**Name**:

Grazioso Salvare’s Animal Rescue Dashboard

**Description:**

This Interactive Web Application Dashboard provides emergency personnel a simpler way of finding rescue trained animals by giving a dynamic view of Austin Animal Control’s data base. Emergency personnel can quickly find available trained rescue dogs to help track someone in need.

**Tools:**

* MongoDB is an important aspect of this application , because it stores the AAC’s database of animals and the application uses this data to create the dash-table, chart and map location.
* Python was used to rite the code.
* Dash framework was use to provide the visual aspect of the dash-table, charts and maps. These can be update. The bar graph in this application tracks status for each animal, which can be change to age.

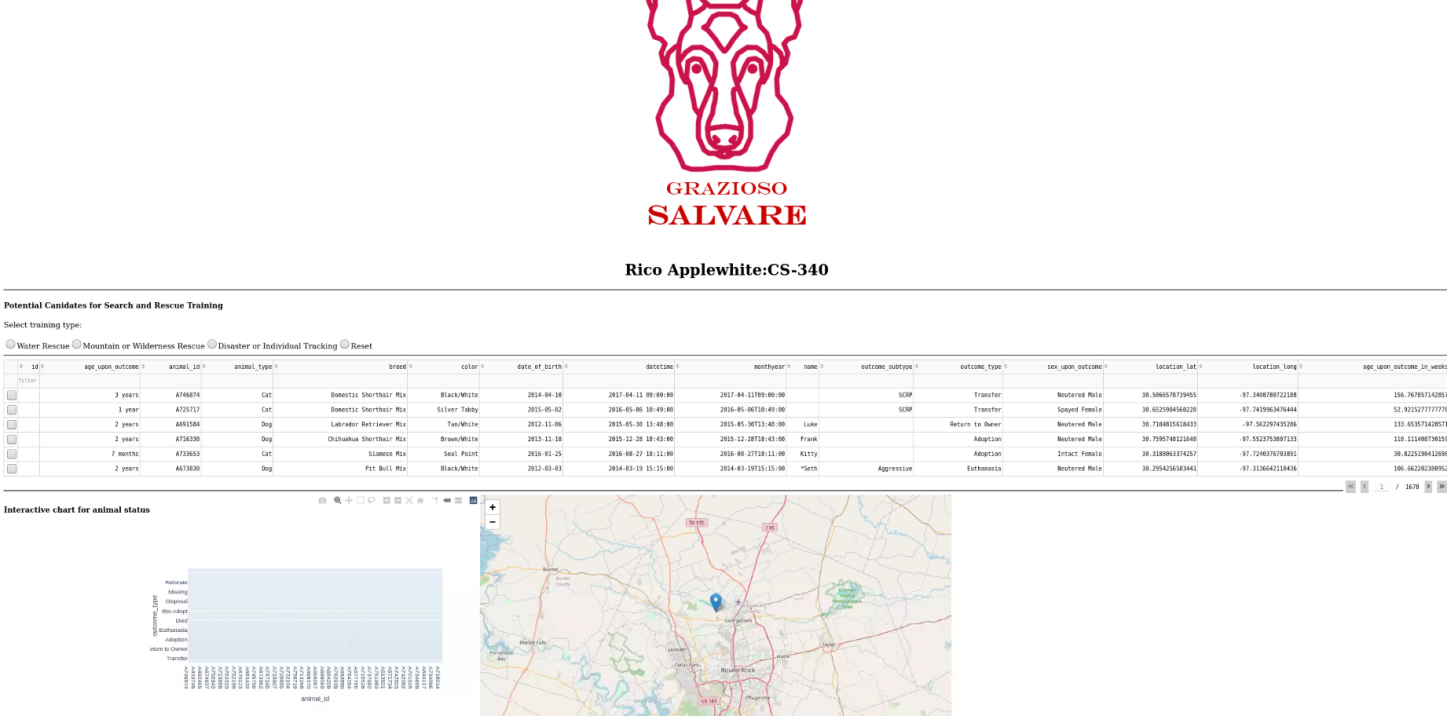
For more graph options or other information about the Dash framework:

https://dash.plotly.com/dash-core-components/graph

## Functionality:

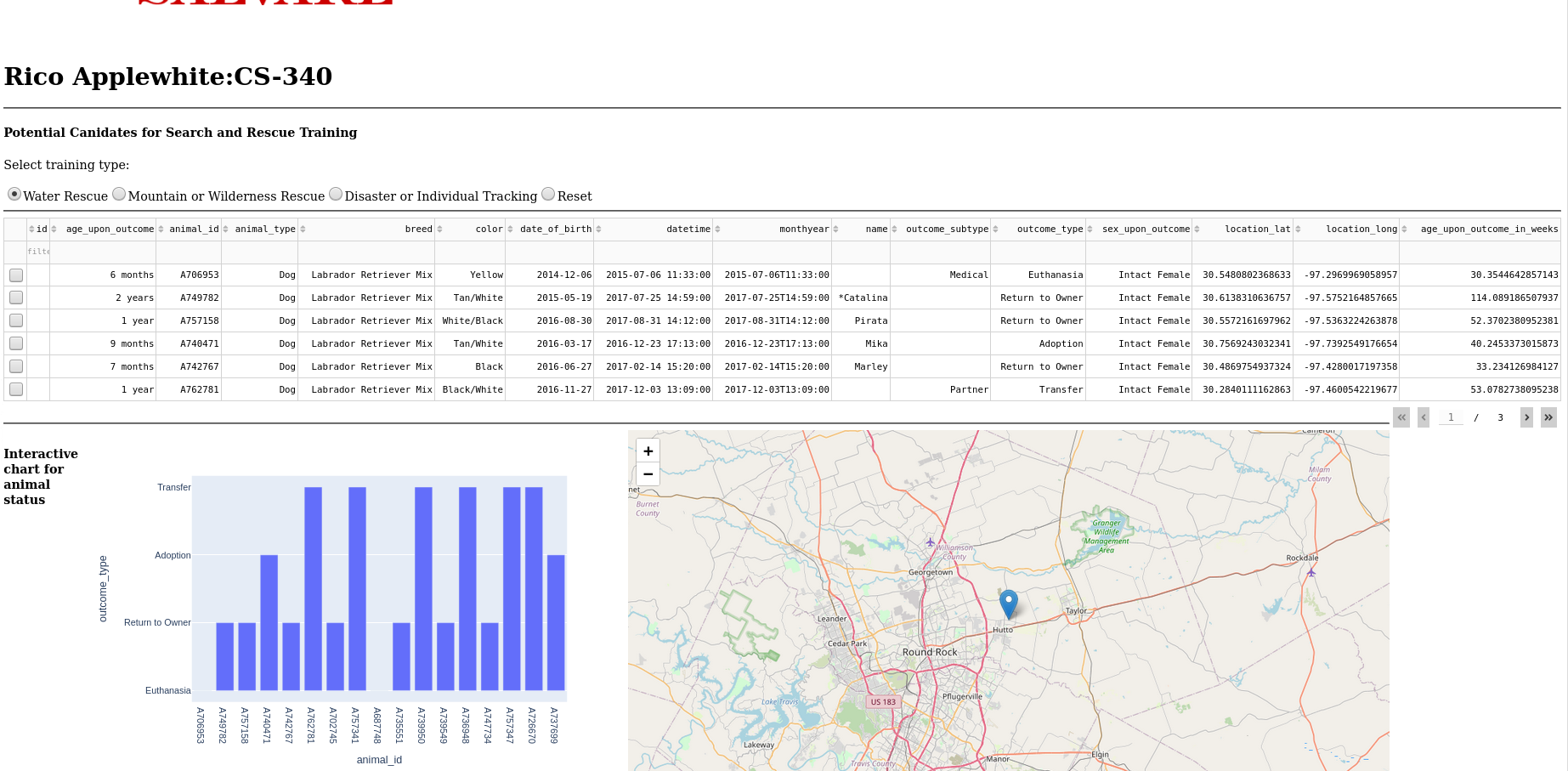
Note: Authorization to AAC’s database I need before you can access this Web Application.

When accessing the Web Application you are greeted to Grazioso Salvare’s log, as well as a dashboard that contain data from AAC’s dashboard, a bar chart that show the status of the database of animals and a map which provides their location. From here you are able to interact with the data table0 by selecting one of the radio item



