# CS 340 README Project 2

## About the Project/Project Title

*This project allows a user to login and load a dashboard that interacts with the database holding all animals’ data. This dashboard shows all the animals and allows the user to manipulate it to filter animals to help find them all. The dashboard only shows 10 at a time to not overload the page with to much and making the interface run slower. The interface also allows users to select and animal in the database and it will show their current documented geolocation on a map. The dashboard was also needed to show a chart of some type I chose a histogram to show the breed vs quantity of that breed of the current page of the dashboard. Meaning as the user went to the next page the histogram would update. This is due to the CRUD .py file we made for project one interacting with the dashboard .ipynb file.*

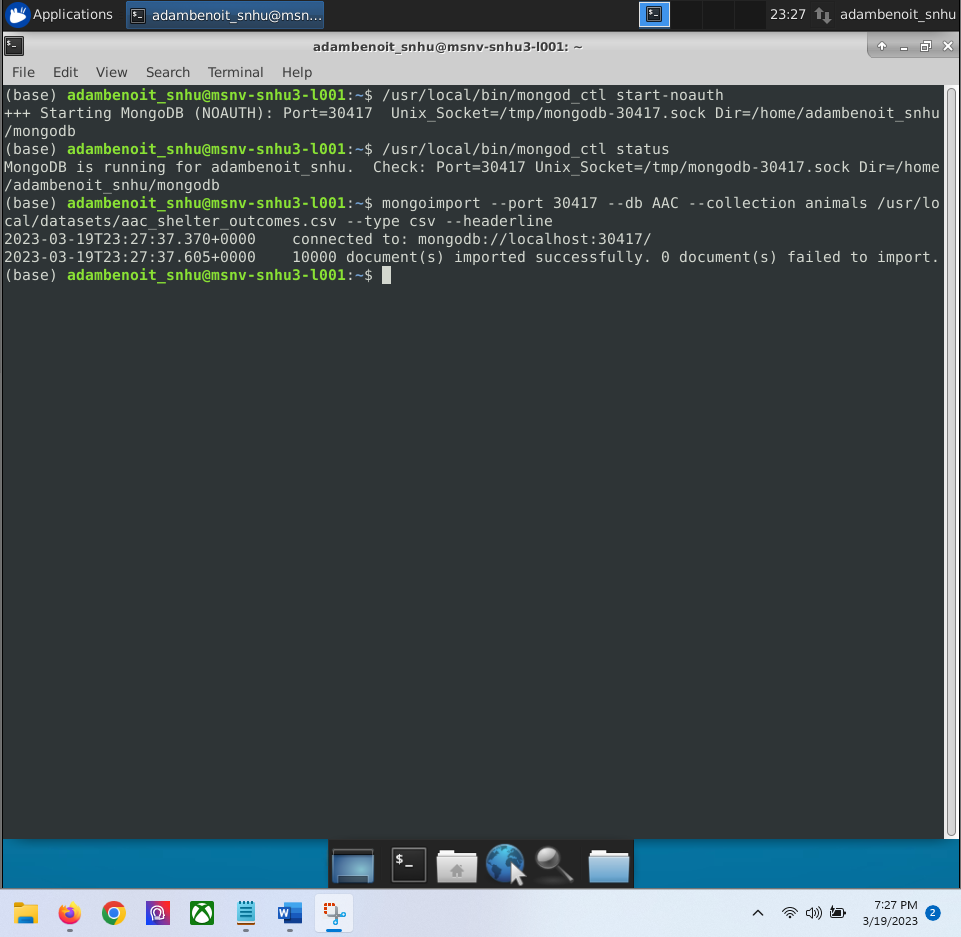
## Motivation

*The motivation behind this project was to build off the first project and implement a dashboard to show the data within the database in an organized and sorted way. This would allow user to filter the animals making it easier to search and find animals they are looking for. For example, I add functions that would allow to filter for specific dogs based on information given by the client to predetermine search requirements that met dogs that would be good in water rescue, mountain rescue and disaster/tracking.*

## Getting Started

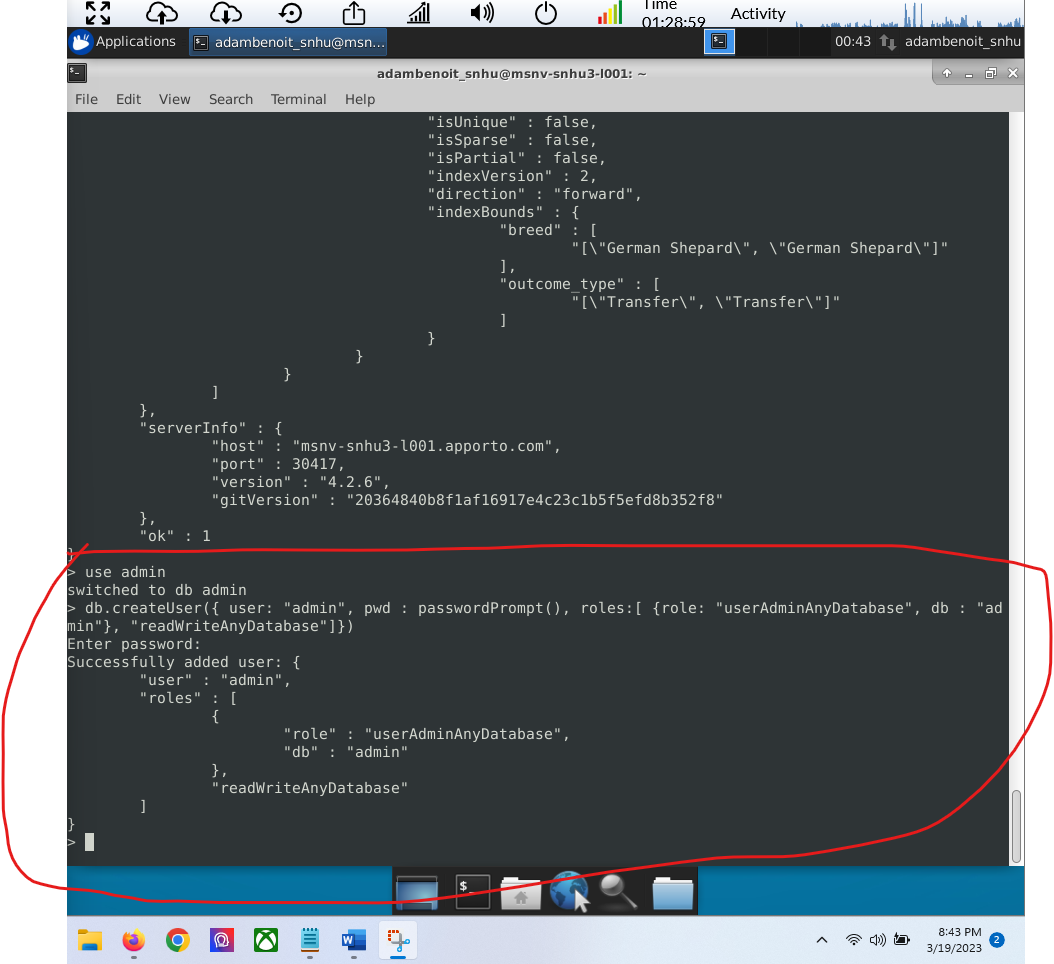
Step one is to start mongo and note what port number was assigned.

