# **Dashboard for Grazioso Salvare – Animal Rescue**

## About the Project

The purpose of this project is to develop a software application that will assist Grazioso Salvare in identifying and categorizing dogs that are suitable for search-and-rescue operations. Grazioso Salvare specializes in training such dogs and requires an efficient method to locate potential candidates for their training programs. The application will allow the client to interact with the server to access data from five animal shelters in the Austin, Texas region. Its primary goal is to categorize and identify dogs that meet certain criteria, such as age and breed, so they can be trained for different types of search-and-rescue operations.

## Motivation

This full stack development project integrates client, server, and middleware components to provide a comprehensive solution. Its core functionality includes filtering and viewing data retrieved from a MongoDB database through a Python module and presenting it in a user-friendly web dashboard. The dashboard is designed to incorporate the company's branding and features visually rich widgets that enhance data presentation. It also enables interactive user engagement by offering dynamic filtering options that instantly update the displayed data.

**Tools Used**

The database used in this project is MongoDB, which is well-suited for scalable applications and projects that prioritize agile methodologies. MongoDB is a NoSQL document-oriented database that provides flexibility and stability, making it an ideal choice for projects with many concurrent users. When combined with Python, MongoDB allows for the development of database applications and modules that are quick and relatively easy to create.

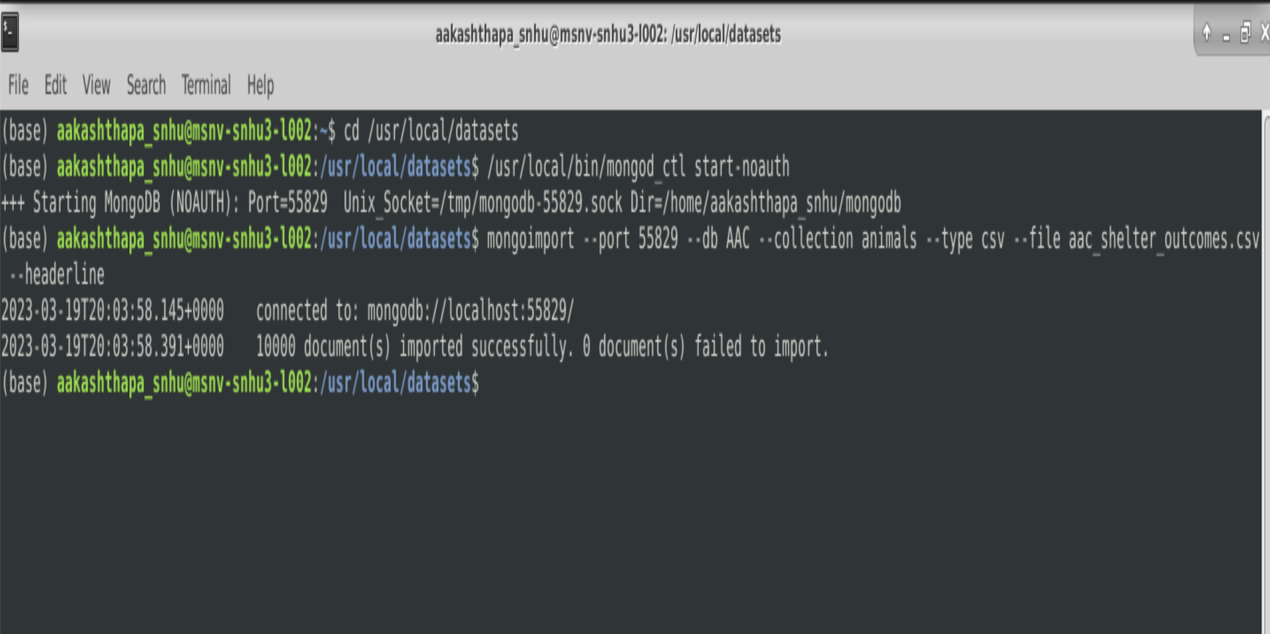
Middleware was implemented using Python, which is a versatile programming language that works well with MongoDB. The PyMongo driver library enables easy manipulation and iteration functionality by providing compatibility between dictionaries and lists.

