**Final Project README**

**Chris Wong**

**CS-340: Client/Server Development**

**October 22, 2023**

**Project Overview**

The Grazioso Salvare Rescue Animal Training Dashboard is designed to provide a user-friendly interface to access and visualize data from animal shelters. The data aids in identifying and categorizing dogs for search-and-rescue training. This project consists of a MongoDB database, a Python-based dashboard, and various interactive components. The goal is to streamline the selection of dogs based on specific criteria such as age, breed, and rescue type, reducing user errors and training time.

**Required Functionality**

The Grazioso Salvare Dashboard project aimed to create a user-friendly dashboard for interacting with and visualizing data from the Austin Animal Center Outcomes dataset. The required functionalities are as follows:

**Interactive Data Table**: An unfiltered data table displaying shelter animal data.

**Filtering Options**: User-friendly filter options to narrow down data by rescue type and preferred dog breeds.

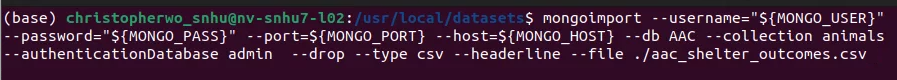
**Interactive Widgets**: Interactive data table and dynamic charts responding to user selections.

**Charts**: The project includes a geolocation chart and at least one additional chart.

**Getting Started**

* **Importing the AAC Shelter Database**

The initial step involves importing the AAC (Austin Animal Center) Shelter database, specifically the "animals" collection. This data is sourced from the "aac\_shelter\_outcome.csv" file and is accomplished using the "mongoimport" command.



* **Creating Simple Compound Indexes**

To improve both readability and database efficiency, simple compound indexes are established within the AAC database. These indexes are vital for optimizing query performance.

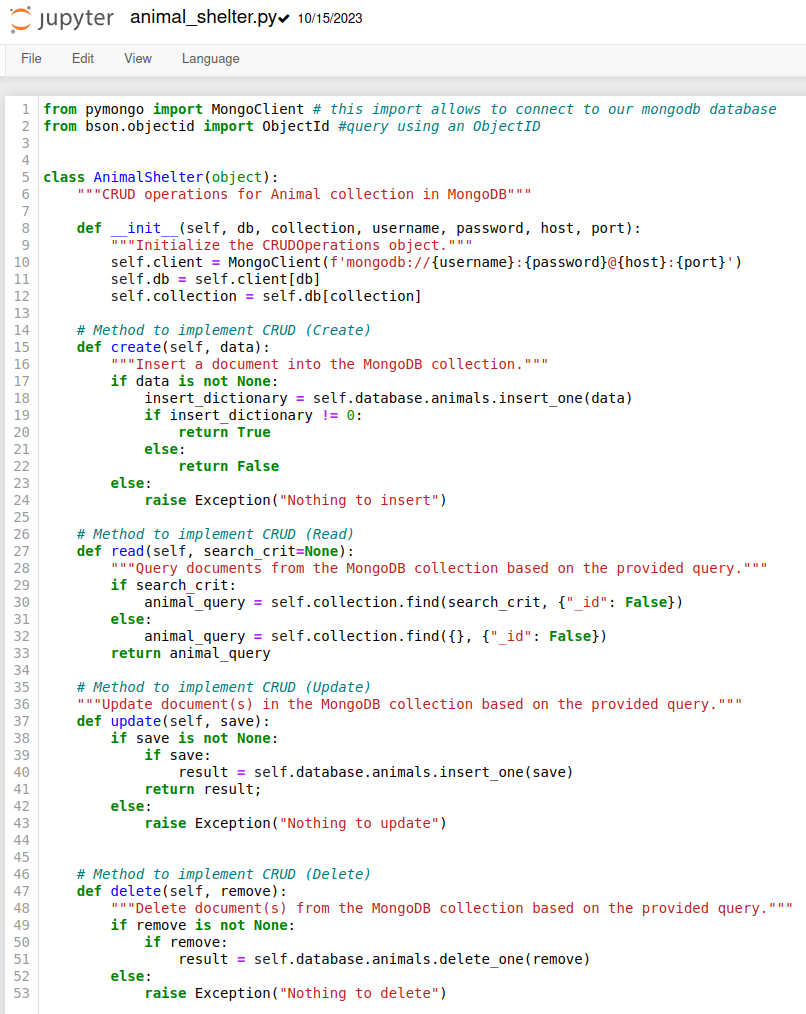
* 3. **User Authentication Setup & Access**

User authentication is a pivotal aspect of the project. It necessitates the creation of a new user account with read and write privileges, enabling CRUD operations within the AAC database and its "animals" collection. Validation of the newly created user account is essential. Ensuring that it has been granted read and write access to the AAC database and its associated "animals" collection is a critical part of the setup.



