**Project One README**

**About the Project/Project Title**

This project aims to develop the create, read, update, and delete methods for an application made for Grazioso Salvare. This Python module implements these methods for the AAC database. This database consists of animals from five different shelters that Grazioso Salvare would like to screen for possible rescue animal training.

**Motivation**

Grazioso Salvare needs to be able to sort through and search its newly acquired database efficiently and effectively. They need to search for specific animals with specific attributes to do specific jobs, as well as maintain the database.

**Getting Started**

To get a local copy running follow these steps:  
1. Install MongoDB

2. Install PyMongo

2. Import the AAC database

3. Create a user in the admin database that has read and write access to the AAC database

4. Download and insert the two Python scripts into Jupyter Notebook

The database AAC consists of many documents describing animals from a large shelter. They detail their breed, name, sex, date of birth, and many other descriptors. The create function is rather simple, as long as the data you prove is not null, it inserts it using the insert() function. It also returns true or false depending on insertion success, but that is mainly for testing purposes. The read function is very similar except it uses find() instead of insert(), and it uses a loop to print any and all documents matching the given data.

**Installation**

List of tools and libraries you need

* MongoDB
* Python
* Jupyter Notebook
* Pymongo
* Byson.objectid

1. Import the AAC database

2. Create a user in the admin database that has read and write access to the AAC database

3. Download and insert the two Python scripts into Jupyter Notebook

4. Import AnimalShelter from the pythonCrud file and you are ready to go!

**Usage**

This project uses PyMongo and MongoDB. They were chosen for their ability to query, retrieve results, and support CRUD functionality. Furthermore, Pymongo recognizes the MongoDB query language, and Python and is installed quickly and easily. Using both makes accessing databases, collections, and documents fast and easy.

Dash is used for the web app solution for the project. Dash works with Pymongo and MongoDB to produce a web application platform for the data in our AAC database. Using Dash gives us access to tables, maps, charts, and much more to use to interpret, interface, and organize our data. The resources that come with Dash are invaluable for interpreting and allowing our clients to effectively and easily utilize the data in their database.

