# Author: Jeba Singh Emmanuel

Date: 4/7/2024

# CS 340 README

Scroll to the End to view screenshots from the dashboard

## About the Project/Project Title

This project is a CRUD app to access and manipulate data about animals in animal shelters. It is a web app using MongoDB for Storage. Security is provided using MongoDB’s inbuilt capabilities and web service authorization. The main file `animal\_shelter.py’ is a python module that provides CRUD functionality

## Motivation

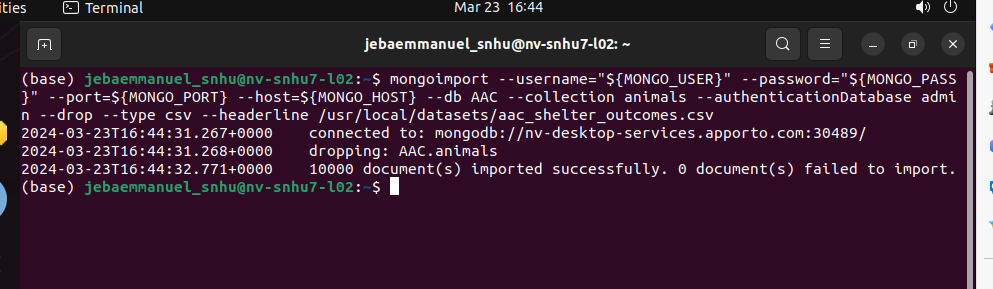
Global Rain has built this software for the innovative international rescue-animal training company, Grazioso Salvare. This software tracks data from animal shelters in Texas and helps Grazioso Salvare’s team identify dogs to train for SAR.

The code is open source and accessible at Github so other organizations may adopt it.