* **Development of Python Module**

In this step, the Python module is created. The ".py" file is developed to enable the utilization of CRUD operations, facilitating interactions with the MongoDB database, making it accessible to Python-based applications.



**Tools Used**

For the successful development and execution of the Grazioso Salvare Dashboard project, the following essential tools were utilized:

**Linux Terminal**

* **Usage**: A Linux terminal is indispensable for running various commands.
* **Rationale**: A Linux terminal provides a command-line interface for executing commands, managing files, and interacting with the system. It is an essential tool for software development and system administration.

**MongoDB Shell**

* **Usage**: Access to a Mongo shell is necessary for database interactions.
* **Rationale**: The MongoDB Shell offers an interactive environment for MongoDB database management. It allows for running queries, performing data maintenance tasks, and interacting with MongoDB databases.

**PyMongo**

* **Usage**: PyMongo, a crucial driver module, is used to facilitate seamless communication between Python applications and MongoDB databases.
* **Rationale**: PyMongo serves as a bridge between Python and MongoDB, enabling applications to interact with MongoDB databases. It simplifies data retrieval and manipulation, making it an ideal choice for this project.

**Jupyter Notebook**

* **Usage**: Access to Jupyter Notebook is required for creating the libraries and test scripts that enable the deployment of the Python files.
* **Rationale**: Jupyter Notebooks provide an interactive and flexible environment for developing, documenting, and testing Python code. They are particularly useful for data exploration, code development, and visualization, making them a valuable tool for this project.

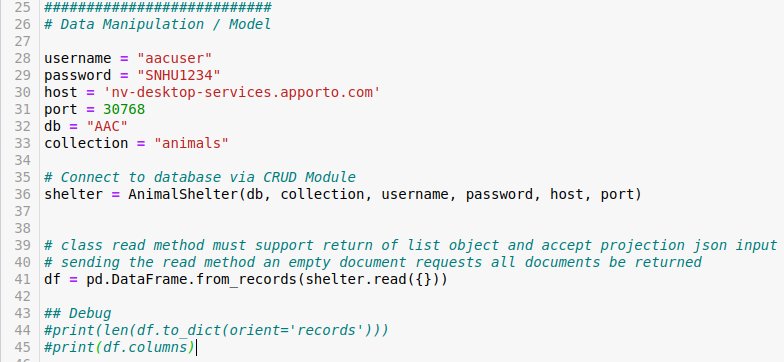
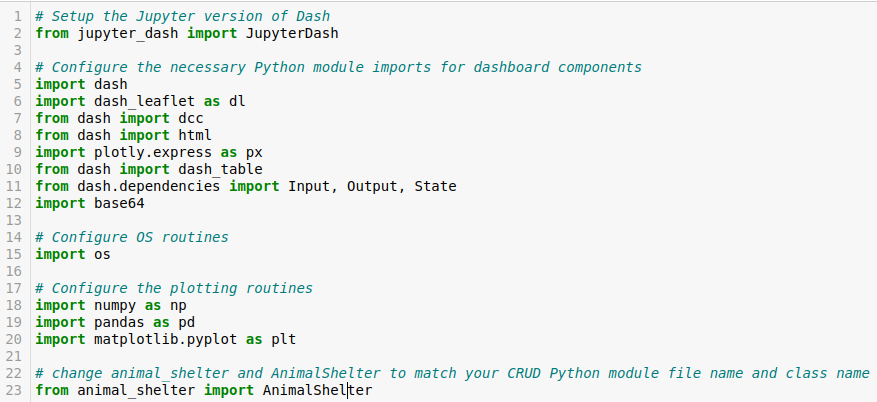
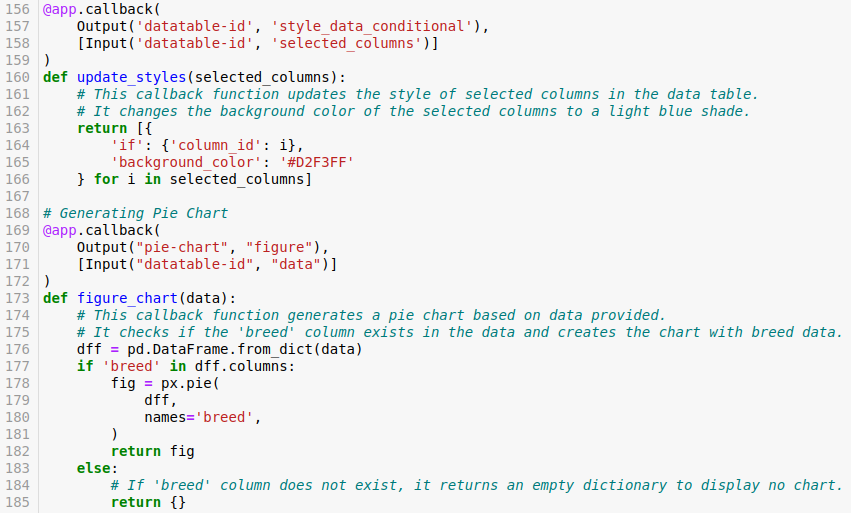
**Dash Framework**

