# CS 340 README NIEBLA

## About the Project/Project Title

The purpose of Project 2 is to create a full stack application with Mongo DB as a data base and leveraging python as a backend and Plotly Dash as a front end. The purpose of the Grazioso Salavare application is to provide a one stop database where users can filter potential rescue animals, visualize data (a bar graph of age), and a map showing where the animals are located.

## Motivation

The motivation behind Project 2 is to metaphorically come full circle. We have worked on mongo, worked on python and in dash. Project 3 puts all of that together to showcase a full stack app. The motivation behind the Grazioso Salavare application is to allow users to access a database of animals that allows filtering and sorting as well as a means to visualize data and map out the location of said animals.

## Getting Started

To get started you will need to set up credentials in the mongodb shell. This is done by accessing the shell as an admin user and setting up a new account. Once the account is set up open Jupytr Notebook and add your new credentials to username and password in the .py project and run.

To run tests in Jupytr Notebook you must add your .py file first then create a .ipynb file. In the ipynb you can write tests to call the python file and get feedback on your operations.

## Installation

Jupytr Notebook, Python, and Mongo DB are required for this project and can be downloaded online.

## Usage

Once a user is logged in and connected they will see a graph with a radio button, a map and a graph. Clicking on the Grazioso Salavare logo on the top of the page takes the user to the homepage [www.snhu.edu](http://www.snhu.edu). The chart by default shows all records, to edit the user can either type a keyword in the chosen column or select one of the pre made filters in the radio button to trigger a query which will update the charts contents. The chart enables you to highlight a column for ease of use as well as give the option to select a row.

When the user selects a row the chart at the bottom left of the screen is updated which highlights the selected row for ease of use. The map also is updated on row selection and displays the tooltip with the animals name.

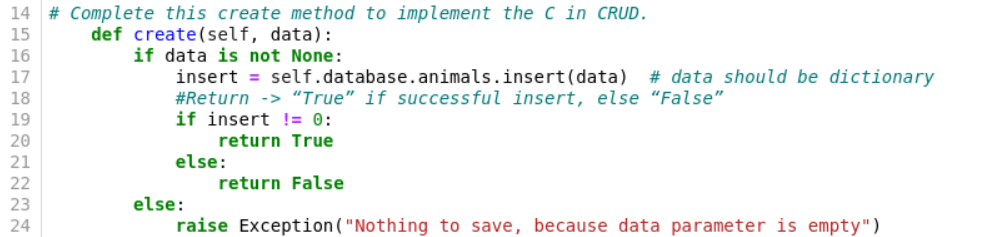
### Code Example

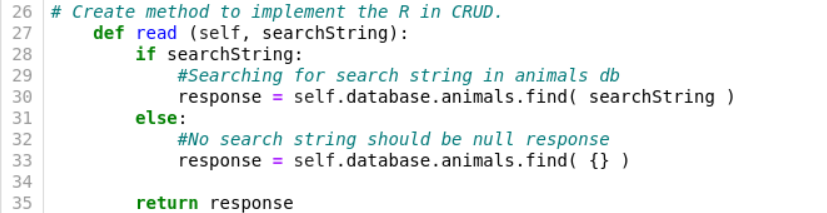
The create method adds data to the mongo db. The read method searches the database by a key value pair. Update finds a specific record by key value pair you provide and updates values on that record (other included parameter). The delete located a record by key value pair and deletes it from the collection. The apply default filter callback is linked to the radio buttons, essentially it triggers a call to the backend to fetch a pre-created query from mongo. The update styles callback enables the highlighting of a selected column. The update map callback updates the map with the contents of the selected row and displays a tooltip with the selected Animals name. Update graph is the graph controller and it takes the 10 records on any given page and displays the animals name and again in a bar graph. The user can then select a row and that row on the graph is highlighted.