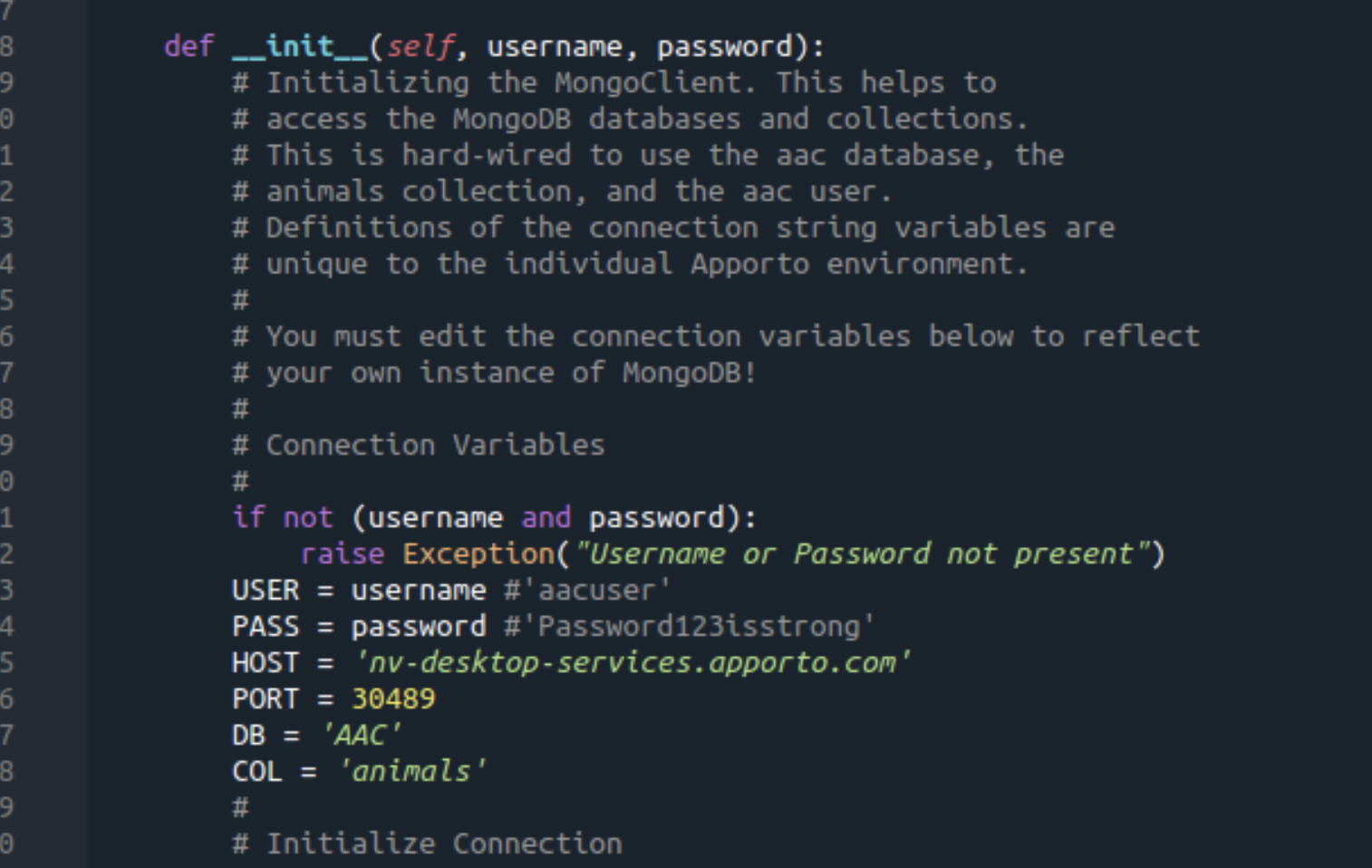
## Getting Started

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

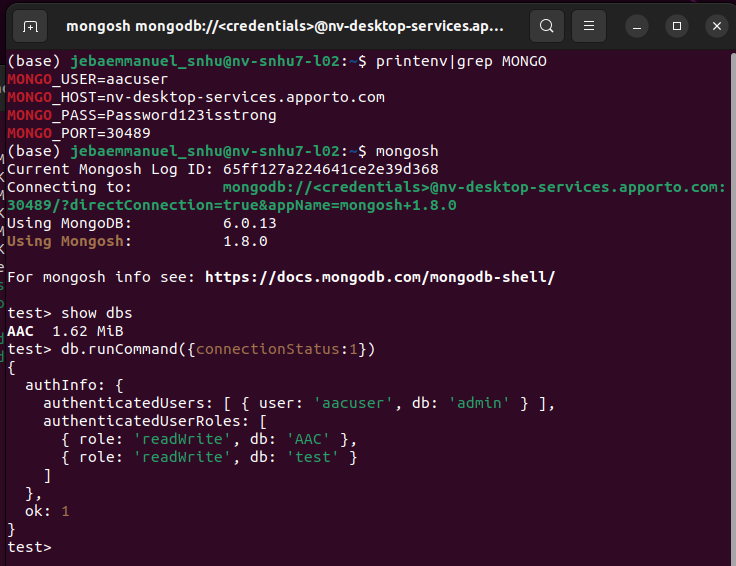
* Setup Mongodb on your system, create a database and import data from the animal shelters using mongoimport. Also setup non admin credentails to access this data.


* Modify the *animal\_shelter.py* file with any configurations changes to connect.



* Place the *animal\_shelter.py* file in the same directory as the Jupyter notebook you will use to run queries.
* Check you can login via mongosh using the credentials in the *animal\_shelter.py.* The class *AnimalShelter* currently hangs if the credentials are incorrect.



## Usage

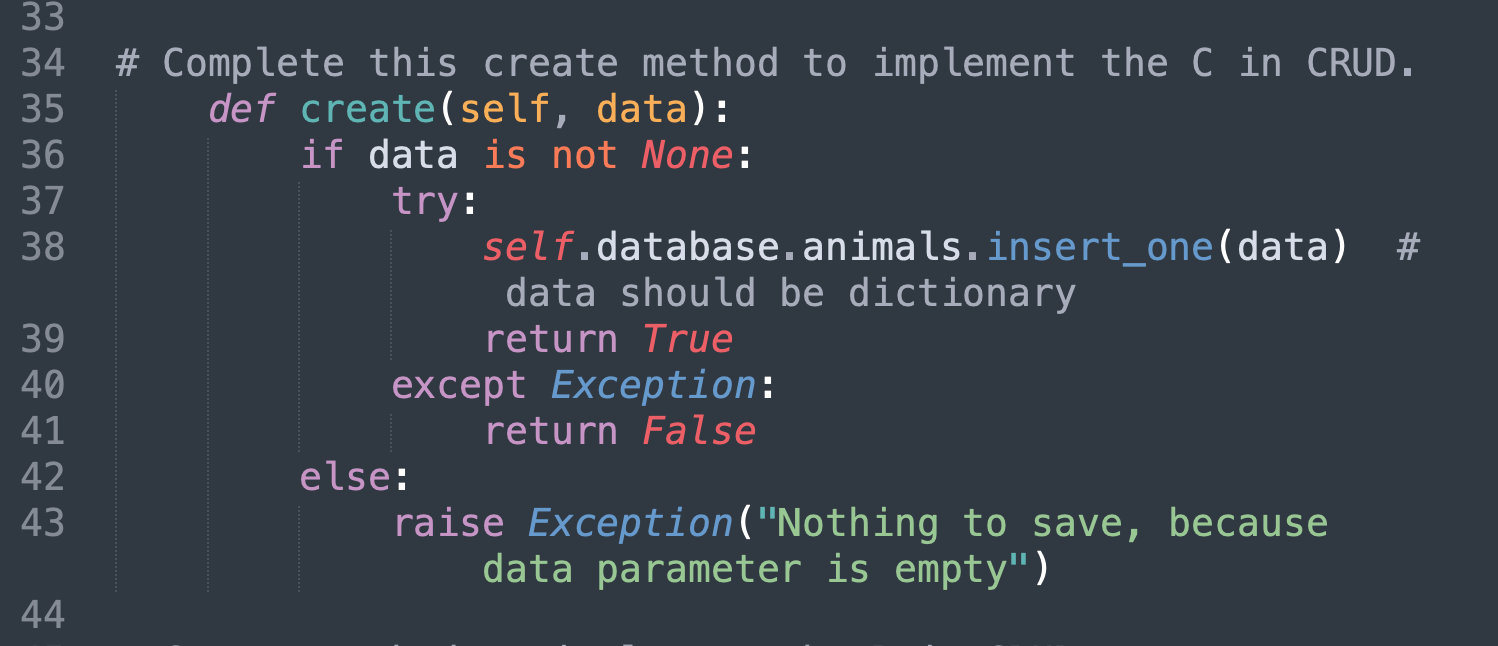
The main AnimalShelter module is a reusable module which provides most of the functionality to access the data from the 5 animal shelters in Austin. See below for example usage.

