# 

## About the Project/Project Title

Austin Animal Shelter dashboard is a Grazioso Salvare product to identify good candidates for working-dogs. When trained, these dogs can perform life-saving duties like Search and Rescue! Grazioso Salvare is partnered with five animal shelter regions, and this product is capable of searching through the partners data to pick out the breeds that are perfect for specific roles.

The dashboard uses AnimalShelter.py to aid in all the database functions that are critical to the dashboards function.

AnimalShelter.py is a python module that implements CRUD functions utilizing Mongodb to manage the ACC collection.

## Motivation

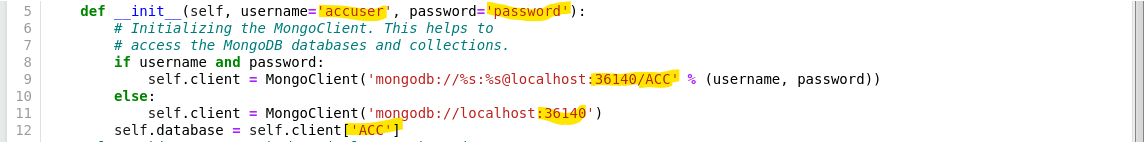
The dashboard is particularly useful at identifying dogs based on prerequisites. Searching for these dogs by hand would be too cumbersome to be practical. It is also useful to identify the location of the dog so that they can be easily adopted from the owning organization.

The purpose of the project is to easily create, read, update and delete entries from the ACC collection using python. This will help speed up database related tasks through an easily repeatable automated process.

## Getting Started

The Austin Animal Shelter Dashboard is quick and easy to use. To get started using the dash board, all that needs to be done is some simple editing of the AnimalShelter.py. This editing will provide the data to drive the Grazioso Salvare Dashboard webface.

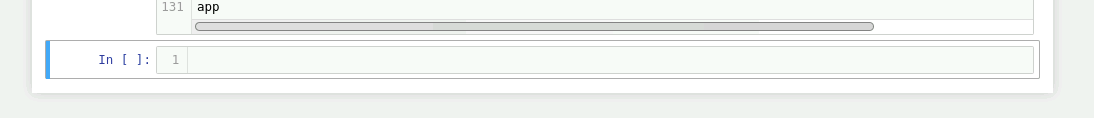
Dependencies: The current AnimalShelter.py is dependent on the existence of the ACC database, the associated accuser, MongoDB, and hardcoded credentials. Once MongoDB has been installed and an ACC database created, edit the credentials to match the desired credentials of your database. If MongoDB is not running on the default port, the port it should be using will also need to be updated. Highlighted below are all the changes that would need to be made depending on the environment of your machine.



## Installation

To start using the Grazioso Salvare dashboard, open the ProjectTwoDashboard.ipynb in your jupyter notebook. AnimalShelter.py will need to be in the same directory as the Dashboard.

Problem 1: Oh no! Where is My Dashboard?

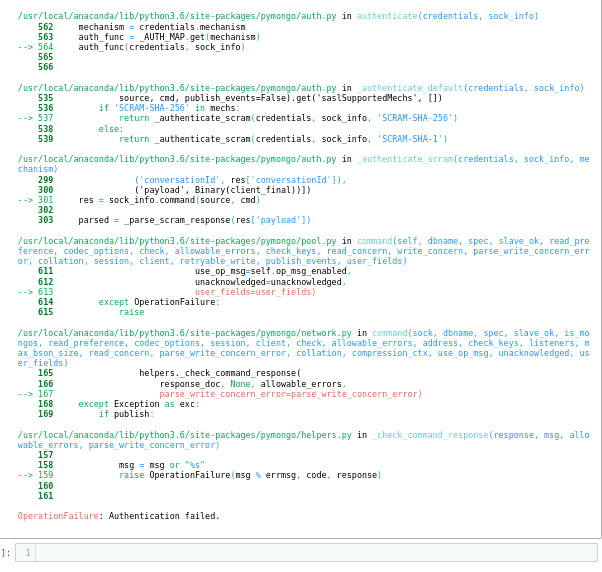


If you press run up at the top of the Jupyter page, and nothing pops up, don’t worry just yet. It is likely that MongoDB isn’t running on your box. The dashboard can’t load data if the database isn’t up. Go ahead and pull up a linux terminal and use,

mongod\_ctl start

to fix this problem.

Problem 2: Authentication Failed.