* **Usage**: Dash, a web-based dashboard framework, was chosen as it streamlines the creation of interactive data visualizations.
* **Rationale**: Dash was selected for its seamless integration with Python libraries, serving as the view and controller component, which guarantees an intuitive user interface for this project. It allows for the development of dynamic, user-friendly data presentation.

## **Additional Resources**

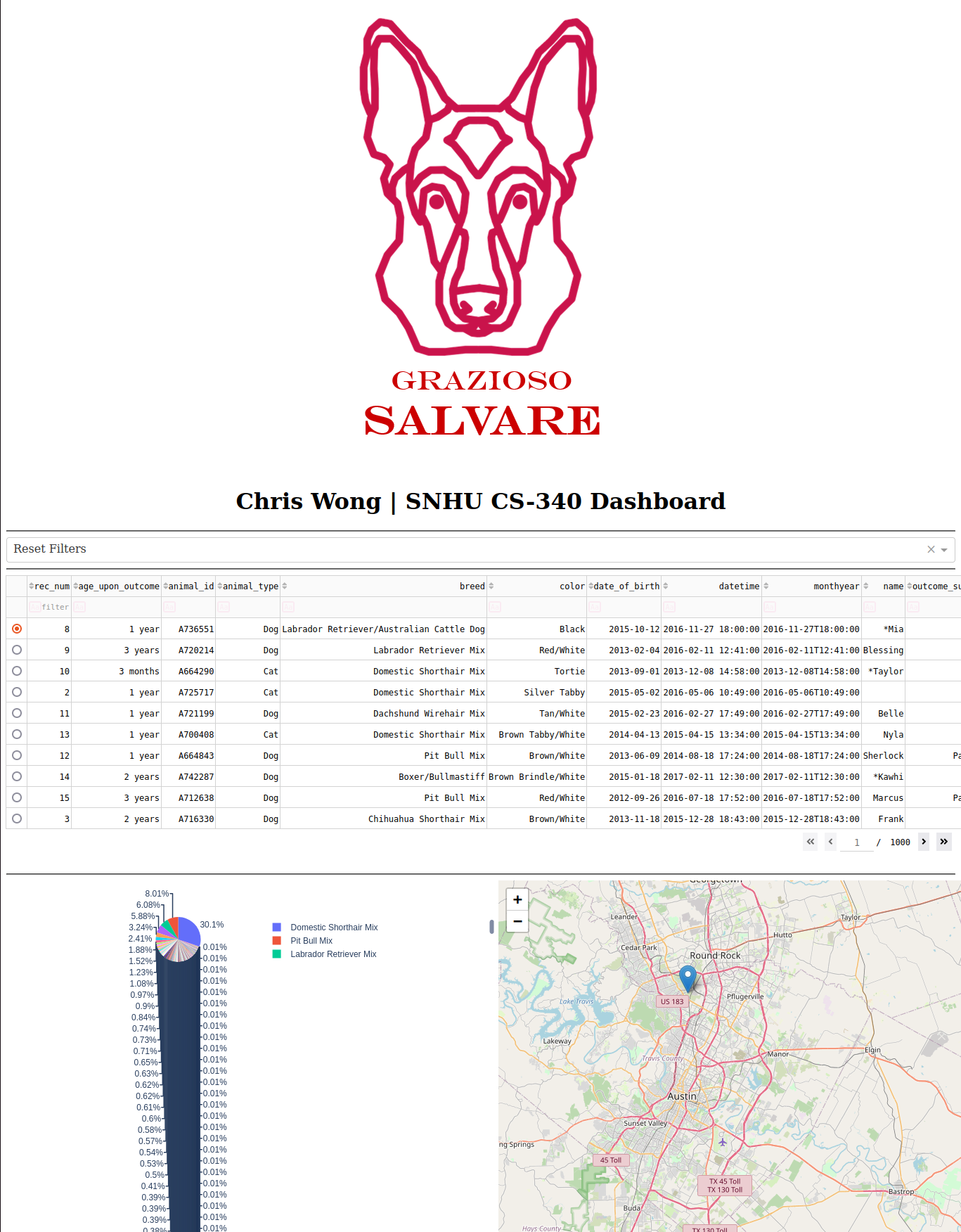
* [Dash Documentation](https://dash.plotly.com/introduction): Access the Dash framework documentation.
* [MongoDB Documentation](https://docs.mongodb.com/manual/): MongoDB documentation for data storage and retrieval references.