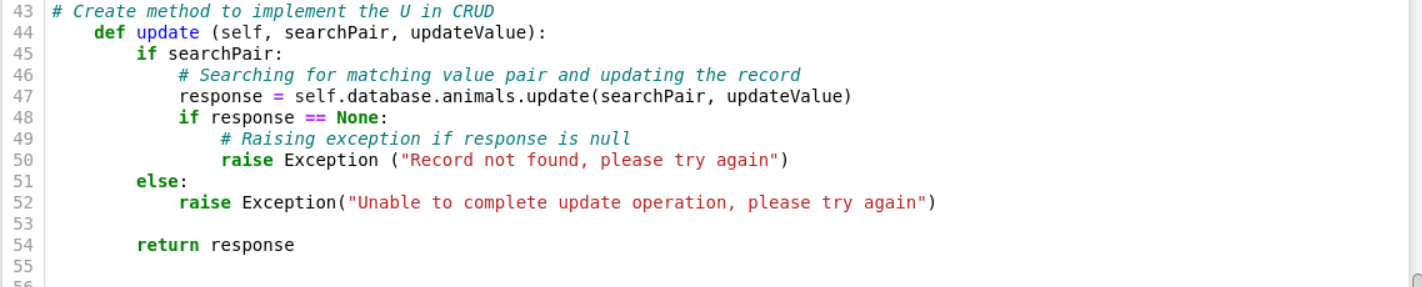
### Tests

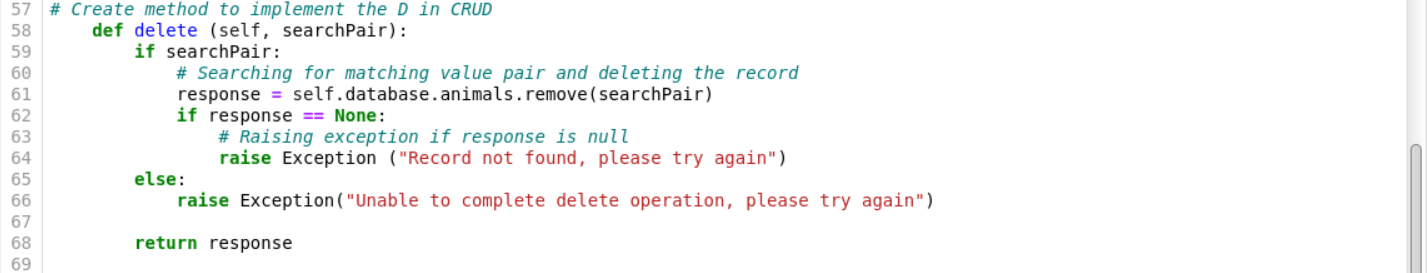
Testing is run through a jupytr notebook .ipynb file. Essentially it creates a testing kernel that we can use call our methods from the .py file to execute CRUD operations