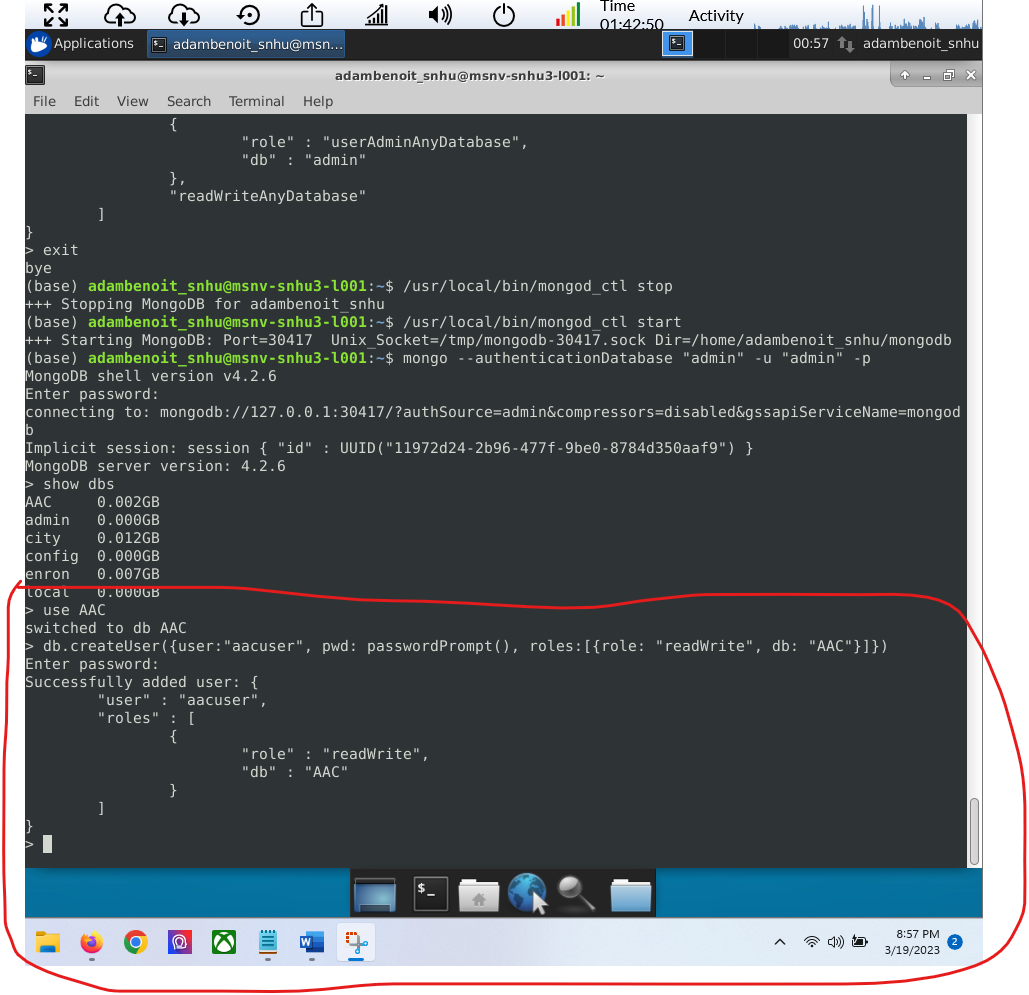
Step two is to import the data set into MongoDB

