**Grazioso Salvare – Austin Animal Shelters Project**

**Contents**

[About the Project/Project Title 1](#_Toc79747019)

[Motivation 1](#_Toc79747020)

[Getting Started 1](#_Toc79747021)

[Installation 3](#_Toc79747022)

[Usage 4](#_Toc79747023)

[Code Example 4](#_Toc79747024)

[Tests 5](#_Toc79747025)

[Screenshots 6](#_Toc79747026)

[Dashboard Screenshots 10](#_Toc79747027)

[Roadmap/Features 14](#_Toc79747028)

[Contact 14](#_Toc79747029)

## About the Project/Project Title

Grazioso Salvare is an international rescue-animal training company that identifies animals that are good candidates for search-and-rescue training. They look for certain profiles in the animals to train. Generally, Grazioso looks for animals that are aged two or younger. In addition, some breeds are more proficient in different types of rescues. A nonprofit that operates five animal shelters around Austin, Texas will partner with Grazioso to find good candidates. The animal information will be uploaded to a database accessible to Grazioso employees via a web-based dashboard. The software will allow Grazioso to identify and classify available shelter animals that may be a good candidate for rescue training.

## Motivation

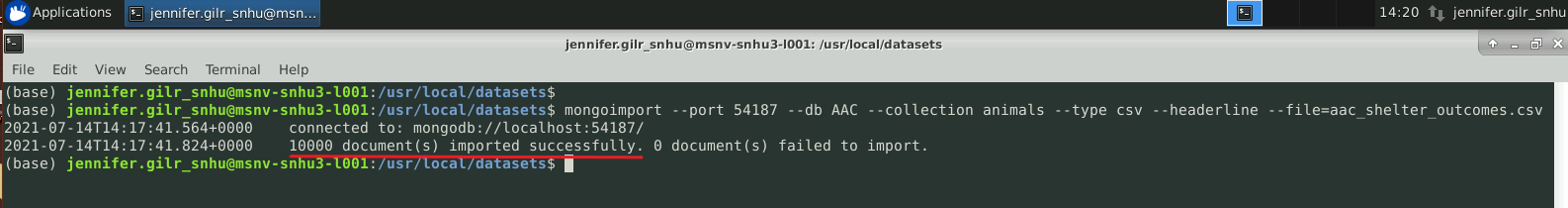
Grazioso Salvare needs an effective way to filter the animal information to quickly find a prospective trainee. Moreover, they need a simple interface so non-technical staff can use the database.

## Getting Started

1. The first step is uploading the data file provided by the nonprofit agency. The format will be a csv file. This only needs to be performed once.
2. Open a terminal window and navigate to the directory where the csv file is located.
3. Start the mongo server using command: /usr/local/bin/mongod\_ctl start -noauth
4. Make note of the port number

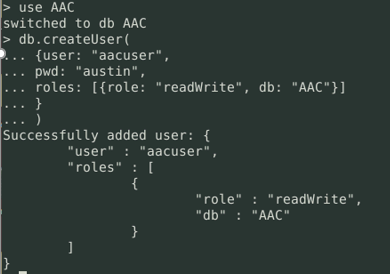


1. Type the command: *mongoimport –port ##### --db AAC –collection animals –type csv –headerline –file=aac\_shelter\_outcomes.csv* (Note: replace ##### with the port number you made note of in the last step)
2. You should see a message that # document(s) imported successfully. If the import fails, verify you typed the commands correctly. It is case-sensitive and must be exact.



1. The database utilizes MongoDB and is equipped with access controls to prevent unauthorized access. Setting up a username and password is done through a terminal window, currently.

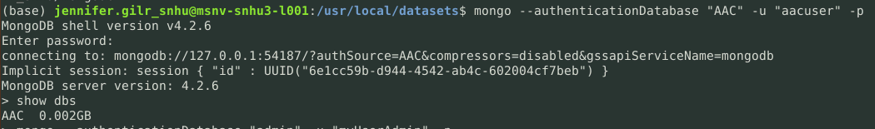
Here is an example of creating a new user:



The new user, *aacuser*, has access to the AAC database with read and write privileges.

