# Project Two: README File

## About the Project/Project Title

*The interface has been designed for Grazioso Salvare, a rescue-animal training company. Specifically, the code uses Python code and an .ipynb file to access a local database using MongoDB via a dashboard. This dashboard allows a user to filter animals by rescue type, shows the percentage of each bread available for each rescue type, and shows visual map data on an induvial rescue animal once selected.*

## Motivation

*Having trained rescues-animals can be a tremendous asset to communities in your state or country during the aftermath of natural disasters and other devastating events. By having the data accessible via dashboard allows for quick searchable result of available animals, so that they might deployed as quickly as possible.*

## Getting Started

*To create this project you will need:*

* *MongoDB*
* *Jupyter Notebook (for .py and IPYNB files)*
* *Import all the necessary dependencies into your IPYNB file*
* *Optional: Python IDE*

*Once you have download what you need, you can import the code into your IPYNB file using the below line:*

*From AAC\_CRUD import AnimalShelter*

*\*\*\*Note: Although you can use any Python IDE to create you python script, you will need to upload the .py and IPYNB file into Jupyter Notebook to access MongoDB.\*\*\**

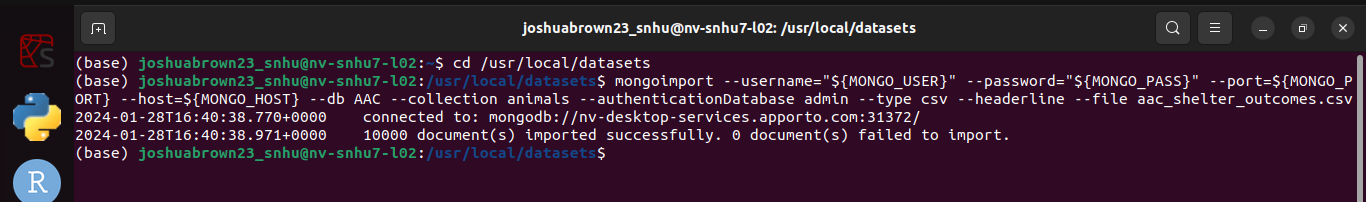
## Installation

*Here are the minimum requirements to use this software:*

* *Processor: Intel 8th Gen i5 or i7*
* *Memory (Ram): 8 GB RAM*
* *Hard Drive: 250+ GB*
* *Graphics Card: Built-in (Intel)*

## Usage

*First, you should import your desired database into MongoDB and log in using the administrator account for your version of MongoDB. An example screenshot is provided below:*

* **

*Next, use the admin account that you logged in with to access the “users” database and create another user account, and password combination to access later on; this account should have readWrite permissions on your imported database. This will be the account that will be used to access the database through a Python script later on, so be sure to remember your credentials. Once you are done, restart MongoDB, and log in to the user account to make sure you have access. A screenshot shows an example of this below:*