### Screenshots

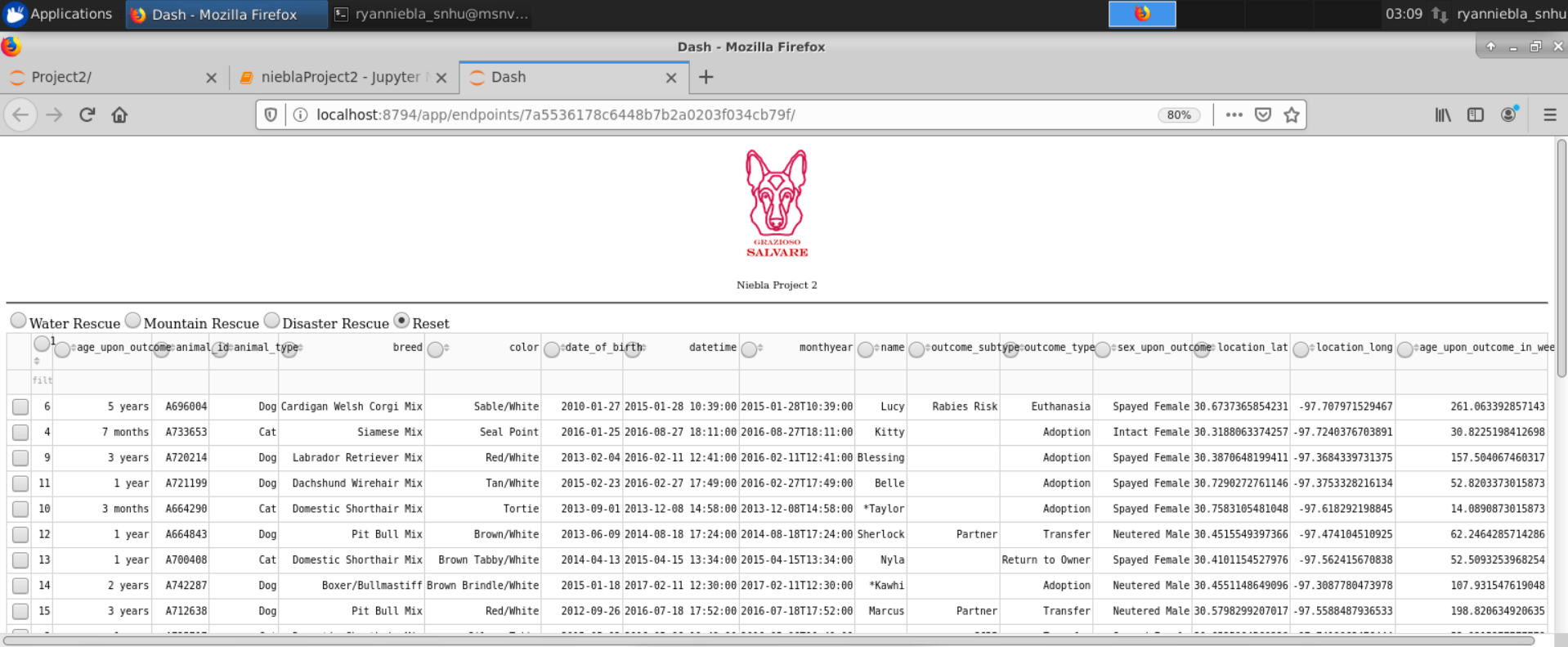
**

**

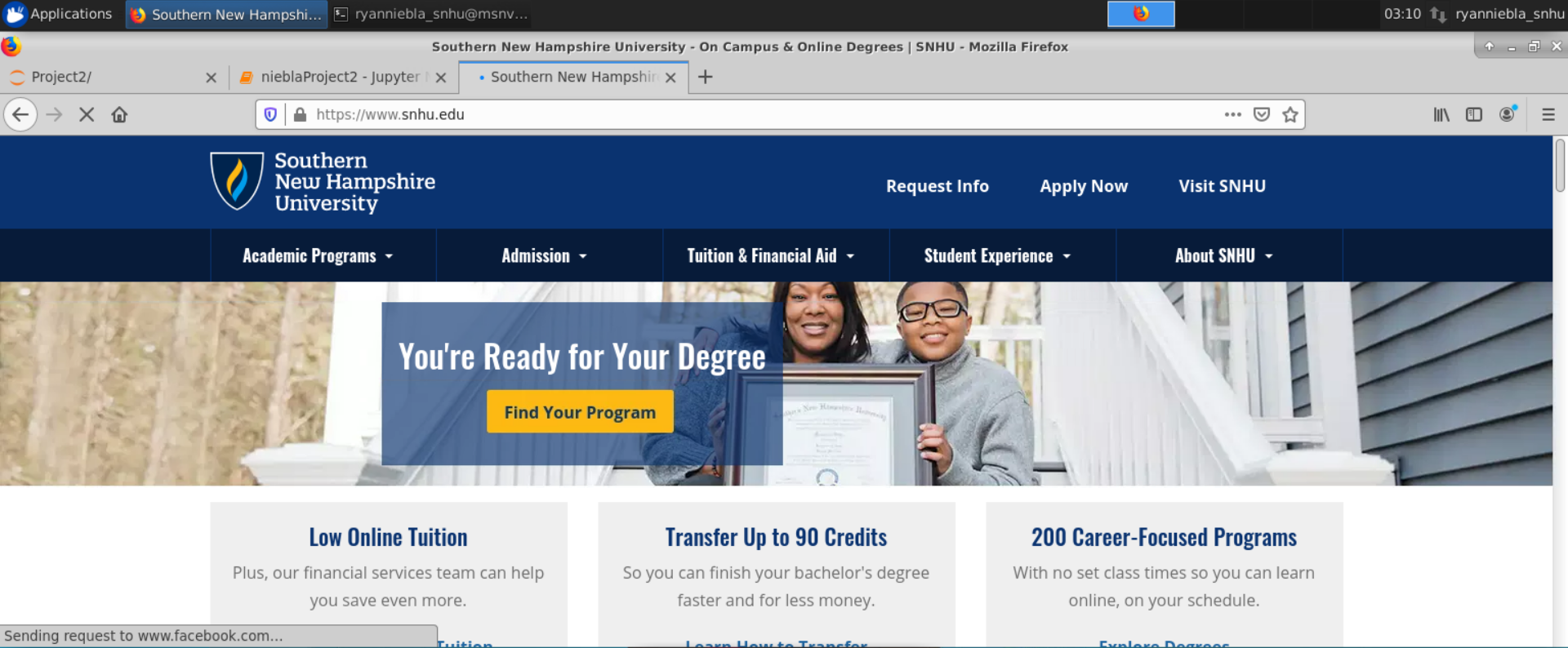
