# CS 340 README

## About the Project/Project Title

This project is being created for Grazioso Salvare, an innovative international rescue-animal training company. Global Rain, a software engineering company, is developing software to help Grazioso Salvare identify dogs that make a good candidate for their rescue training program using a database from the Austin Animal Center.

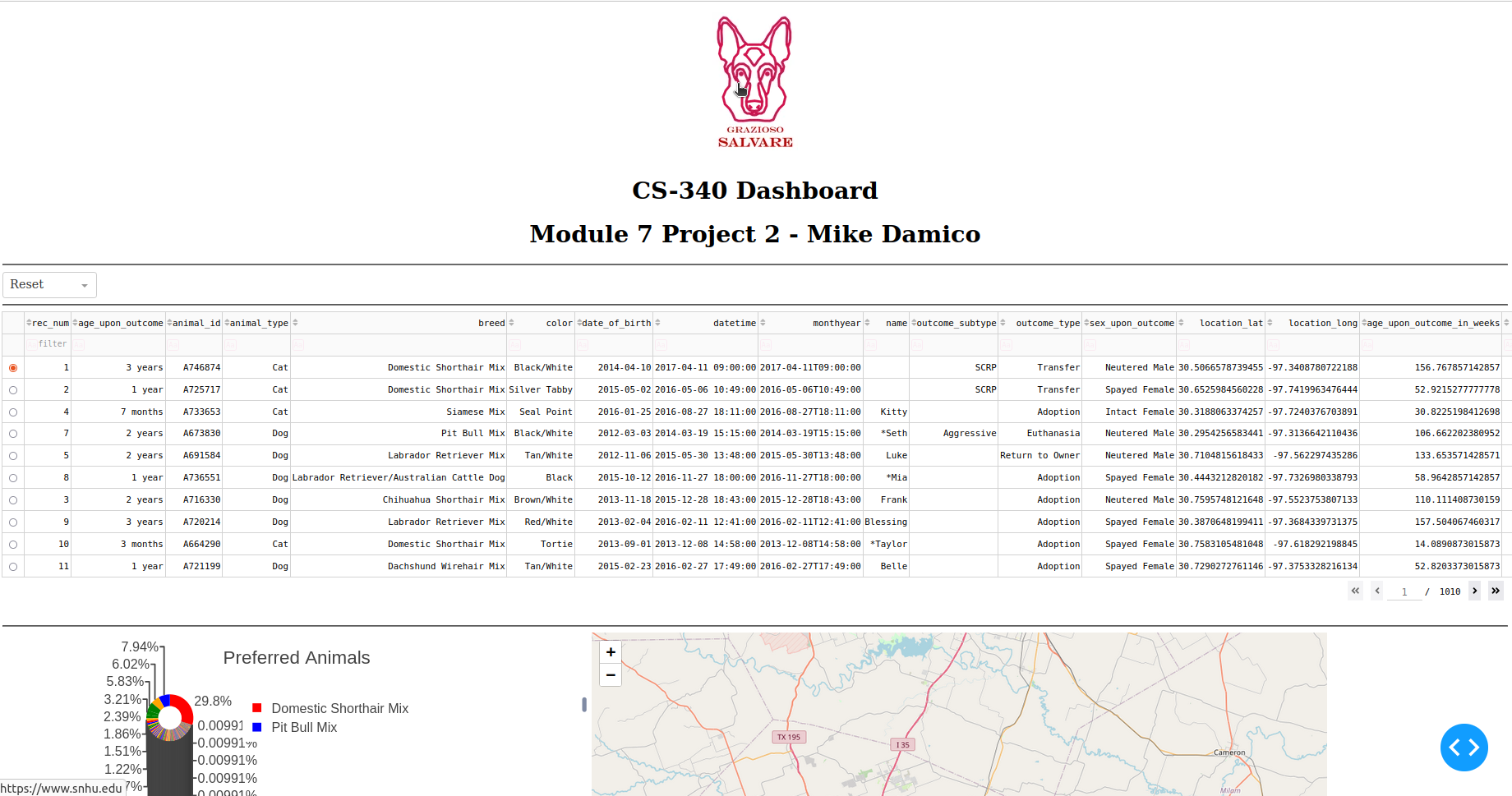
## Motivation

This project was created as part of a full stack application to connect a MongoDB database to a front end interface using a CRUD module. Grazioso Salvare will use this project to maintain a database that is used to identify dogs that would be good for their training program.

