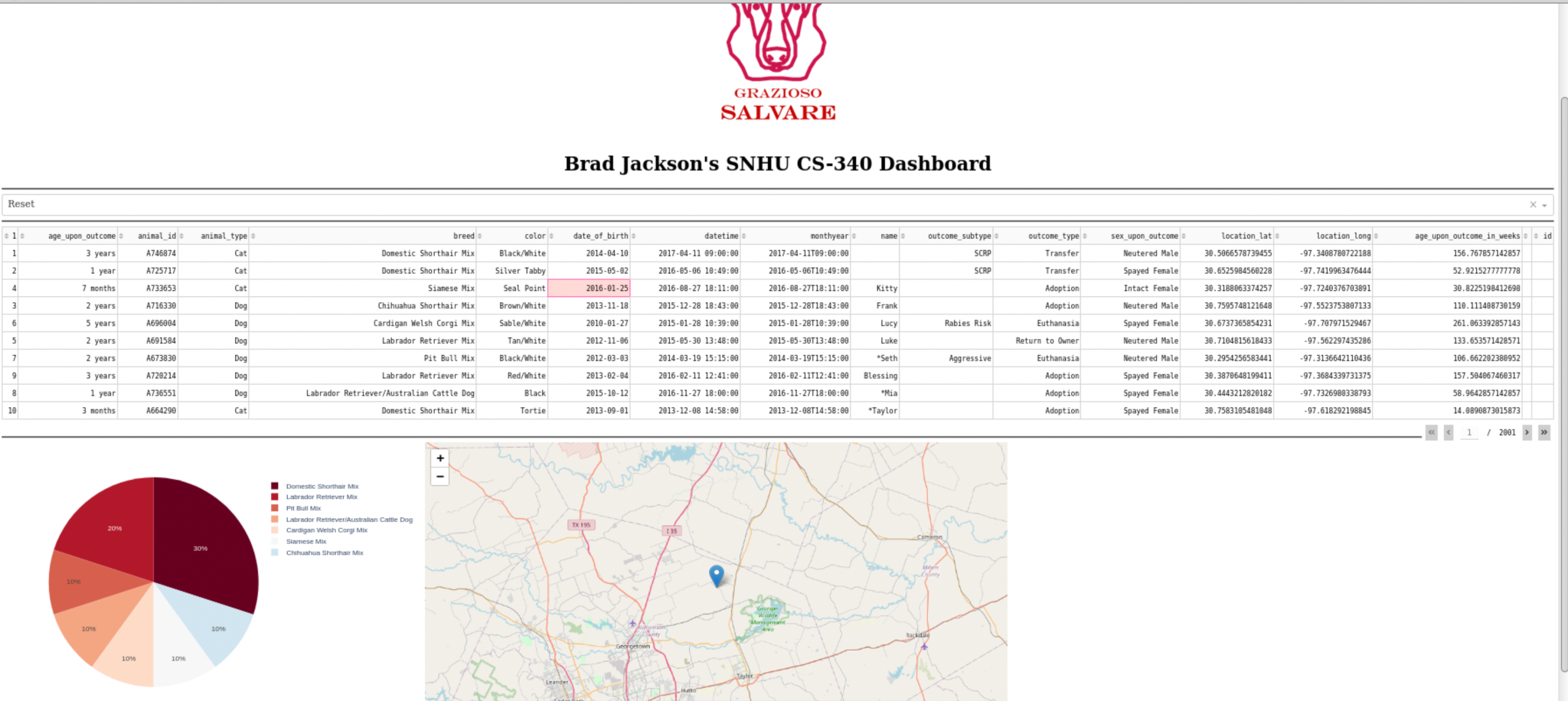
