# John Austin CS 340

# Project Two README

## About the Project/Project Title

This project uses data visualization to process and display data from a database in a clean and organized format. Users can quickly filter through results and display the locations of their selections on a map. There is also a pie chart to allow users to quickly visualize the breakup of the results from queries or even the entire dataset as a whole.

## Motivation

This project was commissioned and made open source thanks to Grazioso Salvare, an international rescue training company. Grazioso Salvare, through their many years of experience, discovered that specific profiles in dogs are best suited to the job of being trained and becoming international rescue -animals. He has shared this knowledge of dog profiles that are best for training in specific circumstances. With this, we developed a way to filter large datasets quickly based on the type of potential rescue animal you are looking for. We also displayed the breakdown of the results to help in your searches such as the location of the animals on a map and a breakdown by breed.

## Getting Started

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

* Download the “DataCRUD.py” file and the “Grazioso Salvare Logo 50.png” file.
* Edit the file on lines 10 & 11 by replacing the words in ALL CAPITALS with their titled values from your MongoDB database:

self.client = MongoClient('mongodb://%s:%s@localhost:PORT#/DATBASE\_NAME') % (username, password)

self.database = self.client['DATABASE\_NAME']

* Save the DataCRUD file
* In the ProjectTwoDashboard.ipynb file edit lines 25 & 26. These are where your credentials for the database collection go.
* Run the script.

## Installation

You must first have MongoDB preinstalled on your system. You will need to then start MongoDB in one terminal. In another terminal make sure you have pymongo installed by entering:

pip install pymongo

You will need this installed to interact with the Mongo database through python. You can use a python IDE such as Jupyter Notebook or PyCharm.

## Usage

### Code Example

The DataCRUD.py file allows users to access a database in which they can quickly add, view, edit, and remove data with simple one-line functions that can be used in a python script for reusability when performing repetitive tasks, such as performing specific search criteria on a dataset once a week. In a python IDE after importing the class to use you can simply input data into a create() function to add a data entry. To find an entry pass your filters as the parameters for the read() function. For the update function, you will pass two parameters, the file you want to update and the change you want to make. Lastly, for the delete function, you pass a filter of what entries you want to remove from the function and a second parameter that determines if the function removes the first matching entry or all matching entries.

* object.create({data})

Adds an entry to the database of the data passed

* object.read({data})

Displays entries matching the data passed

* object.update({whatDataToChange}, {whatToChangeAboutData})

Edits entries matching “whatDataToChange” with “whatToChangeAboutData”

* object.delete({data}, 1)

Deletes the first matching entry of the data passed

* object.delete({data}, 2)

Deletes all entries matching data passed

The ProjectTwoDashoboard.ipynb file is a prebuilt script (given the changes needed to be made mentioned above) that filters data tailored specifically for this use case but with minor changes that could be repurposed for various other applications depending on the dataset.

### Tests

>>> from DataCRUD import AnimalShelter

>>> object1 = AnimalShelter()

>>> object1.create({

"age\_upon\_outcome" : "7 months",

"animal\_id" : "B428934",

"animal\_type" : "Dog",

"breed" : "Beagle",

"color" : "Brown with red suit and blue cape",

"date\_of\_birth" : "1964-10-03",

"datetime" : "2016-08-19 18:11:00",

"monthyear" : "2016-08-19T18:11:00",

"name" : "Underdog Jr",

"outcome\_type" : "Beloved Superhero",

"sex\_upon\_outcome" : "Intact Male"})

>>> object1.read({'name' : 'Underdog Jr'})

{'\_id': ObjectId('638d8530d7eb7f2dbee81d05'), 'age\_upon\_outcome': '7 months', 'animal\_id': 'B428934', 'animal\_type': 'Dog', 'breed': 'Beagle', 'color': 'Brown with red suit and blue cape', 'date\_of\_birth': '1964-10-03', 'datetime': '2016-08-19 18:11:00', 'monthyear': '2016-08-19T18:11:00', 'name': 'Underdog Jr', 'outcome\_type': 'Beloved Superhero', 'sex\_upon\_outcome': 'Intact Male'}

>>> object1.update({'name' : 'Underdog Jr'}, {'$set' : {'age\_upon\_outcome' : '8 weeks'}})