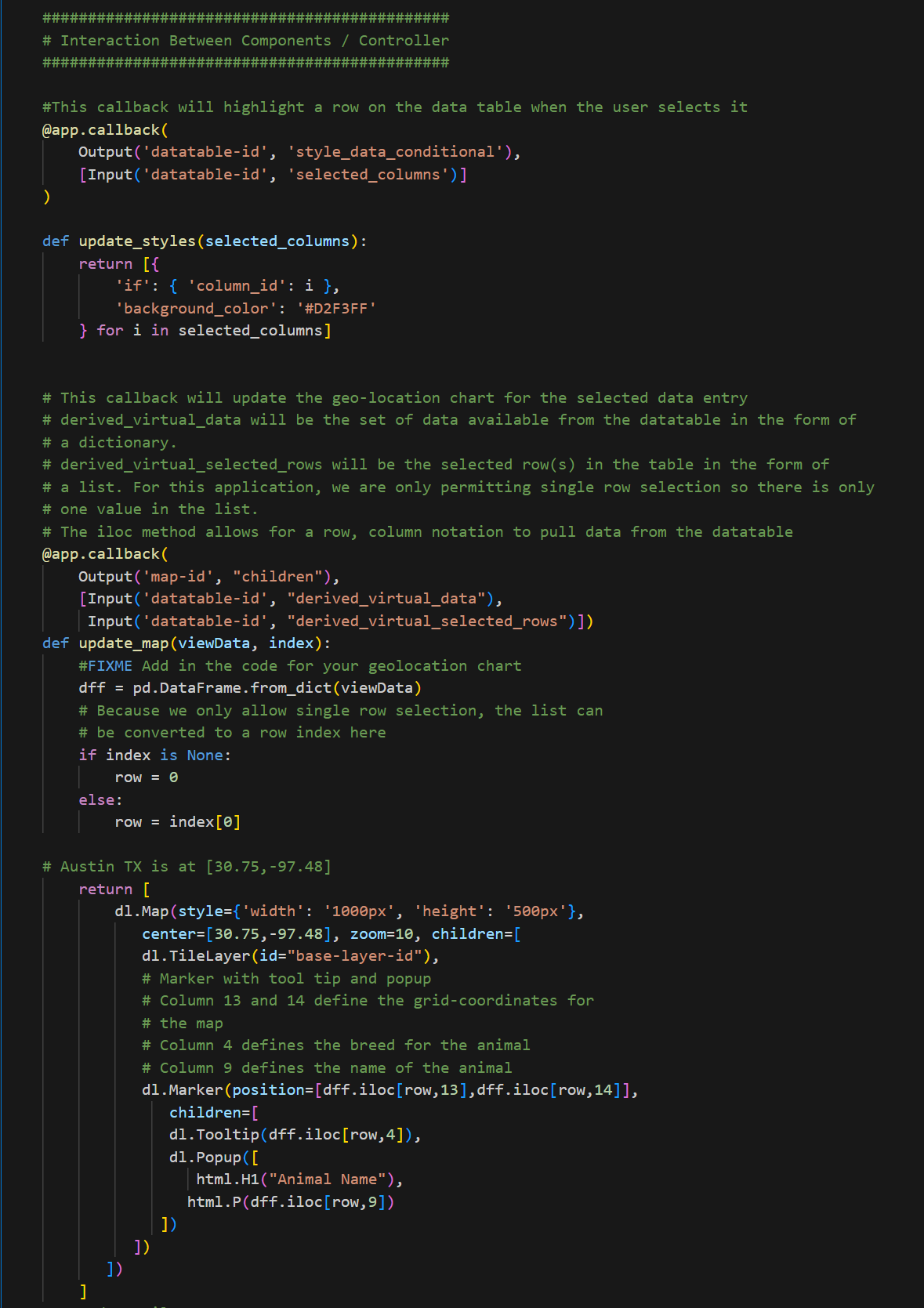
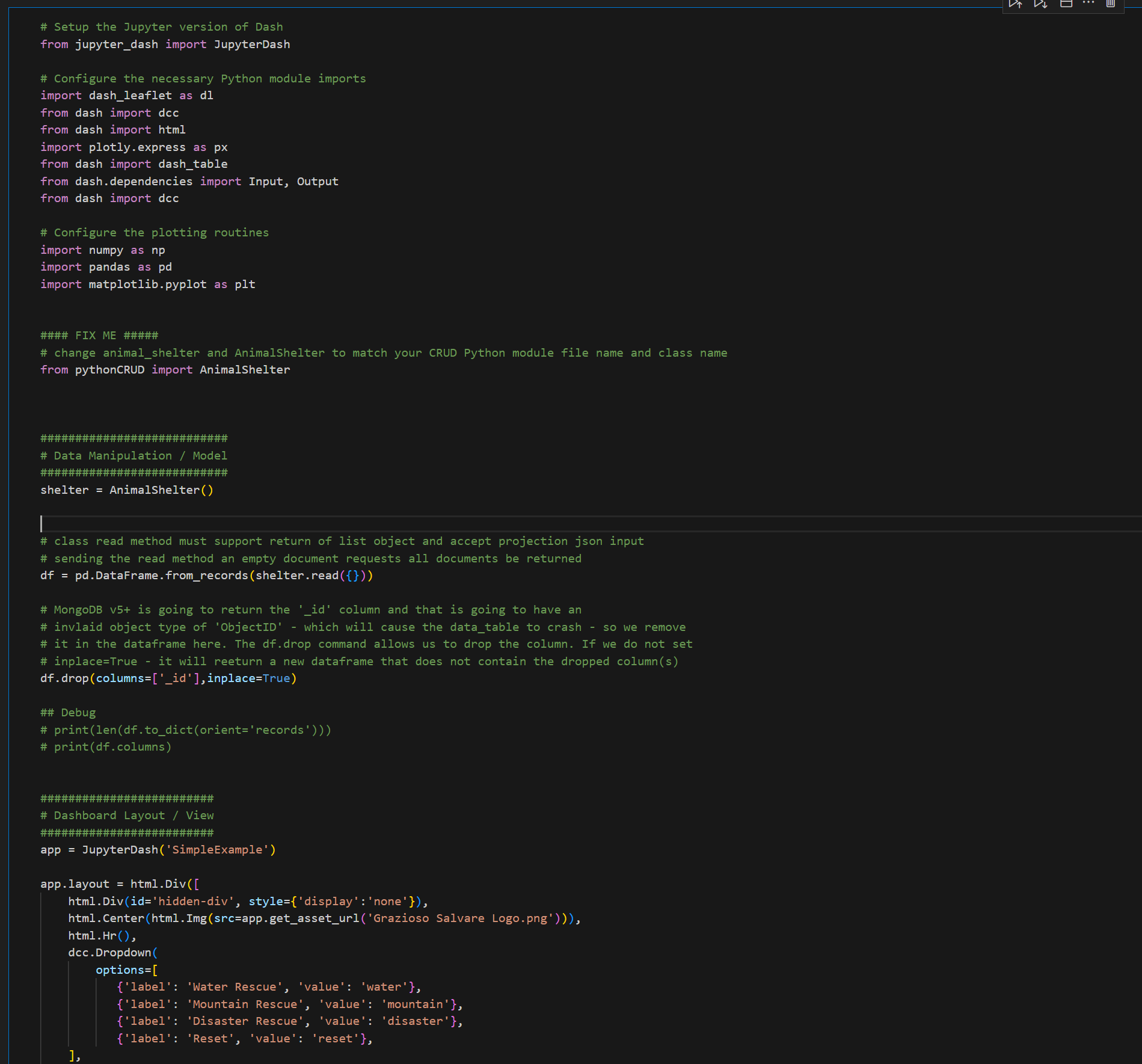
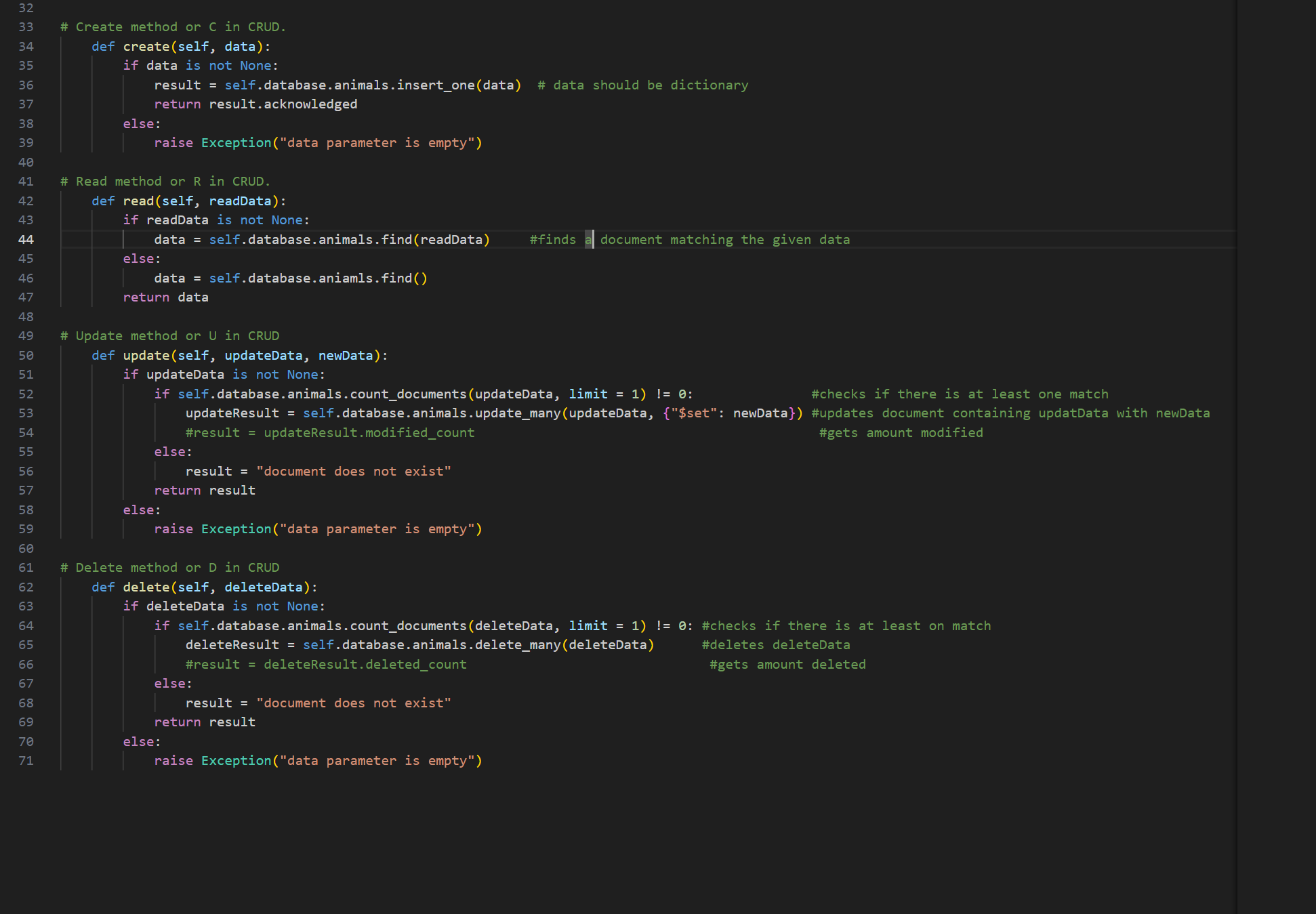
In the backend, most functionality comes down to four methods, Create, Read, Update, and Delete (CRUD). They reside in the pythonCRUD.py file and function as follows:

Create lets you insert a new animal into the database, and read lets you find animals with specific criteria and find them in the console. Both have code included to test the success of the function. The create function does this by returning true or false, and the read function does this by printing the animal(s) found.

Read lets you search the database for a document(s) that have the key/value pair(s) given upon calling the method. It returns a list of cursor objects of the matching results that later get turned into a list of dictionaries. You can put in as many or as few criteria in the method call as you want, as leaving it blank returns all documents.

The update method allows you to update document(s). It uses the first key/value pair given to search for matching document(s) and then uses the second key/value pair to update the document. It also prints the raw results for testing purposes.

The delete method simply lets you delete document(s). It uses a key/value pair to find matching document(s) and delete them. This method also prints the raw results for testing purposes.

**Code Example**

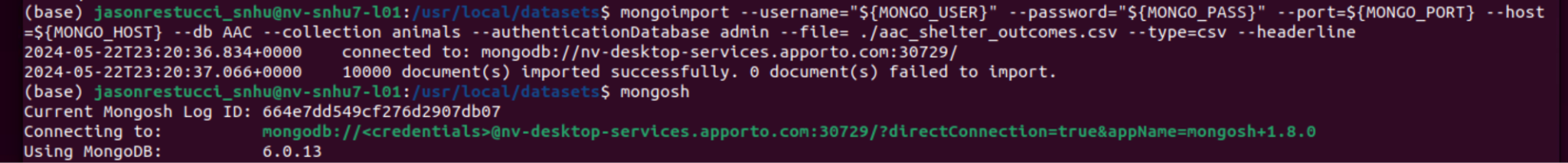
**Tests**

****

To test this you can create a dictionary with the fields, and use that data you have created to test the create and read function like you see above. The create function will return true if the insertion was successful and the read function will print the document if it finds one with the data you provide when calling the function. The update and delete methods simply print the amount of modified/deleted documents.

### **Screenshots**

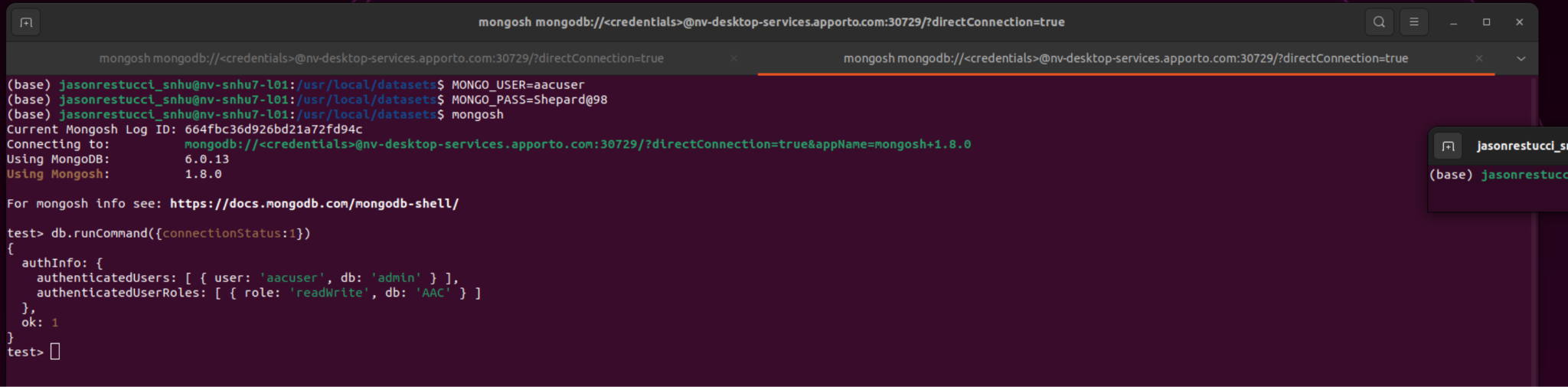
MongoDB Import Execution:



User Authentication Execution:

(Note: password was changed)