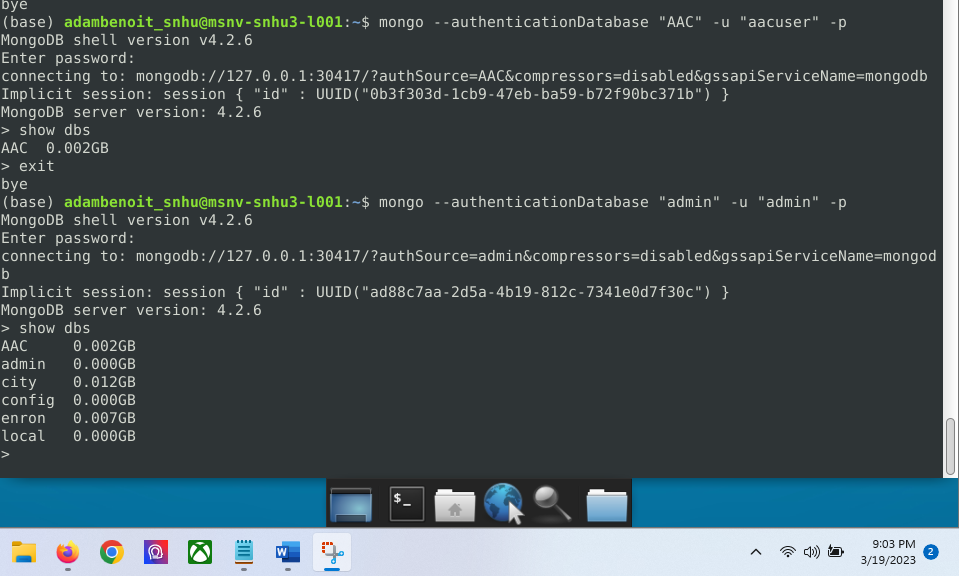


Step three would be to enter mongo into the command to start a session

Step four need to create the admin account and the aac user account for access to the database



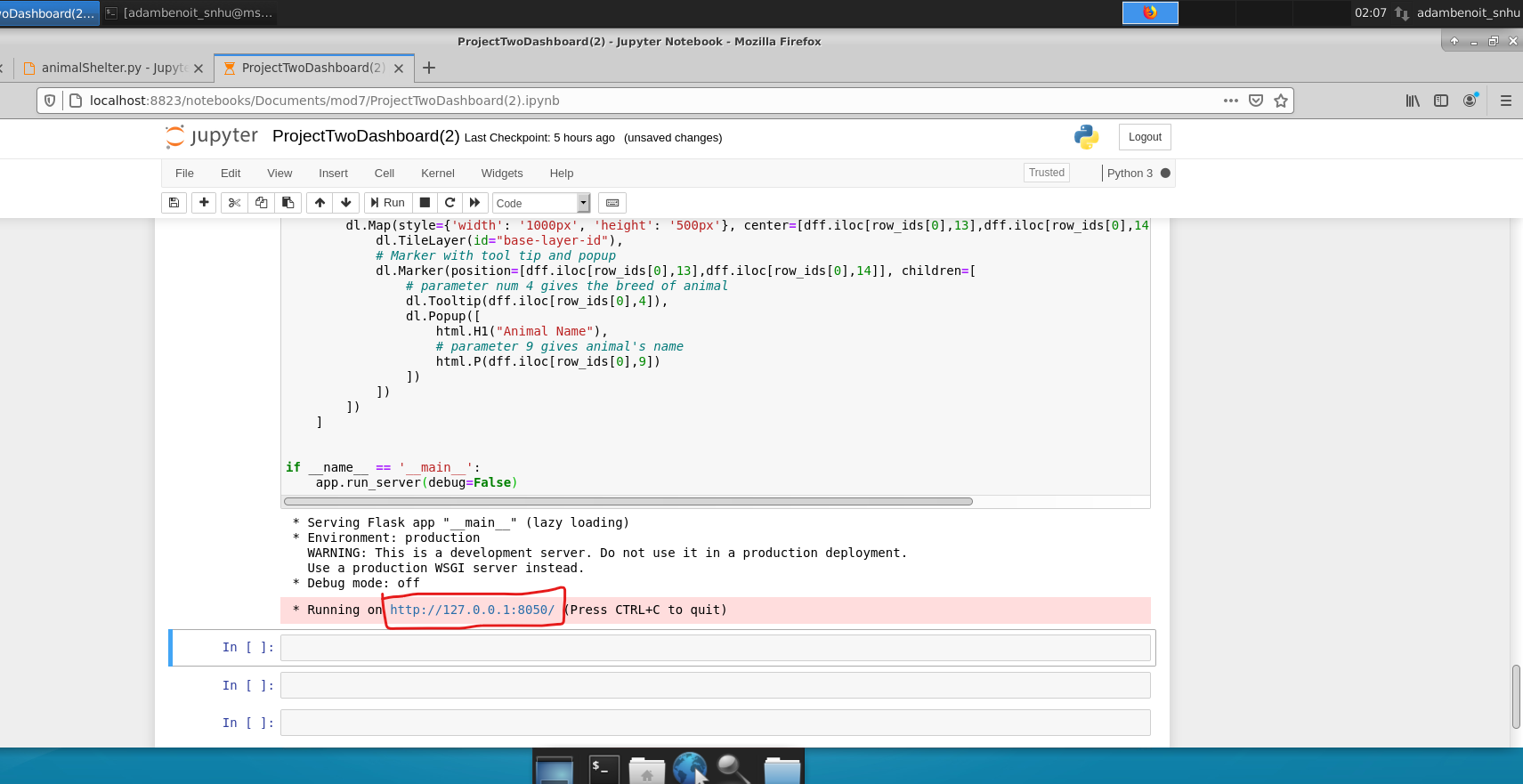




Step five exit mongo and stop the session and make sure you can start it again and log in with the user accounts

Step six switch to Jupiter notebook and create two files one .py for the CRUD method and the other .ipynb for testing purposes to run commands through the .py file to create a dashboard of the data for user viewing.

Step seven run the test script environment and click on the link it produces to open up the dashboard.



Challenges that I had at first was that the dashboard was loading but was completely blank. I over came this by looking through the given resources and doing some independent research and found the answer was not that there was something with my dashboard .ipynb file but with my crud.py file. It seems by passing the print through the read was causing the dashboard to load un populated. I took out the print in my .py file and allowed the .ipynb file to do the printing and my problem was fixed.

