# CS 340 Project Two README

## About the Project/Project Title

This project is a Python notebook that generates a dashboard for interacting with a dataset.

## Motivation

The development of this module was to assist Grazioso Salvare in finding suitable candidates for search and rescue dog training from a large data set.

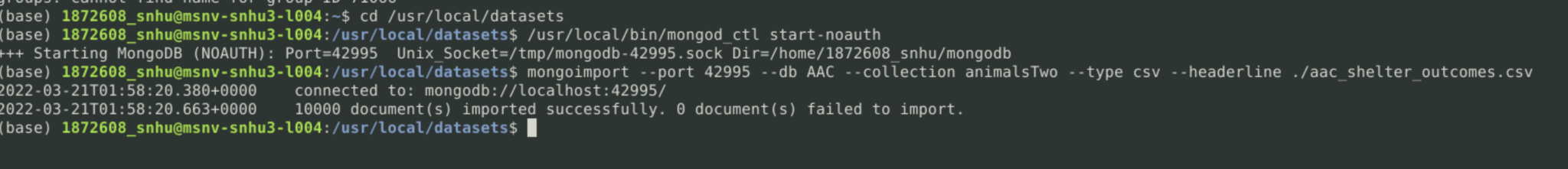
## Getting Started

In order to set up use of the notebook the code will need to be revised with the port number used for connection to MongoDB as well as the username, password, and database name.

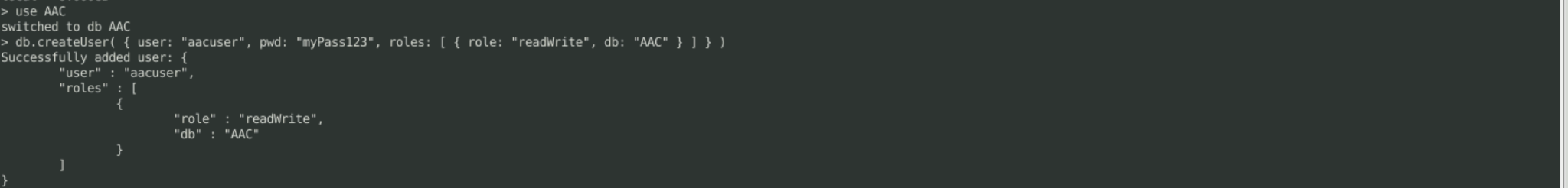
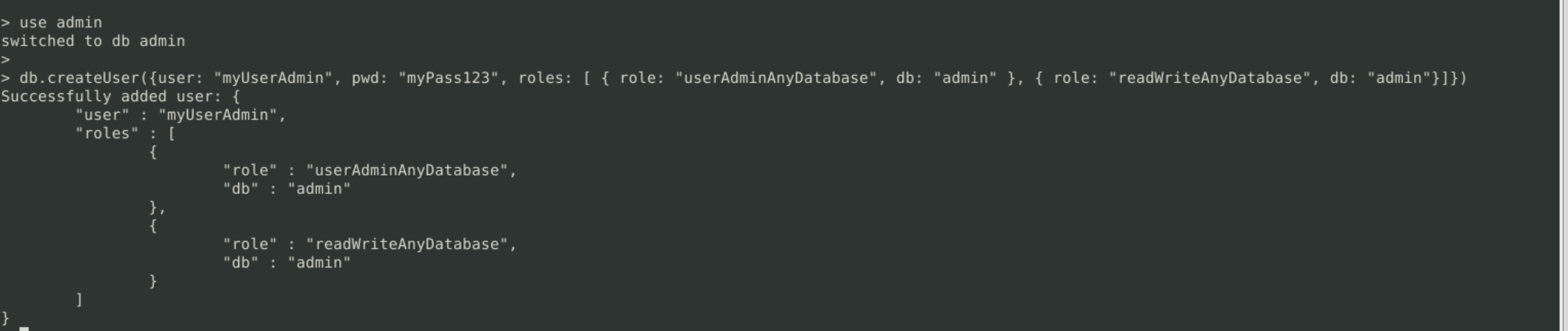
## Installation

Use of the notebook requires installation of Python3 (available at <https://www.python.org/downloads/>) and installation of MongoDB (available at <https://www.mongodb.com/docs/v5.0/installation/>). In addition to the python notebook you will also need to download the AnimalShelter.py CRUD module.

