# CS 340 README: Project Two

## Project Two: Animal Shelter

*This project involves building a dashboard for Grazioso Salvare, a rescue-animal training company, that interacts with data from multiple animal shelters. The dashboard provides users with an intuitive interface to view and filter animal data based on certain criteria, such as breed, rescue type, and outcome. The goal is to help identify dogs that are suitable candidates for search-and-rescue training.*

## Motivation

## *The motivation behind this project was to develop a user-friendly application for Grazioso Salvare that can filter animal data to identify the best candidates for search-and-rescue training. By integrating a MongoDB database with a Python dashboard, the application provides real-time access to animal profiles from regional shelters, helping the company make data-driven decisions in their animal rescue efforts.*

## Getting Started

*To get a local copy up and running, follow these steps:*

1. *Import Animal Data: Import the aac\_shelter\_outcomes.csv file into MongoDB using the mongoimport tool.*
2. *Create Indexes: Create simple and complex indexes to improve search efficiency in the database.*
3. *User Authentication: Set up an Admin account and an aacuser account to authenticate and access the database.*
4. *Install Python: Install Python on your computer to run the .py files and Jupyter Notebook files.*
5. *Run the Dashboard: Start the Dash app and connect by entering the local address provided.*

## Installation

1. *Python: Ensure Python is installed to run both the .py and .ipynb files.*
2. *MongoDB: MongoDB is required to store and manage the animal shelter data.*
3. *Libraries: Install the necessary libraries using pip:*

* *PyMongo: For MongoDB integration*
* *Dash: For building the dashboard*
* *Plotly: For visualizing data*
* *Pandas: For data manipulation*
* *Dash Leaflet: For embedding maps in the dashboard*

## Usage

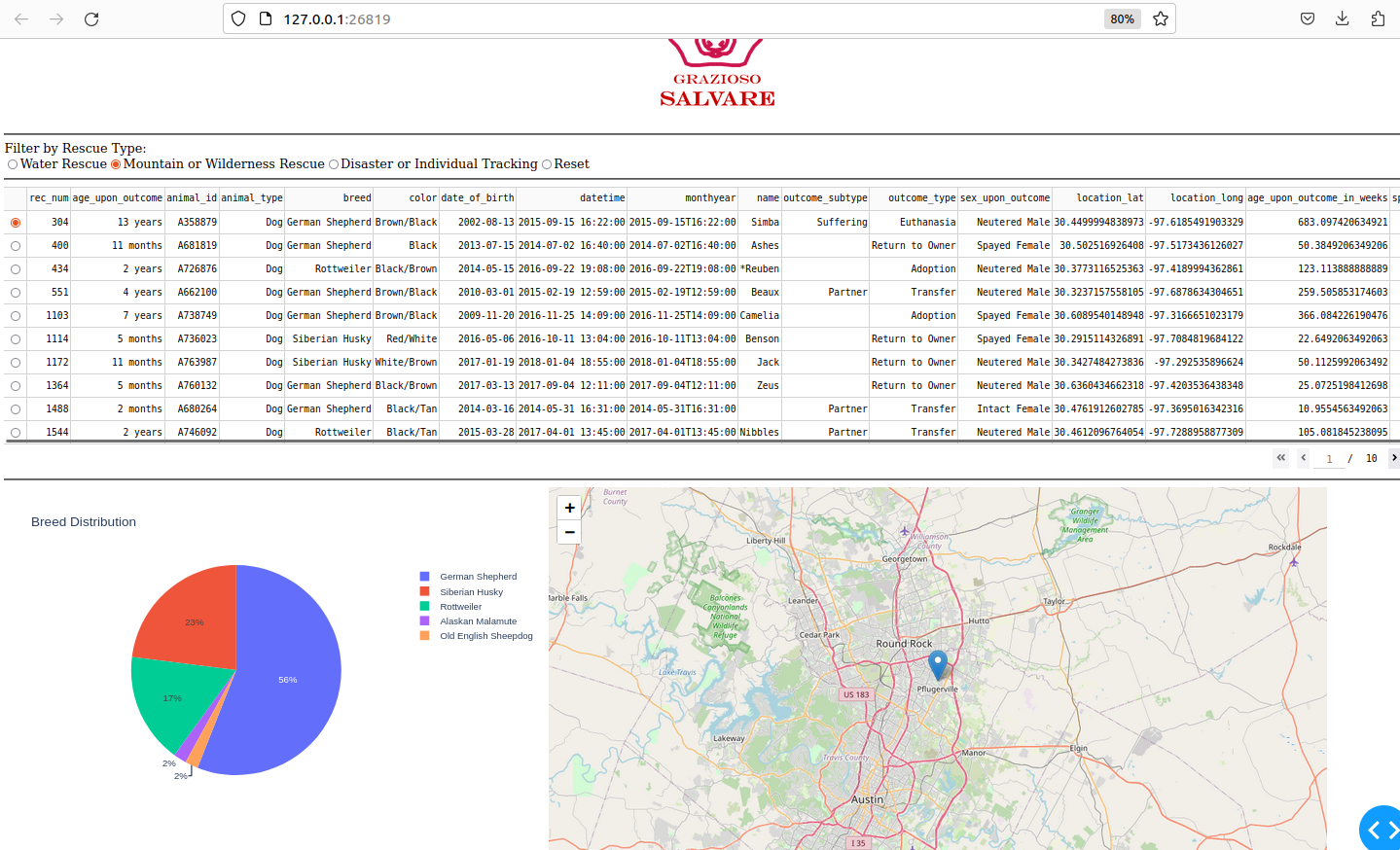
*To use the dashboard:*

1. *Access the Dashboard: Run the Dash app, and open the provided local address in your web browser to view the interactive dashboard.*