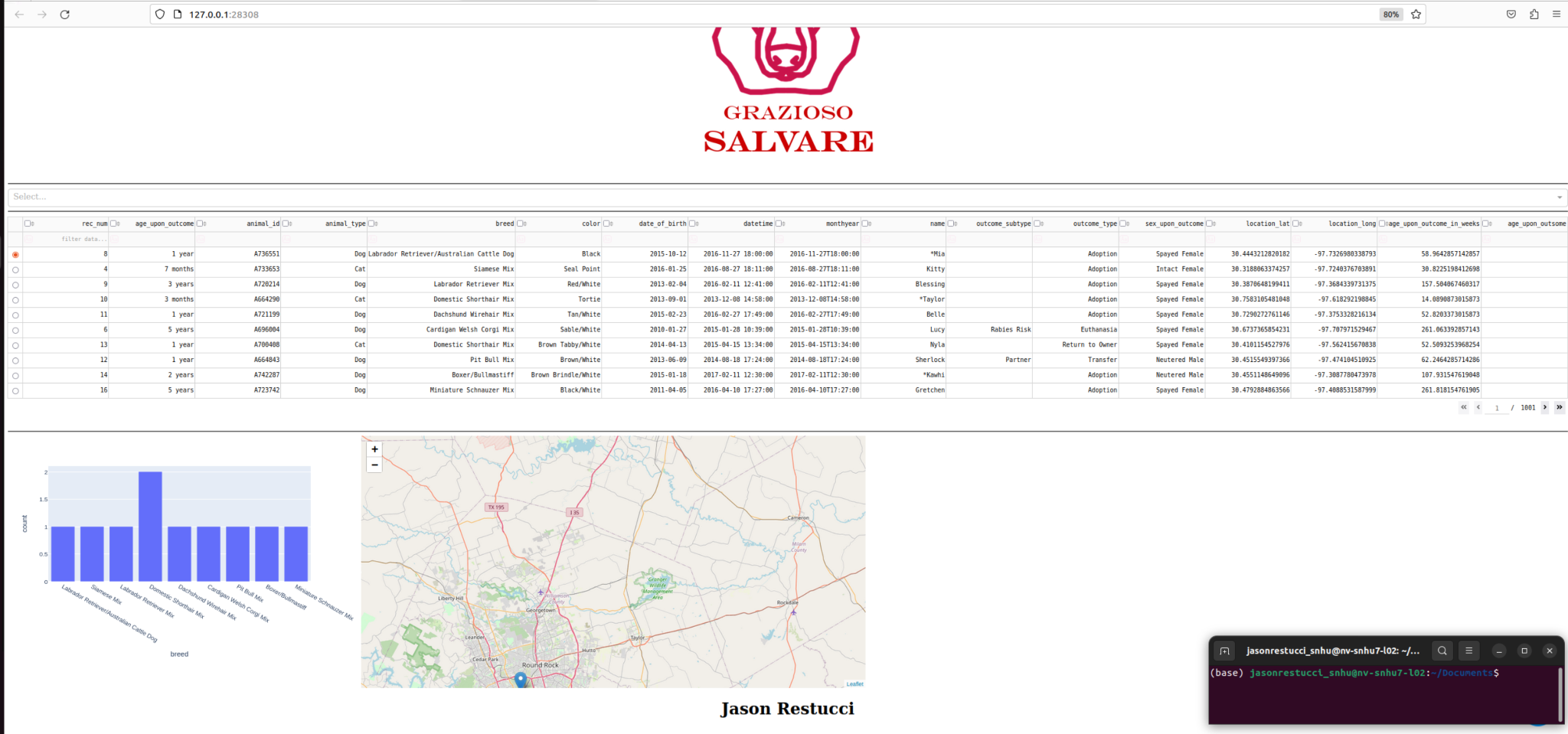




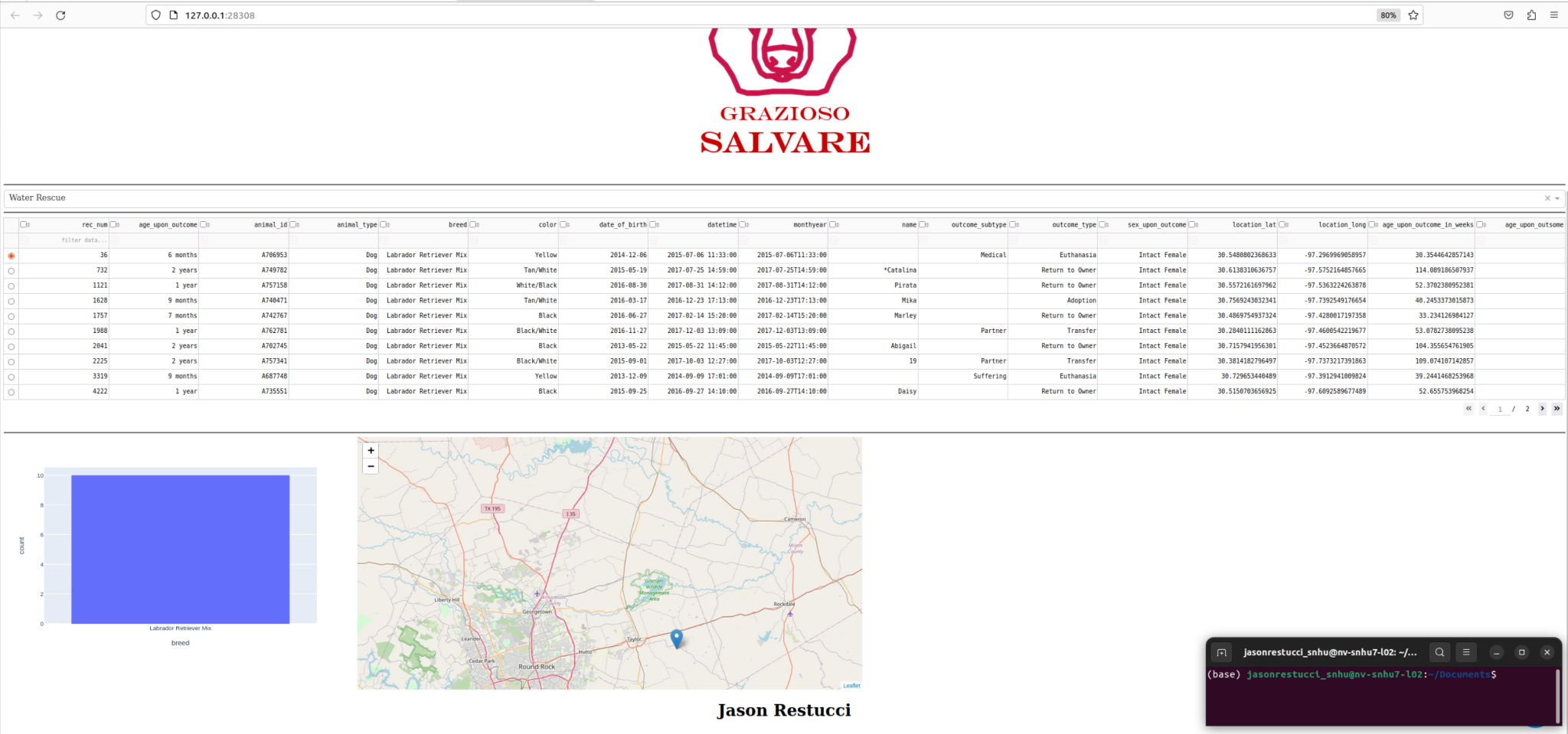