## **Usage**

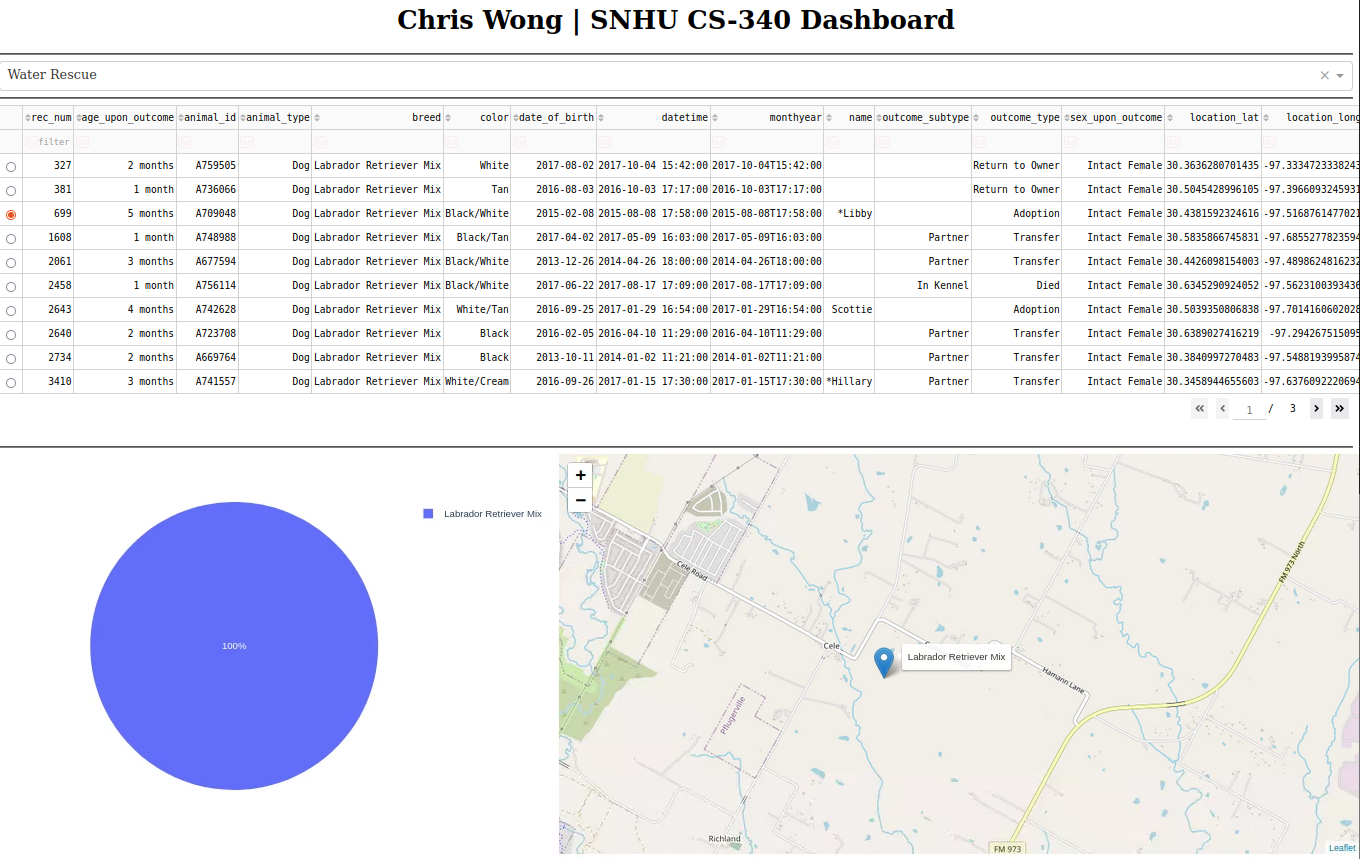
First, we import the necessary libraries / functions going to be utilized in this project and establish access to the AAC database through our newly created user:  
Second, we establish the dashboards features, including the filters and layout:  
Third, we define the multiple callback functions that dynamically update various components based on user interactions with the dropdown menu and dashboard:  