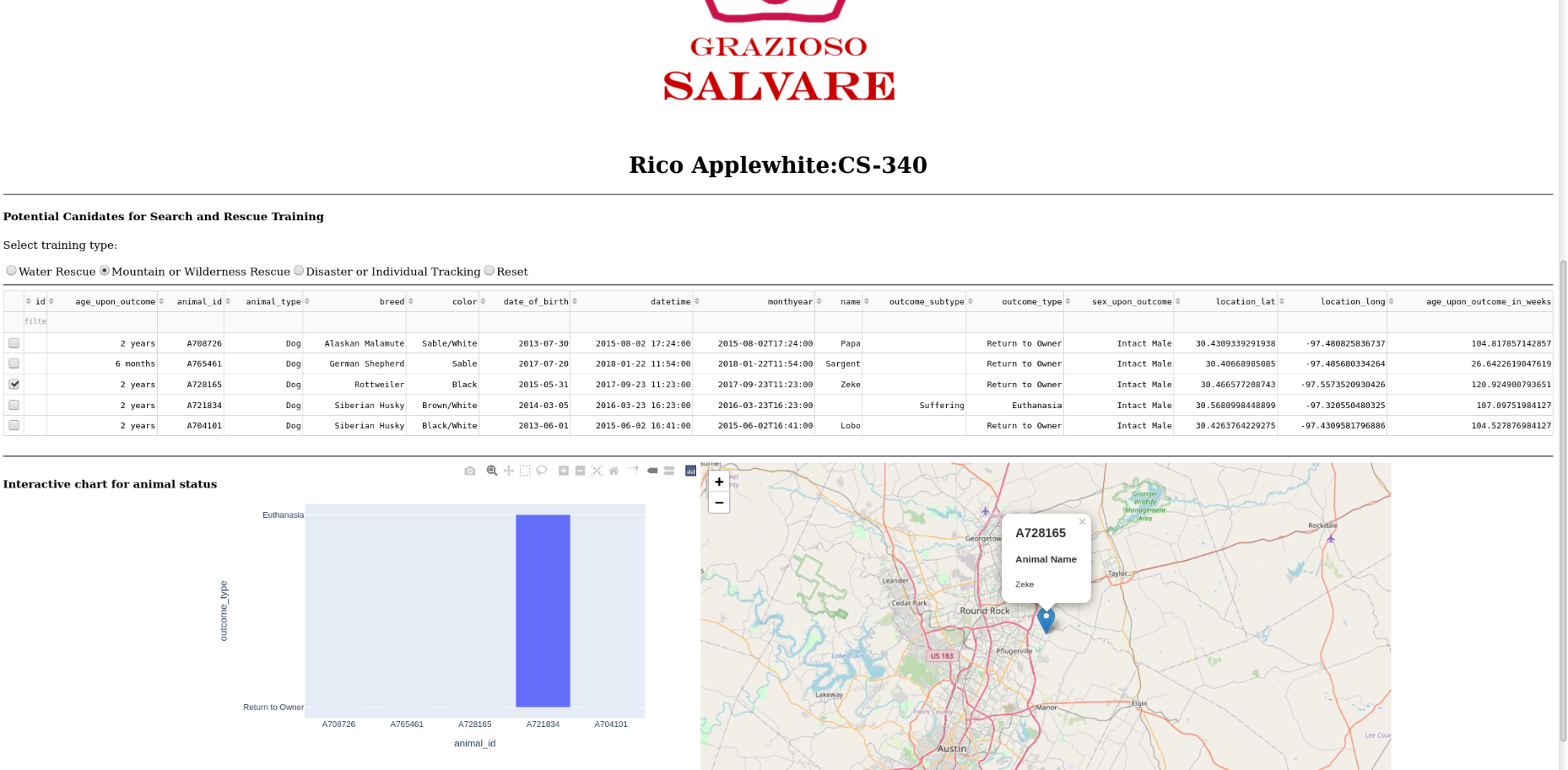
* Water Rescue

By selected Water Rescue the database search through hundreds of animals to narrow down and find those that fit the require criteria. Compare the page number of over 1600 to the trim number of 3 after searching fo Water Rescue animals. The Bar provide a quick overview of the animal’s status so you can see which animals are availble for service.