## Installation

*You will first need to have MongoDB cable computer and install it. Mongodb was used because it allows for easy and quick setup of a database from an excel sheet (.csv file). It is also a very friendly interface to integrate with python which is also a very user friendly programming language to use in comparison to some other languages. These reasons allowed for a faster implementation and project development than that of a SQL database directly.*

*You will also need a current version of Python to be able to run and create .py and .ipynb files.*

*You will need the database you plan on manipulating also installed and also uploaded into the MongoDB*

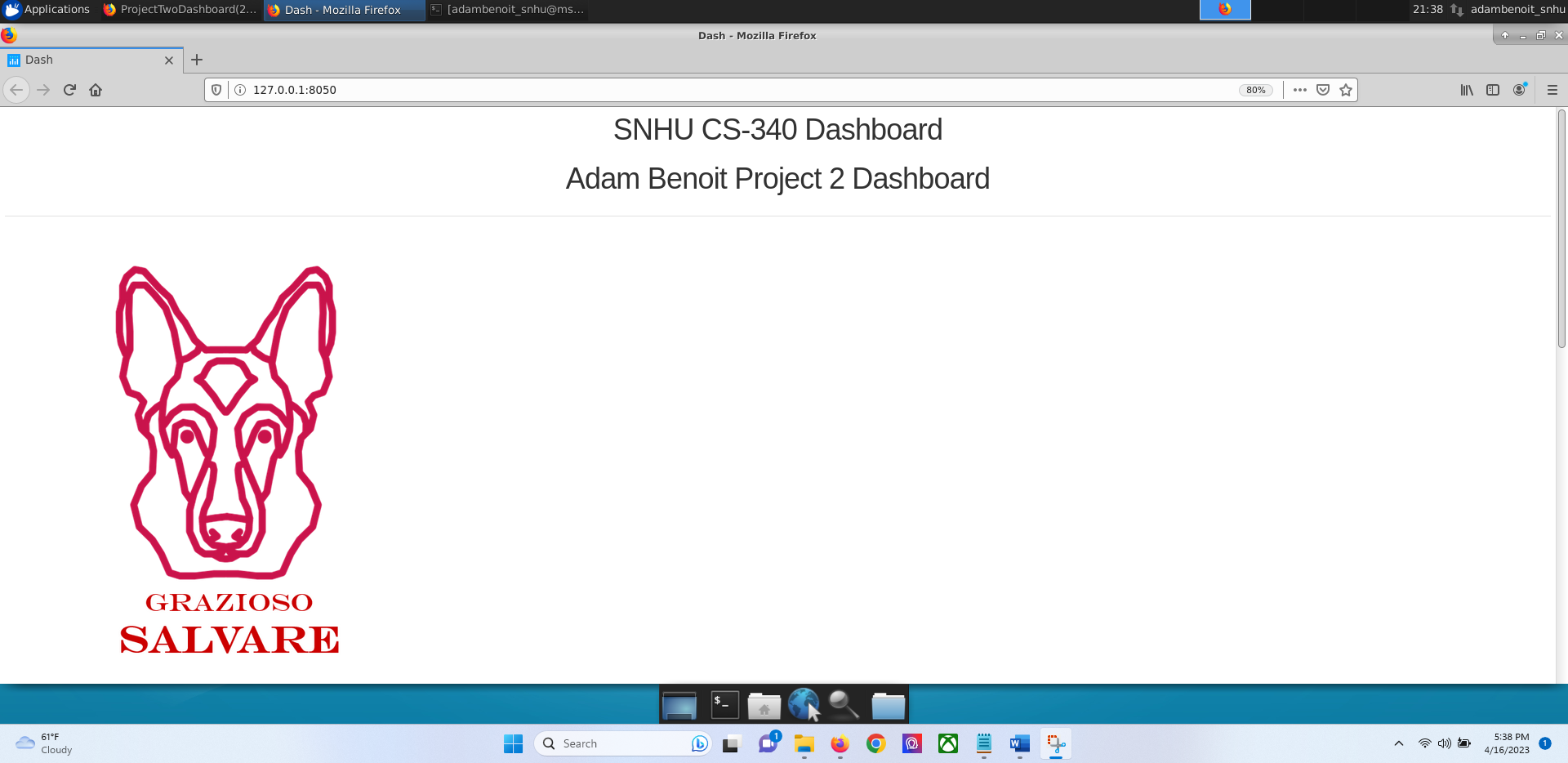
*You will need an IDE for writing the scripts and I suggest installing jupiterbook just like used.*

*You will also need to use dash framework to get the dashboard to show and work properly. The reason dash is chosen for this project is because it controls outputs to the segments and then correcty updates the target inputs from the specified app callbacks. This allowed me to create a dashboard that was interactive with a user and allowing it correctly update and show what the user wanted by correctly pulling the correct data from the database.*