### Code Example

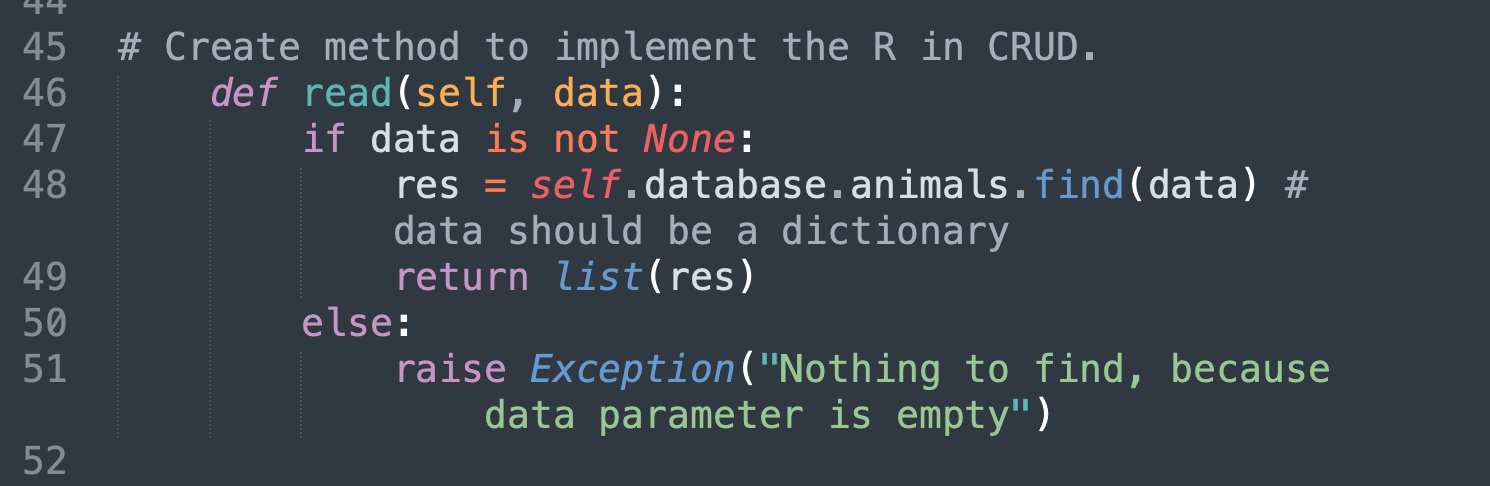
The class has hard coded credentials which will be modified later.

The crud methods are the following

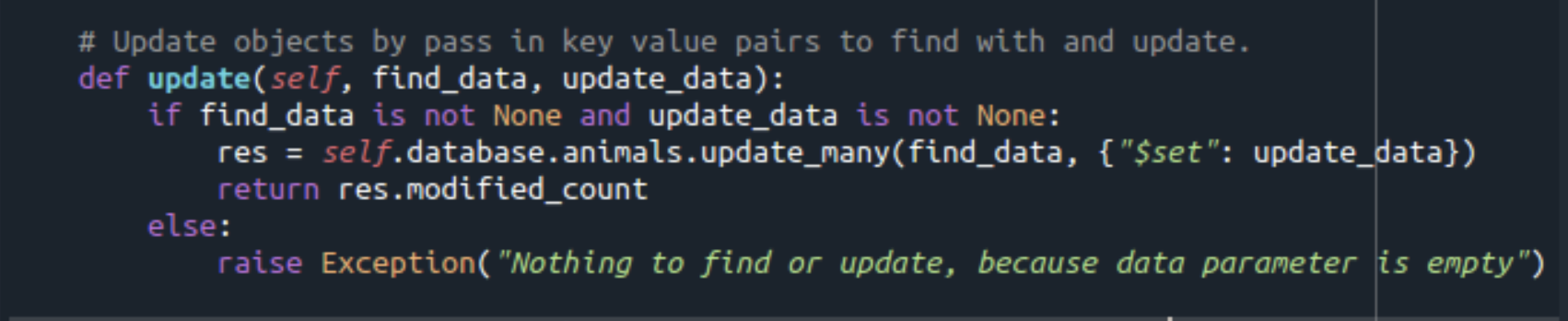
**Create** – which takes in a data map as a parameter and inserts into mongodb