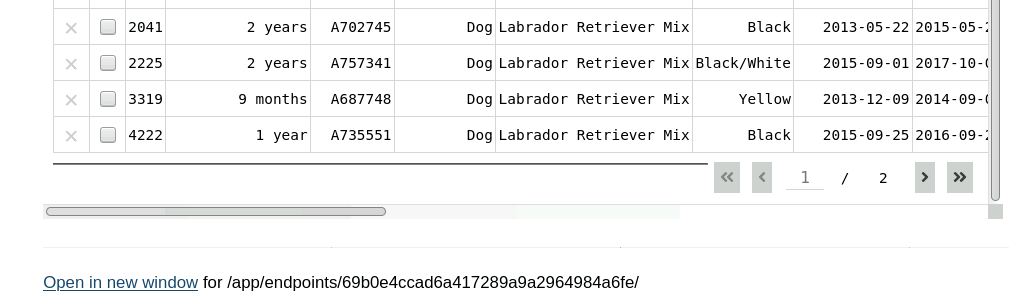
Authentication failures happen when your username and password is wrong. Double check the username and password on the Mongodb and change lines 22 and 23 to your credentials!

Problem 3: Loading…???



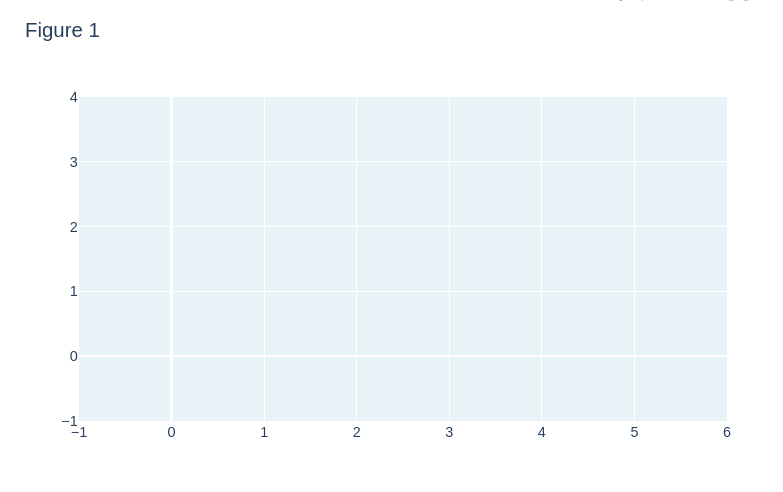
Oftentimes it takes a minute for the Jupyter Notebook to load all the data. In this case, just try being patient. If it is still not loading, it is likely a problem with Jupyter. Refreshing the page or pressing the run button can help.

Problem 4: No Map?



Sometimes when you run the application the graphic, the map, or both will not load. The easiest fix is to use the radial buttons to run a query. This will populate the map and graphic. However, I don’t usually have these issues when using the Open in new window feature, and recommend using that.

Problem 5: No Graphic?



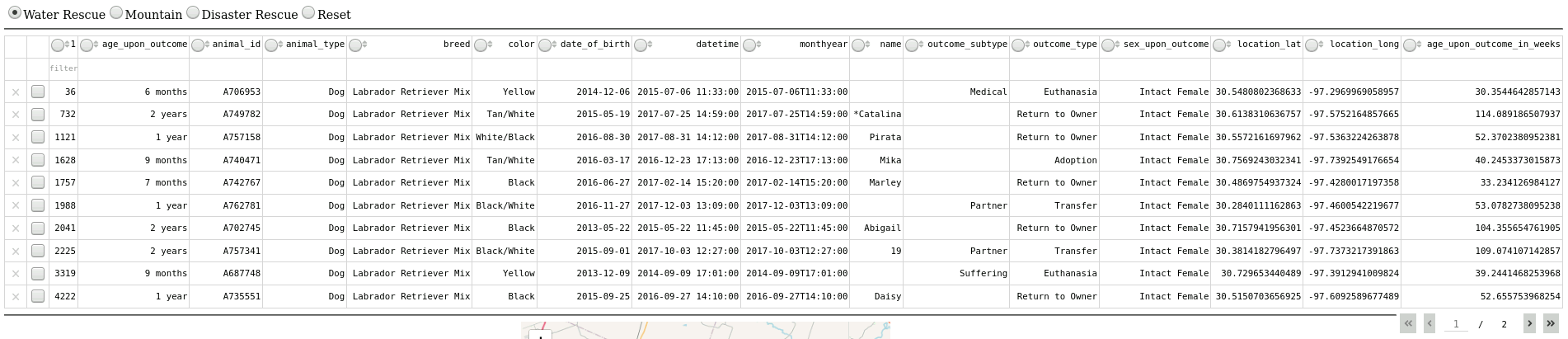
The No Graphic issue is a known issue. After several debugging attempts, I was never able to get a pie graph to render, even when using static test data not associated with the datatable. This coordinate plane is the closest I’ve got to having it rendered appropriately.