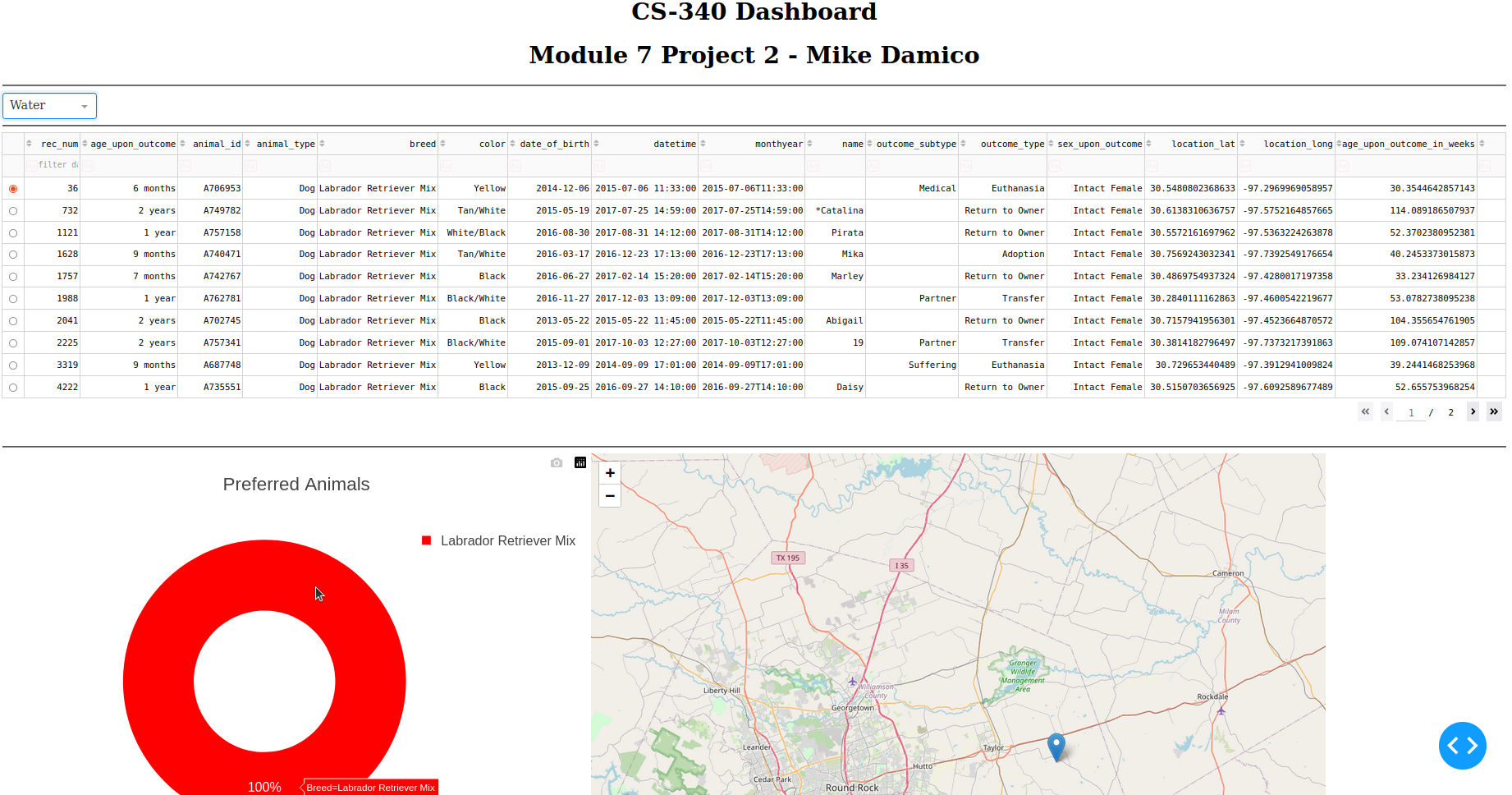
## **Required Functionality**

This project requires a front end for searching for dogs using four filters (Reset, Water, Mountain or wilderness, Disaster or Individual Tracking). These filters are chosen via a drop down menu above a main table that starts by displaying all the dogs in pages of ten dogs at a time. It must have the Grazioso Salvare logo (with a link to [www.snhu.edu](http://www.snhu.edu/))and two charts that display useful data for users to help find good training dogs. In this application the two charts are a map to show the locations of the dogs and a pie chart that shows how many of each breed there is to choose from.