**

**

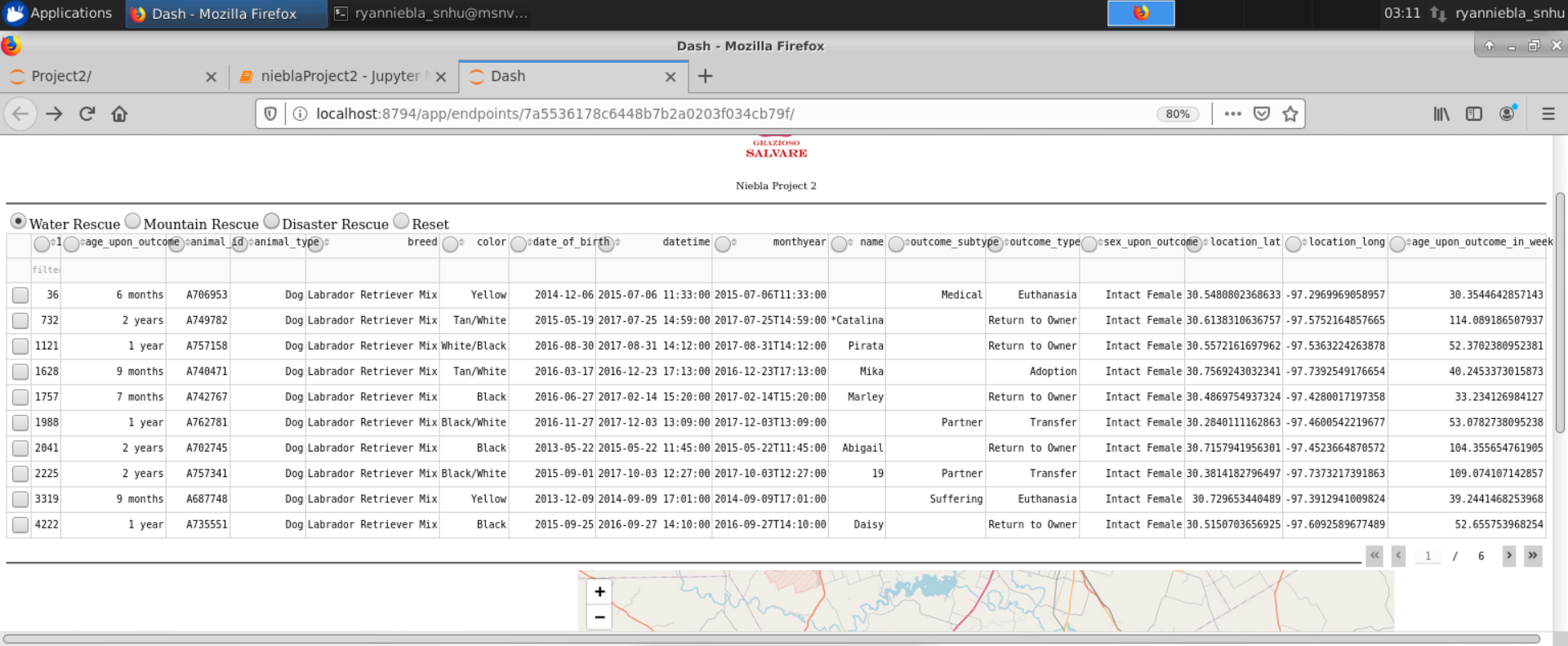
1. Landing page



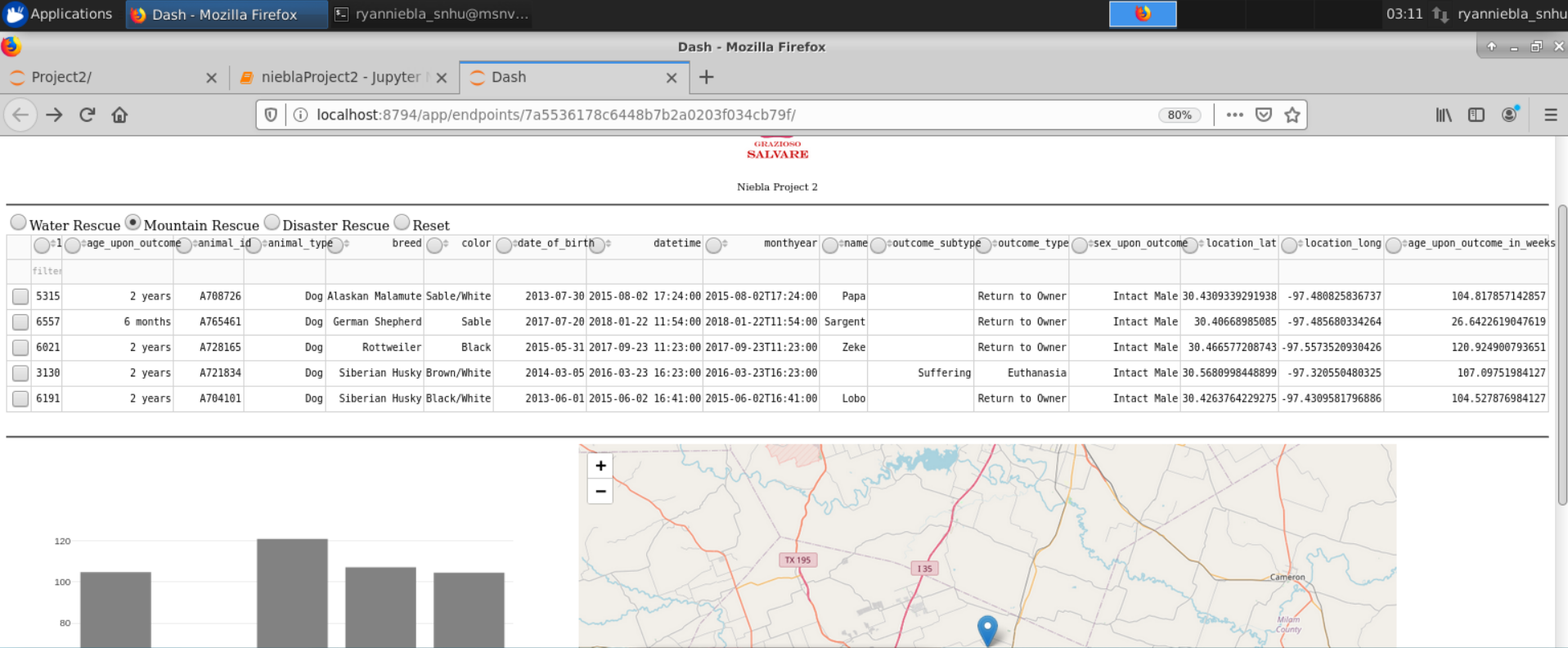