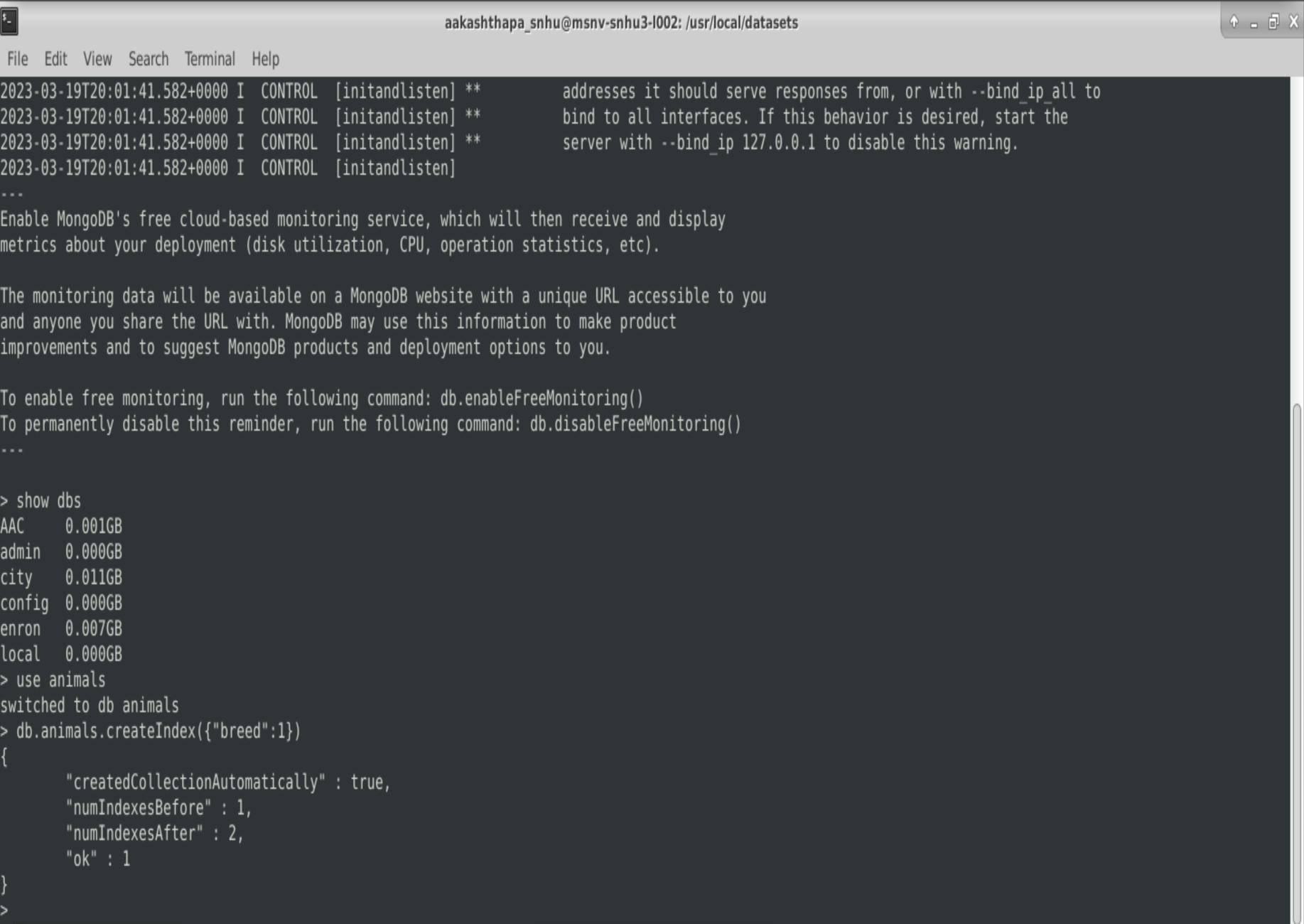
The client-side dashboard was developed using Jupyter Notebook, which is a web-based tool that offers a blend of an IDE and a data science environment. Jupyter Notebook supports several programming languages and provides easy access and usability as a web application.