1. Normal operations – The first two steps are for setup. For day-to-day operations, you will normally start here.
2. Start the mongo server using command: */usr/local/bin/mongod\_ctl start*.
3. Type *mongo –authenticationDatabase “AAC” – u “[username]” -p* (Note: replace [username] with a valid username). Using the example user above: *mongo –authenticationDatabase “AAC” -u “aacuser” -p*
4. Enter the corresponding password. Note: the cursor does not move while typing the password.
5. Press Enter
6. To verify connection to the database, type *show dbs*

Screenshot showing *aacuser* has access to the database, AAC:



## Installation

The application requires the following software (for each item, click the provided link, select your operating system, and follow the instructions):

* Python:
  + Interpreter:

<https://www.python.org/downloads/release/python-396/>

* + Python Package Installer (pip):

<https://pypi.org/project/pip/#files>

* + PyMongo:

<https://pymongo.readthedocs.io/en/stable/installation.html>

* MongoDB:
  + Shell:

<https://www.mongodb.com/try/download/tools>

* + Server

<https://docs.mongodb.com/guides/server/install/>

* JupyterLab:

<https://jupyter.org/install.html>

* Dash:

<https://github.com/plotly/jupyter-dash>

In addition, you will need the Austin Animal Shelter module and the testing script:

* Python module distribution file:

After you have downloaded the distribution file, navigate to inside the folder. Open a terminal window and type the following command:

* + - Windows: *py -3 -m pip install aac-1.0.zip*
    - Unix: *sudo python3 -m pip install aac-1.0.tar.gz*



This will install the module in the site-packages so the software can access it.

MongoDB was selected for its scalability and query-response time. It is an established platform with many developers, so support will be readily available. Python integrates well with Mongo and does not need to be compiled, meaning it can run anywhere. Furthermore, Python is open source, so the program will be easy to distribute.

As per Grazioso’s request, the code for the Python module, JupyterLab testing notebook, and Dash dashboard are available on GitHub.

## Usage

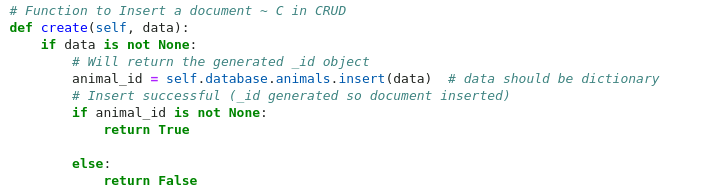
The application has full CRUD functionality. You can create a new document or search for all records that meet the specified criteria. In addition, you can perform a mass-update or deletion.

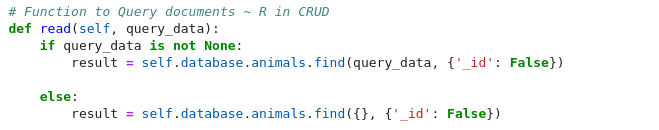
Access controls are available to ensure information is secure. This requires the user to enter valid credentials to access the database.