1. Clicking Icon



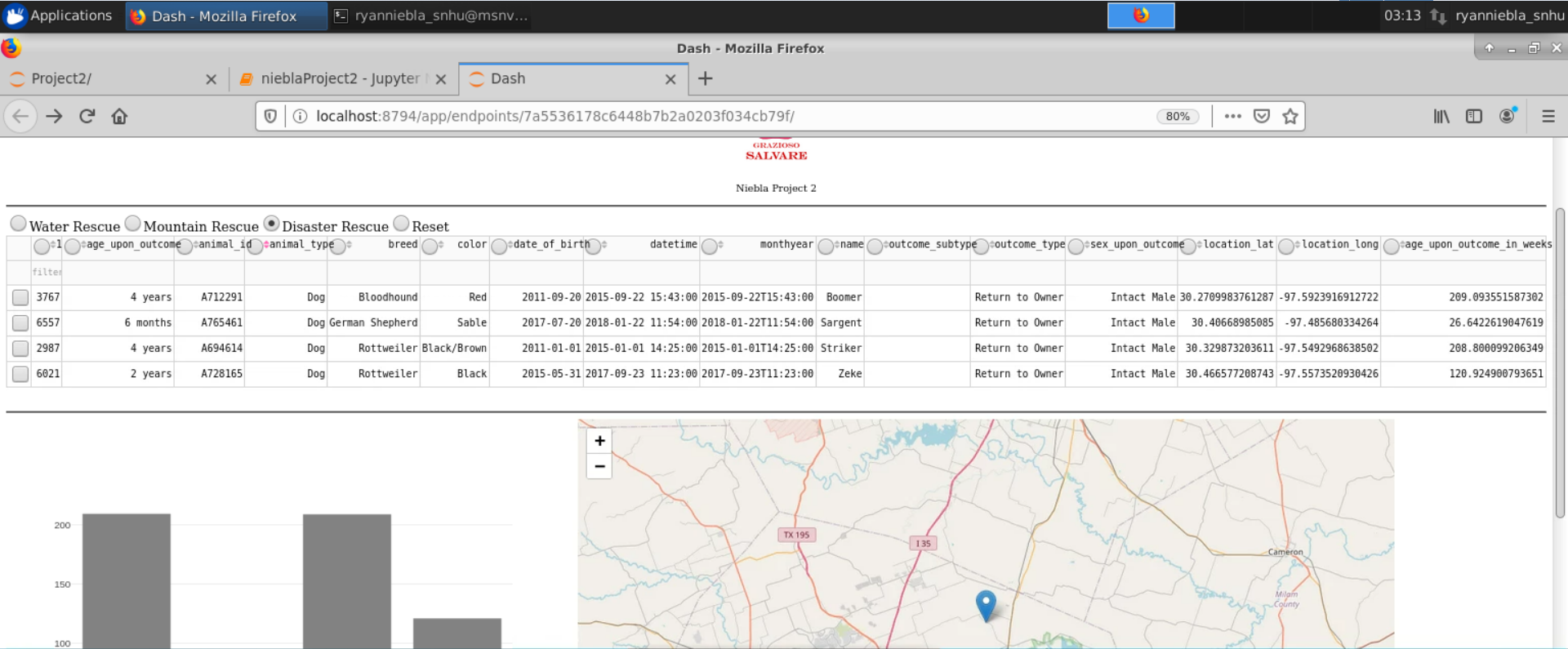
1. Water Rescue Query