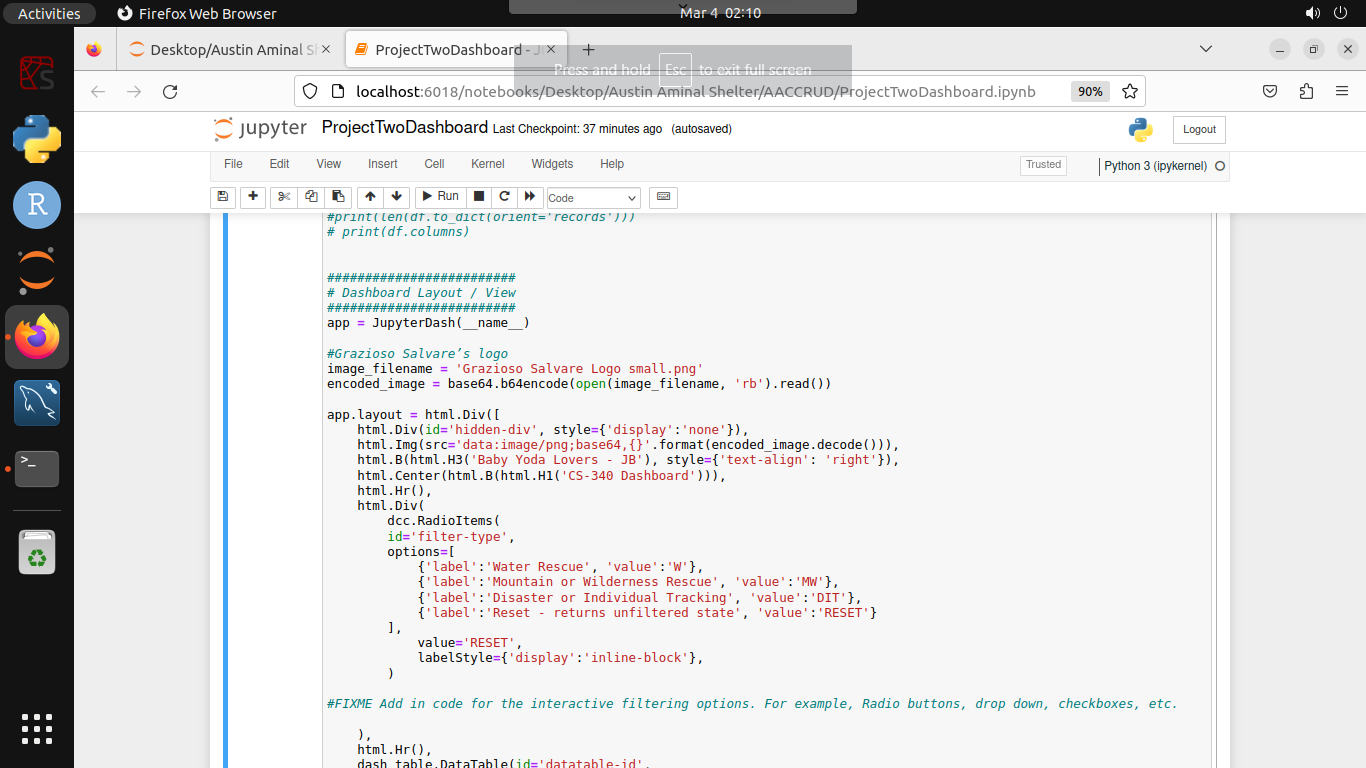
* *A screenshot of a computer screen

  Description automatically generated*

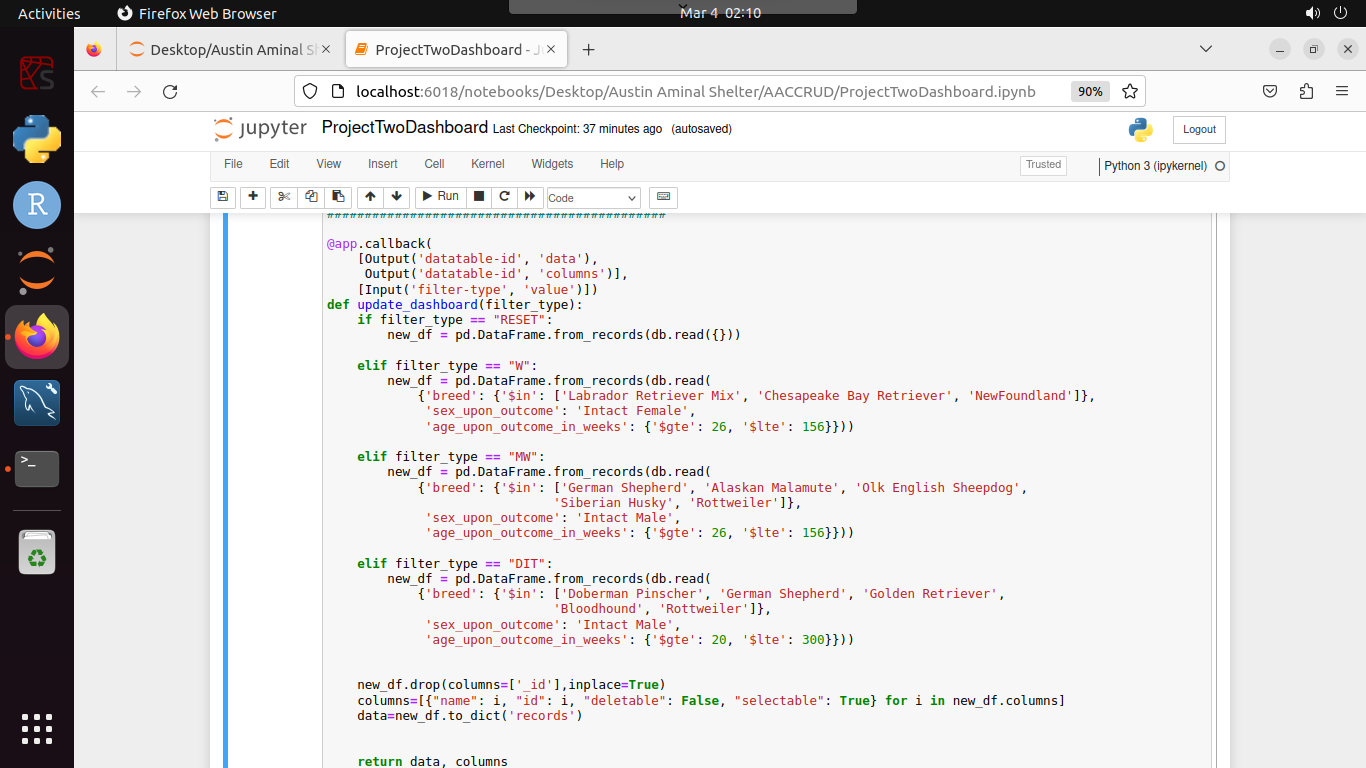
*Next, if you haven’t already, create an IPYNB file and import Animal Shelter from AAC\_CRUD. Afterwards import all the required dependencies and initiate an instance of AnimalShelter using the username and password combination that you created for your user credentials in MongoDB. Screenshot below:*