CRUD Functionality Test Execution:



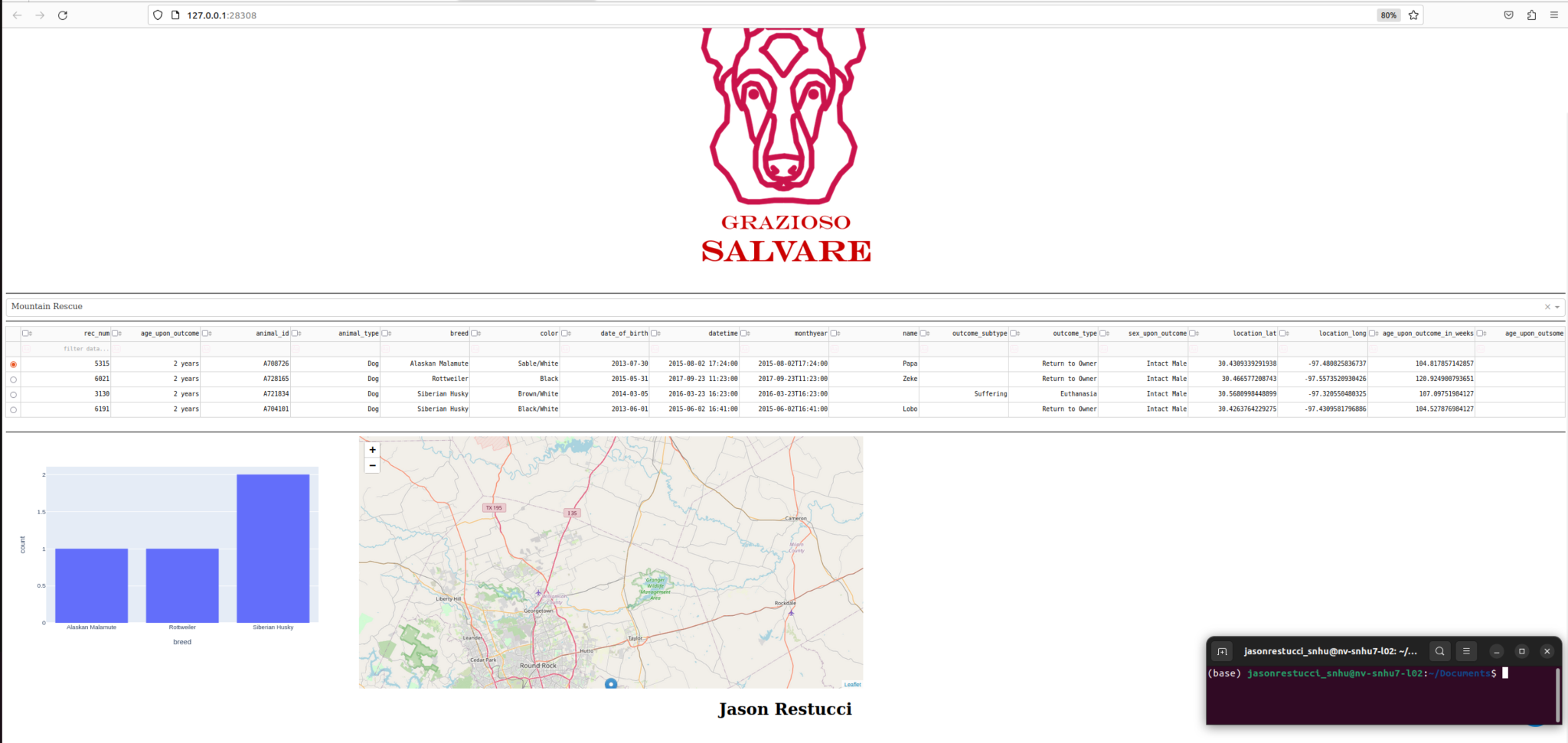
Default State of the dashboard:



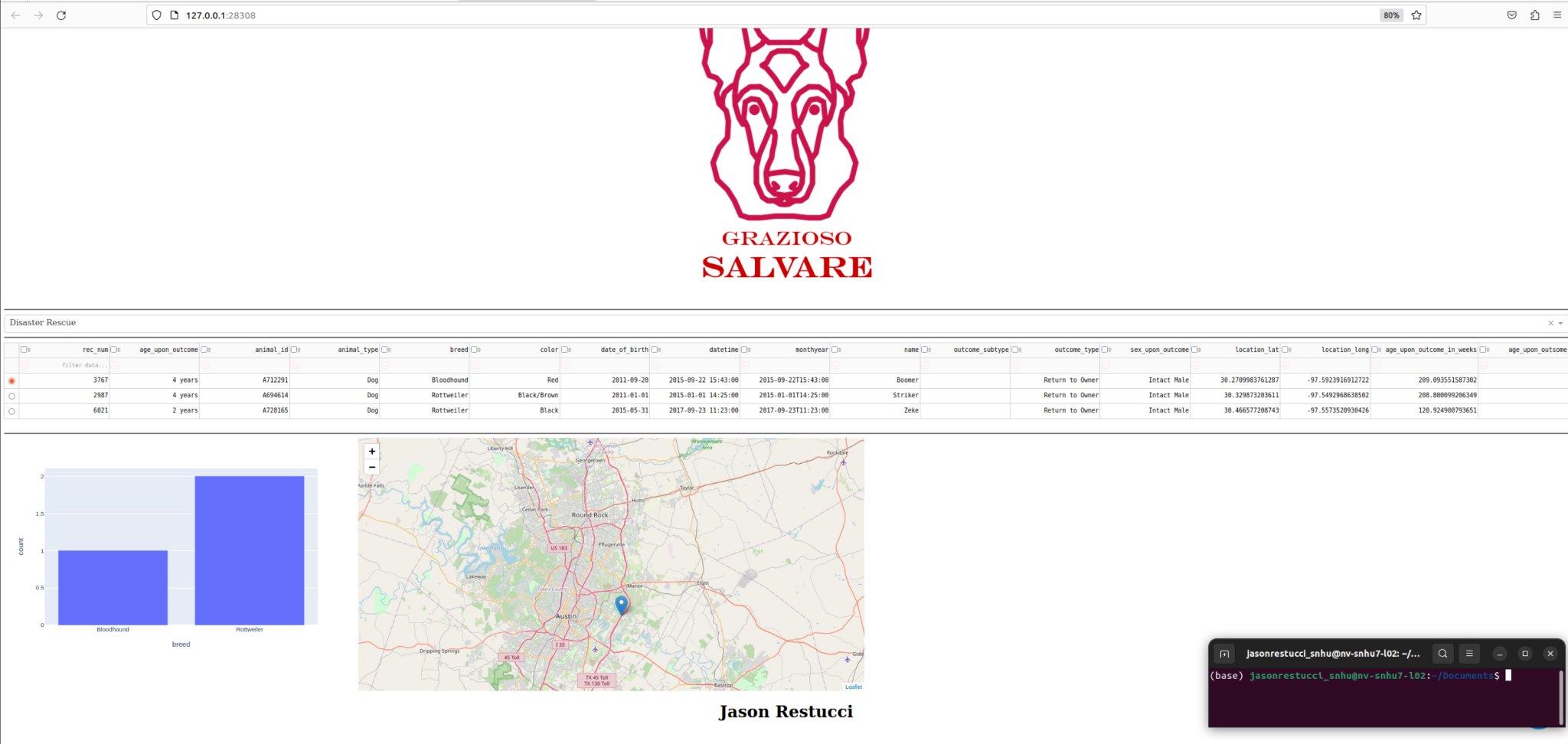
Water Rescue Filter:



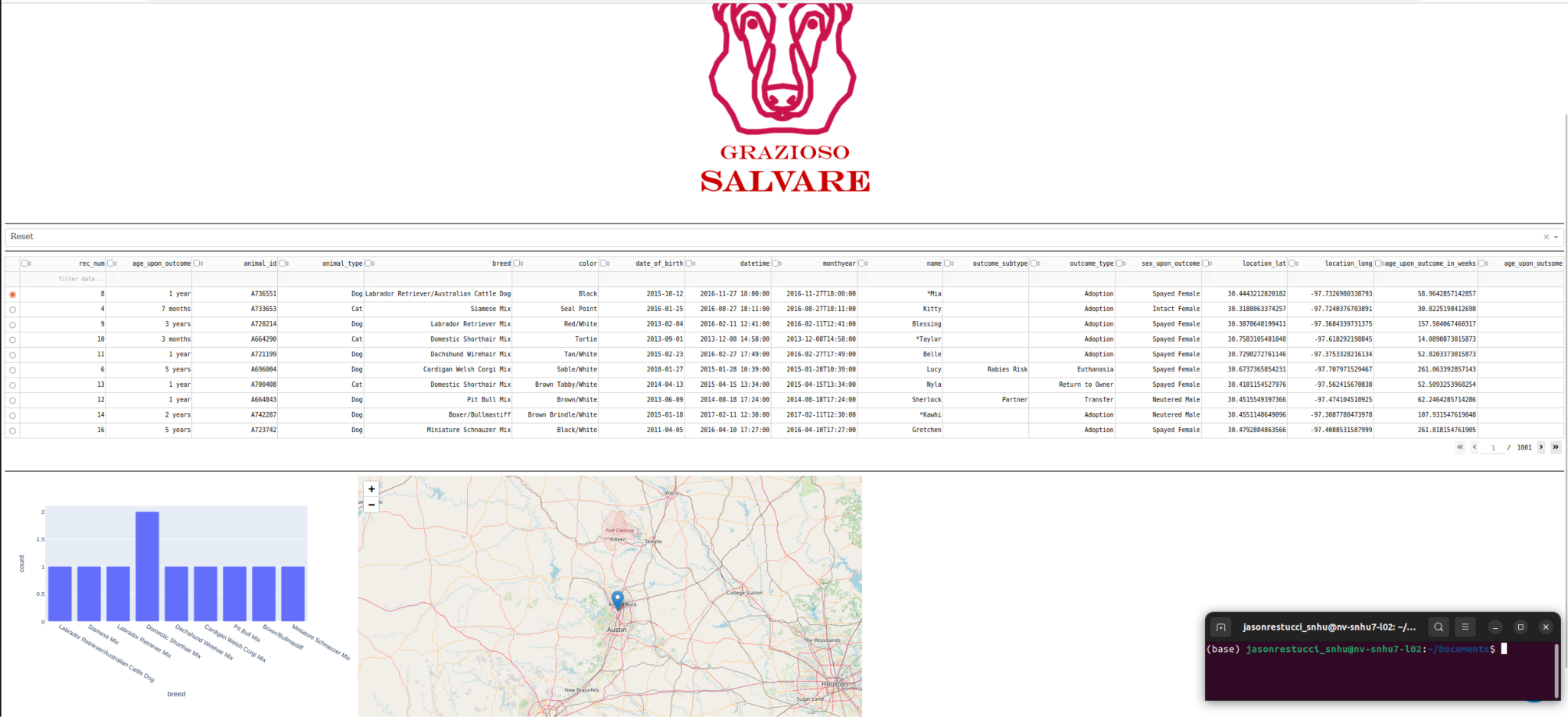
Mountain Rescue Filter:



Disaster Rescue Filter:



Reset or Clear Filter:



**Contact**

Jason Restucci

jason.restucci@snhu.edu