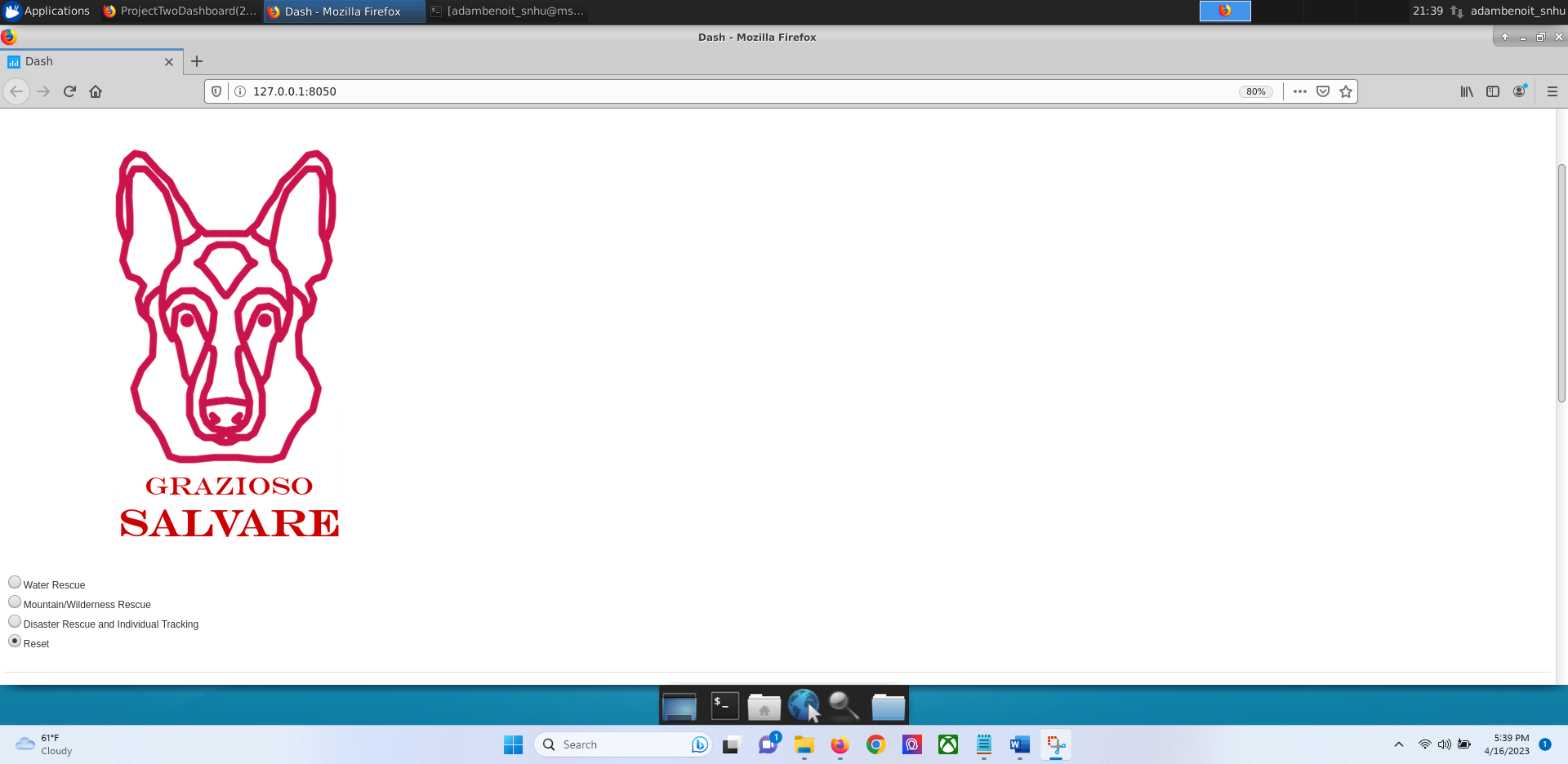
## Usage

### Screenshots

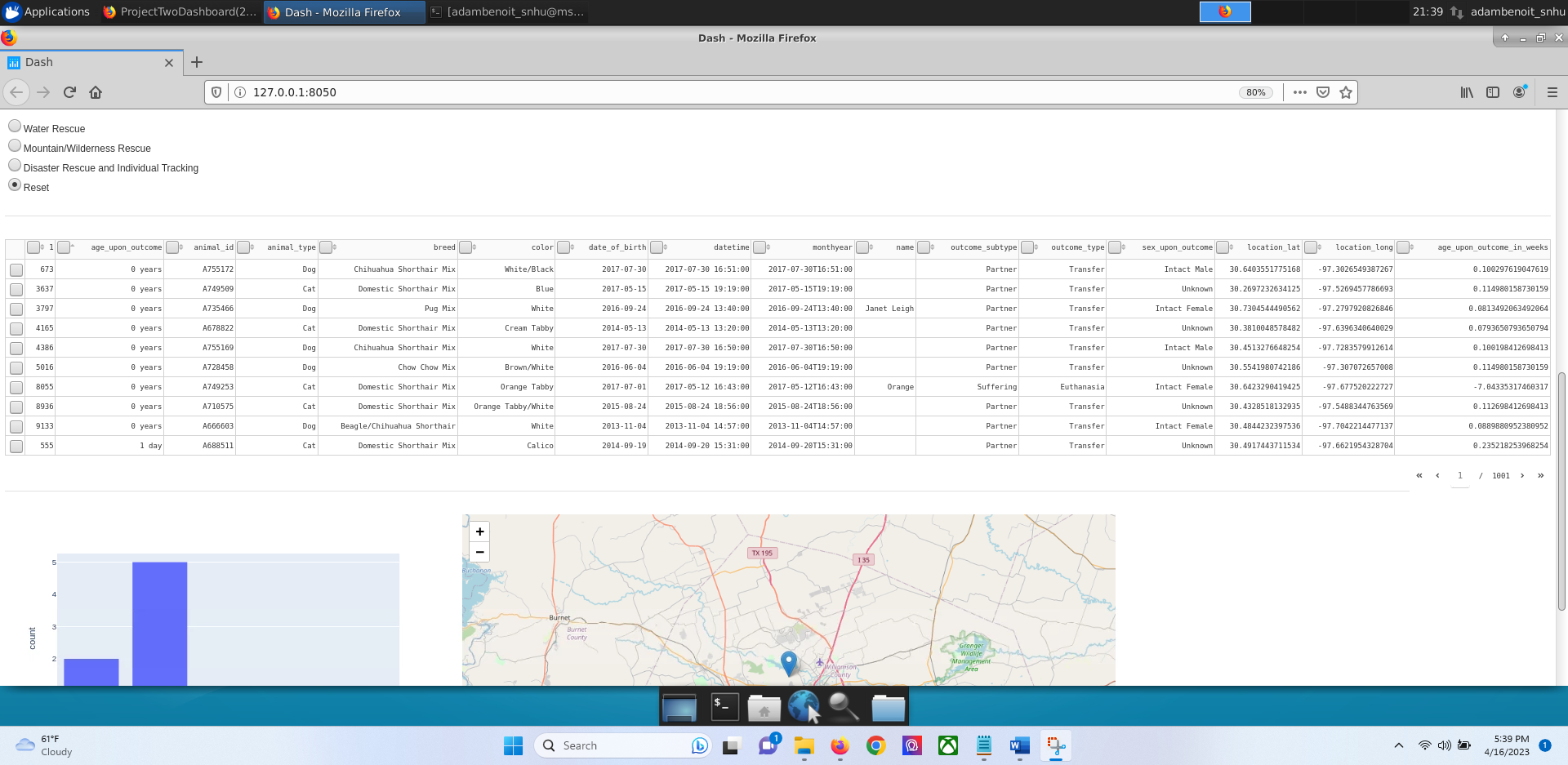
Dashboard loads.