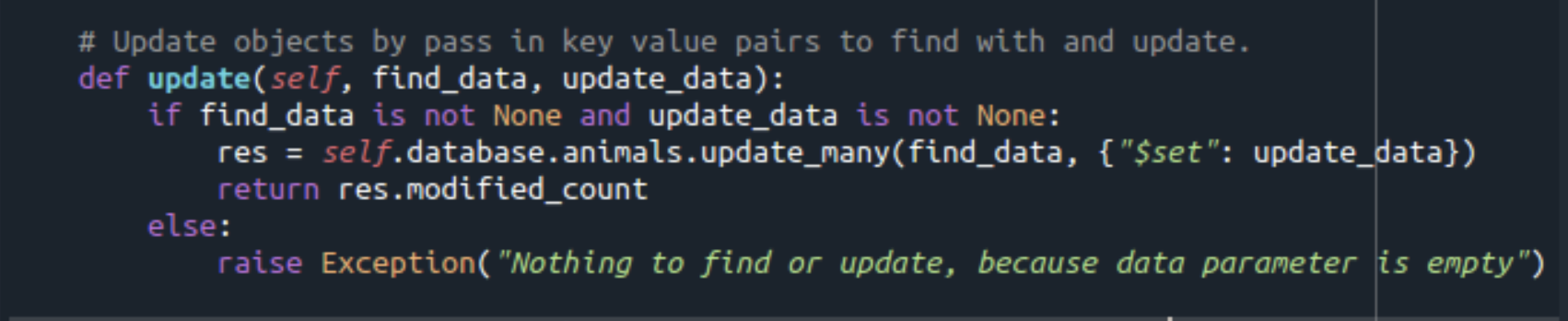
**Read** – takes key value pairs and searches in the mongodb database