Now that those potential issues are solved, the Dashboard can be interacted with. The first thing that you will see is :



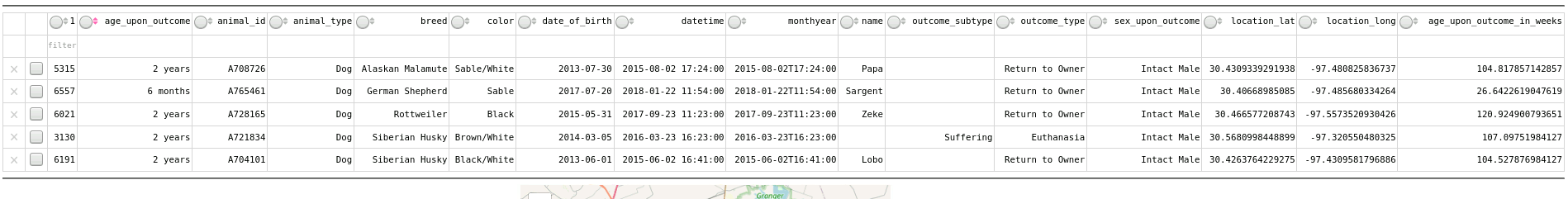
## Usage

Awesome, that means that it is working. In full screen mode, the image can become offset. Scrolling we find the data table. By default the data table shows the dogs which would be most suited for water rescue.

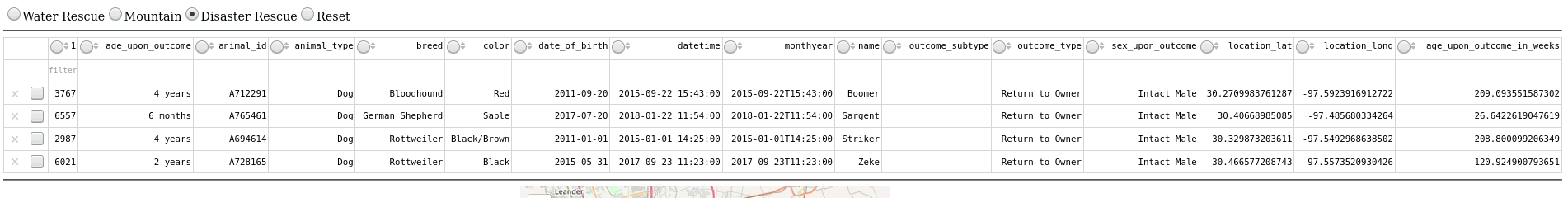


Selecting the radial buttons will change the query. Selecting Mountain will change the query to select dogs best suited for mountain or wilderness rescue. Selecting Disaster Rescue will find dogs best suitable for Disaster or Individual Tracking. Reset will show no filtering on the datatable.

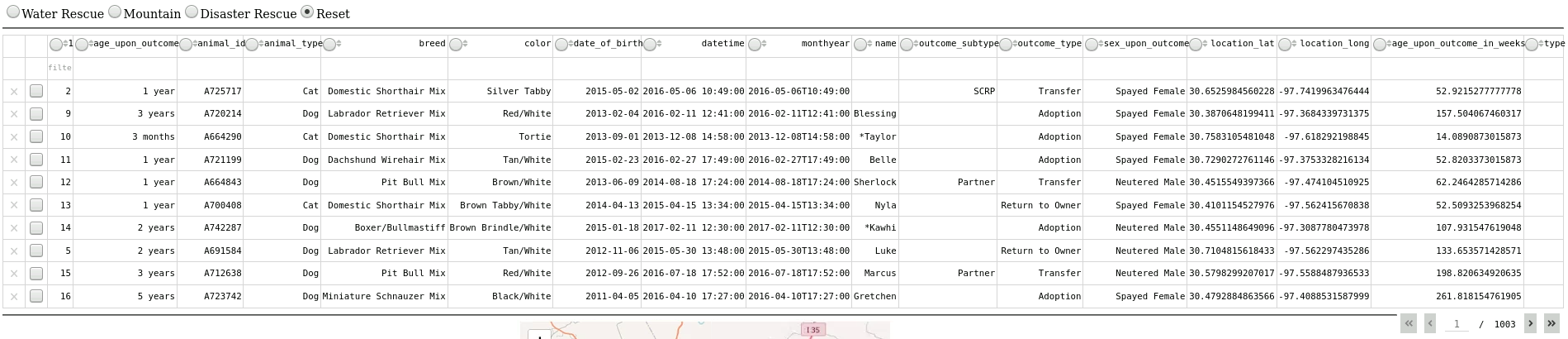
To get started using AnimalShelter.py paste a copy of the code in your workspace. You will also use MongoDB and the ACC database. Once the AnimalShelter.py is in the directory you wish to work with, its functions can be imported in any of your scripts that will manage the ACC collection.