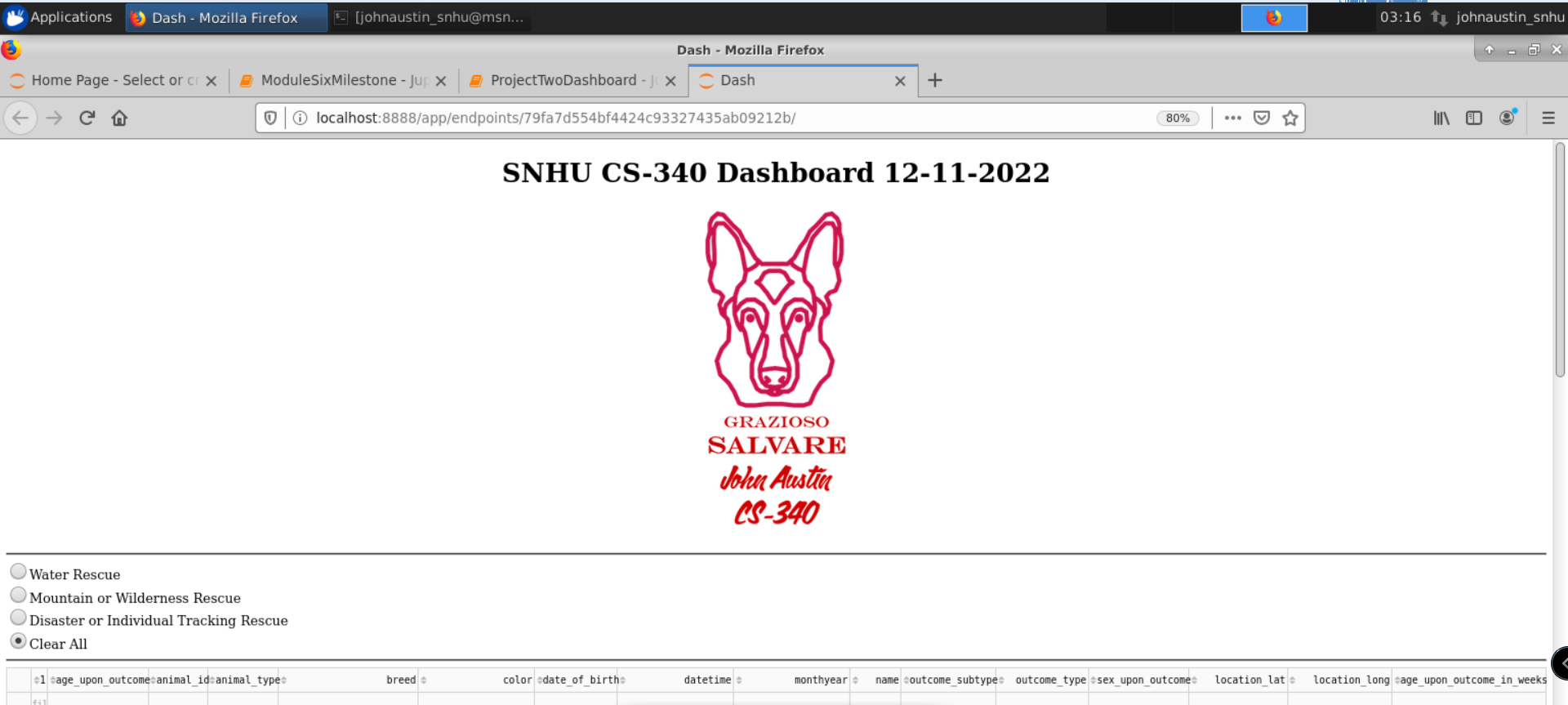
{'n': 1, 'nModified': 1, 'ok': 1.0, 'updatedExisting': True}

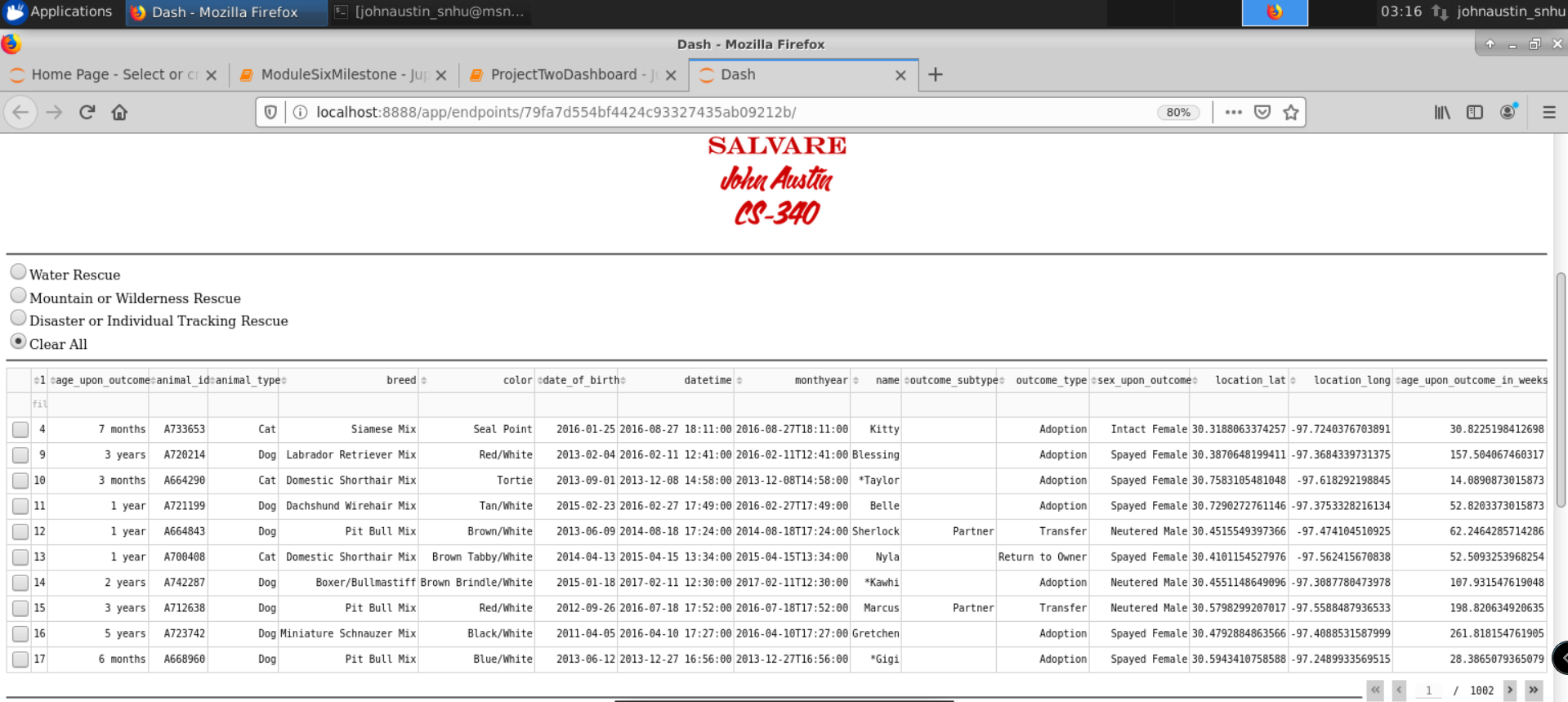
>>> object1.delete({'name' : 'Underdog Jr'}, 1)