### 

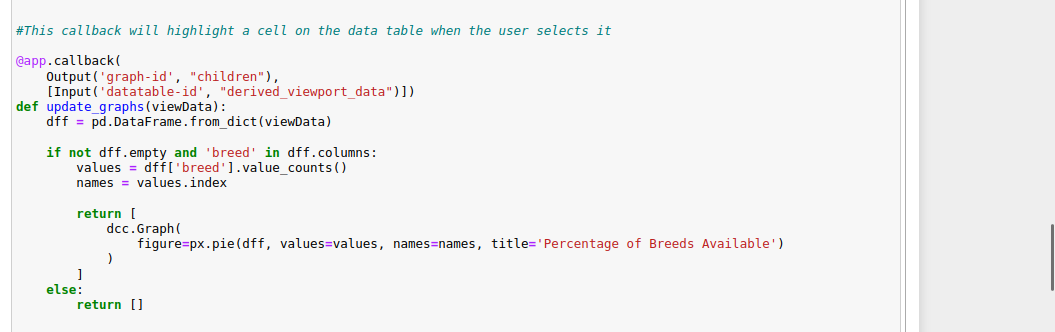
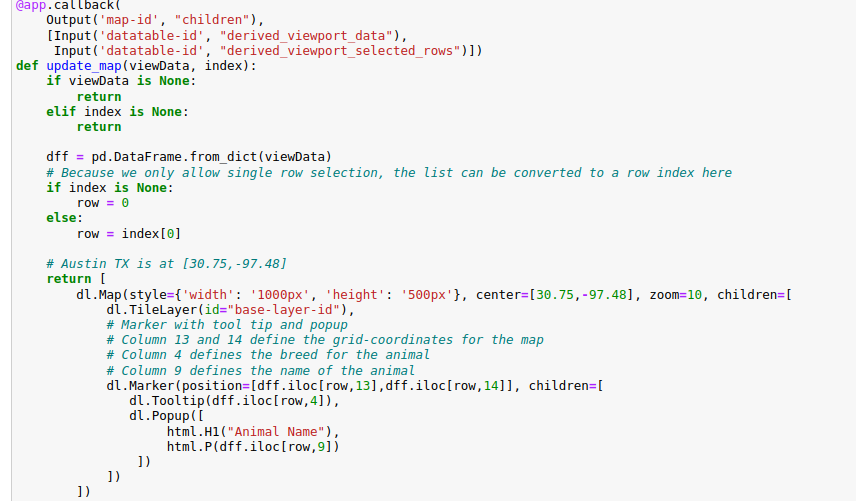
*When setting up the dashboard layout, you should put in option using html (radio, dropdown, etc.). For the example below I chose to go with radio buttons setting their id to filter-type. Be sure to label your buttons and assign value to each label. These value will be used in the next step to set filters for your data. An example is below:*