*Mountain Radial In Use*

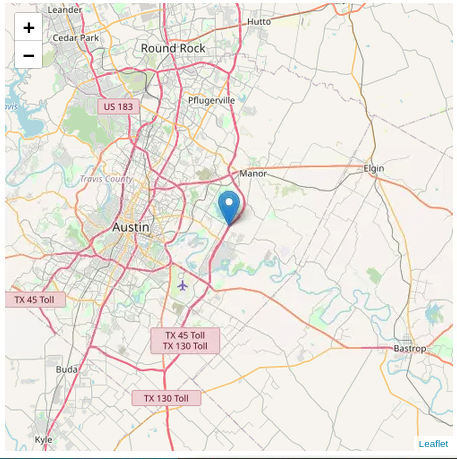
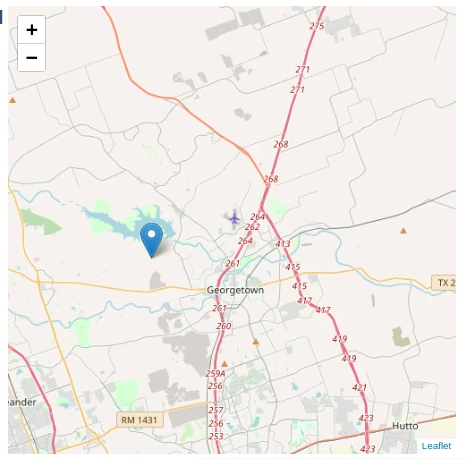
**

*Disaster Rescue Radial In Use*

**

*Reset Radial in Use (no filter)*

These queries are where the usefulness of the Dashboard shines as it can narrow down several thousands of pages of animals to just the ones that are useful for rescue operations. One additional function mentioned is the graphics that accompany the queries is a map and a graphic. The map shows the Animal Shelter the animal is currently in.



*Different Uses of the Map*

Currently the Map only shows the location of the Animal that happens to be at the top of the search results.

**Using AnimalShelter.py:**

Currently, AnimalShelter.py has four incredible functions that can be used: Create,Read, Update and Delete. Create allows a user to insert a new document into their mongo database. Since MongoDB uses JSON, the data needs to be in a JSON format for best results.

### Code Example

The world renown create function in use:



*(remember to import AnimalShelter before use!)*

AnimalShelter can also read items in a database. The read function takes a query in a JSON format as an argument. Below is an example of reading the MongoDB documents just created above:

**Code Example**

The stunning read function in use:

### 

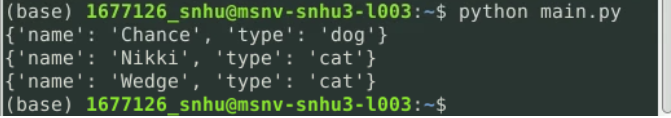
*(a is what we called our AnimalShelter object)*

### Tests

Once you’ve created a script that takes advantage of the AnimalShelter.py functions, simply compile the python script using any compiler you like.

### Screenshots

Below are the results of the example methods compiled and ran from the Linux command line:



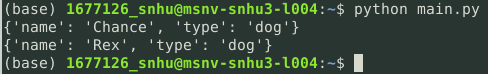
AnimalShelter has been recently updated to use the update function which allows records to be changed. The Update function accepts a query and new values using the traditional mongodb syntax. In the example below the new values are saved to a newValue variable.

## 

The other function added is the handy delete function. This function will delete the first result that is returned with the query supplied to it. Below is an example of deleting the record of a dog named Rex!



With the addition of Update and delete, it is now possible to fully manage the ACC database with AnimalShelter.py. Below is an example of the entire module being used. First a record of a dog named Chance will be created. Then that record will have the name field updated to Rex. Lastly, the record of the dog name Rex will be deleted; results below!



## Roadmap/Features

Known Bugs:

A query must be performed before the Map shows.

The graph is rendered but no data shows.

## Contact

Aaron J Walls

aaron.walls@snhu.edu