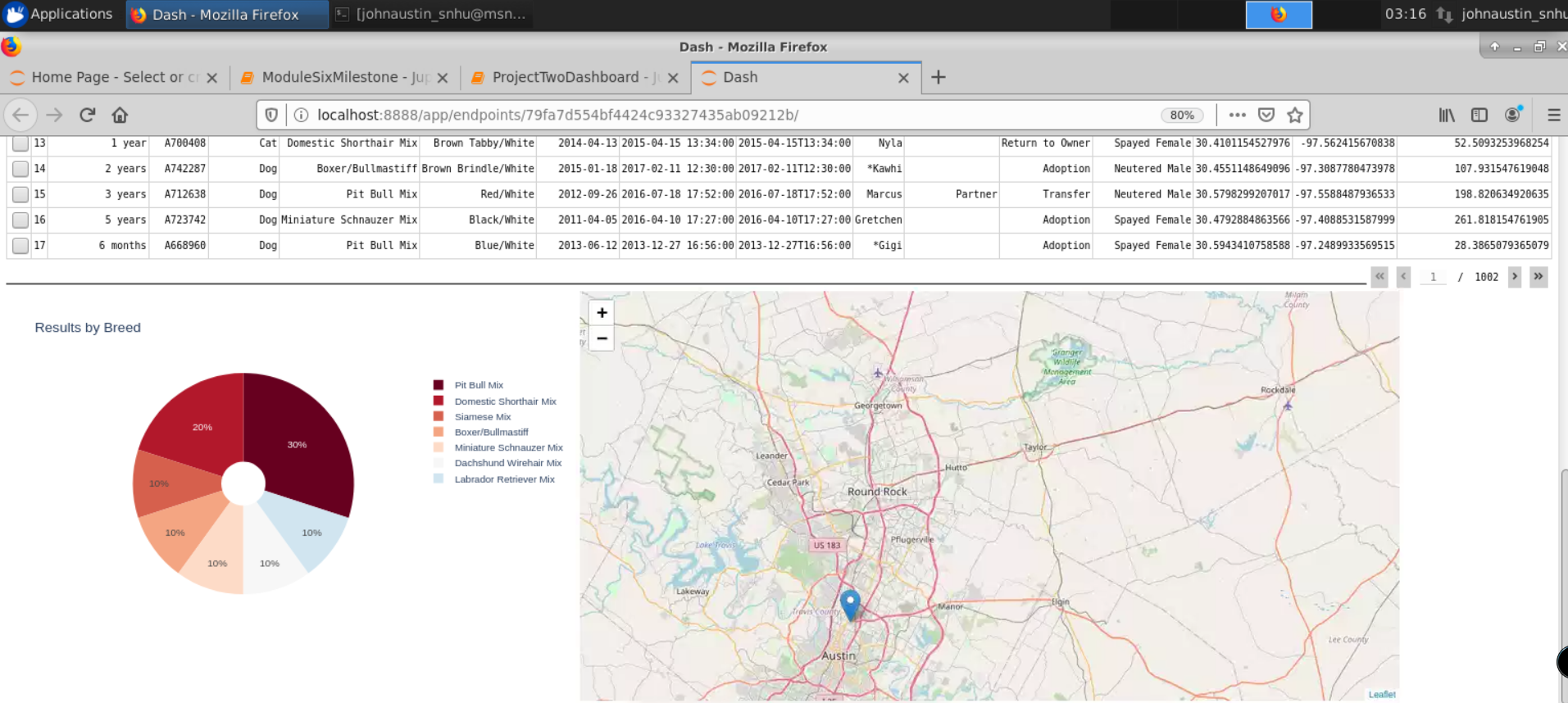
<pymongo.results.DeleteResult object at 0x7ff5b18b9688>

### Screenshots

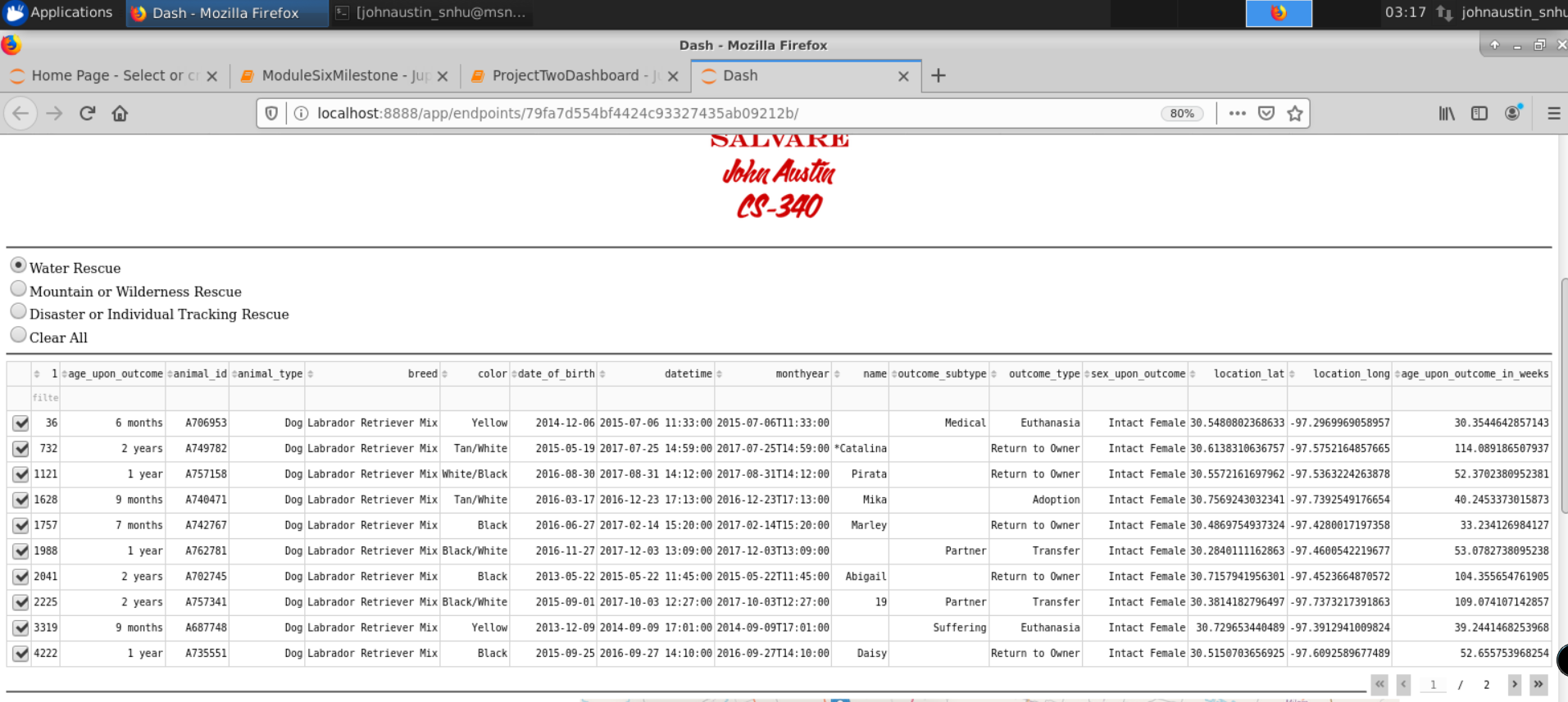
*Provide screenshots that demonstrate your work.*