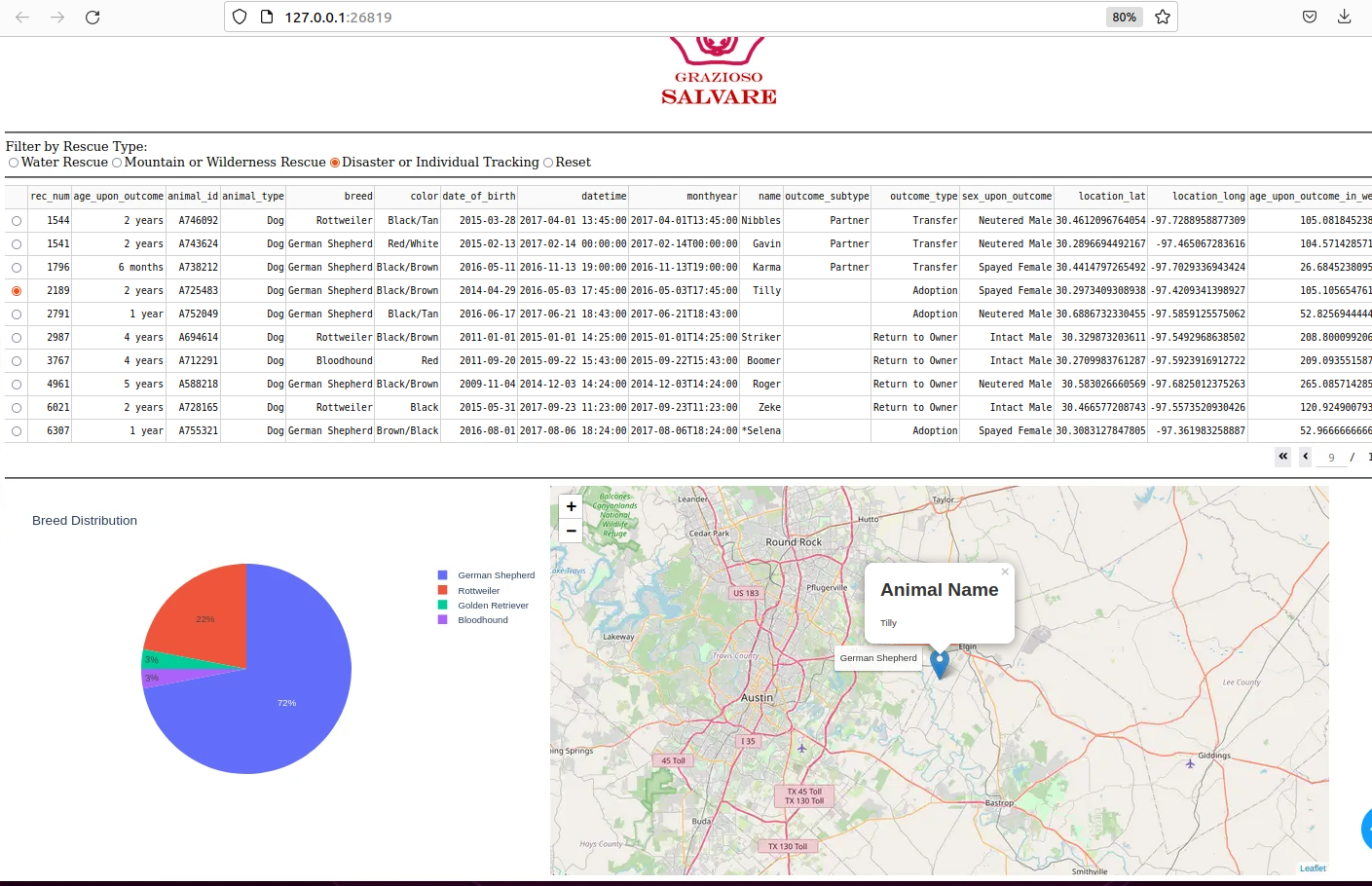
*A screenshot of a computer

Description automatically generated*

1. *Filter Data: Use the radio buttons in the app to filter the data by rescue type and see the results in a table and a pie chart.*

**

1. *View Map: Select a row in the data table to view the location of the selected animal on the map.*

**

## Challenges:

*One challenge I faced during this project was correctly filtering the data based on the outcome\_type field. Initially, when I tried to filter by a general term like "Adoption" or a specific rescue type, the query returned no data. This happened because the query was too specific and didn't account for all possible variations of the outcome\_type. To resolve this, I used the $in operator to include all possible values of the outcome\_type, ensuring the query would match any of the defined rescue types. Here's an example of the updated query: query = { 'outcome\_type': {'$in': ['Adoption', 'Transfer', 'Return to Owner', 'Euthanasia', 'Died', 'RTO-Adopt', 'Missing']}}. This approach allowed the data to be filtered correctly, ensuring that no data was missed.*

## Contact:

*Larissa Rojas*