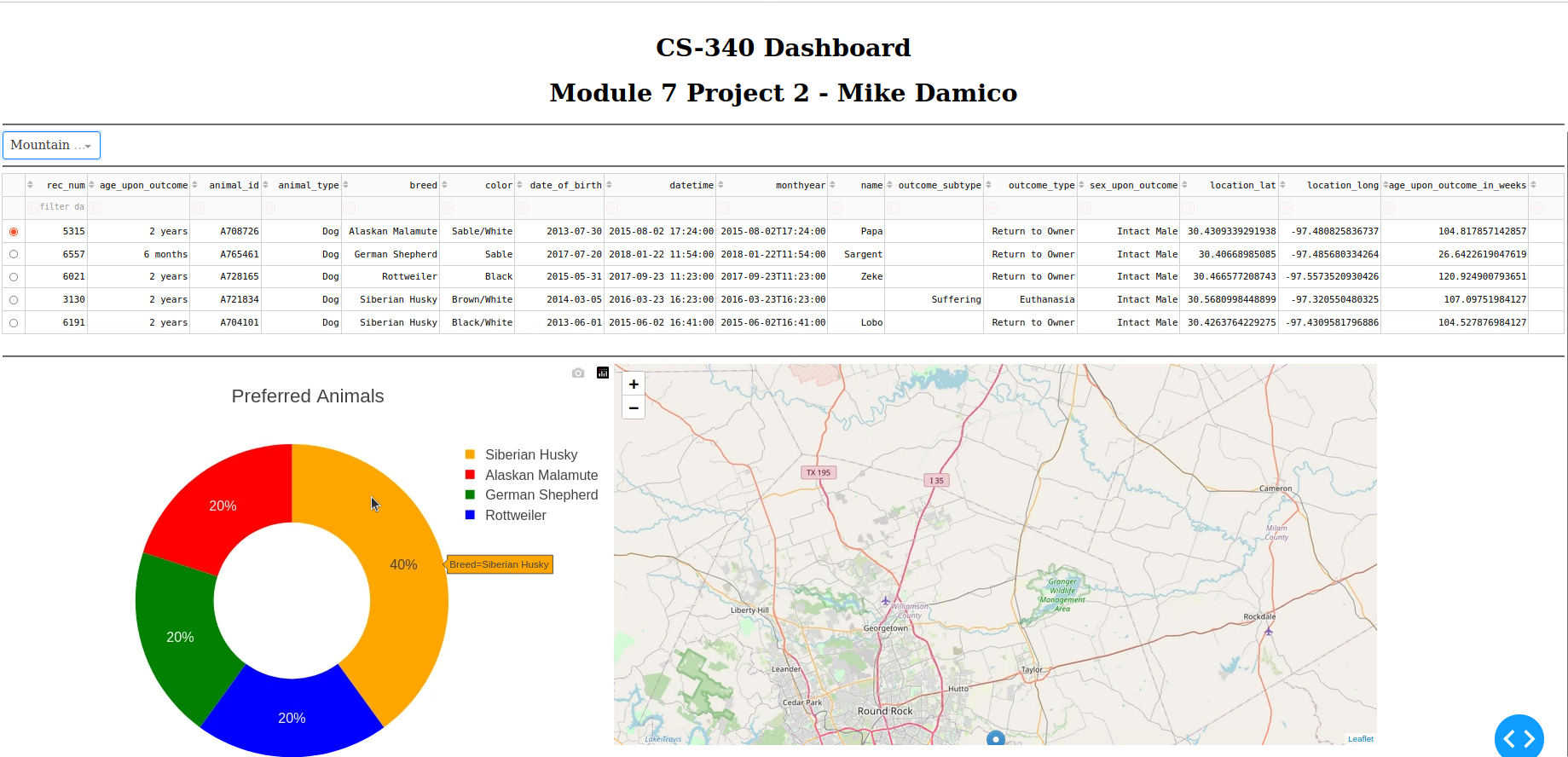
**Main/Start Screen**



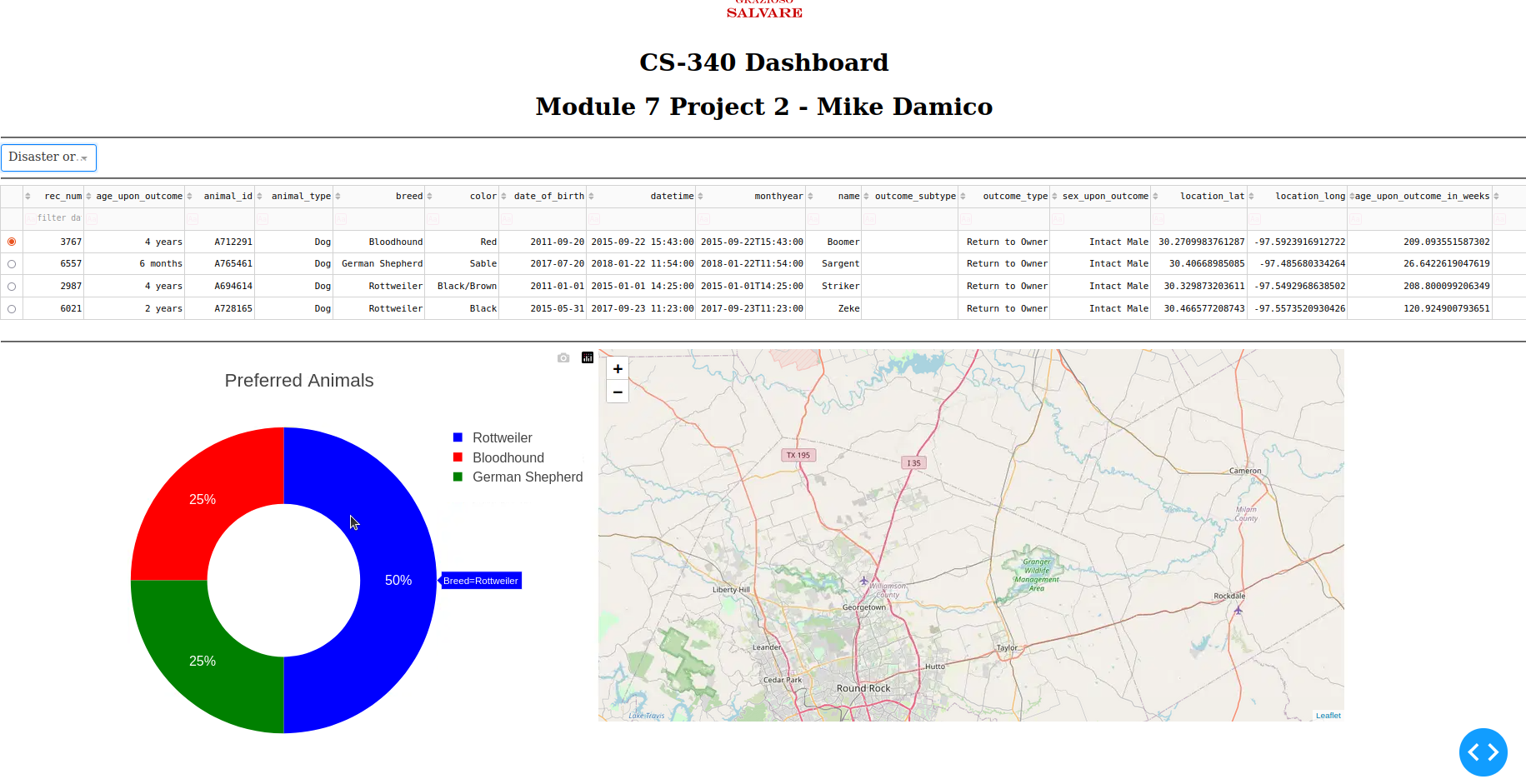
**Filter Option: Water**



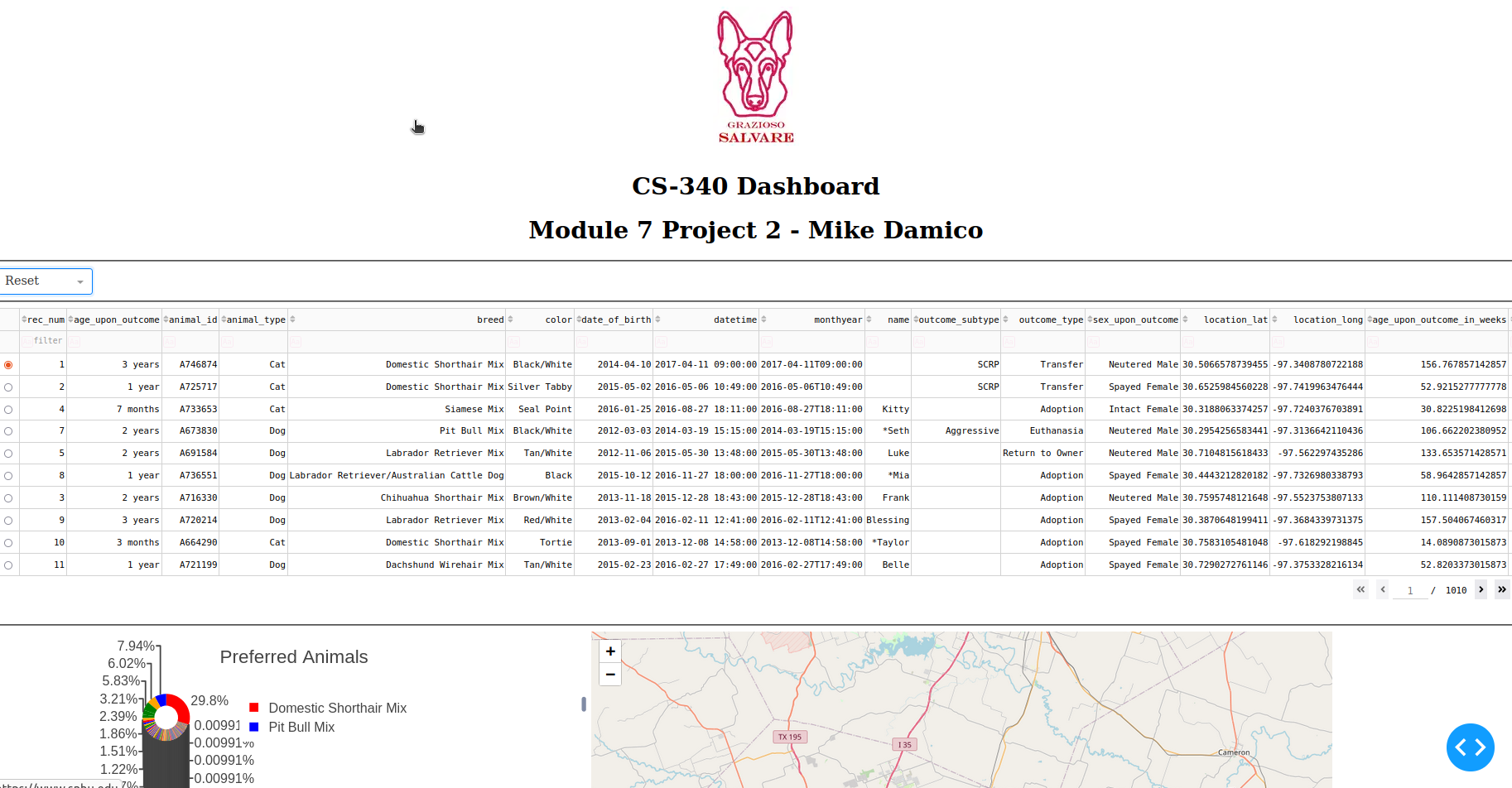
**Filter Option: Mountain or Wilderness**



## Filter Option: Disaster or Individual Tracking



**Filter Option: Reset**

****

## Tools Used

**MongoDB** – MongoDB is a NoSQL database management system that is easily used with Python via Pymongo. MongoDB is used in this application because it does not have strict schema constraints like SQL making it easier to use.

**Python** – The Python programming language has simple syntax and has libraries that makes it easy to rapidly develop an application using MongoDB and Dash framework.

**Dash Framework** – Provides tools to create the front end interface. This is great for adding in visuals such as charts to display data in meaningful ways to the users. Dash is great for quickly creating custom user interfaces.