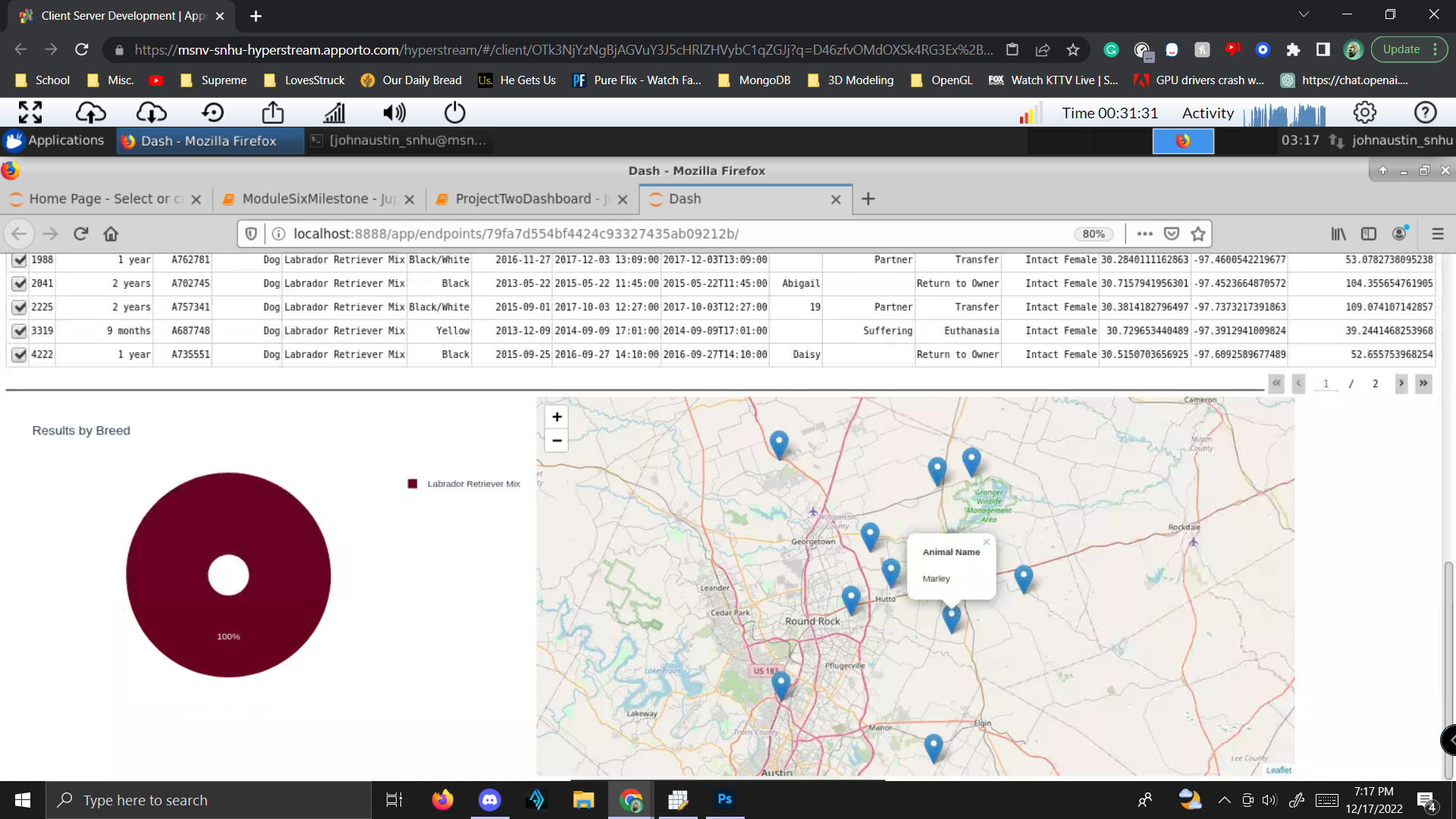




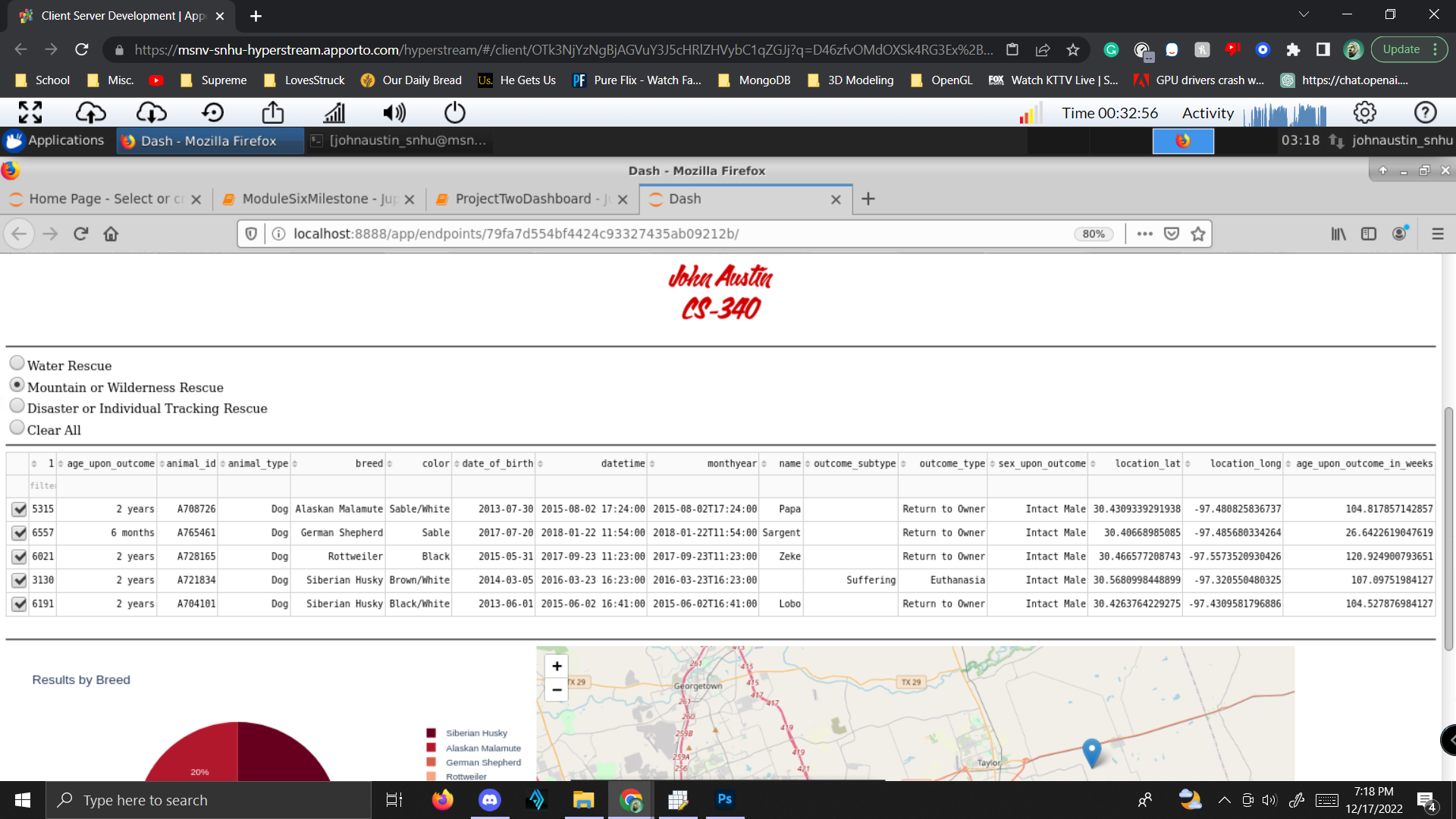