To create interactive maps, graphs, and user interfaces, Plotly and Dash were utilized. Plotly is a graphing and analytics library that is functional with many programming languages, including Python. Dash, on the other hand, is a Python framework used for building web-based applications. Together, these tools offer a web-based dashboard with visualizations and widgets.

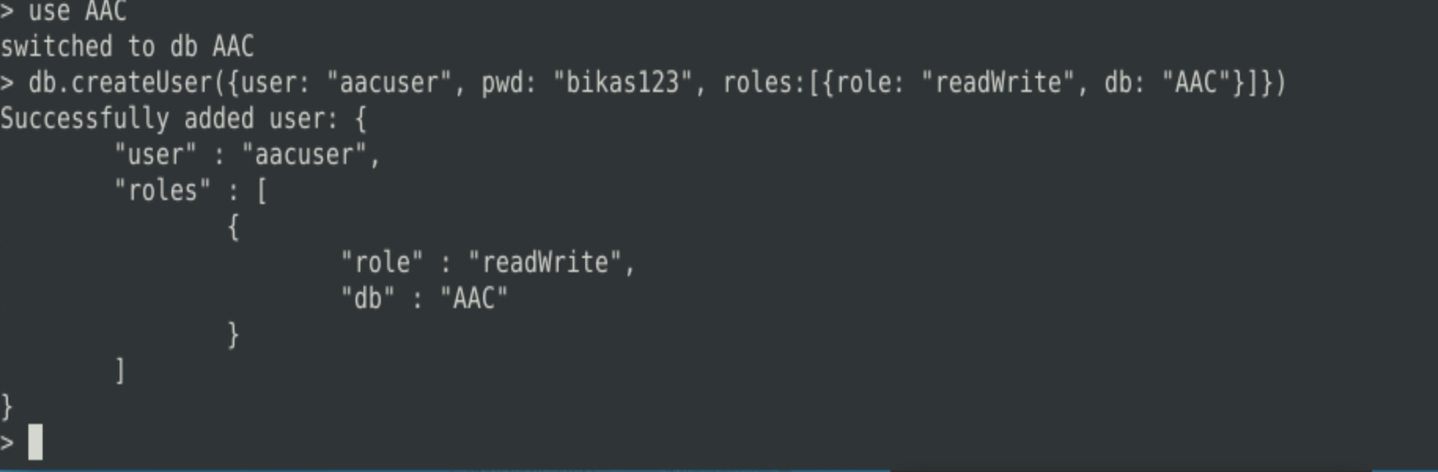
## Getting Started

To create a suitable setup, we utilize the MongoDB import tool to transfer the Austin Animal Center (AAC) Outcomes dataset into MongoDB. This involves uploading a CSV file, which can be accessed from the "/usr/local/datasets/" directory in Apporto and is identified as "aac\_shelter\_outcomes.csv". During the importing process, we designate the database name as "AAC" and the collection name as "animals". By doing so, we ensure that the dataset is accurately categorized and organized within the database.