**Update**



**Delete**



### Tests

You can use Jupyter notebooks to import the module and test the functionality



## Installation

*The following software is needed*

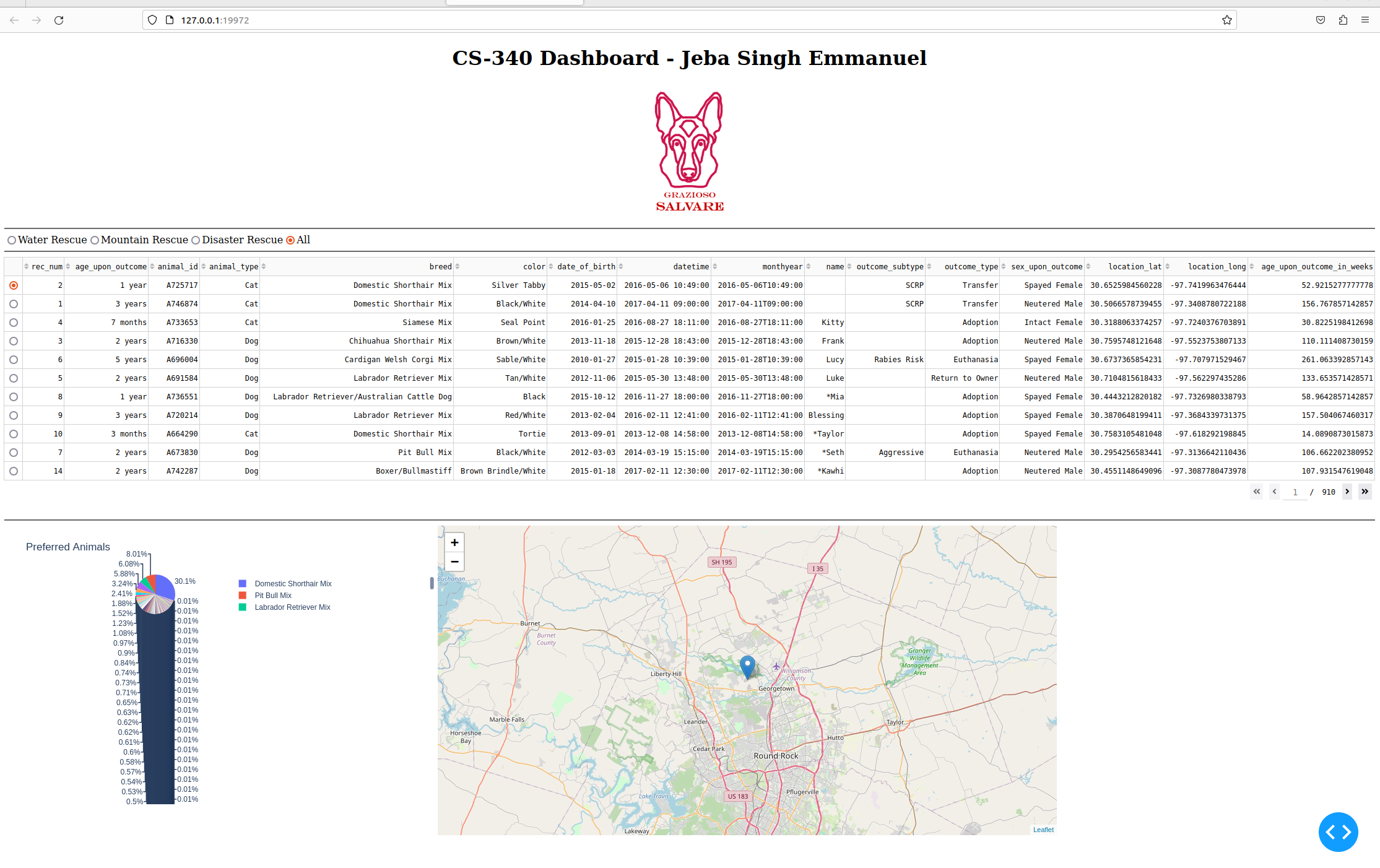
* [MongoDB](https://www.mongodb.com/resources/products/fundamentals/why-use-mongodb#:~:text=Using%20MongoDB%20can%20provide%20many,modern%20databases%20such%20as%20transactions.) - MongoDB is a powerful NoSQL database. It provides a way to store structured data and allows to query it quite effectively. It is also powerful at data manipulation and works very well with Python.
* [Dash](https://github.com/plotly/dash) – Dash is a Python UI framework that provides an MVC architecture. MVC architectures are a great solution for any UI project since it provides separation of concerns for business logic, data access and UI rendering. Dash leverages reacts real time capabilities with other parts of the stack to provide a unified session experience that is real time and powerful.
* [Jupyter Notebooks](https://jupyter.org/)
* Python 3
* PyMongo