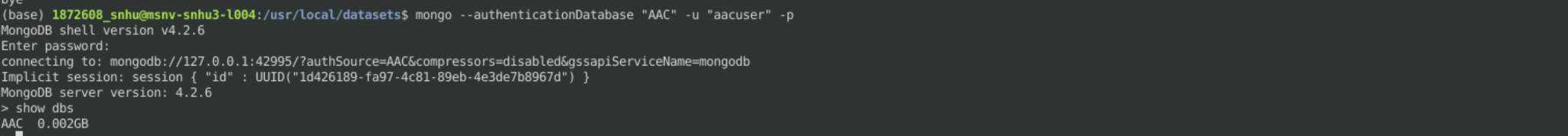
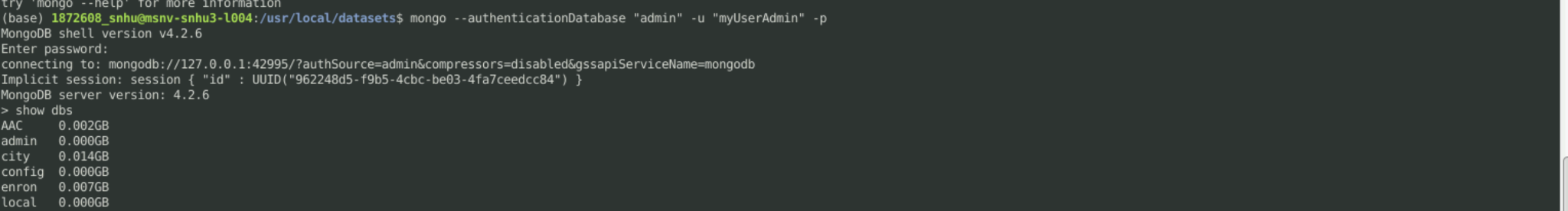
The documents will need to be imported from a csv file into a Mongo collection:



Once the documents have been imported admin and user accounts will need to be created within MongoDB in order to implement authorization:

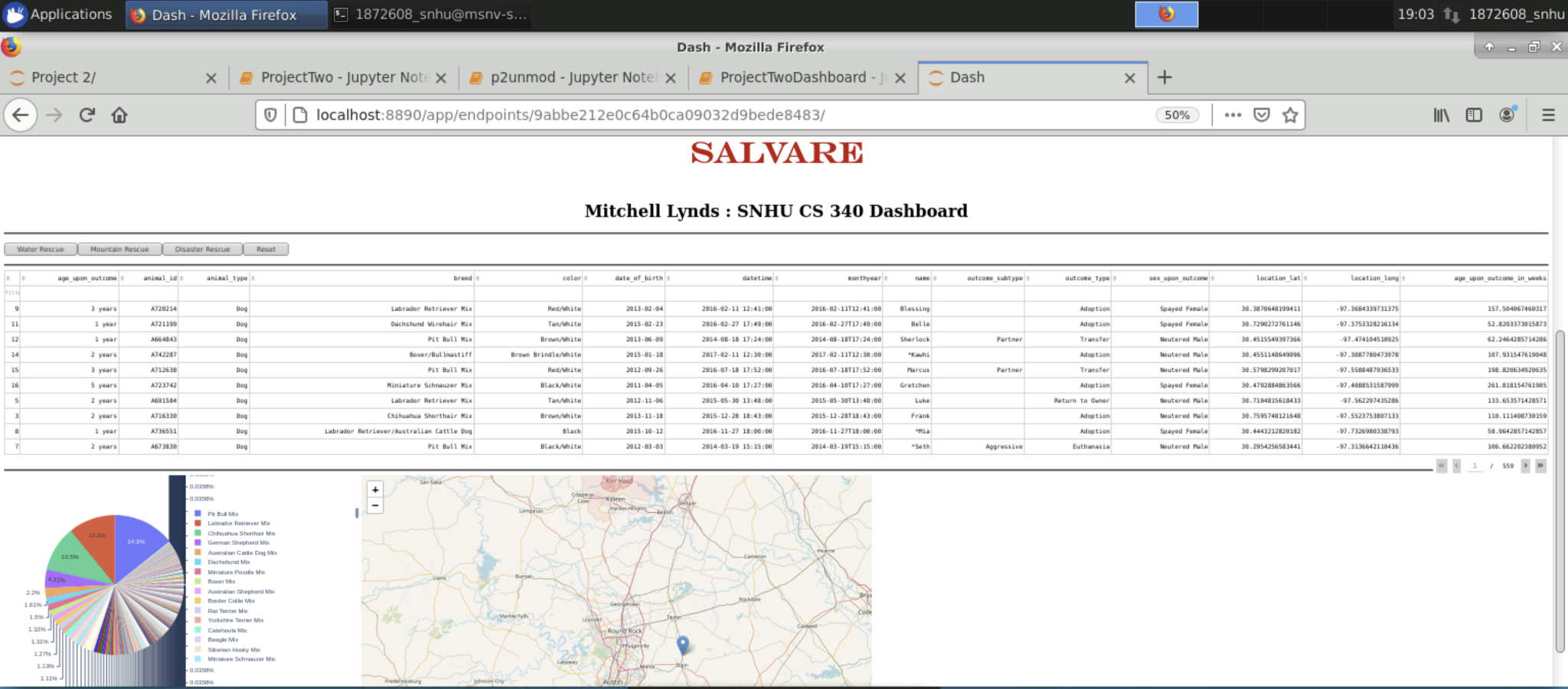


After the users have been created we can test connecting with authorization:

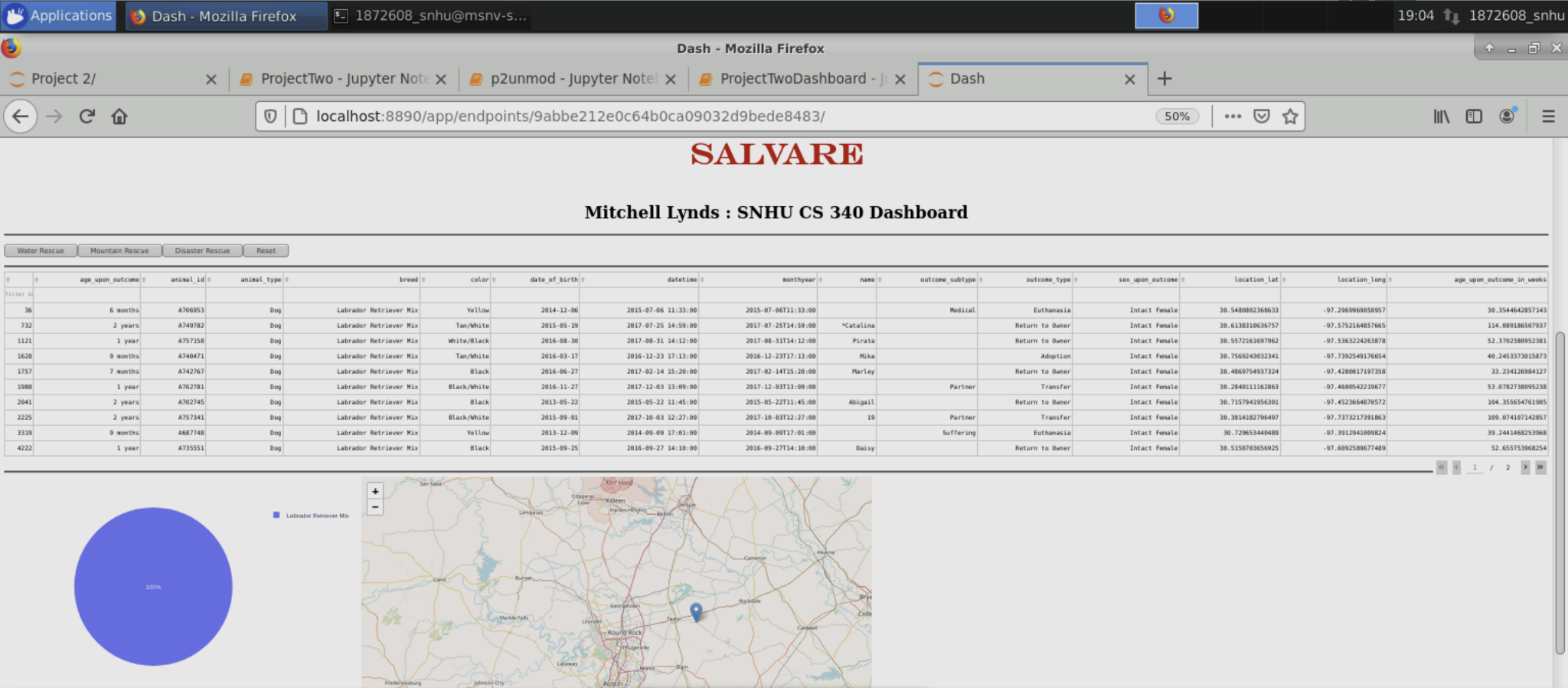


## Usage

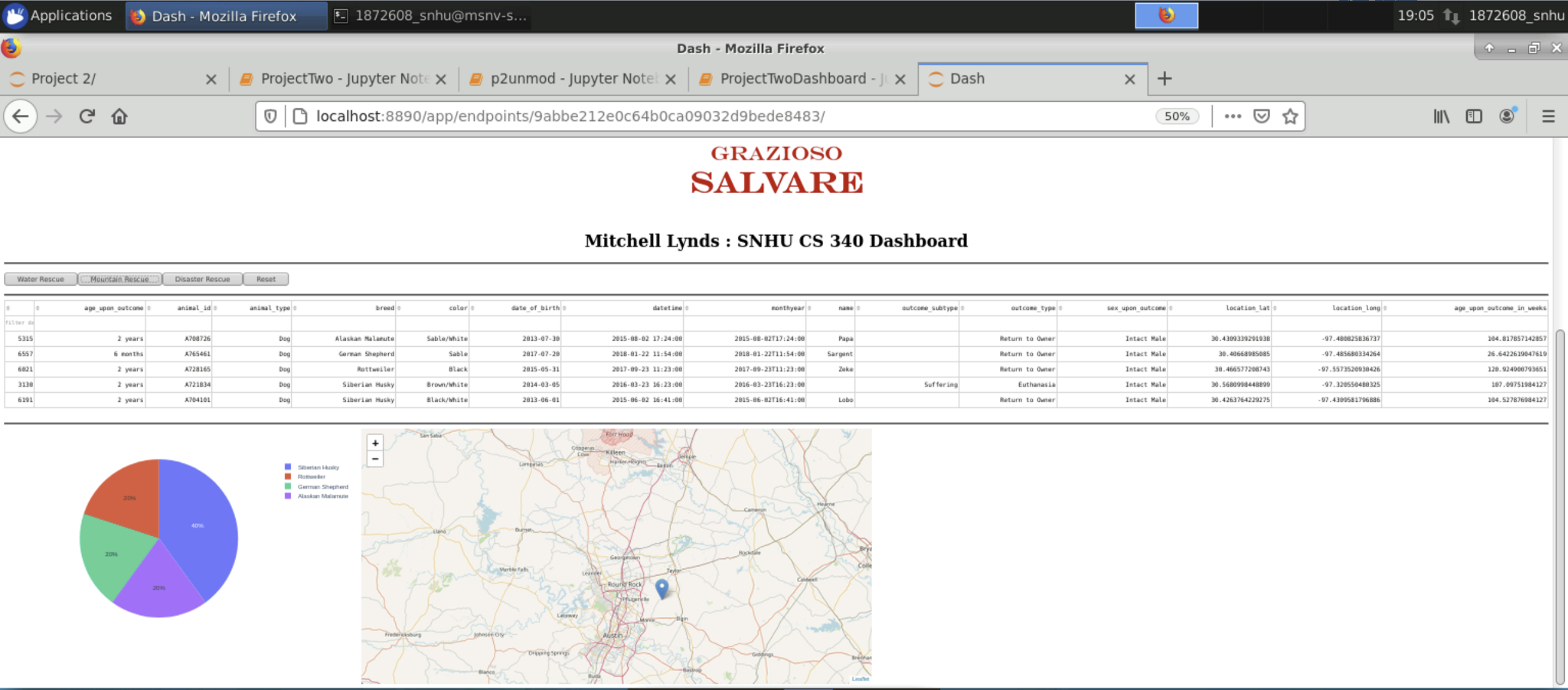
When run the python notebook will generate an interactive dashboard with a table and graphics.