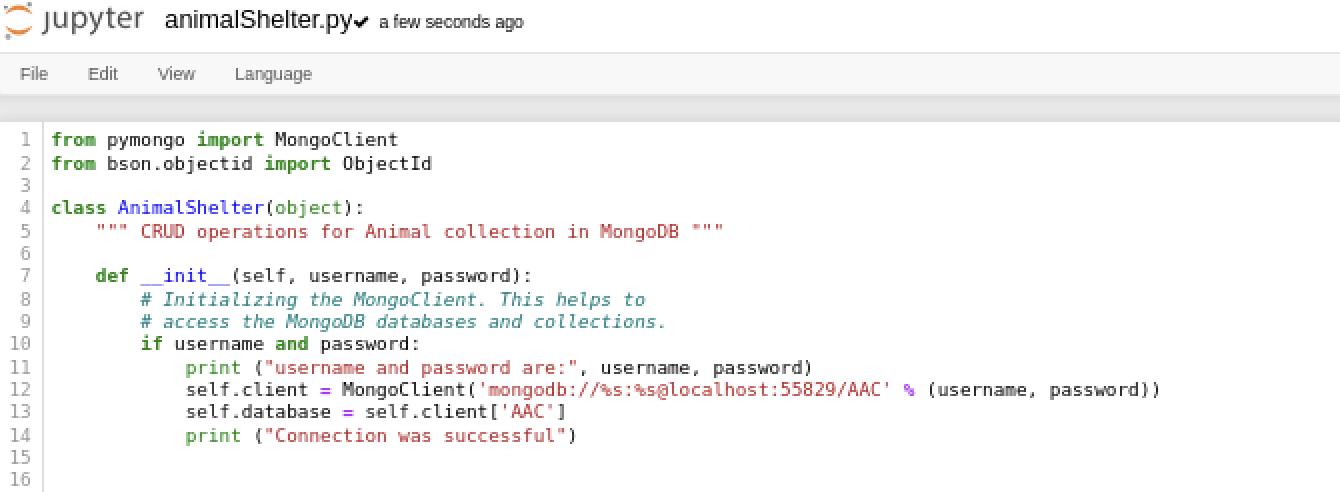


We created a new user account called "aacuser" for the AAC database within the mongo shell. This account was generated from scratch to ensure that it has the necessary permissions to access and manipulate the database.



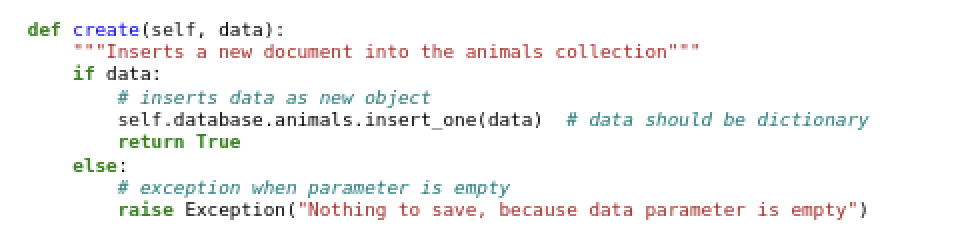
## Usage

To utilize the module, it is necessary to instantiate a new AnimalShelter class instance by providing the MongoDB username and password.

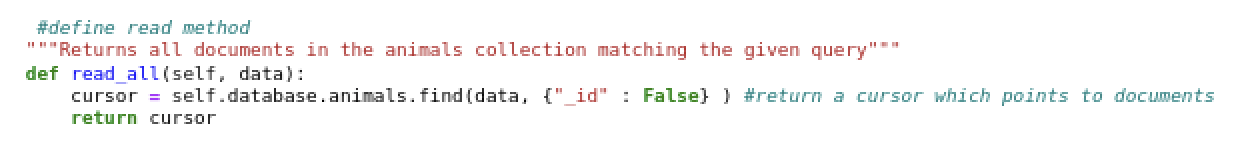


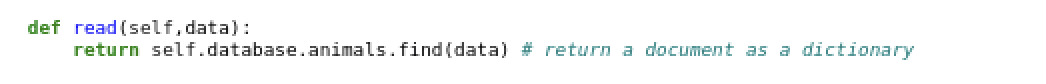
This section details the core functionality of the Python module, which acts as the intermediate layer for the software. The module supports the standard CRUD operations - CREATE, READ, UPDATE, and DELETE - that enable users to manipulate and view the database's contents. Screenshots depicting the methods for these functions are provided below.