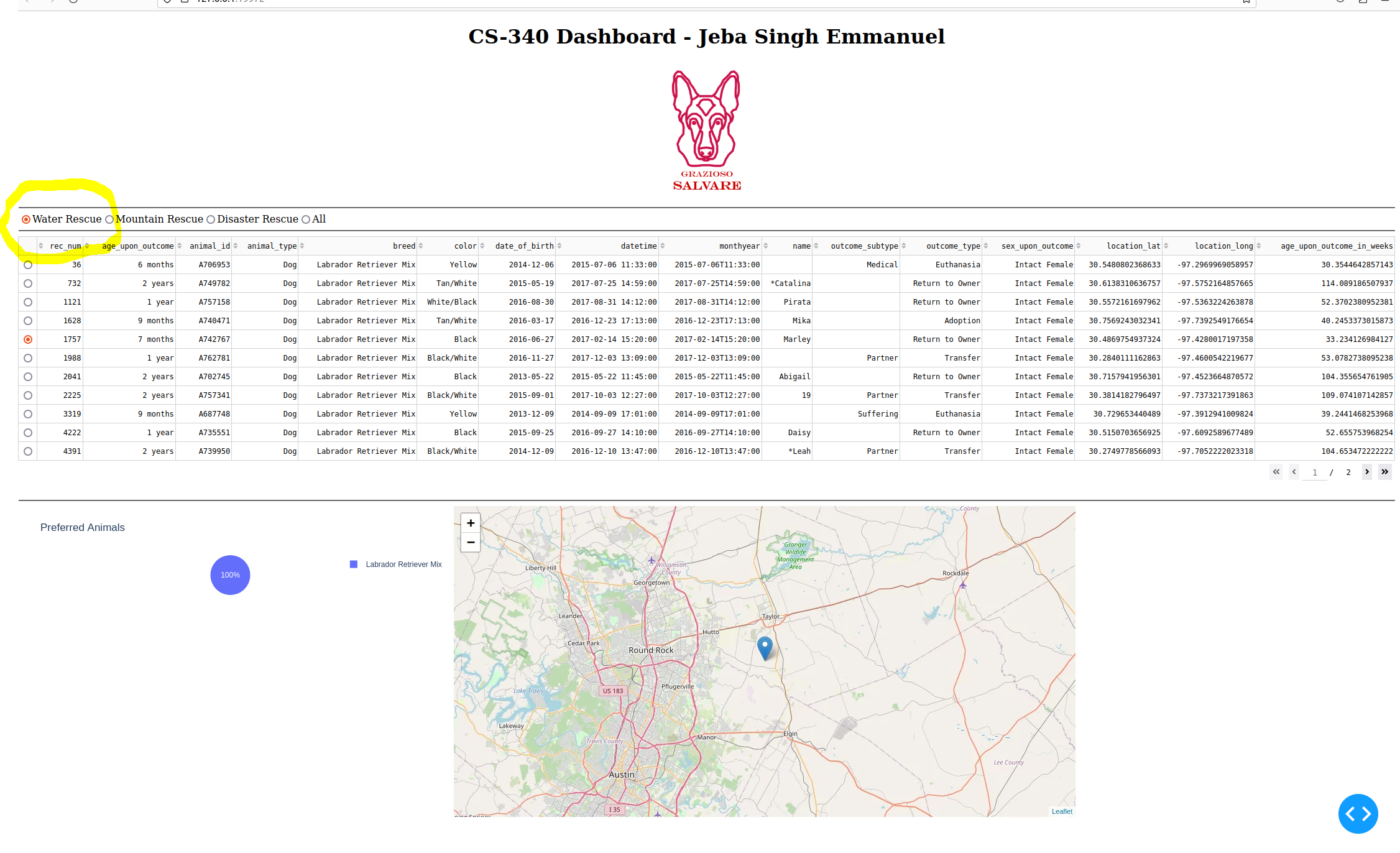
## Using the Dashboard

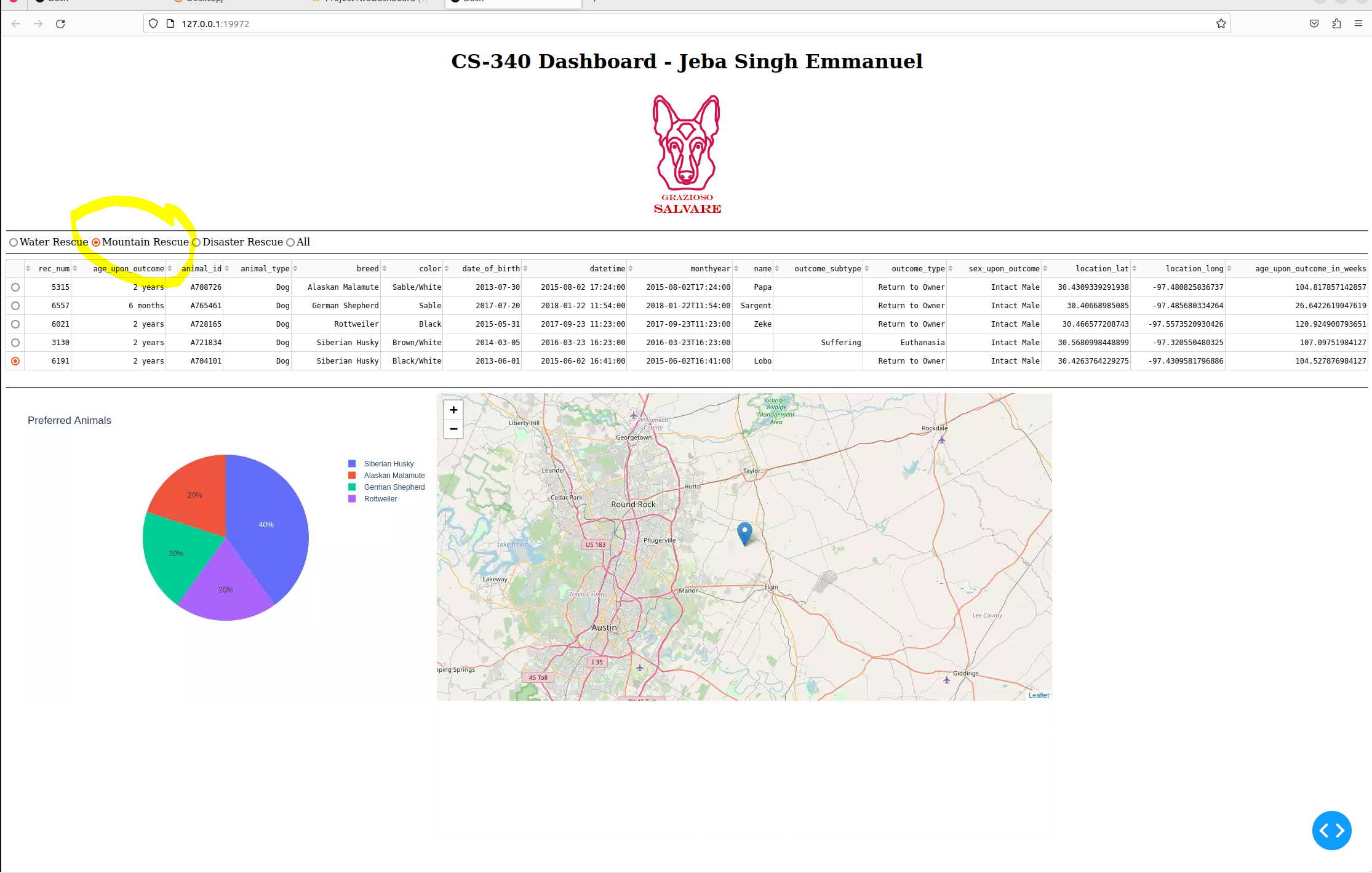
Use the filters at the top to identify animals that are useful for those scenarios. You will see a list of animals and breakdown of different breeds in the pie chart.