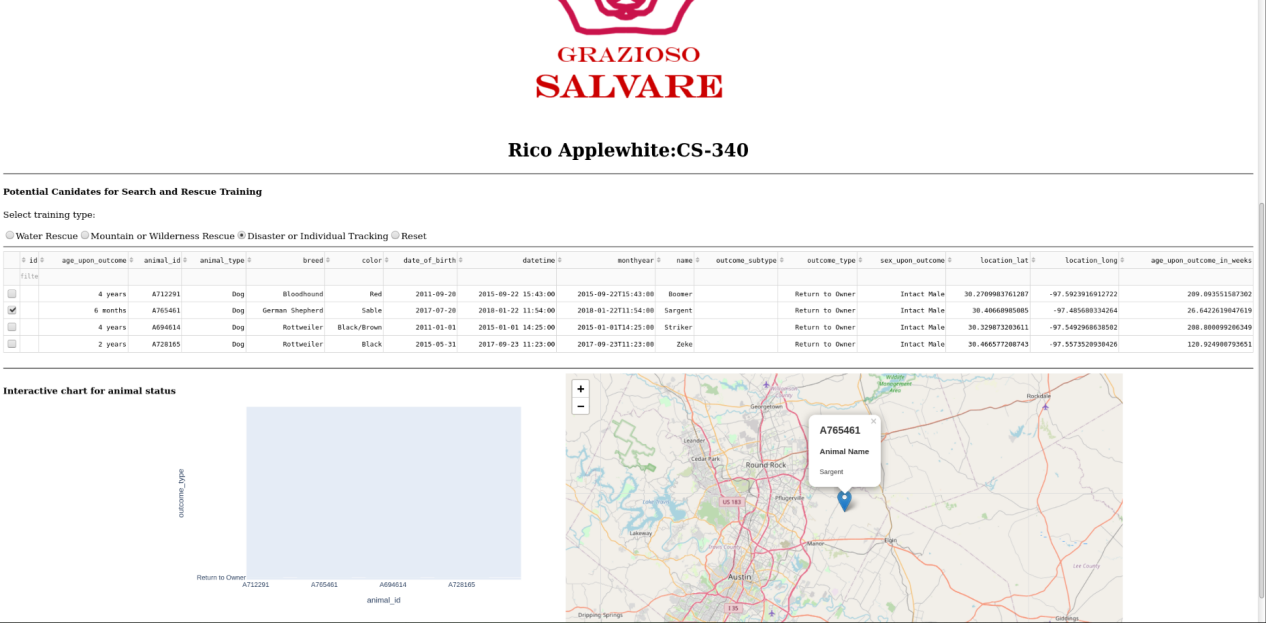


* Mountain or Wilderness Rescue

Some searches may return a small number of results, but because the dashboard is connected to AAC database, it’s always up to date and more animals are add often.

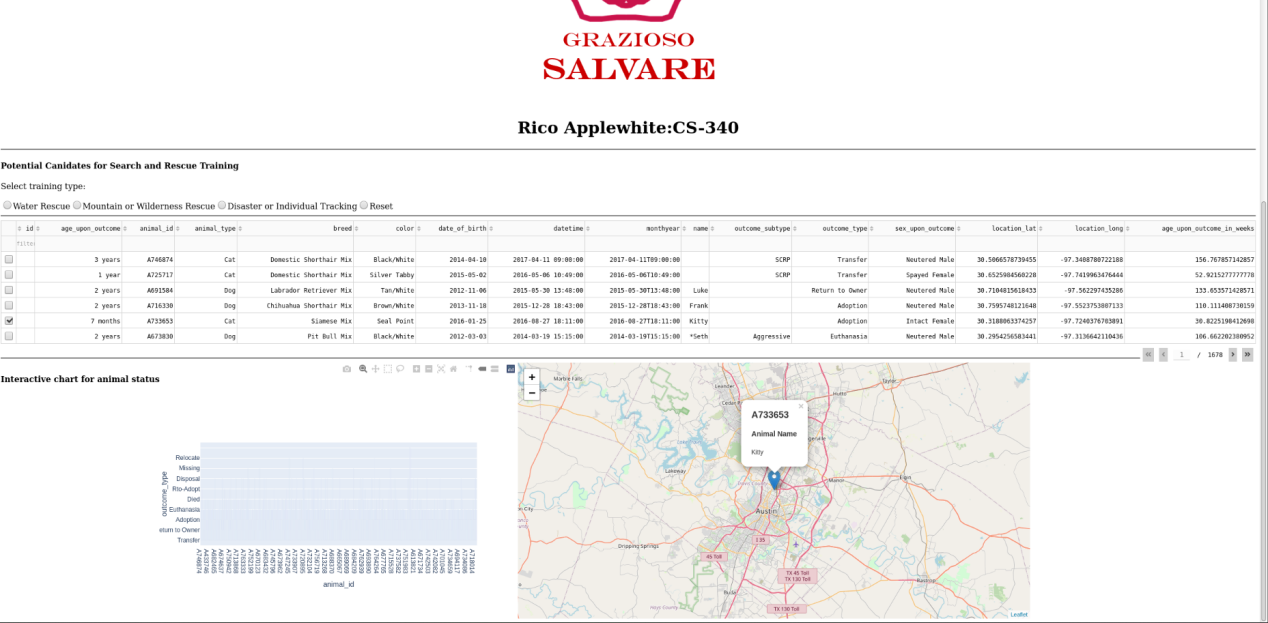


* Disaster or Individual Tracking



* Reset

Hitting reset will bring back the main dashboard view of all the animals.



**Reproduction Instructions:**

To reproduce this, you need to have access to a Mongodb. You can import one from cvs or other spreadsheet filetype. Having authentication to use the database is also important.

Once you have a data base , your need step will be creating a CRUD (Create, Read, Update, and Delete) python file. This will allow easy access and modification of the data base.

You can then begin writing your dash program. I used Juypter notebook for this application. dash.plotly.com has tons of resources to help you get started.

Pay close attention to the callback aspect of the code, while coding my application it took many tries for me to get my graphs and map to show up, because of the input and output in the callback.