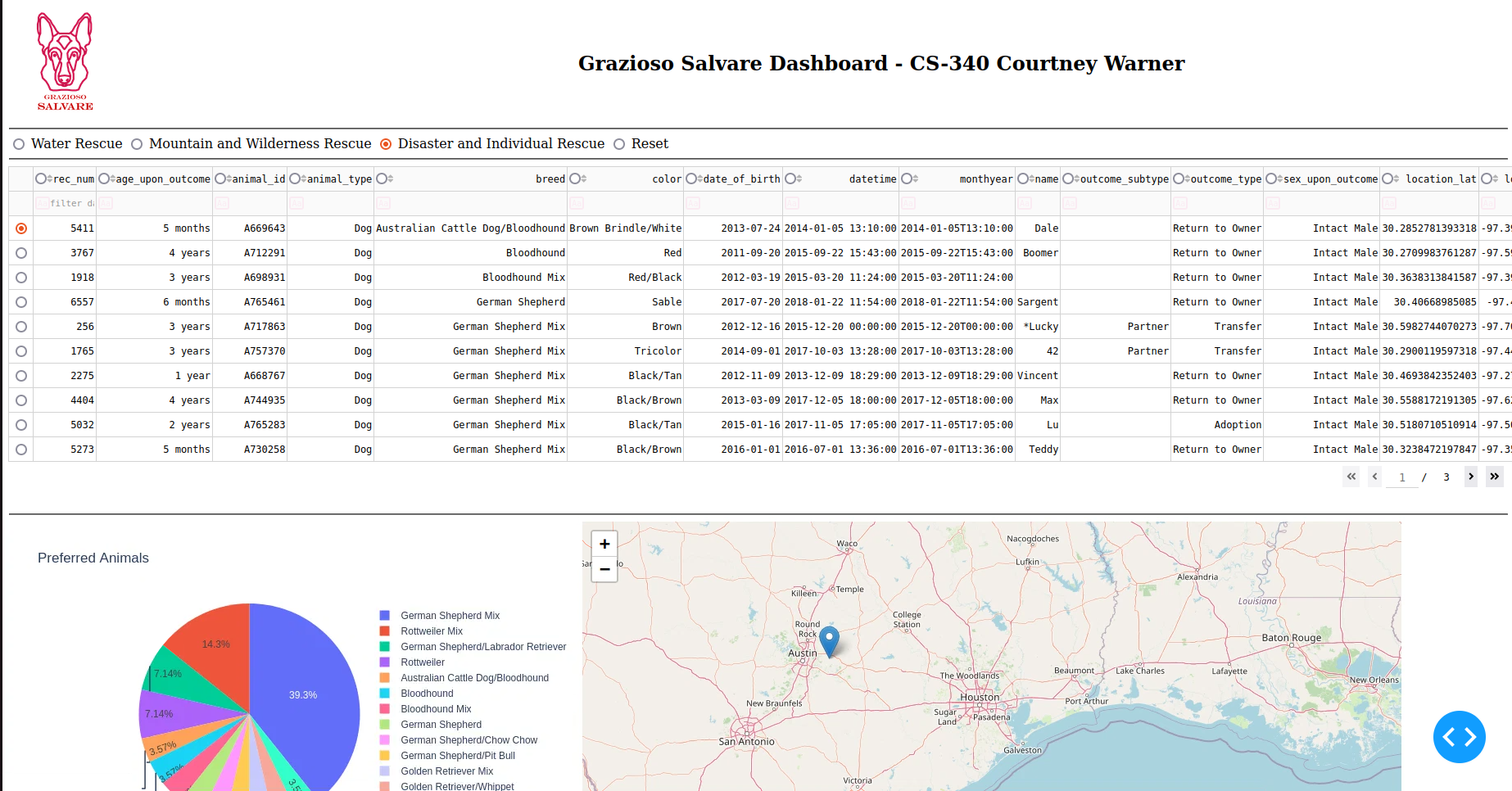
Geolocation charts and pie charts provide insights into the distribution of dogs and the effectiveness of different breeds for various rescue operations, supporting informed training strategies.