**Project Functionality Examples**

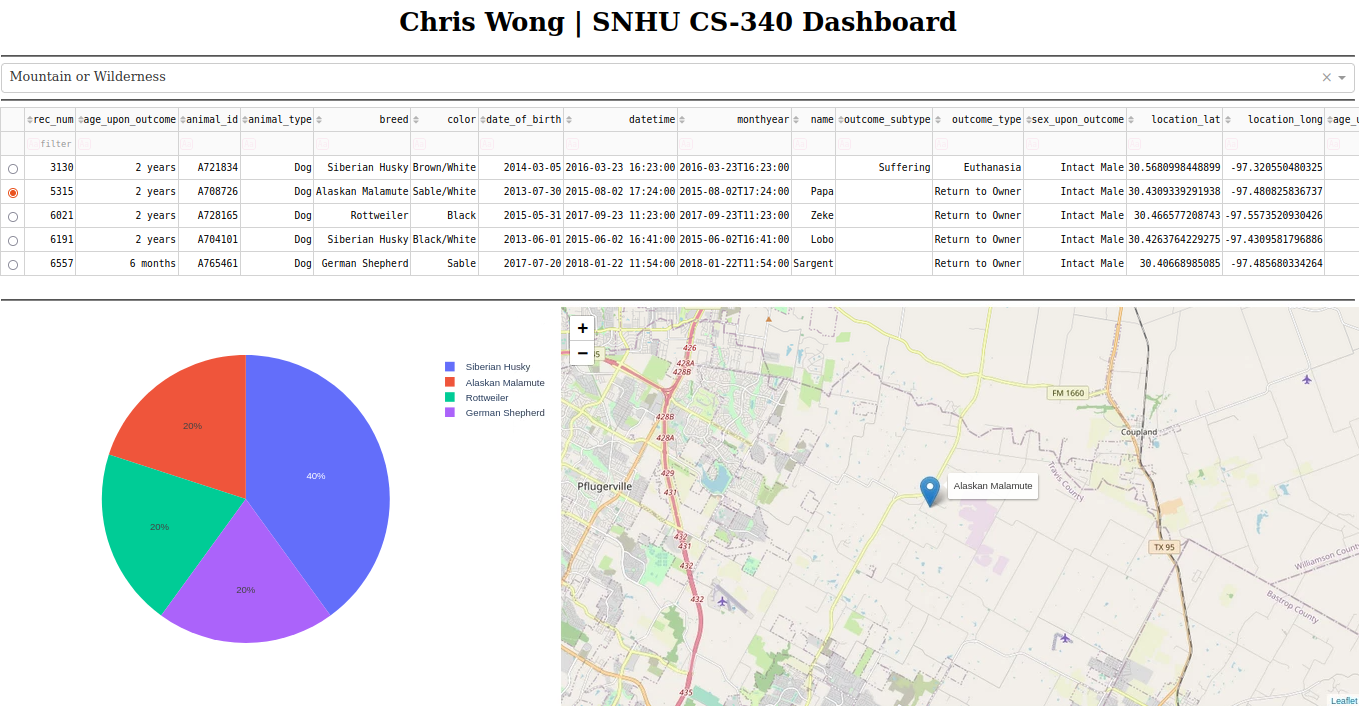
**Reset Filter:**

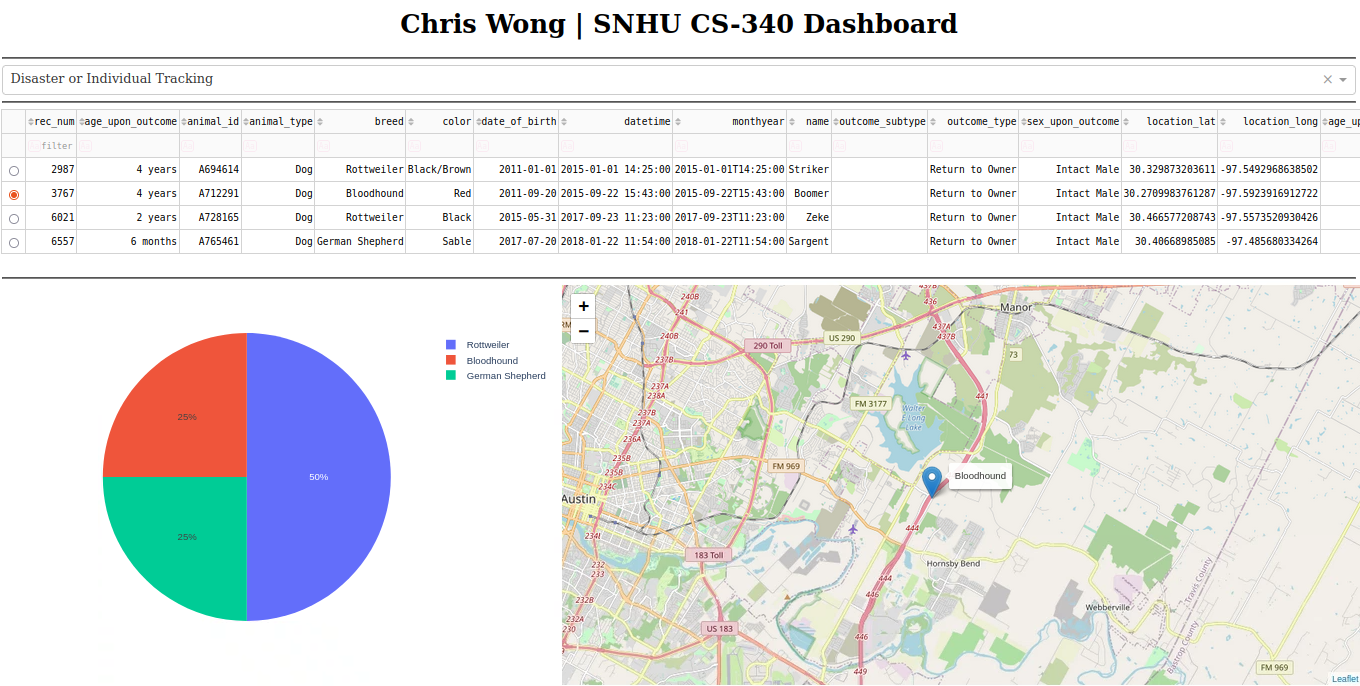


**Water Rescue Filter:**



**Mountain or Wilderness Filter:**

  
**Disater or Individual Tracking Filter:**



**Project Challenges**

* **Callback Debugging:** I resolved issues with callbacks by carefully reviewing their definitions and matching inputs and outputs to dashboard components.
* **Data Formatting:** Correct data formatting was achieved through Pandas documentation and experimentation.
* **Dependency Management:** Documenting package versions and using virtual environments helped manage dependencies.
* **Data Filtering Logic:** Careful testing and logic adjustments ensured proper data filtering.
* **Geolocation Chart Integration:** Challenges with geolocation chart were addressed by referring to documentation and experimentation.

**Project Conclusion**

The Grazioso Salvare Dashboard project successfully met the client's requirements and provides an efficient tool for managing and visualizing data. It enhances the ability to make data-driven decisions, making it valuable for Grazioso Salvare and similar organizations.

**Contact**

Name: **Chris Wong**

Email: [**christopher.wong1@snhu.edu**](mailto:christopher.wong1@snhu.edu)