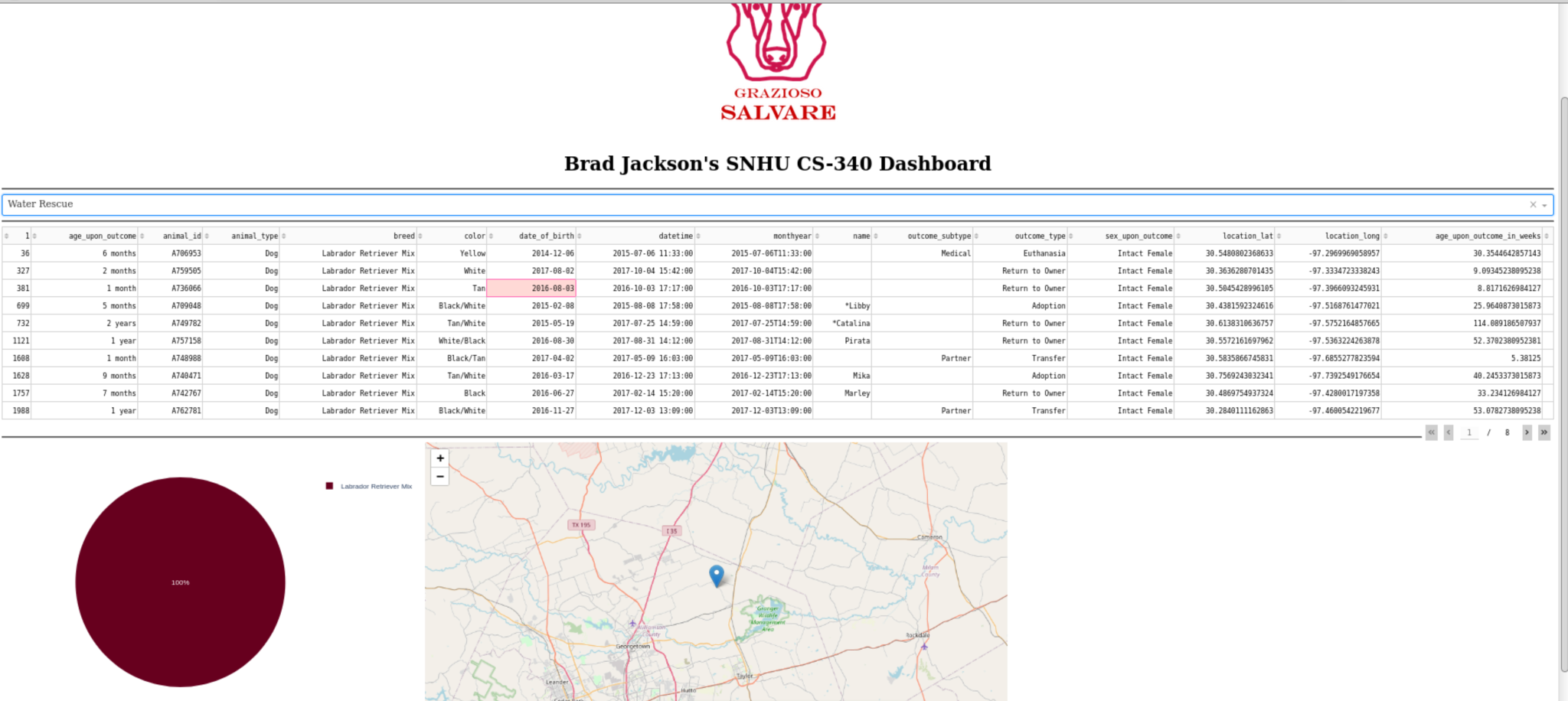
**Project Two README**