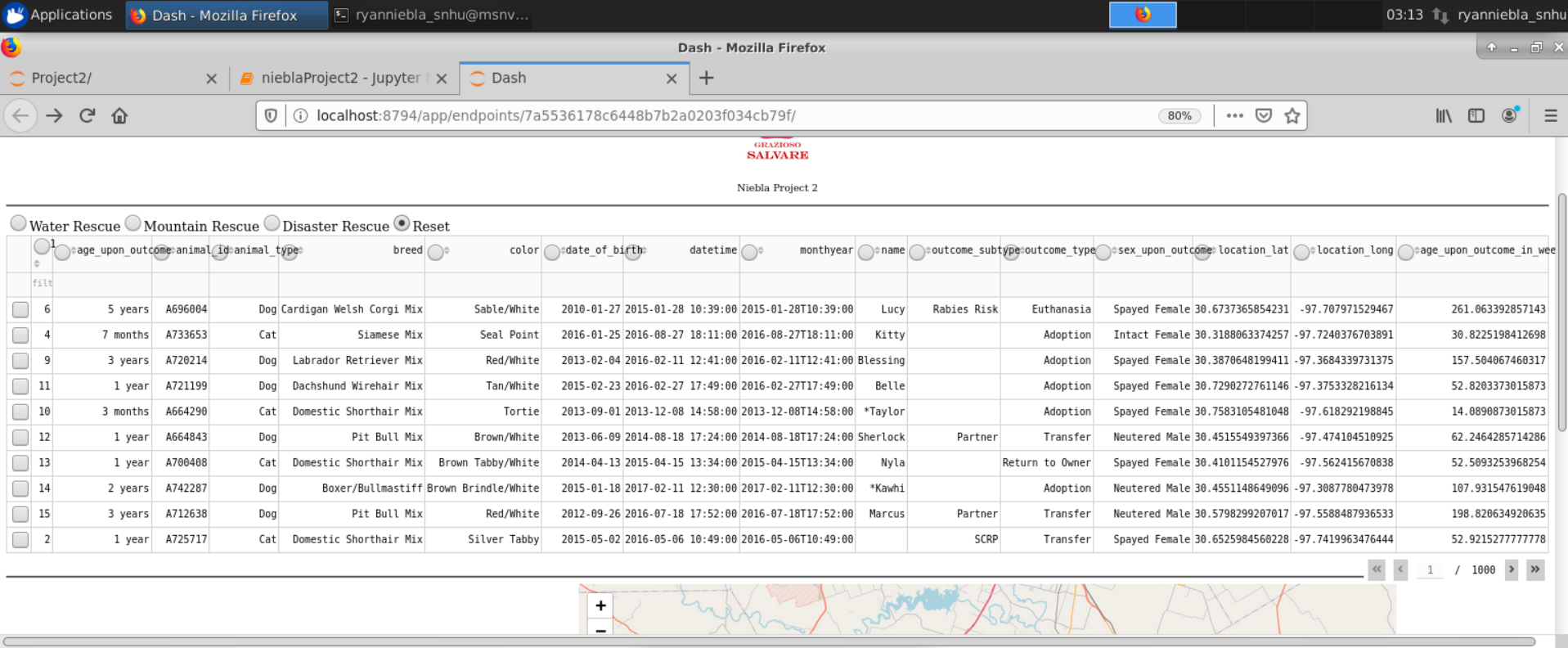
1. Mountain Rescue Query



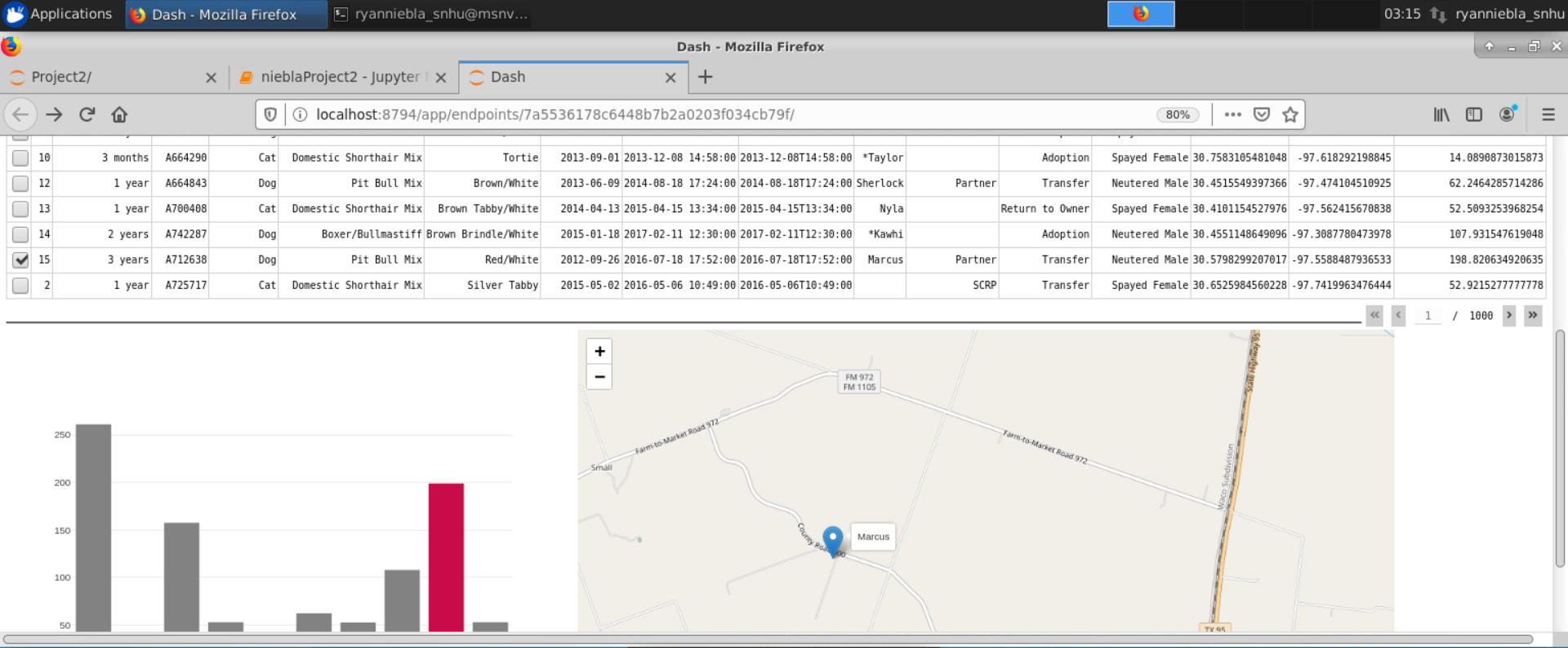