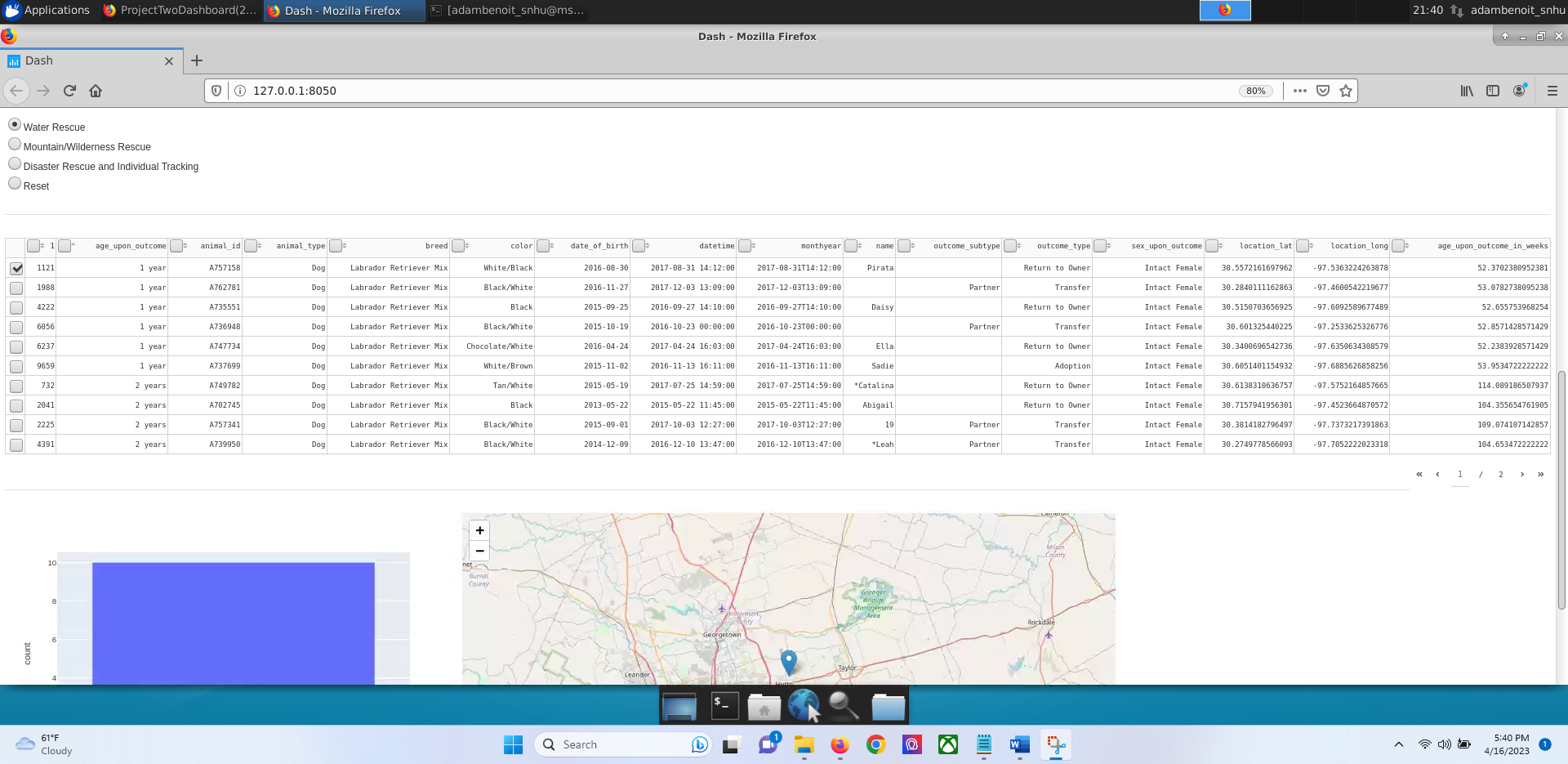
Scrolled down to see the predetermined filters for types of rescue dogs