This initial screenshot shows the dashboard before the widgets and filters are implemented:  


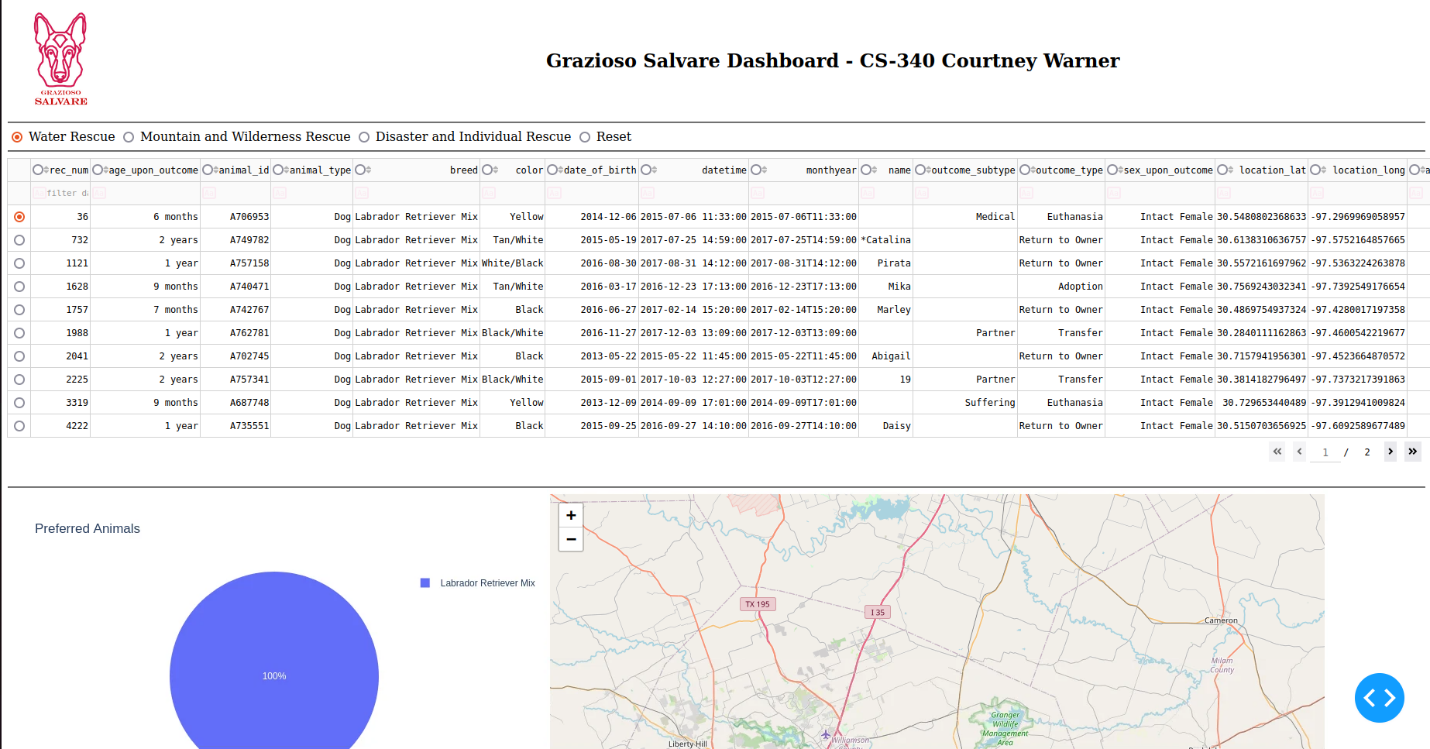
These screenshots show the dashboard with the default reset, the disaster and individual rescue, mountain and wilderness rescue, and water rescue filters respectively.

A screenshot of a map

Description automatically generated





  
This video shows the widgets and filters when selected and the anchor tag function:  


## Contact

Your name: Courtney Warner