1. Disaster Rescue Query



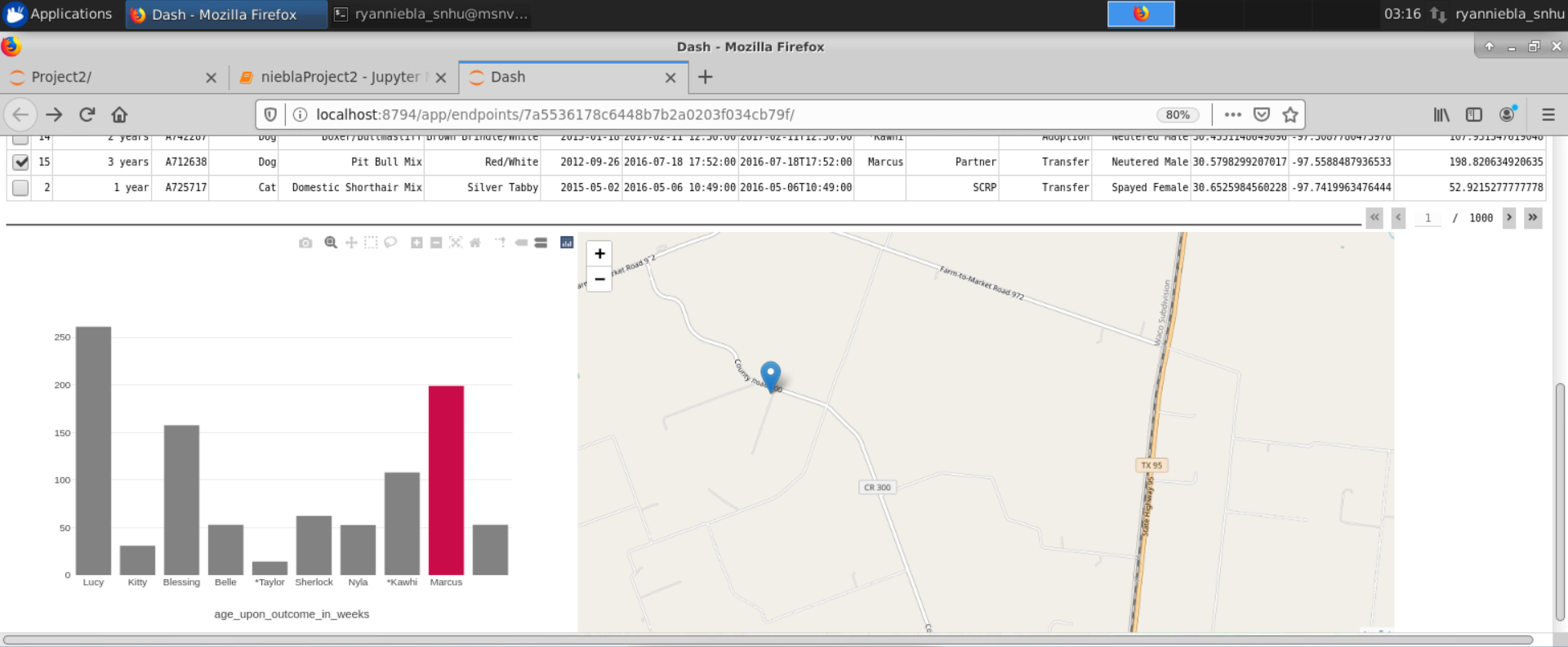
1. Reset Query (all)