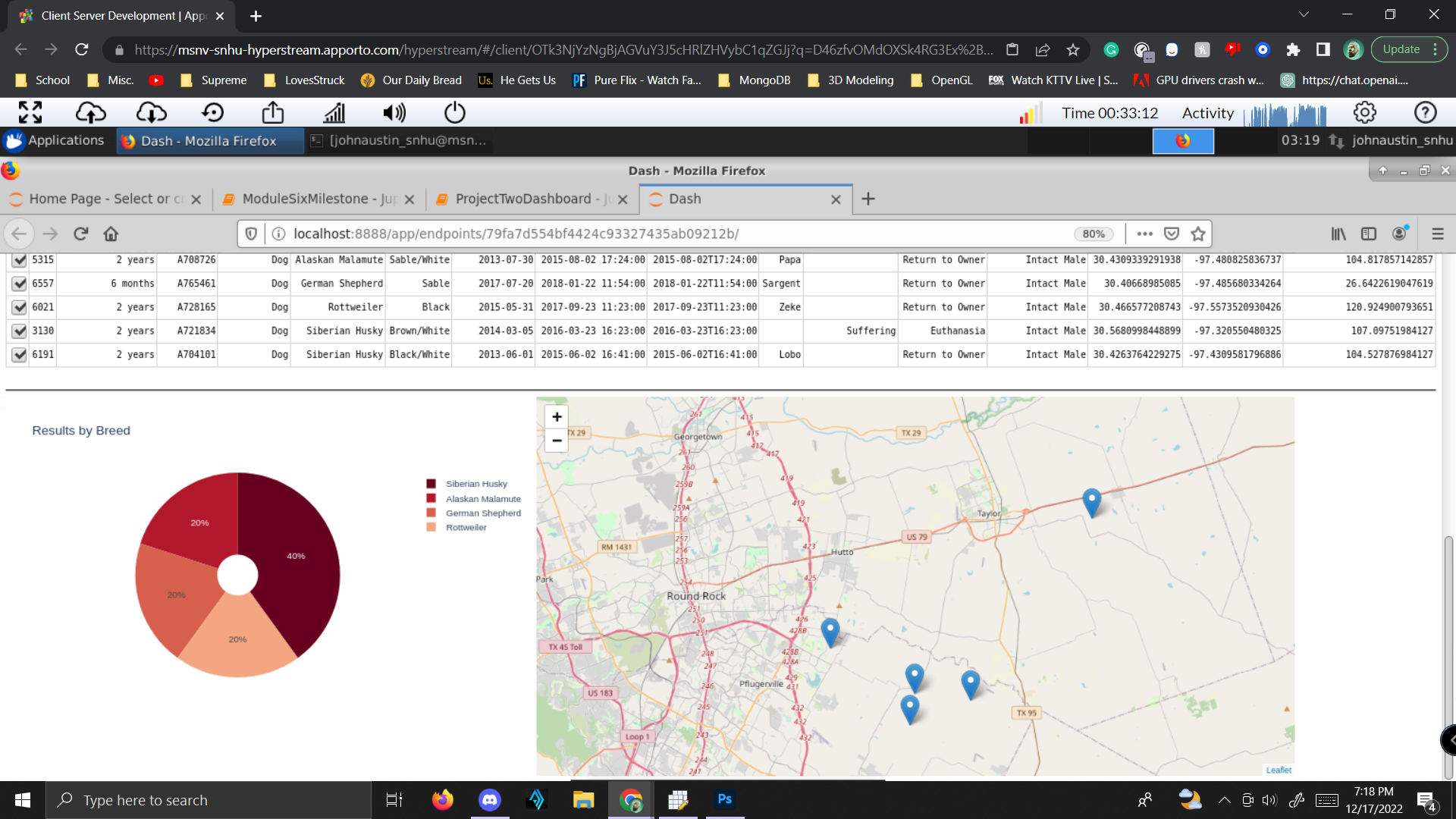
* Water Rescue Search



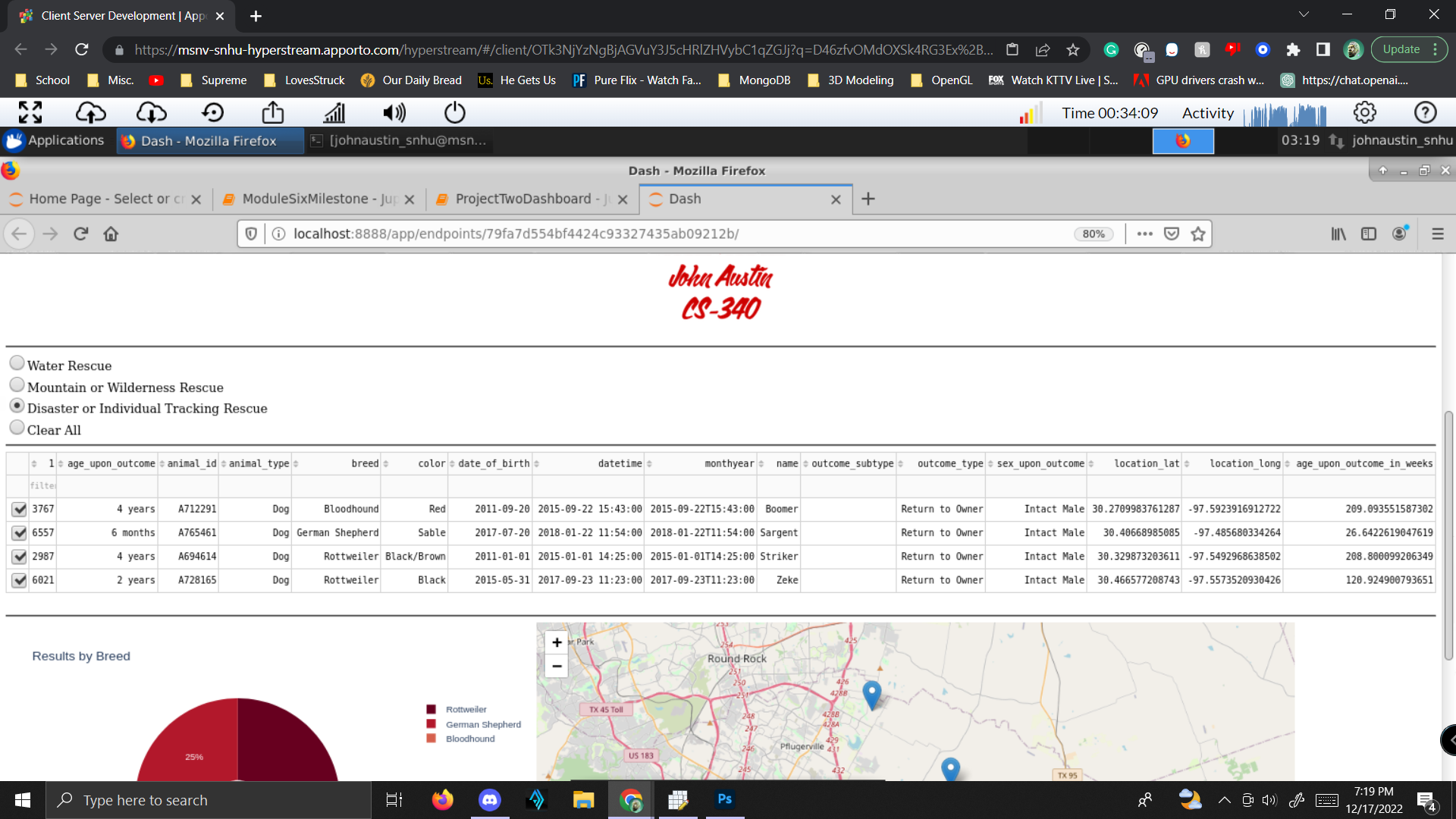