### Create

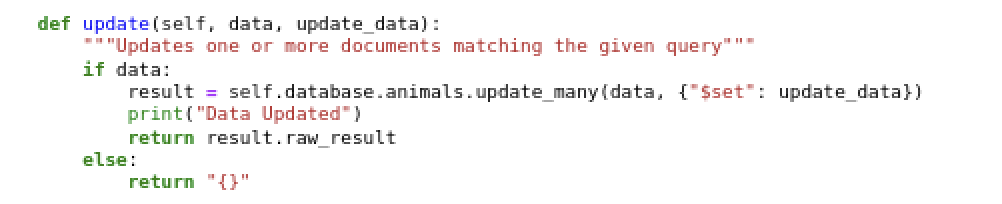


### Read





### **Update**



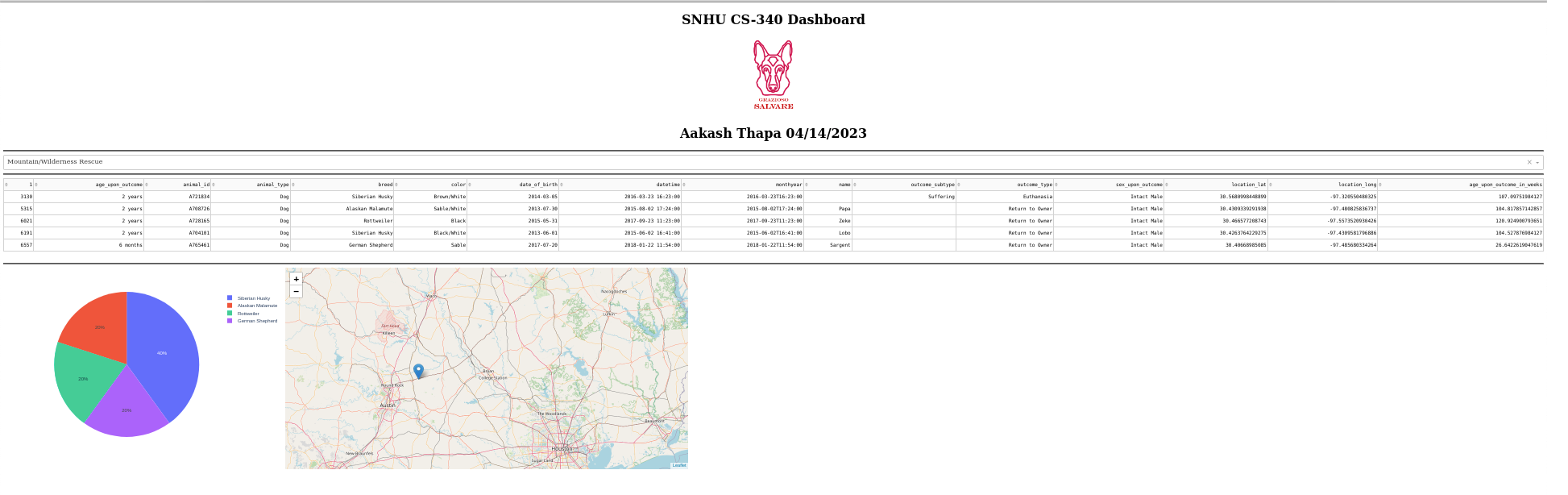
### **Delete**

### 

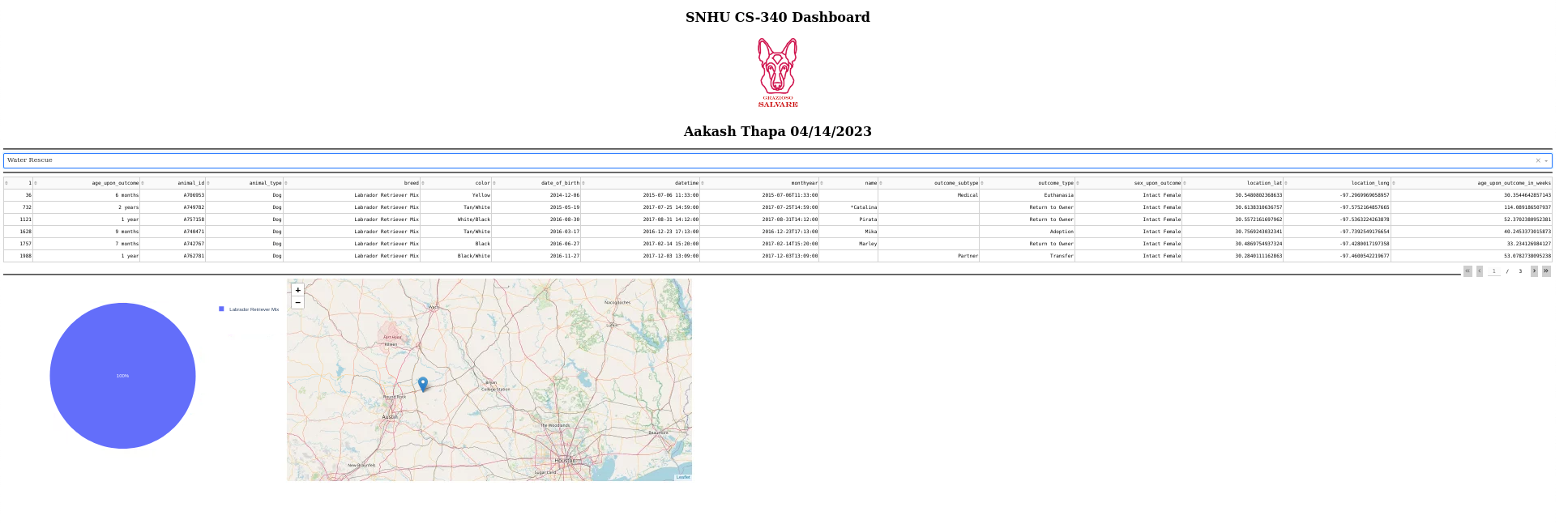
**Dashboard Interactivity**

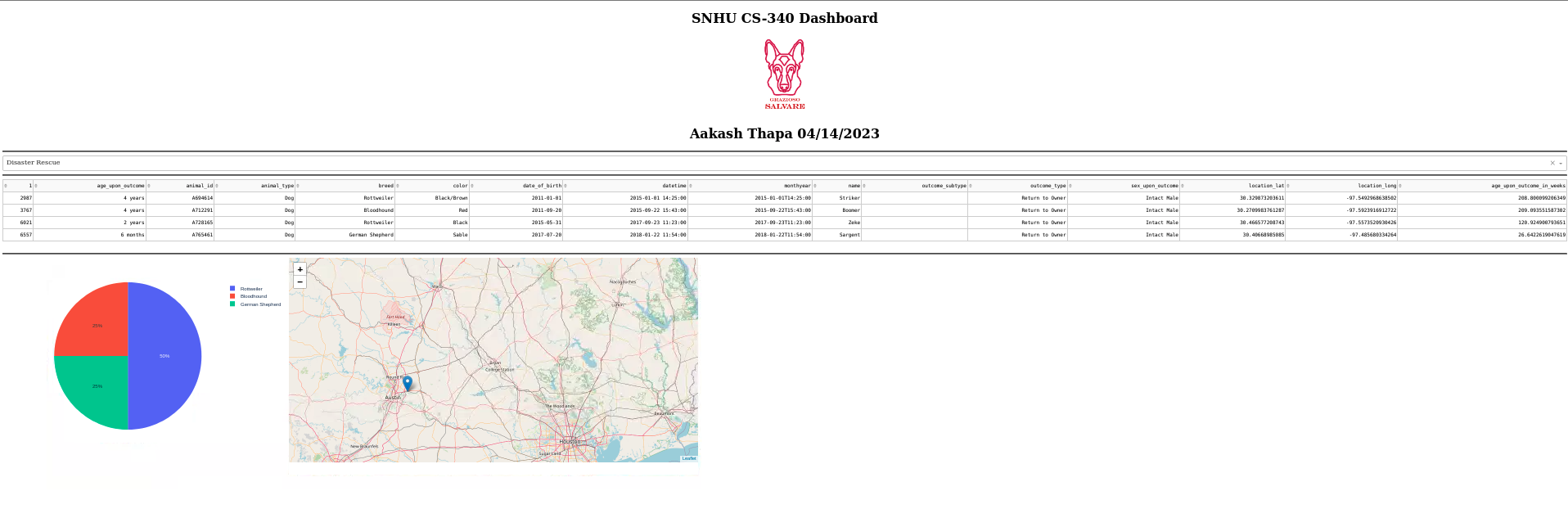
This section describes the interactive features of the client-side dashboard. Users can interact with the dashboard by navigating around the map and selecting specific data points. The dashboard also offers filtering options based on five key criteria: Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and a reset function that returns all widgets to their original state. These filtering options are illustrated below:

**Mountain/Wilderness Rescue**

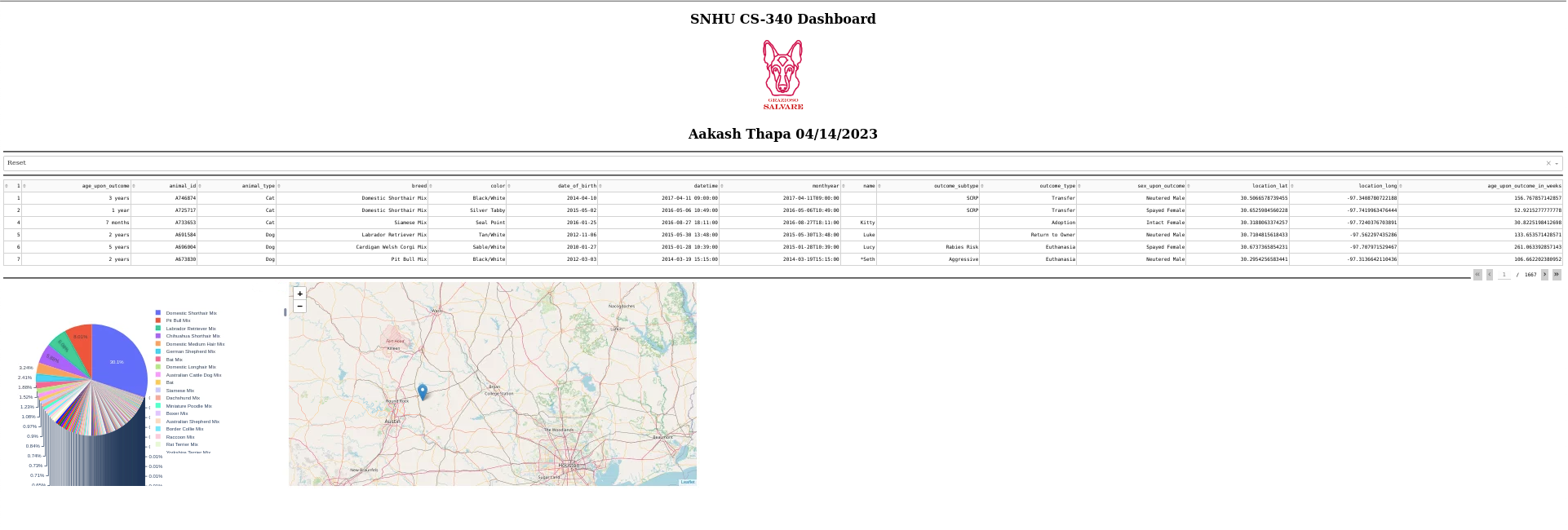


## **Water Rescue**

**Disaster Rescue**



**Reset**



## Challenges

Gaining a comprehensive understanding of MongoDB and Jupyter Notebook was a challenging aspect of this project that required significant effort and time investment. Additionally, testing the various CRUD operations for their correct functionality was a time-consuming process that involved a great deal of trial and error.

The primary challenge faced during the project was related to the dashboard's functionality and intermittent connection issues between the dashboard and the database. To address these issues, consider the following solutions for authorization/database connectivity problems:

* Confirm that the username and password are correctly specified.
* Verify that the username and password exist.
* Double-check the host and port information.
* Ensure that the dashboard and/or database processes are terminated successfully. If not, locate the process ID and terminate it.

## Contact

## Aakash Thapa

## aakash.thapa@snhu.edu