**Challenges Encountered**

I did have many small challenges while learning about MongoDB and Dash. These challenges were all resolved by reading the errors and researching. My most time consuming issue was that my port number to access MongoDB was wrong and that prevented my CRUD from accessing the database.

## Getting Started

Overview

To setup the MongoDB, open your MongoDB shell and create the database. Use the MongoDB import tool to import a CSV file which will add your documents to the database. Then, setup your user with a username and password. The code requires these credentials as well as the connection string and port.

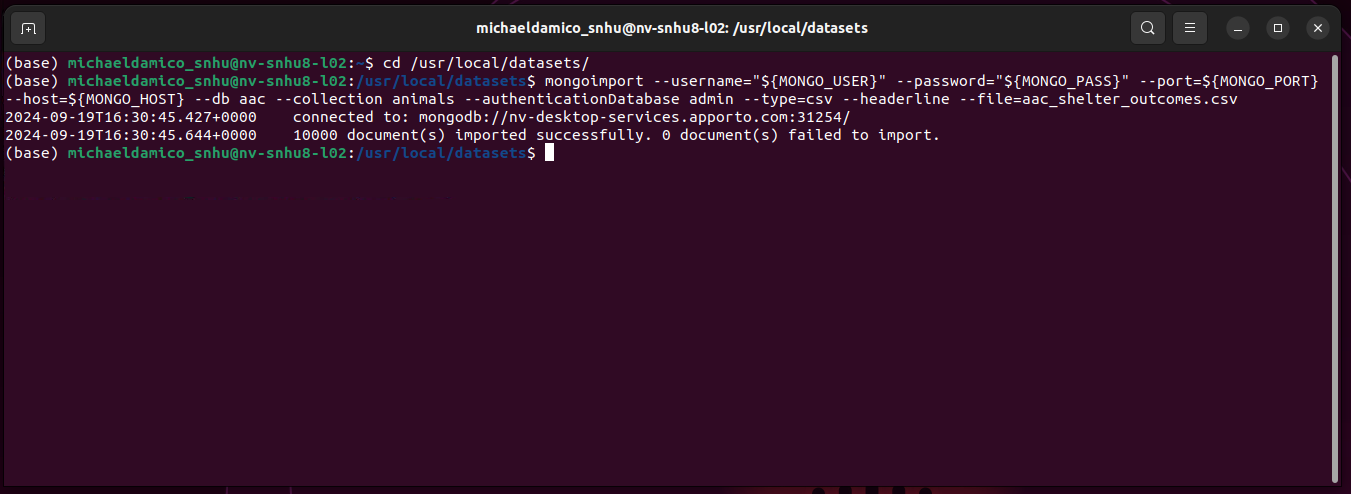
**Step-by-Step**

Import dataset

