**Grazioso Salvare Search and Rescue**

The program presents the user a list of potential search and rescue animals currently housed in shelters near the Austin TX area.

**Installation**

Include the included mongodb dataset along with the included python file to run the Jupyter file to launch the application. Launch mongo db to give the Jupyter file access to the needed records.

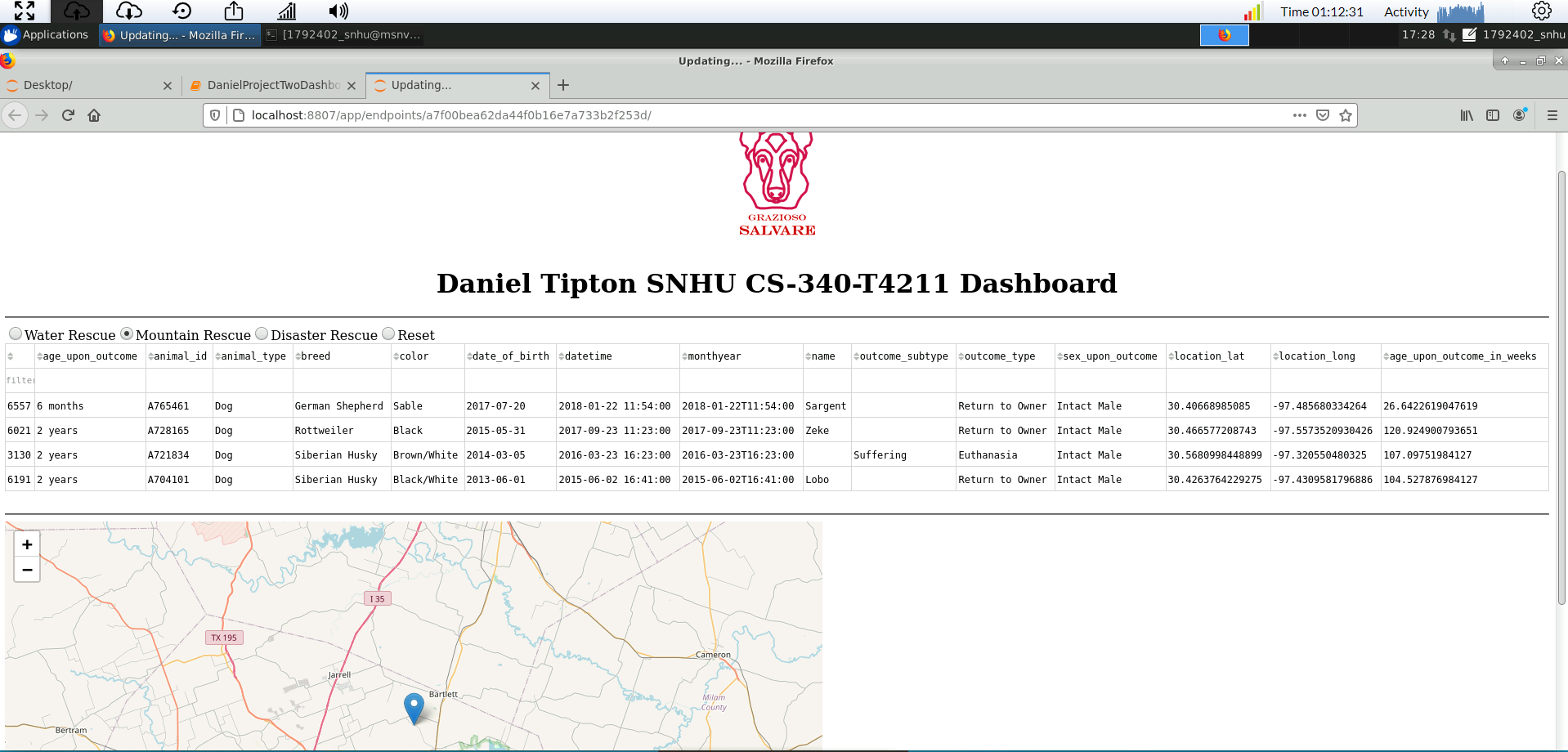
**Usage**

The program presents the users at first with a full view of all animals currently available.