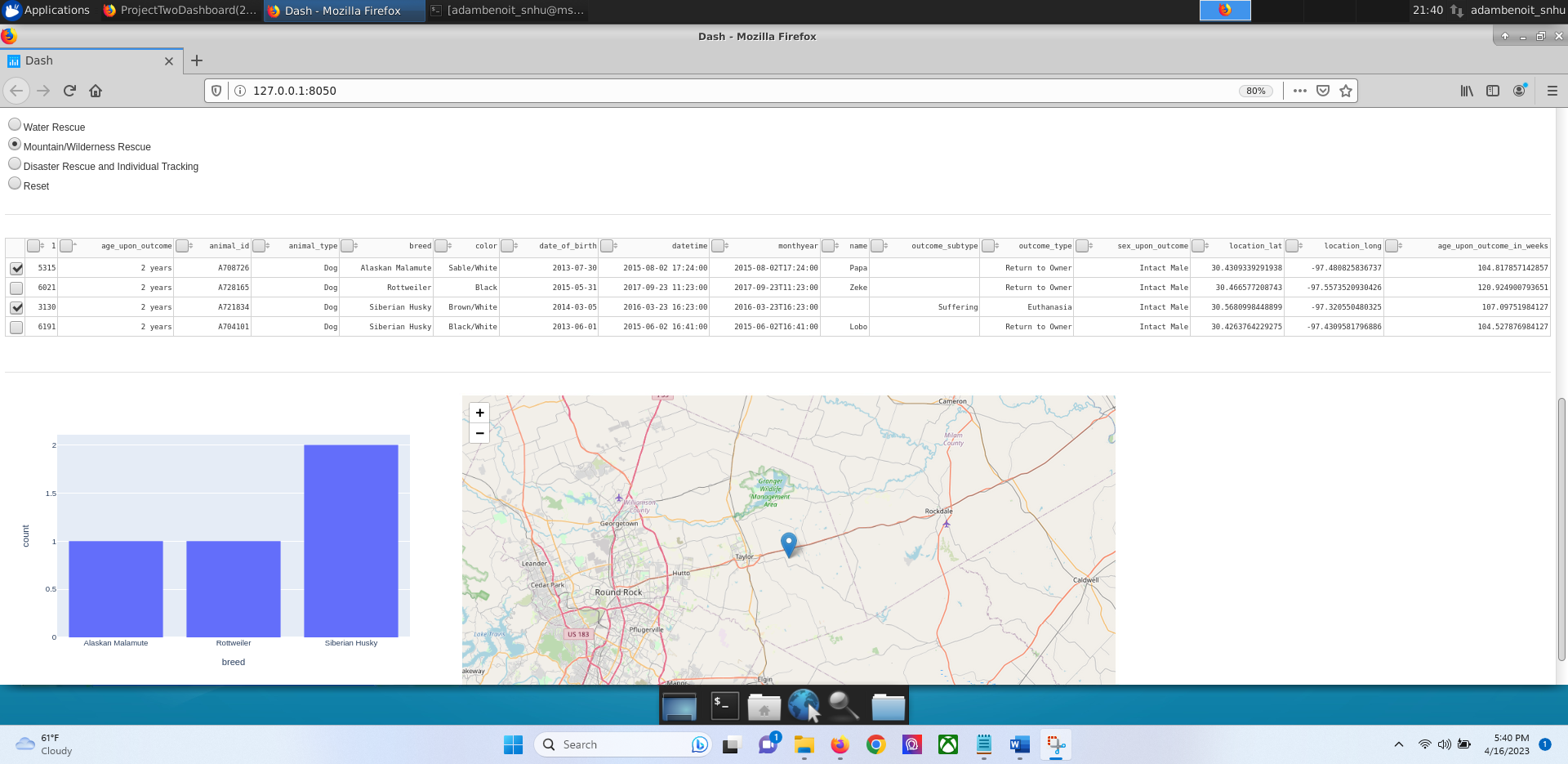
Scroll down to show default view of dashboard showing all animals. Top of histogram and geolocation map can be seen also



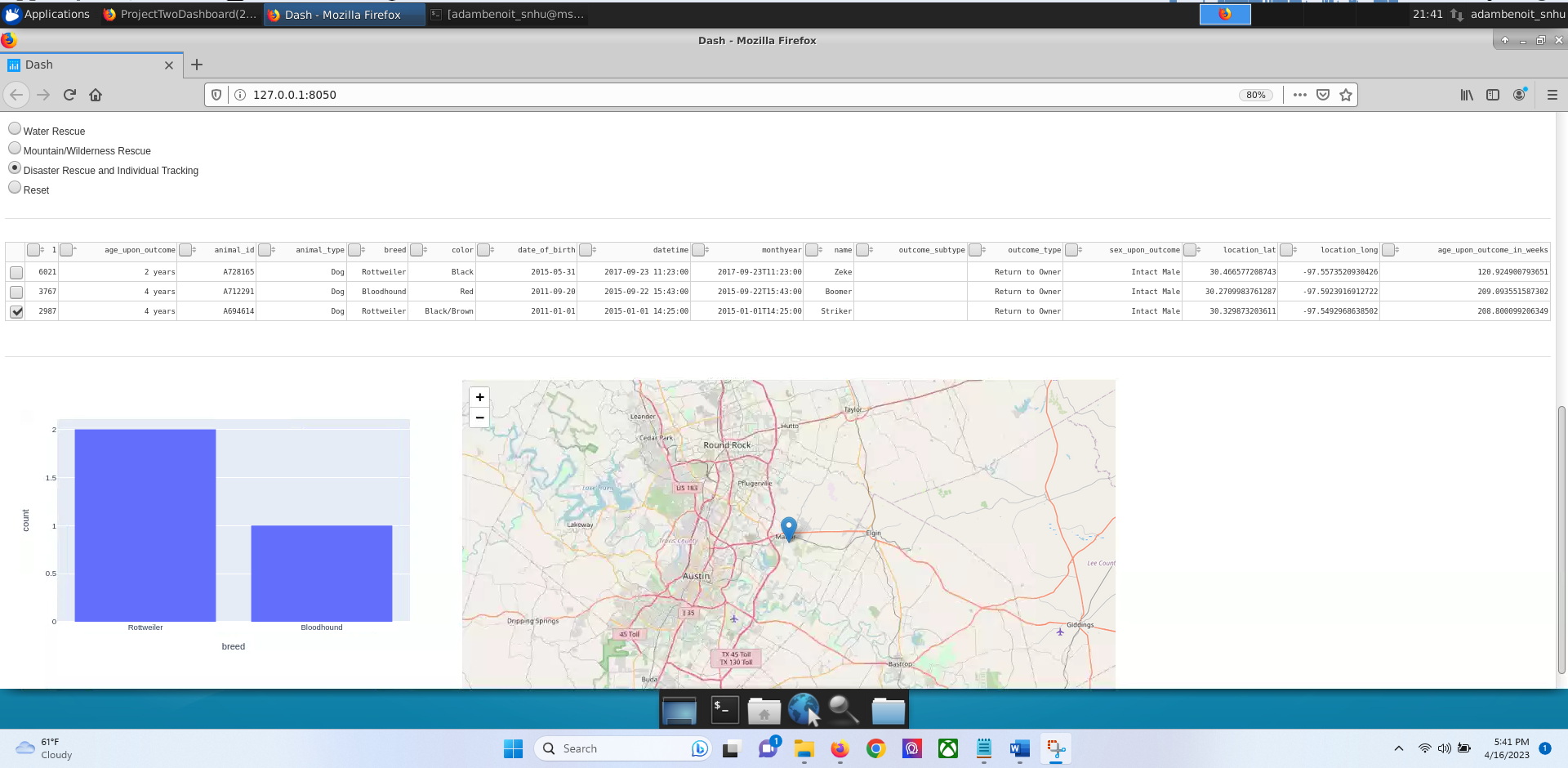
First filter of water rescue and selected with the expected reults the histogram is also seen to have been updated to what it had been previously and the geolocation as changed due to the animal that was selected has changed.