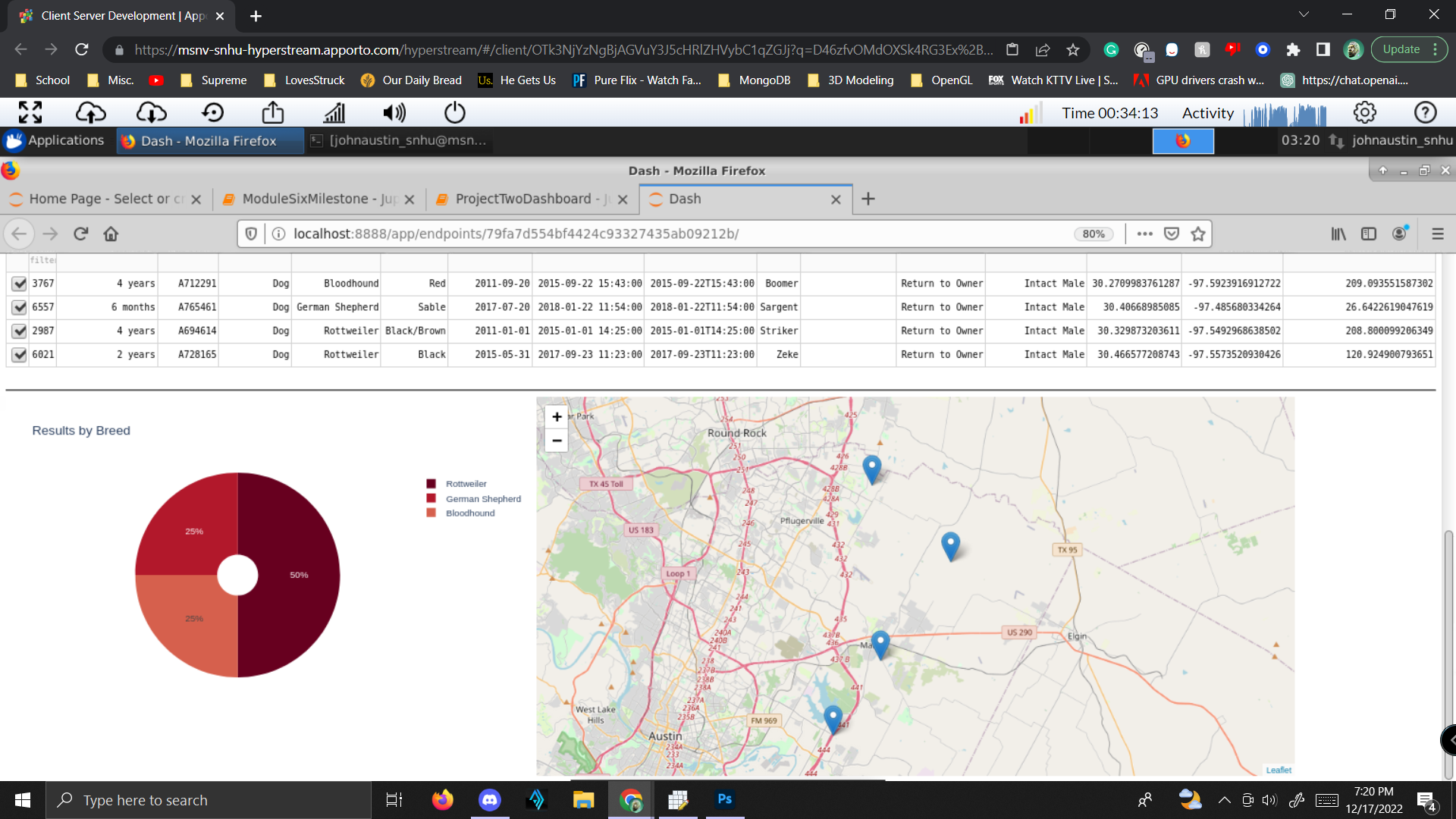
* Mountain or Wilderness Rescue