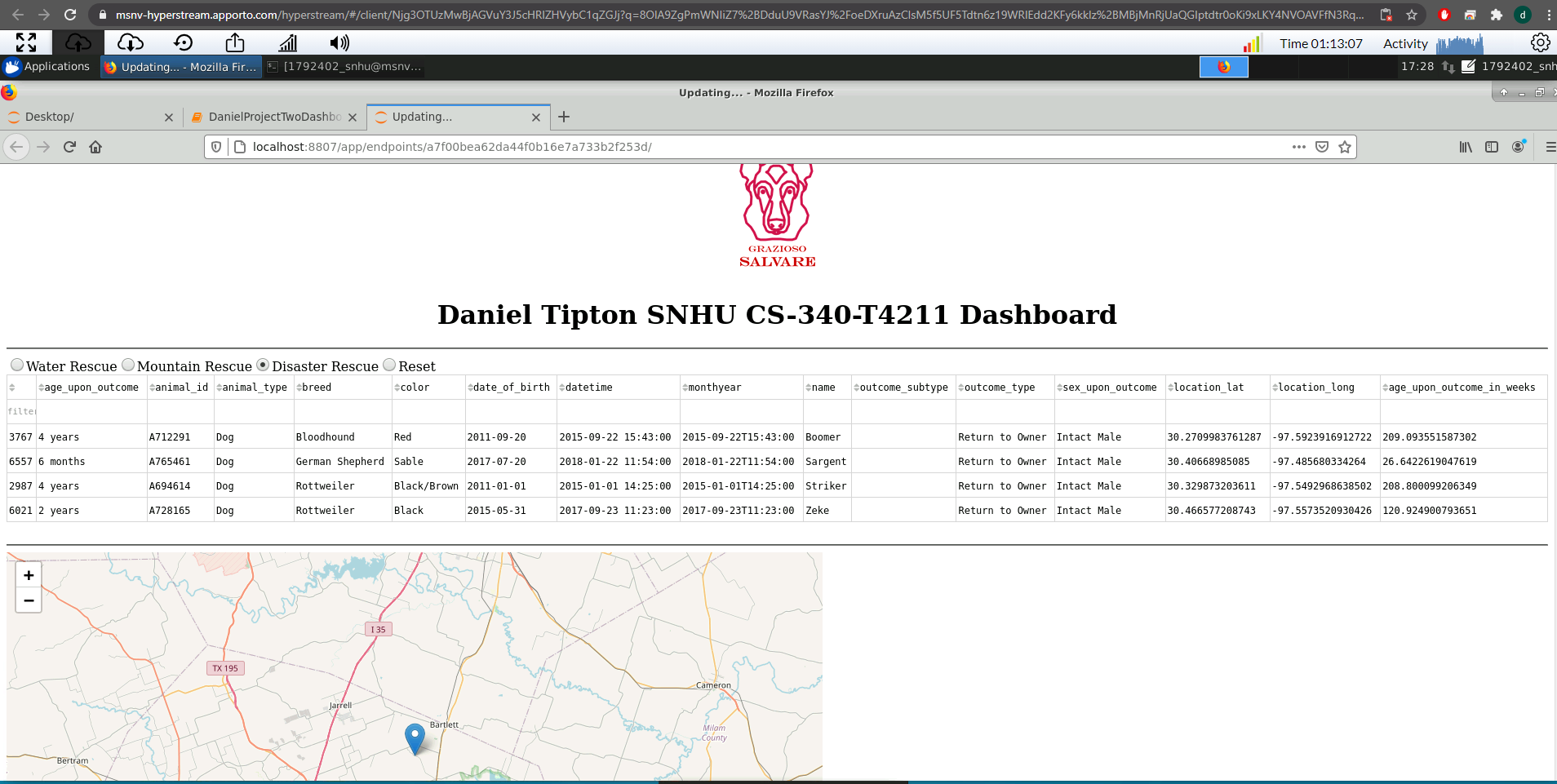
Graphical user interface, text

Description automatically generated

As seen in the screenshot above there are well over 1000 records , so to help better filter and allow users to figure out there needs faster we have added filtering options that greatly reduce the time needed to look.



By selecting the different options it lets you get a better idea of the variety available.



Graphical user interface

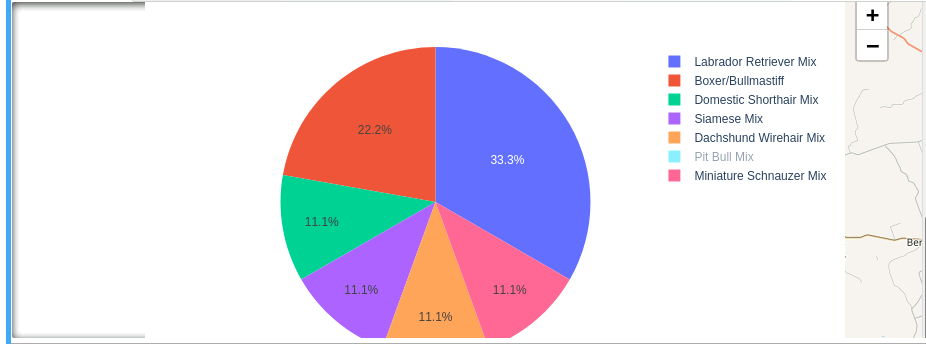
Description automatically generated

And finally we added a reset to take you back to the start.

Graphical user interface, text, application, email

Description automatically generated

By selecting any of the options it allows you to see a map of where the shelter is located along with a breakdown of the different breeds of the shelter in the form of a pie graph



**Tech Used**

The tech used to build out this project were as follows: Frontend was built with the Dash framework, the backend was built with python and a mongodb database. We went with Dash as it allows us to quickly build out data components needed to show off the data in both a column graph along with pie graphs or the geo map we included. We went a python backed as it connects nicely with both dash and with mongodb. The easy human readability will make any updates easier to implement. Finally we used mongodb to store the data. We went with mongodb because it offers several benefits. Being schema less we can better load in information on a variety of dog and animals breeds without having to have it all. It also has a deep query-ability which allows us to speed up processing on the most used searches, and finally it’s easy to scale as the database needs to grow we don’t have as much to worry about as compared to a more traditional rational db.

**Challenges**

A big challenge on this project was finding the right way to display the data. Going in I had the idea that this would be the easy part and actually implementing a CRUD would be much more difficult. Figuring out how to render out the tables and the map were quite the challenge, but thanks to the variety of documentation given along with the dash documentation and python head first I was able to get this to render out.