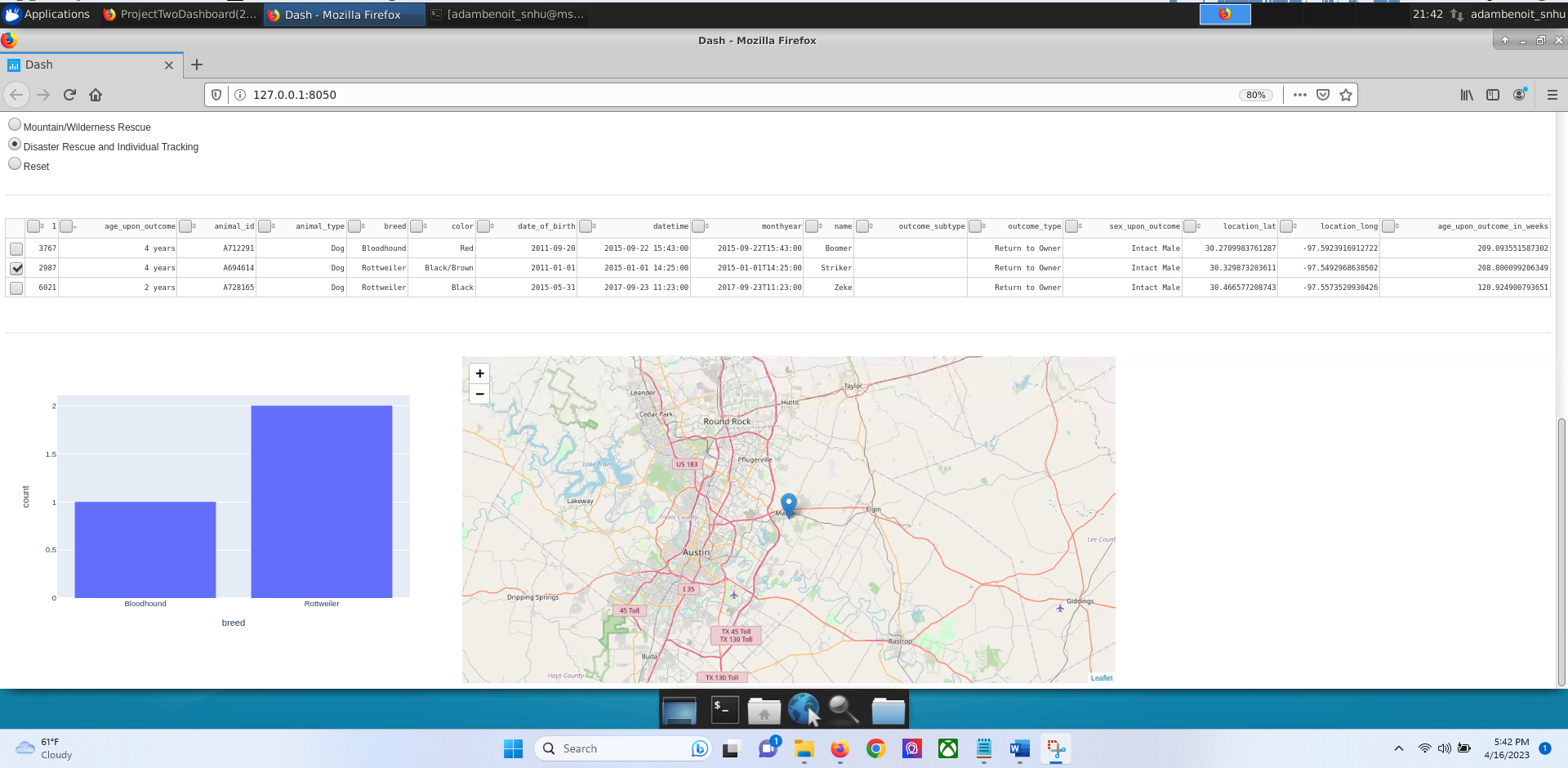
The next predetermined rescue filter and as shown you can the amount has changed to the expected amount based on the dogs for this filter. Geolocation and histogram properly change as well



The next predetermined rescue filter and as shown you can the amount has changed to the expected amount based on the dogs for this filter. Geolocation and histogram properly change as well



Showing that the individual columns of the dashboard can respond and filter animals based on the column in this picture I chose to make age go from oldest to youngest instead of how it was in the picture above with youngest to oldest.



## Contact

Adam Benoit