* 

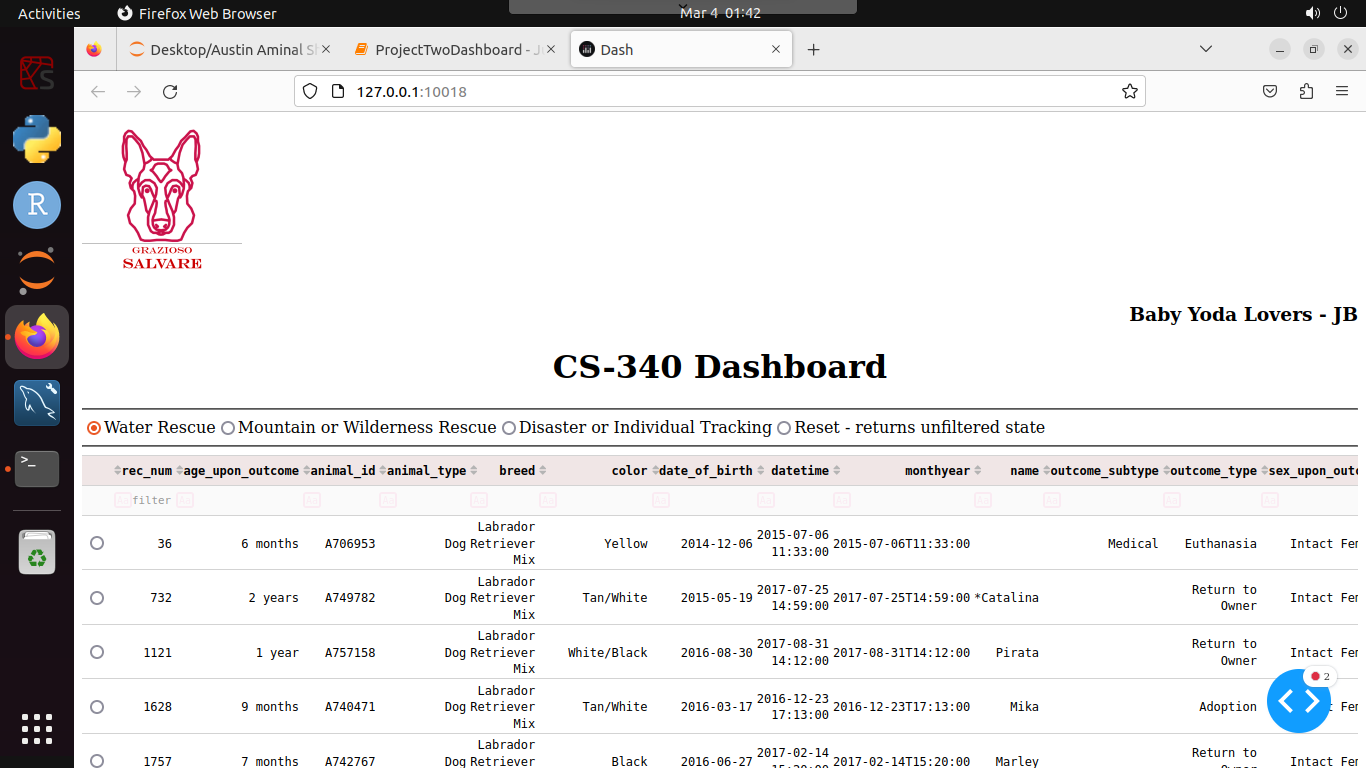
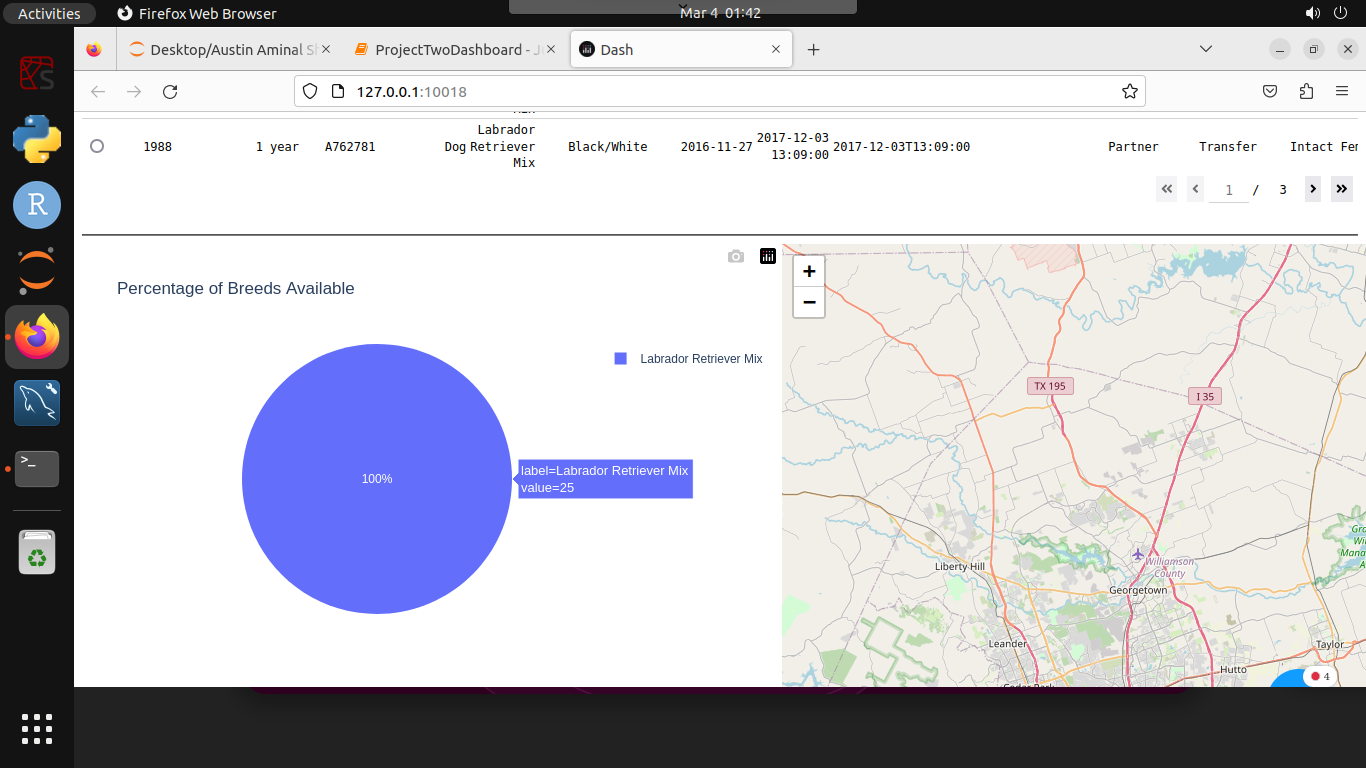
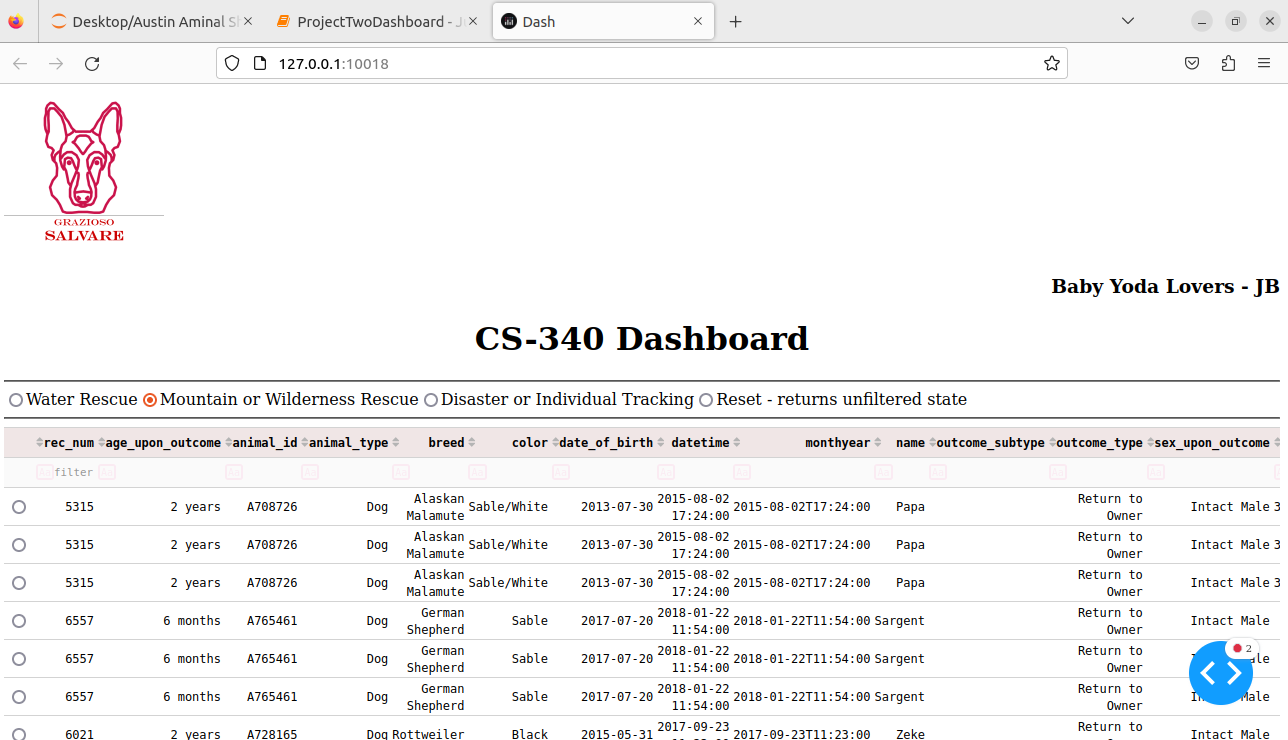
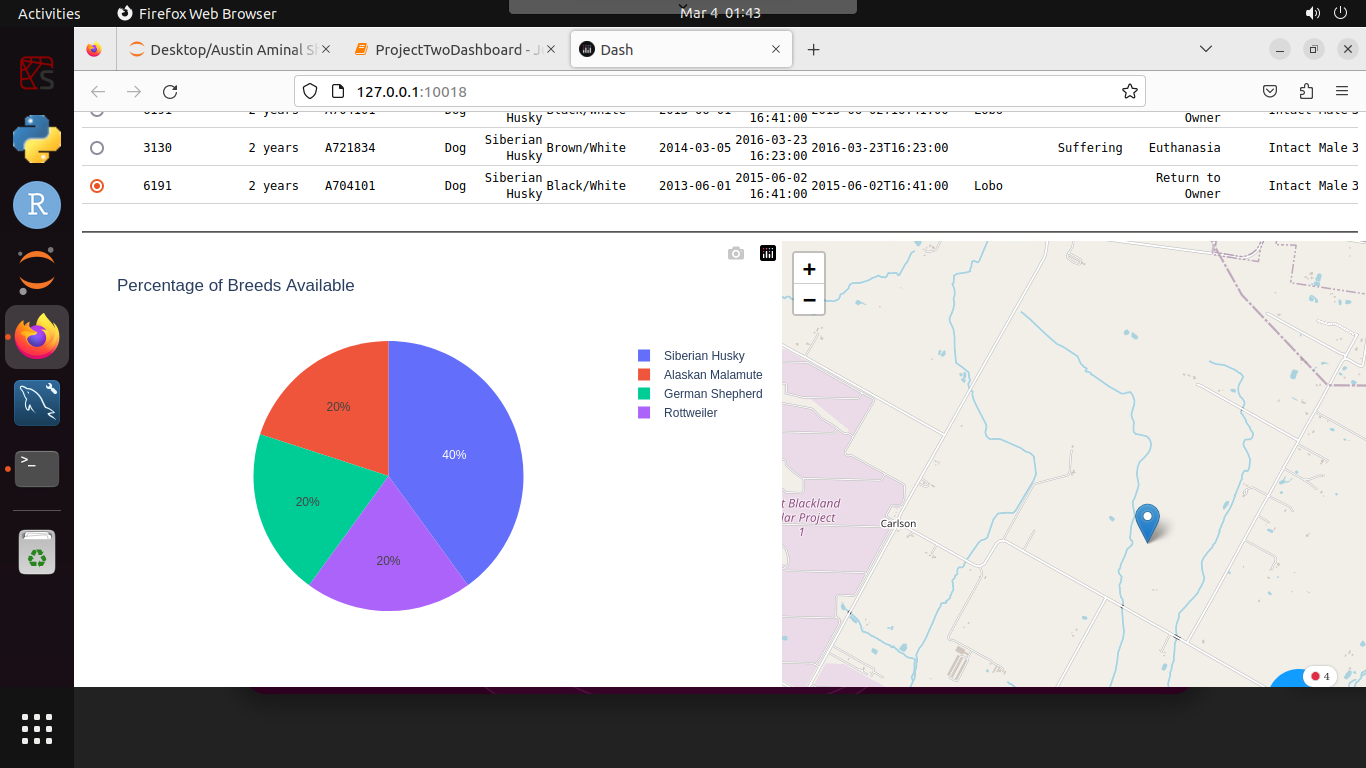
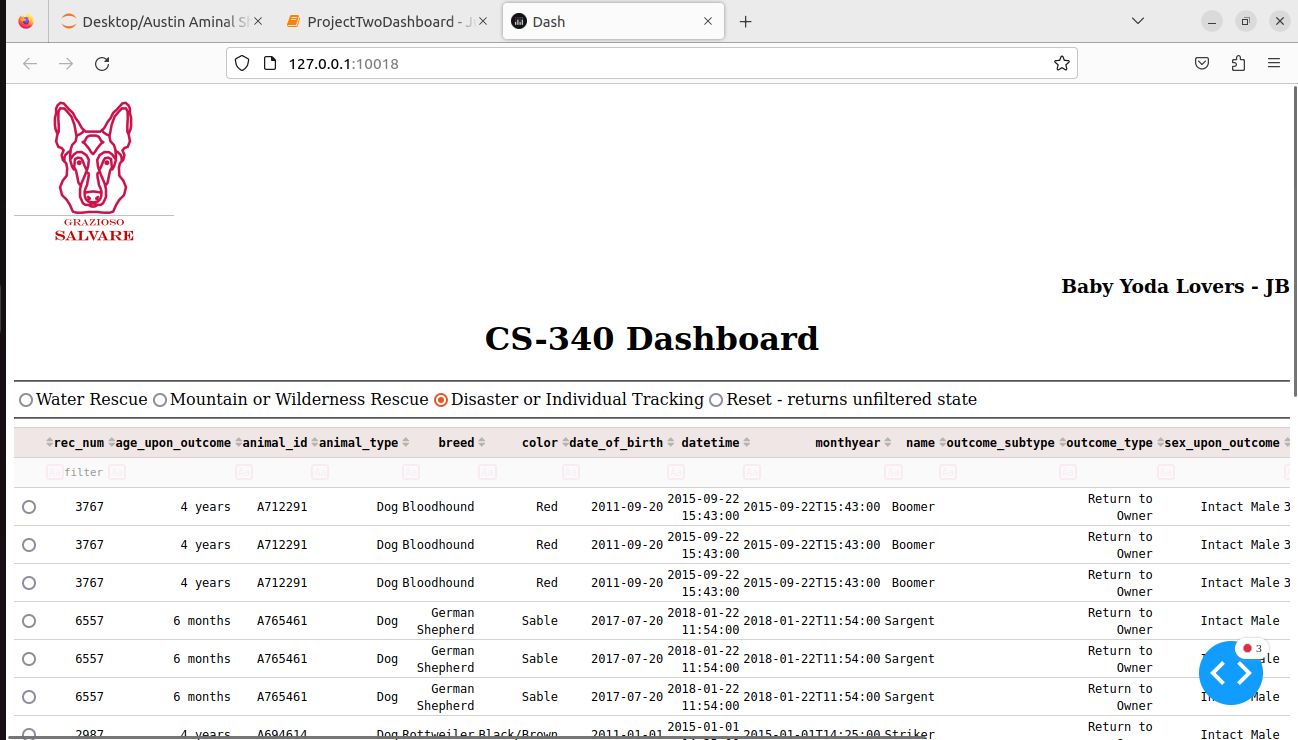
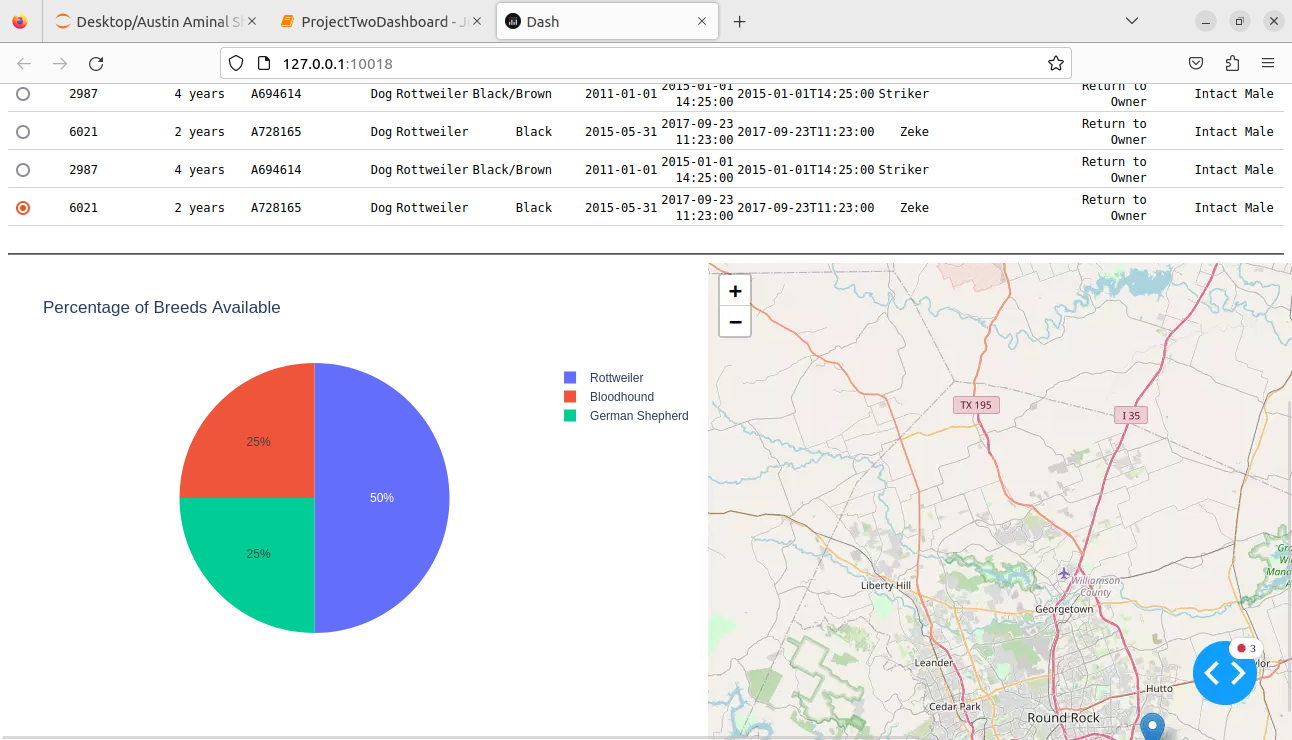
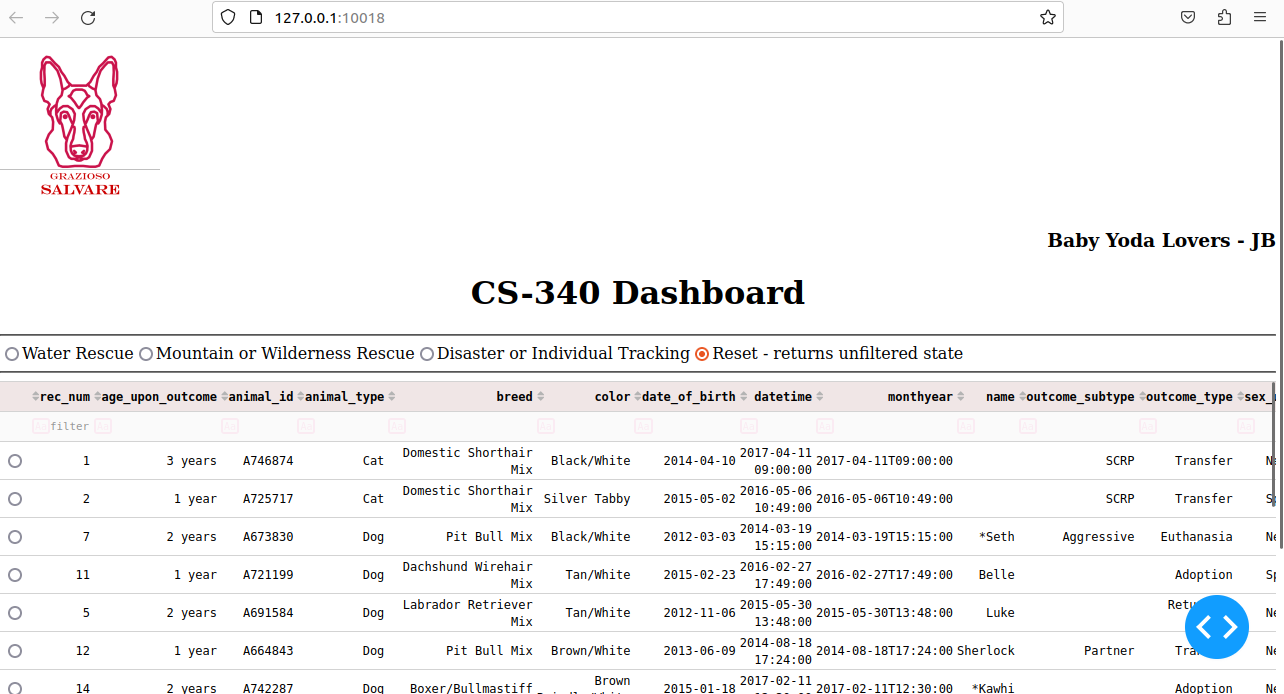
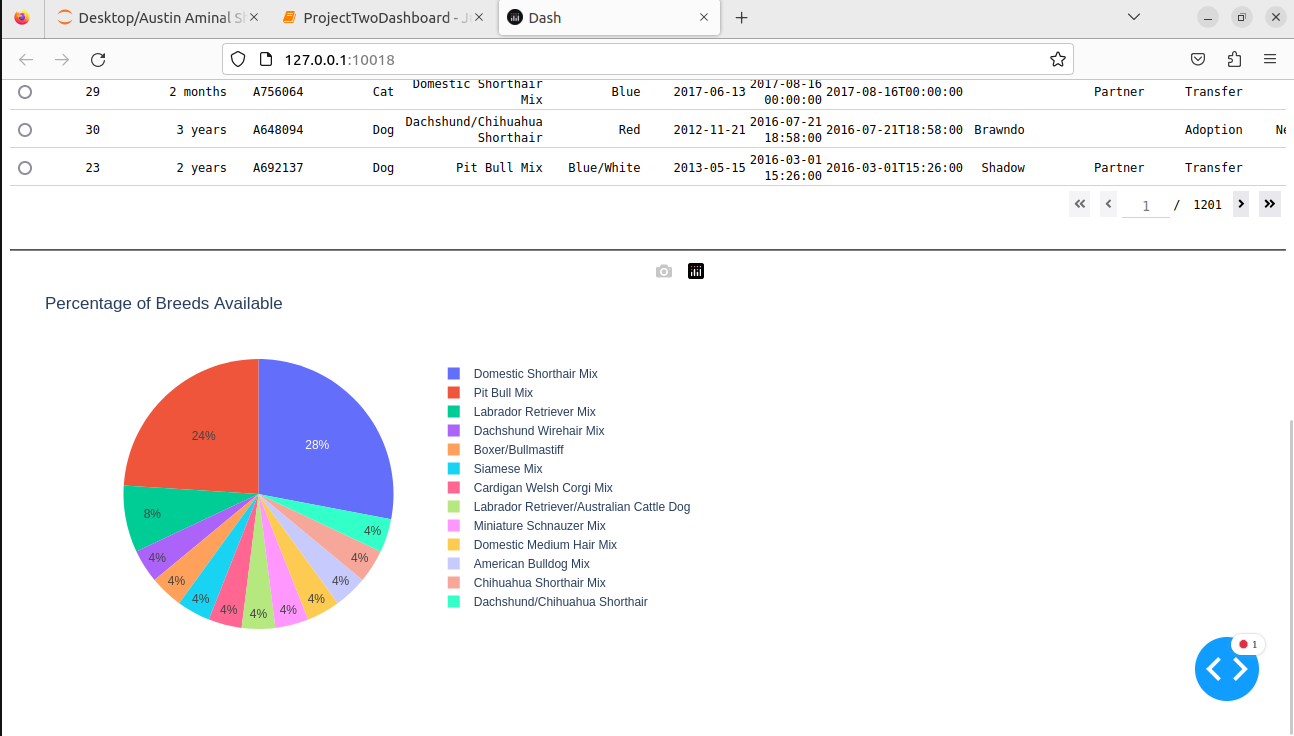
*Once your filtering option are in place, you can create a method to update your dashboard. Using the filter-type as input and the value that we created in the previous step we will output the data and columns. For the function, will accept the filter type as a parameter, and use that to create conditional if statements for each branch making a read for each one to fit rescue type. For my example I set the default value of the dashboard to the RESET – unfiltered value. An example is below:*

* **

*From there you can create an update graph to form visuals of your filtered results, and an update map function to show a geolocation chart for a selected animals location. Screenshot of both of these functions are below:*

* *Update Graph:*
  + **
* *Update Maps:*
  + **

*The result should be a datatable showing your data from MongoDB on your dashboard, with radio buttons that allow you to filter through your data. You can find screenshots of the filtered database along with an example of visuals for each selection below:*

* *Water Rescue:*
  + **
  + **
* *Mountain or Wilderness Rescue*
  + **
  + **
* *Disaster or Individual Tracking:*
  + **
  + **
* *Reset:*
  + **
  + **

**Contact**

Your name: Joshua Brown