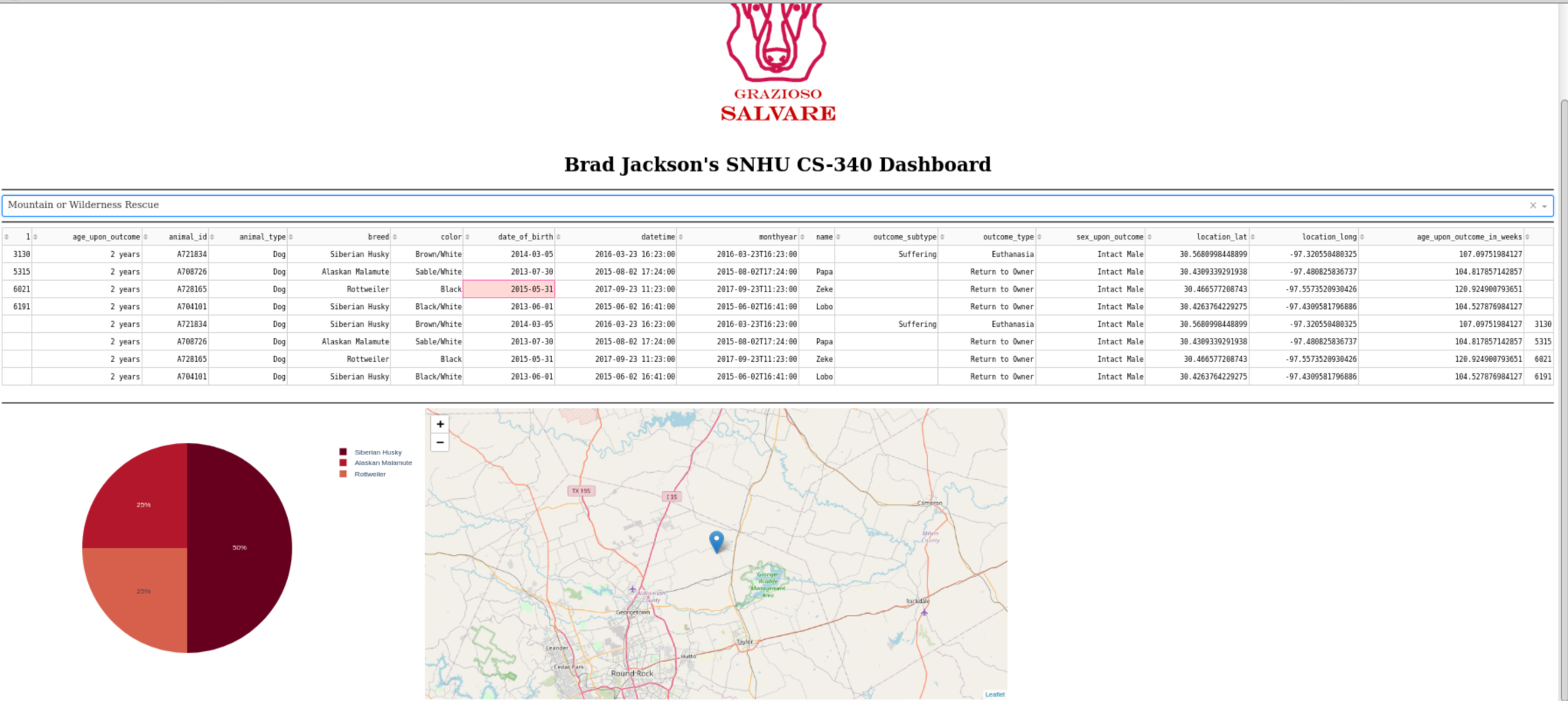
PURPOSE

As part of its work, Grazioso Salvare identifies dogs that are good candidates for search-and-rescue training. When trained, these dogs are able to find and help to rescue humans or other animals, often in life-threatening conditions. To help identify dogs for training, Grazioso Salvare has reached an agreement with a non-profit agency that operates five animal shelters in the region around Austin, Texas. This non-profit agency will provide Grazioso Salvare with data from their shelters. This application can work with existing data from the animal shelters to identify and categorize available dogs. Through the dashboard, users can view the database (below).