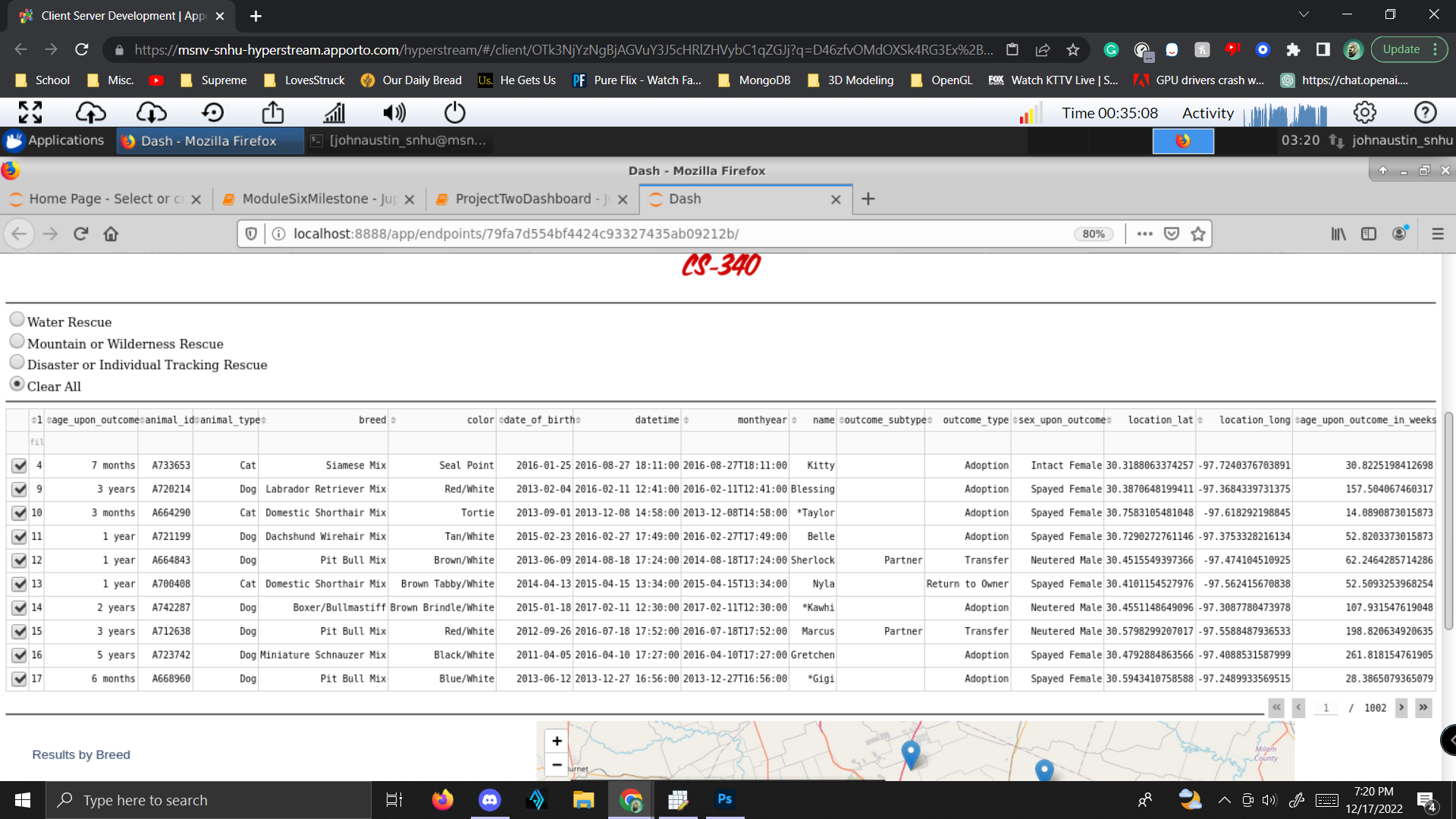


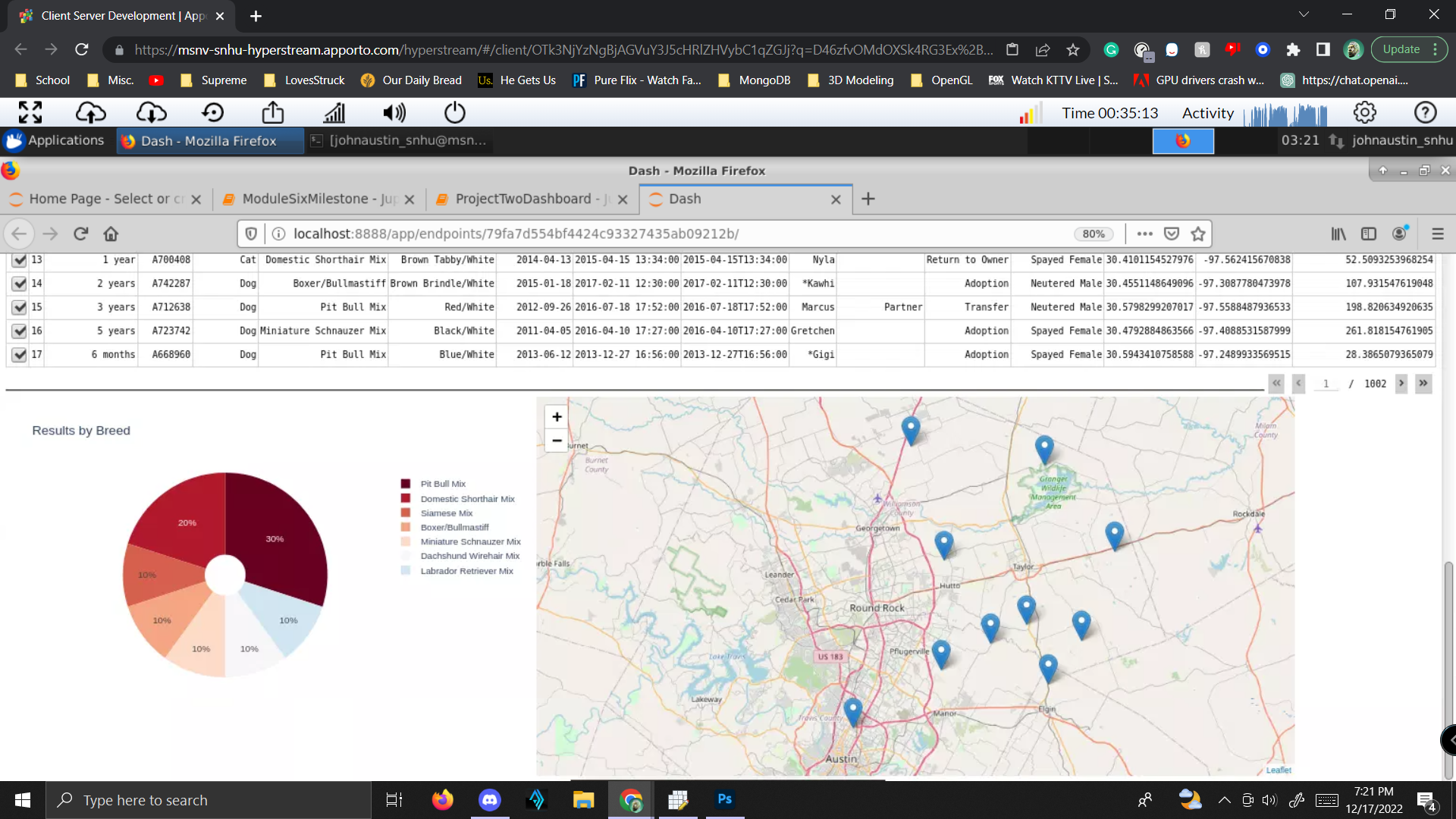
* Disaster or Individual Tracking Rescue





* Clear All (all selected)





## Contact

Your name: John Austin