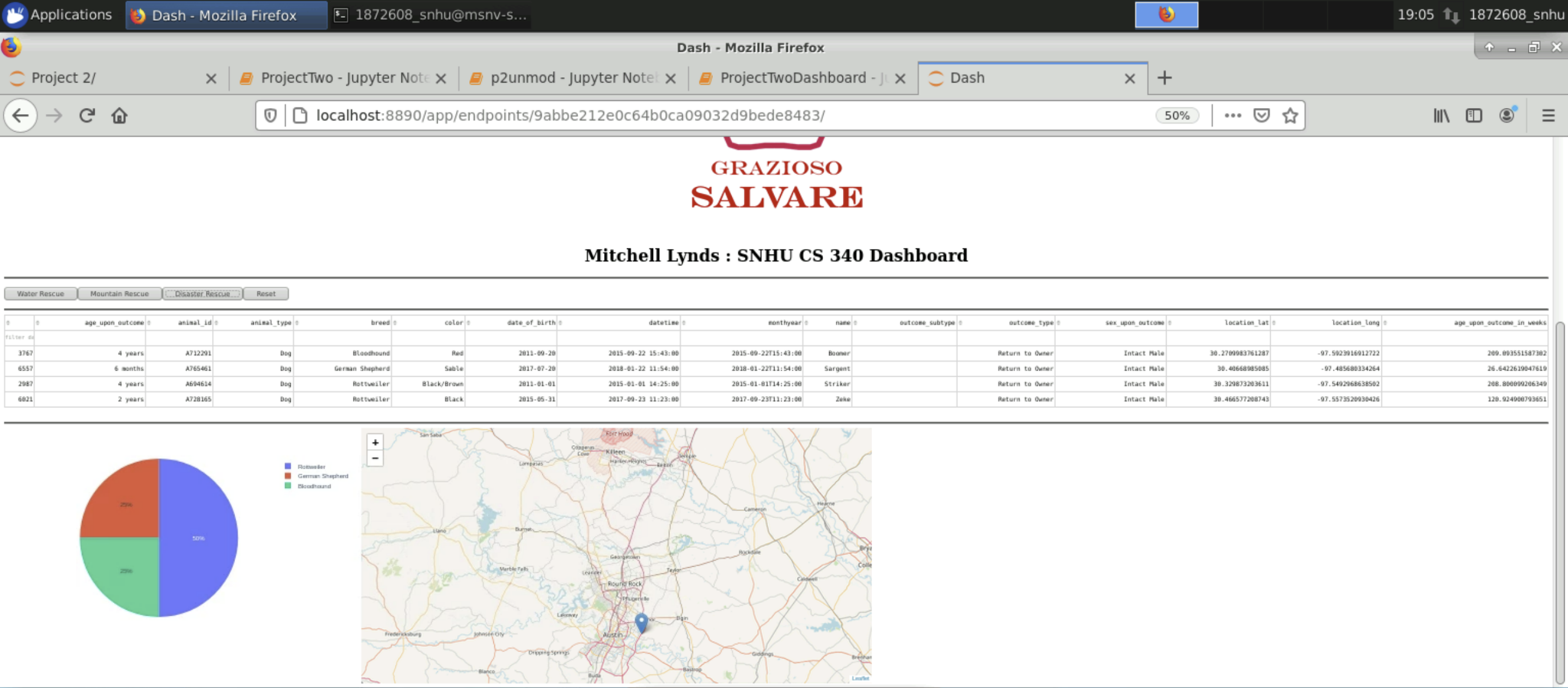
Pressing the “Water Rescue” button will filter for dogs suitable for water rescue training.