1. Enter mongo shell using “mongosh” in the terminal

2. Use “cd” to change directory to the location of the dataset

3. Use the mongoDB import tool to import the dataset into mongodb



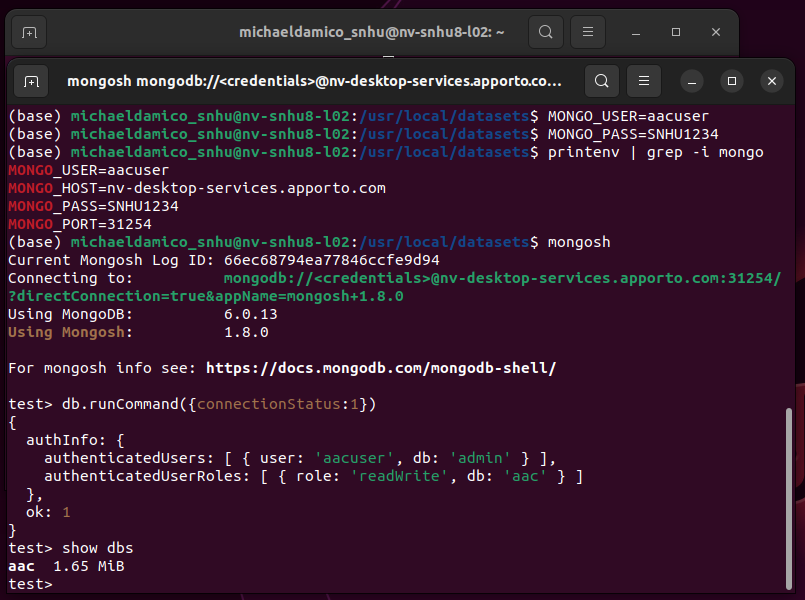
Create User

1. Enter mongo shell using “mongosh” in the terminal

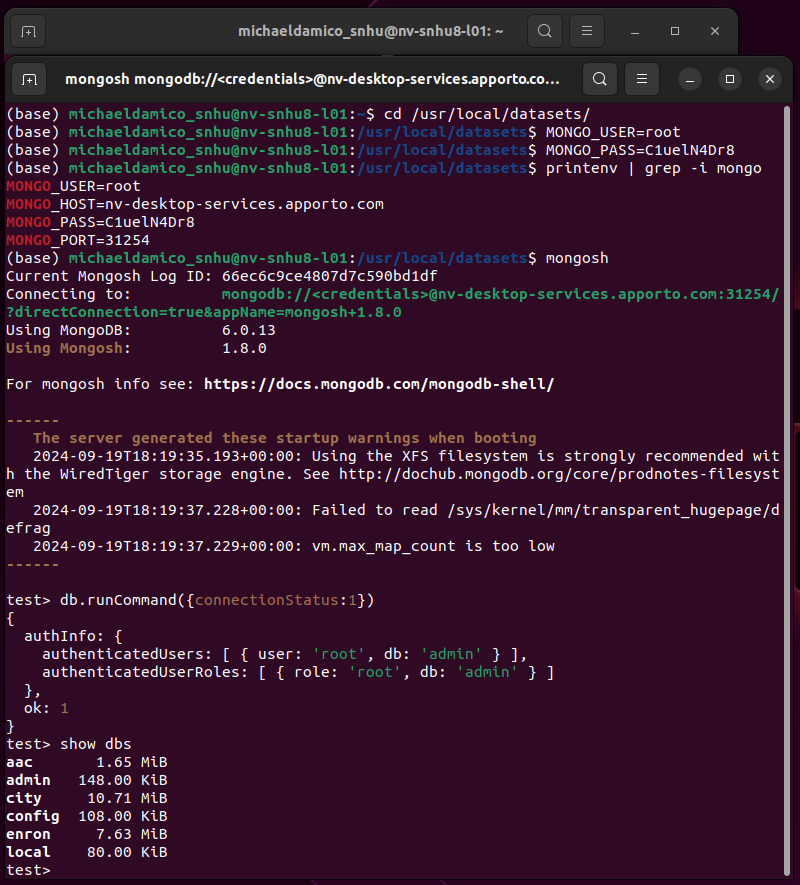
2. Enter “use admin”

3. Enter “db.createUser({user: “aacuser”, pwd: “SNHU1234”, roles: [ { role: “readWrite”, db: “aac” } ]})

Login process for aacuser accounts.



Login process for Admin

****

## Installation

*MongoDB – NoSQL database*

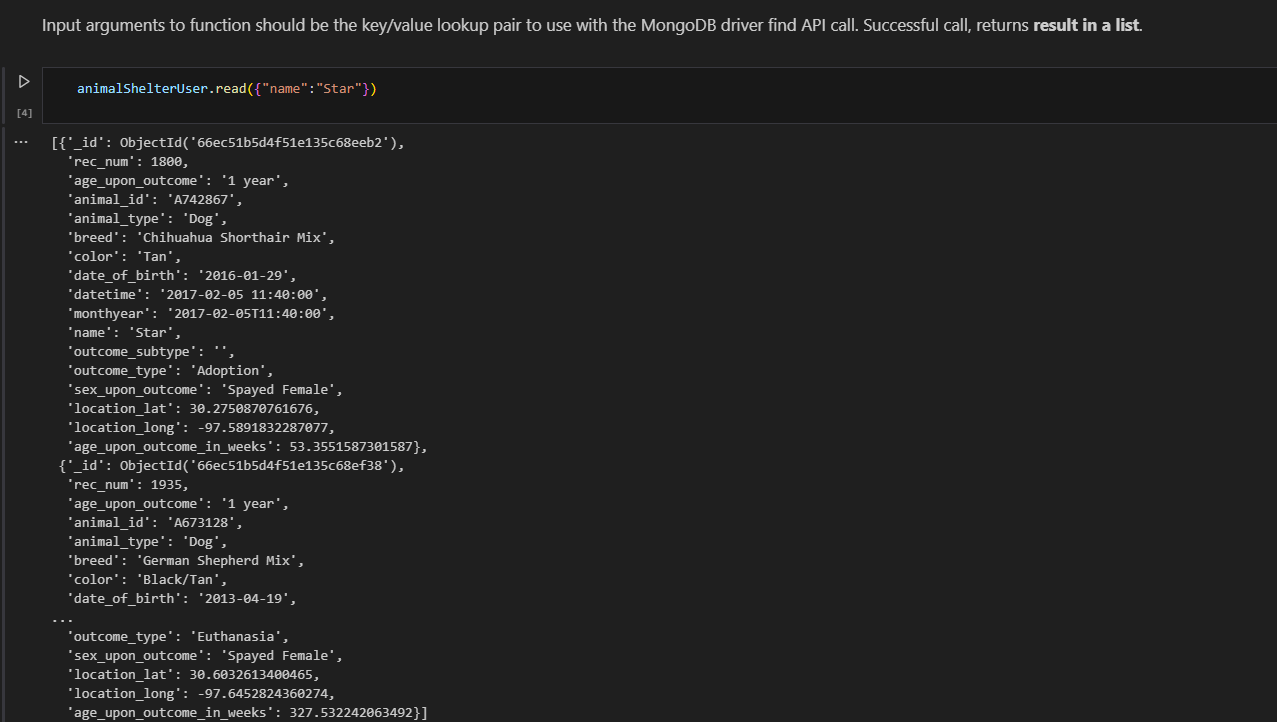
*Jupyter Notebook- IDE*

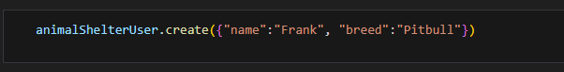
*pymongo - Official driver to connect to MongoDB using Python.*