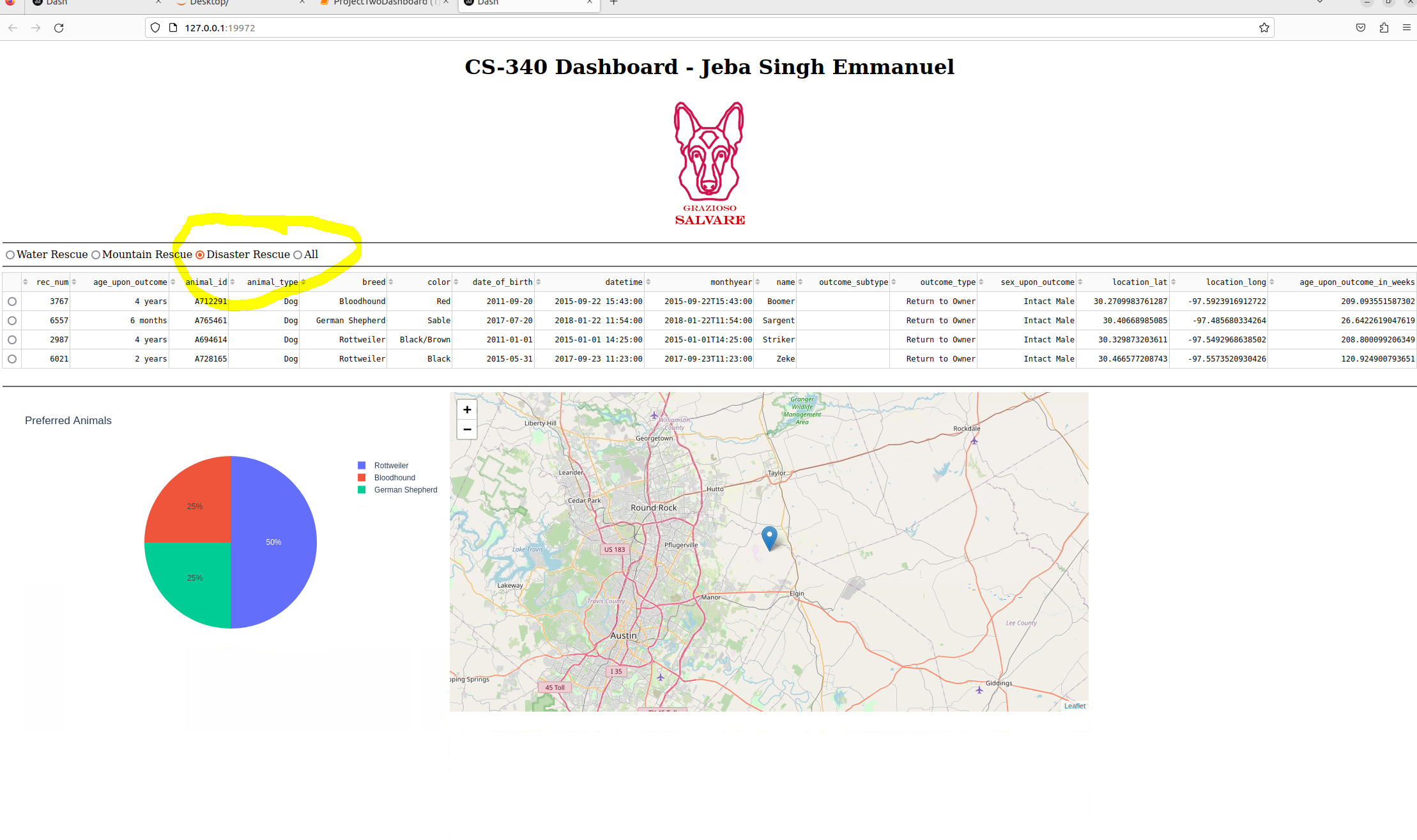
You can select individual rows using the radio buttons on the left and the location of the animal will show on the geo chart.