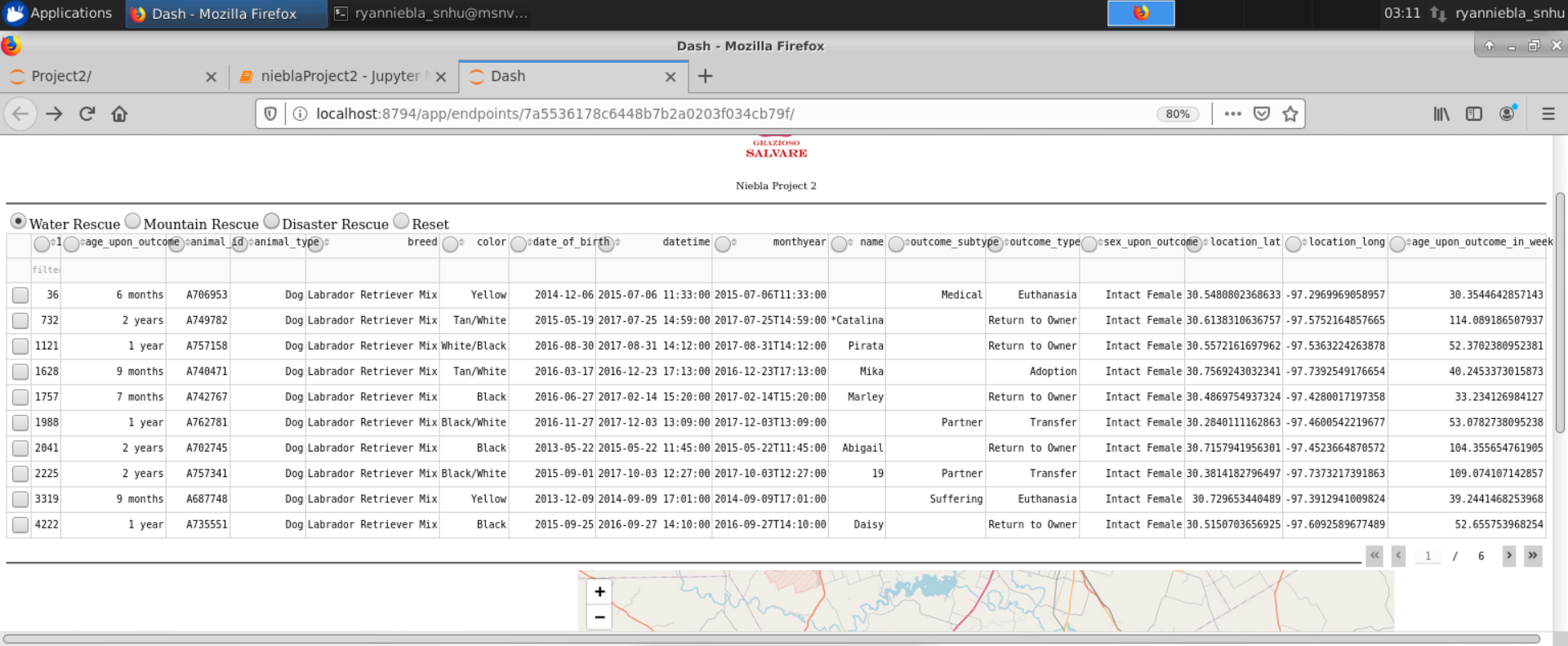


1. Map showing selected row name on tooltip



1. Graph showing the selected row highlighted





**Helpful Links**

Dash: <https://dash.plotly.com/>

Mongodb: <https://www.mongodb.com/docs/manual/>

Pymongo: https://pymongo.readthedocs.io/en/stable/

**Steps**

The project was completed iteratively. I was able to apply a lot of what we learned in previous weeks to the completion of the project.

**Challenges**

By far the biggest challenges for me were the tech stack. Working in Jupytr notebook is somewhat challenging compared to a more traditional IDE, it doesn’t give you prompts, there is not linter, and no real warnings that something is wrong. Furthermore when something inevitably goes wrong, there is not really a way to debug(from what I have seen). Dash also presents a steep learning curve. The html is not standard html so it is challenging to relearn how to write an anchor tag for instance. Finally Python itself. I am not very familiar with Python so this is a needed exercise but it is challenging navigating a new language.

## Contact

Your name: Ryan Niebla [ryan.niebla@snhu.edu]