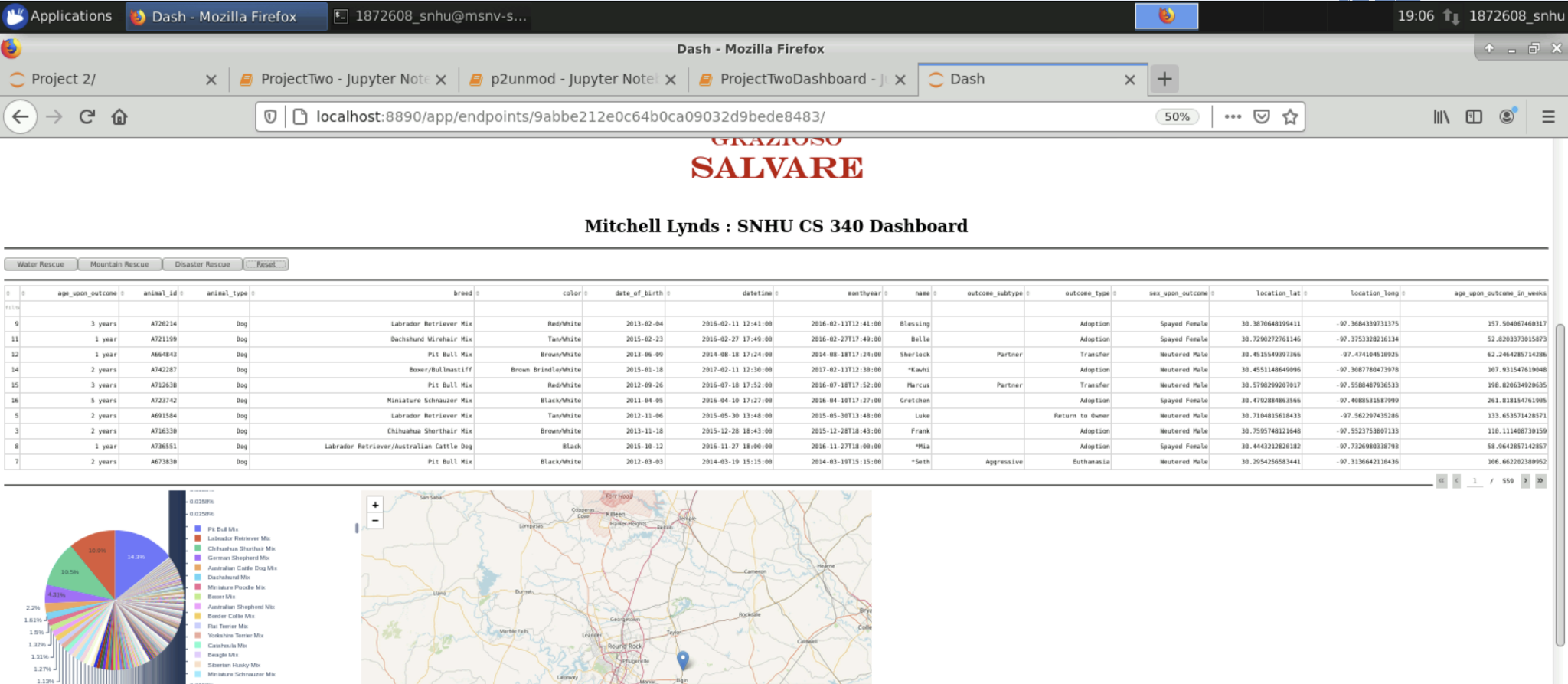
Pressing the “Mountain Rescue” button will filter for dogs suitable for mountain rescue training.



Pressing the “Disaster Rescue” button will filter for dogs suitable for disaster rescue training.



Pressing the “Reset” button will remove all filters and display the original data set.



**Tools Used:**

**MongoDB:**

MongoDB was chosen for its query flexibility and ease of integration with python thanks

to the pyMongo driver.

**Dash:**

The dash framework was chosen because it is an approachable tool for visualizing

datasets. Dash includes the DashTable as well as the core component used for the geolocation chart. Dash allowed the use of an HTML layout section combined with callbacks enabling interactivity. The callbacks allow the implementation of python methods to update the dashboard.

**Plotly:**

Plotly was used because of its compatibility with Dash and the Pandas DataFrame objects. Plotly provided the plotly.express.pie tool used to generate a pie chart from

the data set.

**Creating the project:**

The first step in creating this project was the CRUD module used to interact with the MongoDB database. Once that was complete the dashboard could be built. The dashboard needed to import the python CRUD module and be updated with the username and password for the database. Next, the logo for Grazioso Salvare was added to the top of the dashboard and set up as a URL link to SNHU.edu. After that, the settings for the Dash data table were selected. Then, buttons were placed for the interactive filters. With buttons in place it was then necessary to write a callback that would activate each time a button was clicked. The method for the button callback was written so that it would determine the most recently clicked button and apply the corresponding filter to the data set. After the buttons were finished the method for displaying the geolocation chart was added. Finally, the method for the plotly pie chart was implemented so that it would update each time a new filter was applied.

**Challenges:**

I struggled for a while to decide how to determine which button had most recently been clicked in order to implement the correct filter. Eventually, I realized that the button has a timestamp attribute storing the time when the button was last clicked and comparing these values makes it easy to determine the correct filter to use.

I also struggled with the syntax for the plotly express pie chart. I knew that I needed to count the number of occurrences for each dog breed but was unable to get it to work until finding an example that used the .value\_count() feature.

## Contact

Mitchell Lynds