**Initial Load**

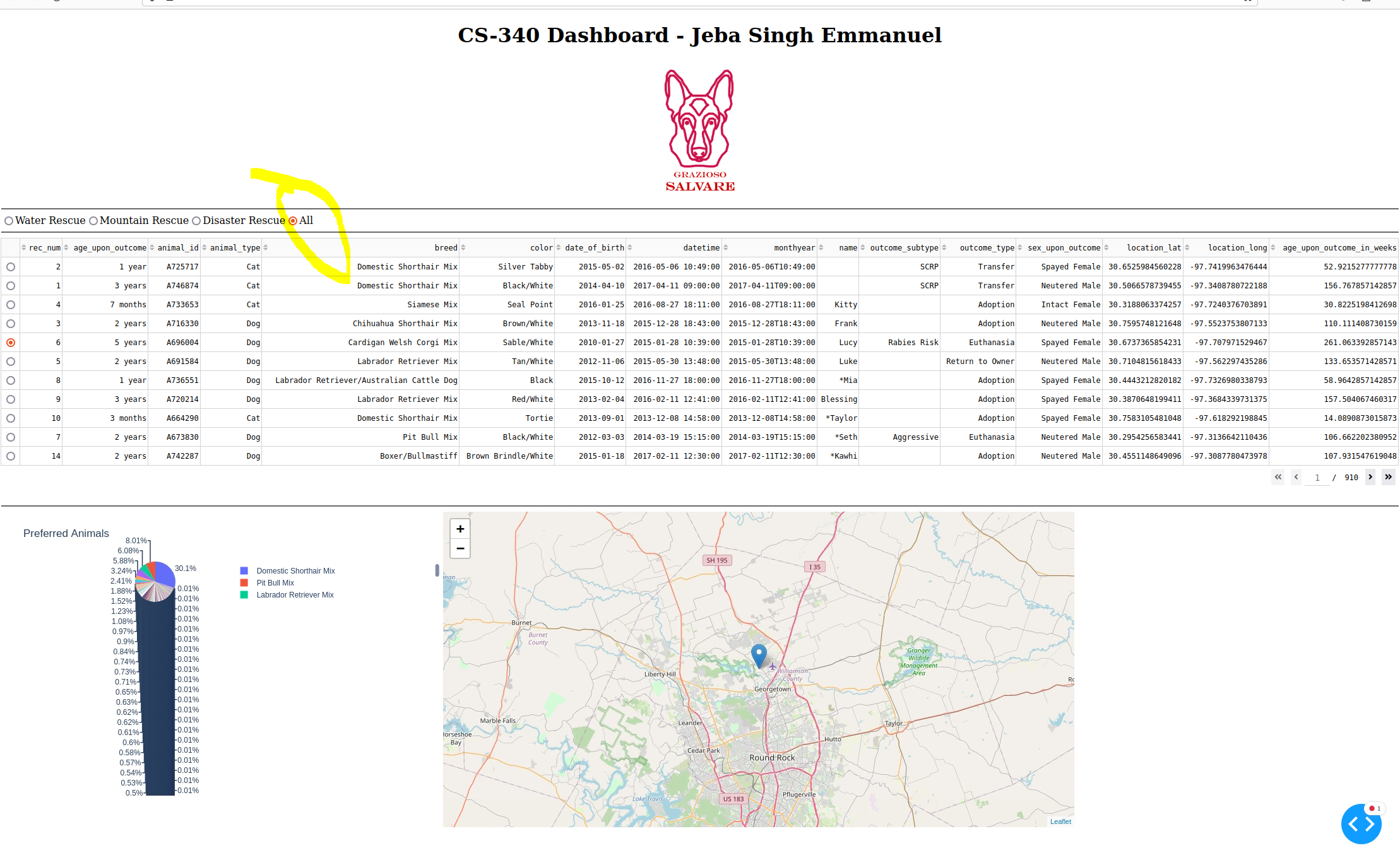
Water Rescue Selected

Mountain Rescue Selected

Disaster Rescue Selected



Reset back to default State



## Development Experience

* Dash is a pleasure to use.
* Dash’s documentation is amazing and so is MongoDBs. They are intuitive to use and work as expected.
* I finished each requirement one by one and since the earlier modules had done a lot of the prework, this project was about stitching them all together.

## Challenges

* Different versions of Dash. The sample had code that didn’t work in newer versions of dash. It took some development time to do the necessary research.
* The scaffolding for filter callback was incorrect. I tried to keep the scaffolding (return columns, data) as much as possible but after spending a week trying to debug it, I finally discarded it and rewrote the function myself.
* The database contains species other than Dogs and this is unnecessary storage cost. We can filter out animals other than dogs.

## Roadmap/Features (Optional)

* Handle incorrect authentication credentials
* Allow passing in connection details in constructor
* More sanitisation around data structure and values that can be edited once written.

## Contact

Your name: Jeba Singh Emmanuel

jeba.emmanuel@snhu.edu