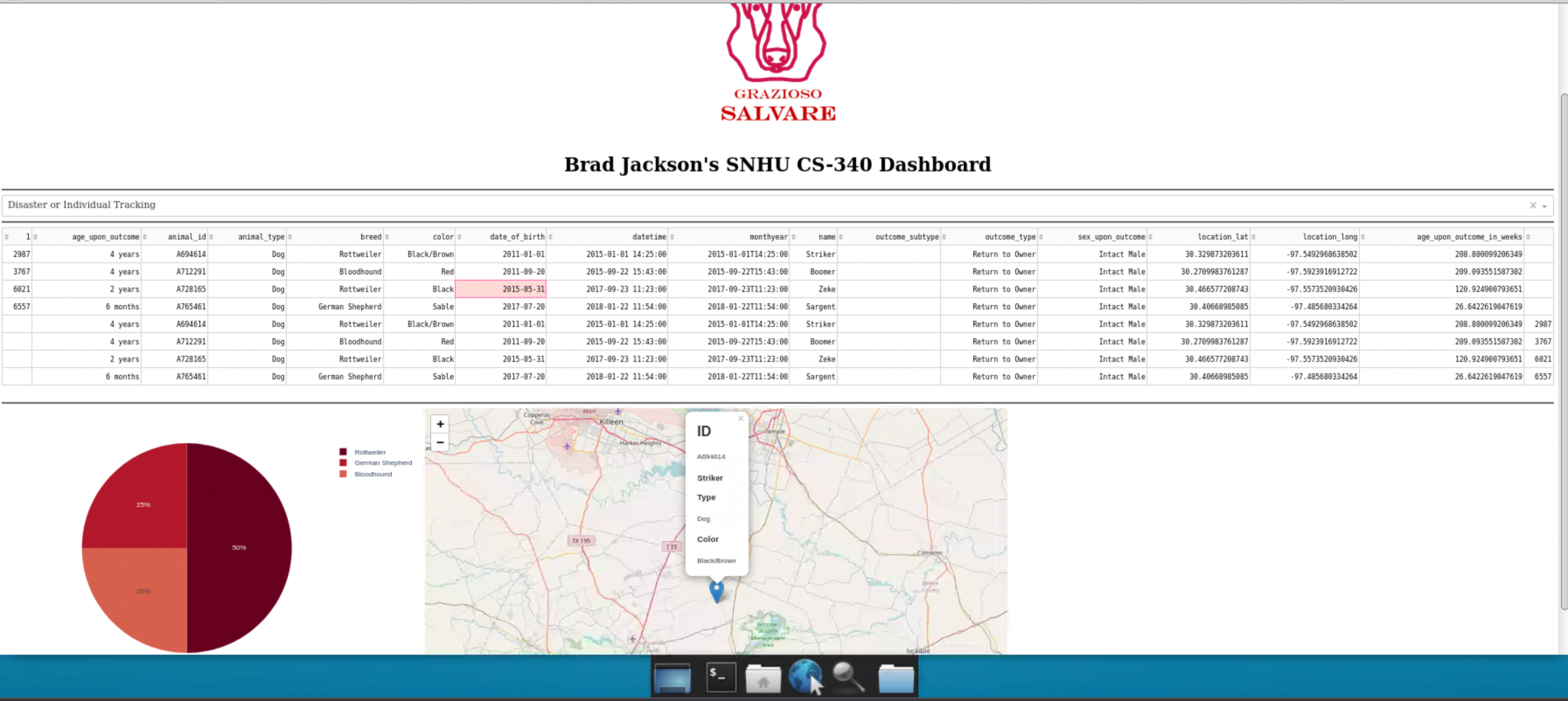


Users can also filter through the animals by:  
**Water Rescue**

**Mountain or Wilderness Rescue**



**Disaster or Individual Tracking**



Visualizations are also available for the user. The pie chart shows the ratios of different breeds for the current page of data. The map shows the location of a selected animal.

**Requirements**

* [Python3.6+](https://www.python.org/downloads/)
  + [Dash 1.10](https://pypi.org/project/dash/)
* [MongoDB4.2+](https://www.mongodb.com/)

The main application runs in python, using a library derived from plotly called dash. These were used due to the rapid speed of development python provides and the powerful tools dash offers. Dash provides a lightweight framework for creating dashboards. We just have to dynamically add the data which can range in format. For our database we use MongoDB. MongoDB is a NoSQL document database. In our case, we dealt with animal shelter data which could be treated as objects with attributes associated with them. Using python we created a model for create, read, update, and delete methods in MongoDB. From there we could load data to dash via the model using callback functions.

**Steps**

Initially we were asked to initialize the Mongo database given data from an animal shelter. We then created a database user for the database. Using the credentials provided from the user creation step we created a model to interact with the database in python. We then used that model to populate a simple dashboard. We were then provided with more software requirements which necessitated a more fully-featured dashboard.

**Challenges**

The most challenging part of the project was learning how to use dash. I struggled with getting the callbacks to work for the filters in project two. I also remember having trouble with it in the first dashboard. I am not sure what exactly caused my problem but eventually I achieved the desired result.