*Bson.objectid – used to handle bson documents in MongoDB*

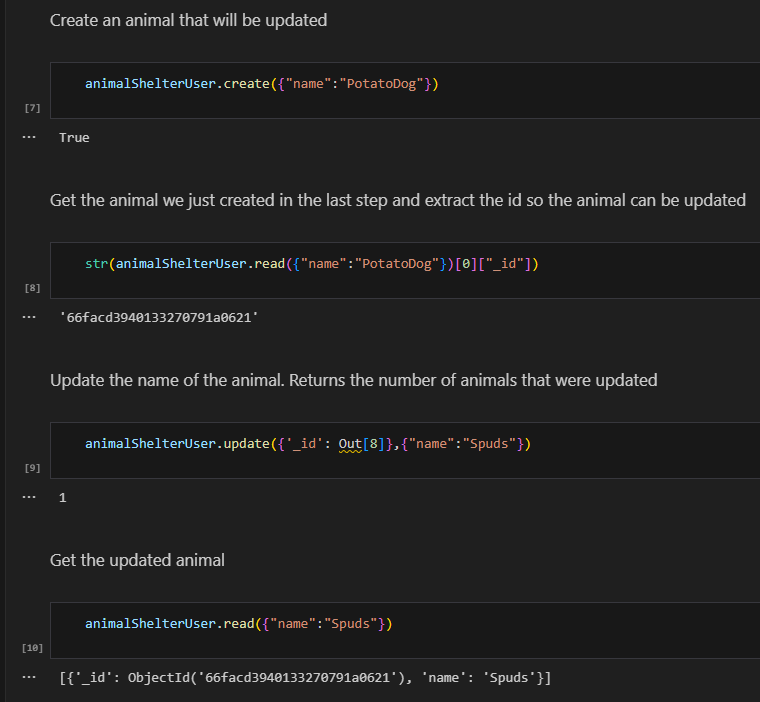
## Usage

### Code Example This project allows you to search the database and insert new resources into the database. The Read function takes a search term and returns any case-sensitive name matches from the database. The following example searches for any animals named Star.

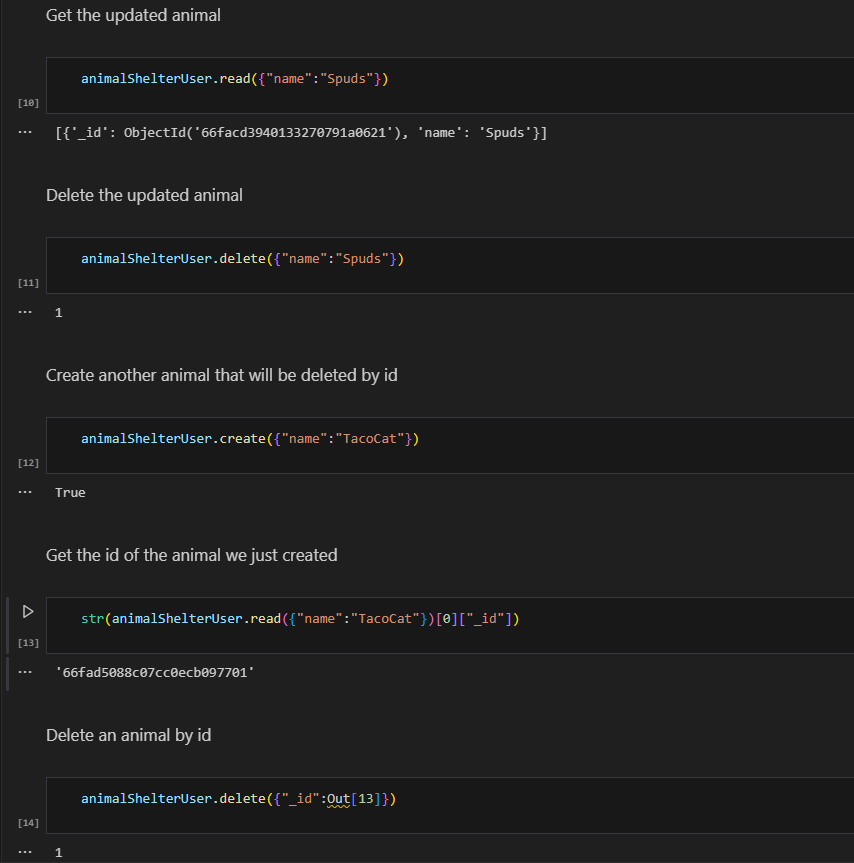
  
The create request returns True if the creation was successful and False otherwise. The following example creates an animal with the name Frank and a breed of Pitbull.



### The update request returns a number equal to the number of documents updated. Note: documents can be found by \_id.

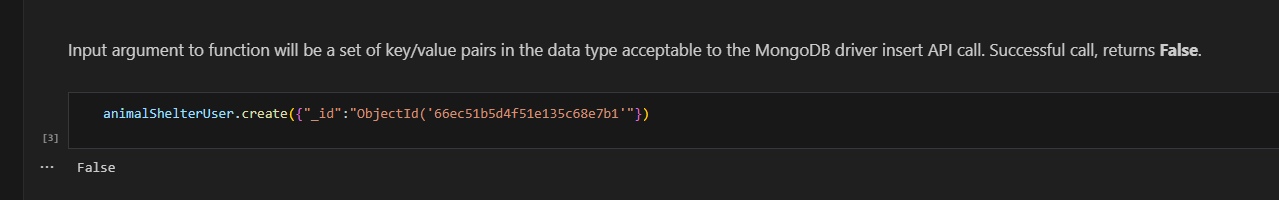


### The delete requests returns a number equal to the number of documents it deletes. Note: Documents can be found by \_id.

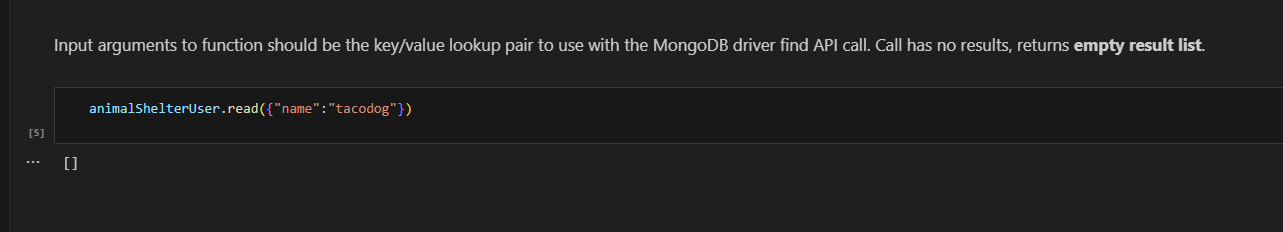


### Tests

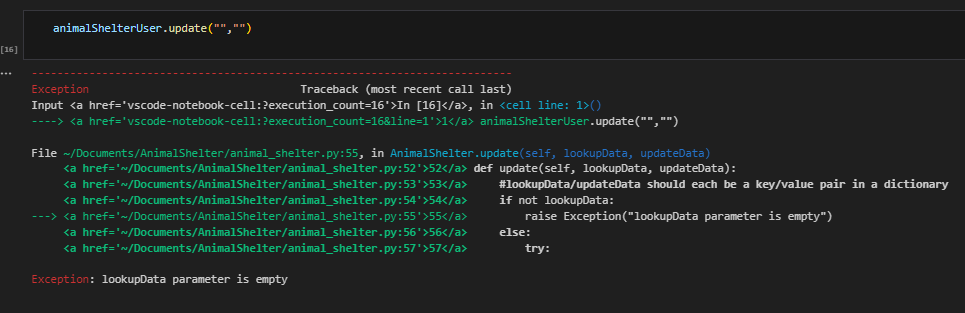
*In the last section working examples were shown. In the following example the create method is called using a id that is already taken in the database. This returns False and does not get entered into the database.*

**

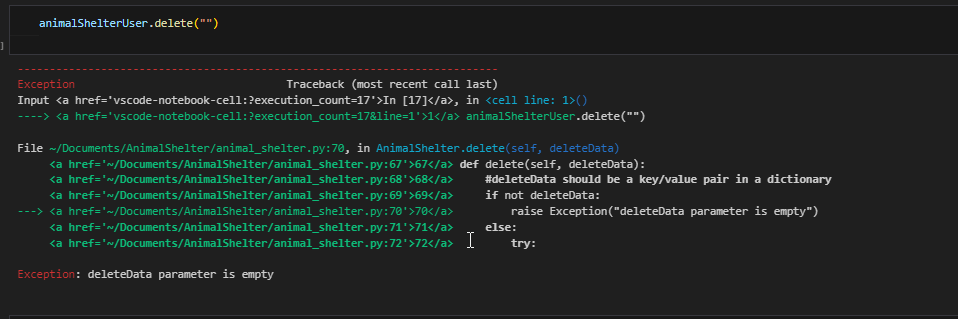
*In the following example, the read method is used and returns an empty list because there is no animal named tacodog.*

**

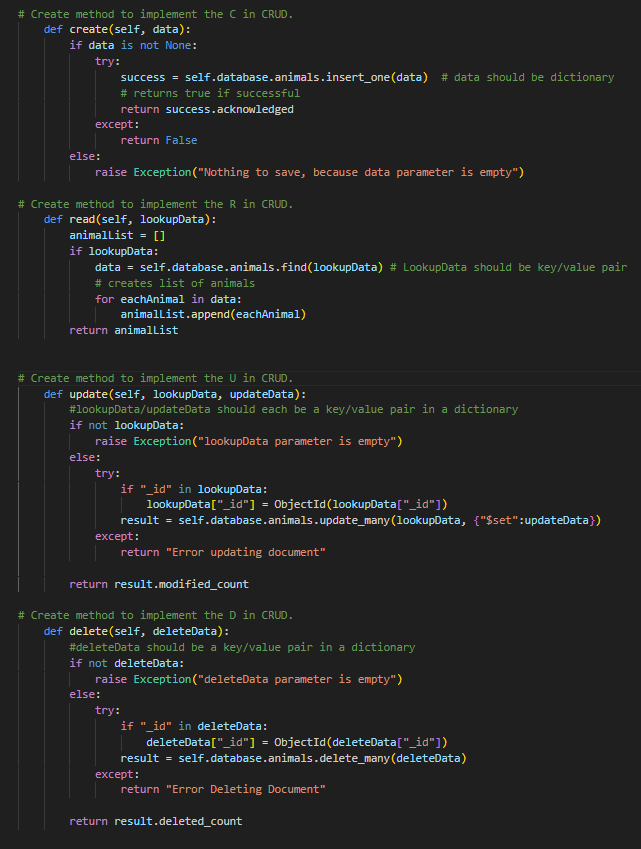
### Example of a failure to update



Example of a failure to Delete



### Screenshots

**

## Contact

Michael Damico

1(800) DONT-CALL-ME