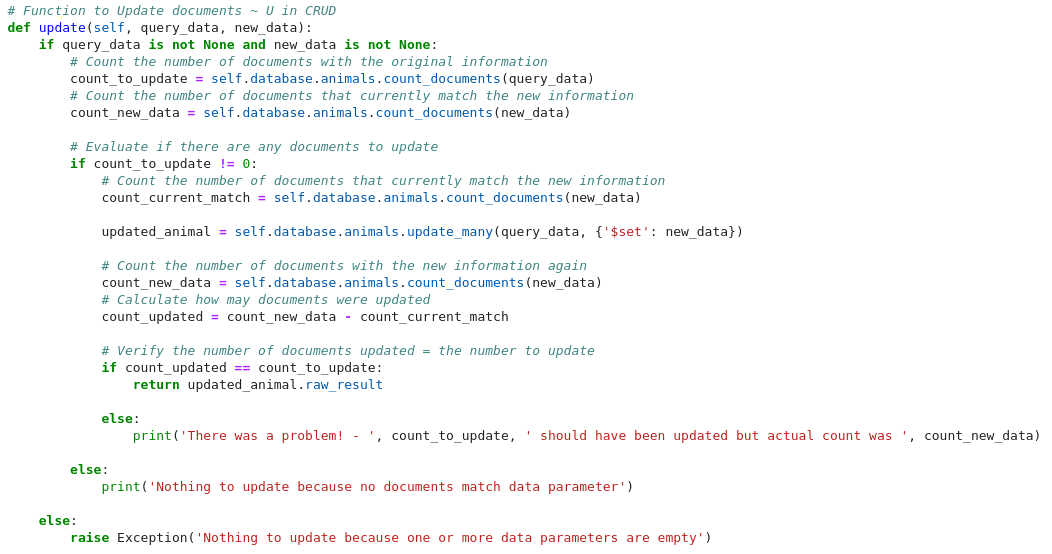
After the initial setup, users can search for training candidates using the online dashboard. The dashboard features an easy-to-navigate interactive table. Since the table is large, the user can also narrow the search by filtering by the type of rescue training. In addition, there are charts that update based on the table.

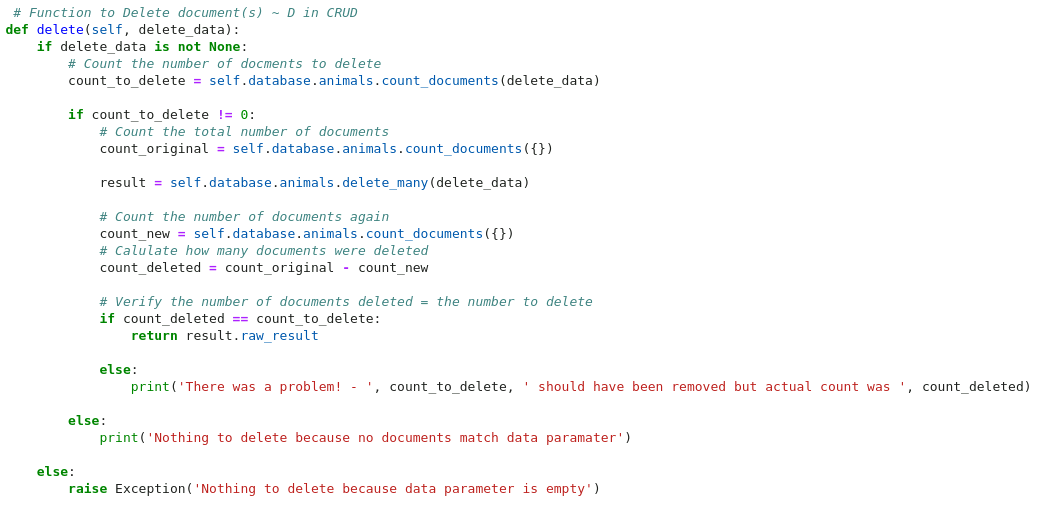
## Code Example

These screenshots are from the aac.py module to demonstrate how CRUD is implemented:



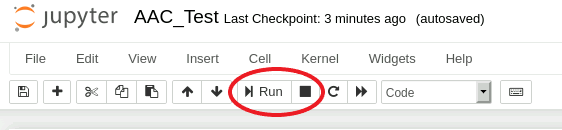






### Tests

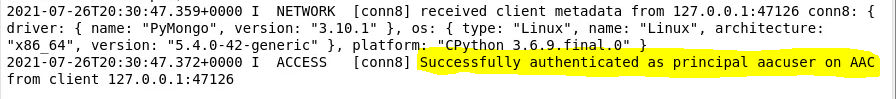
Unit tests are available through the Jupyter Notebook *AAC\_Test.ipynb*. Open the notebook and click the Run button:

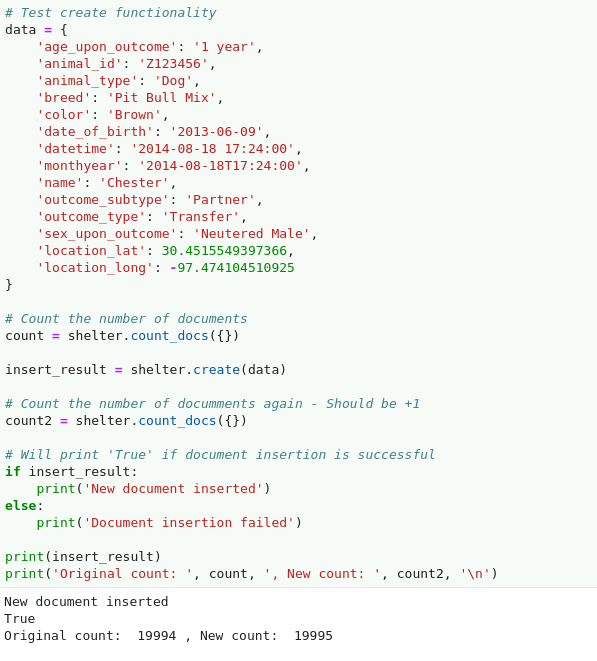


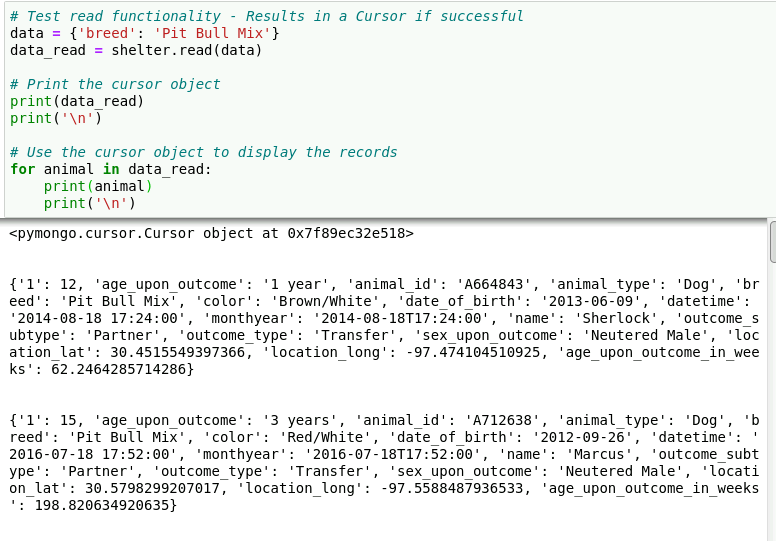
### Screenshots

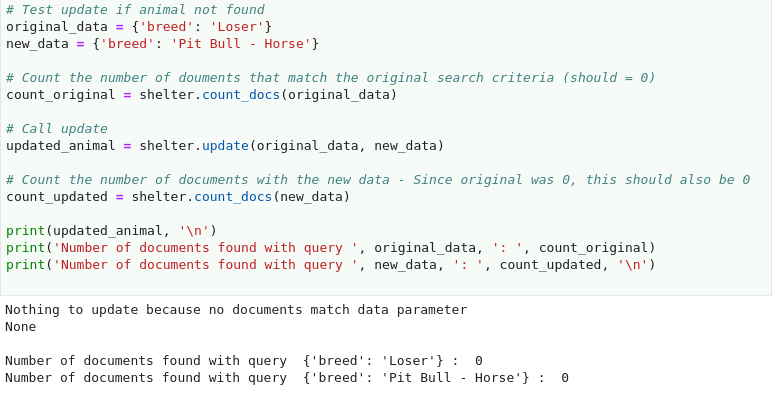
Here are some screenshots showing that the Python module is accessible and functions correctly:

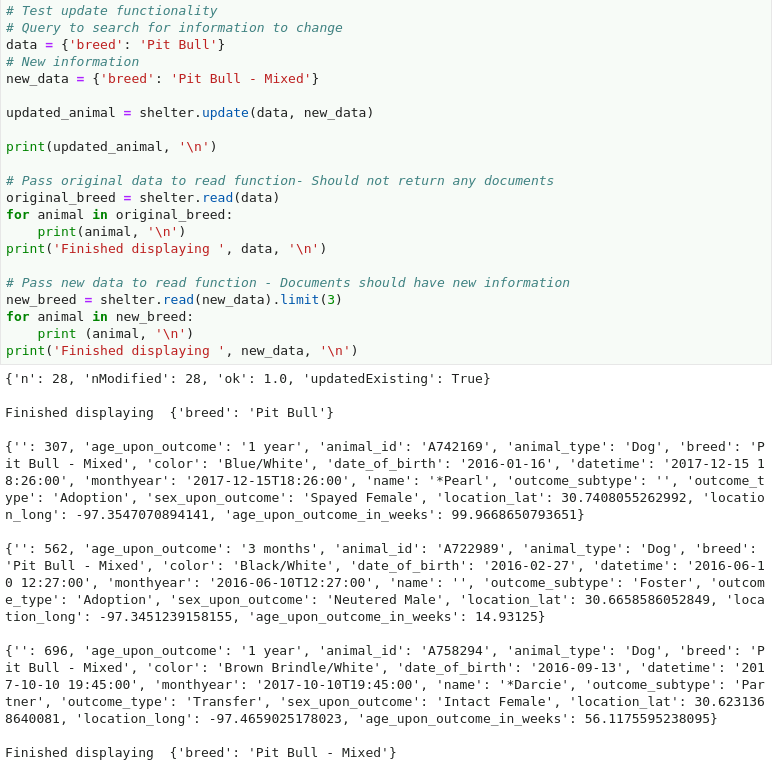


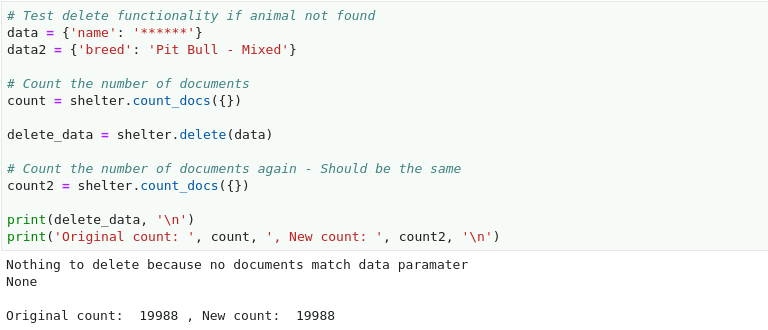


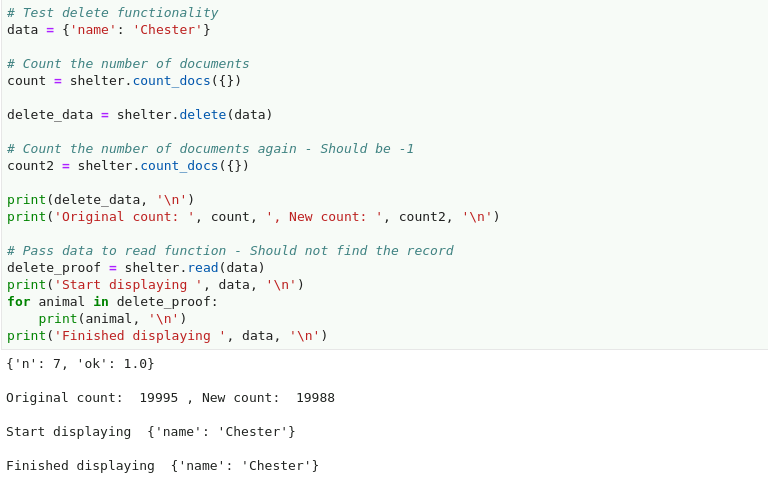








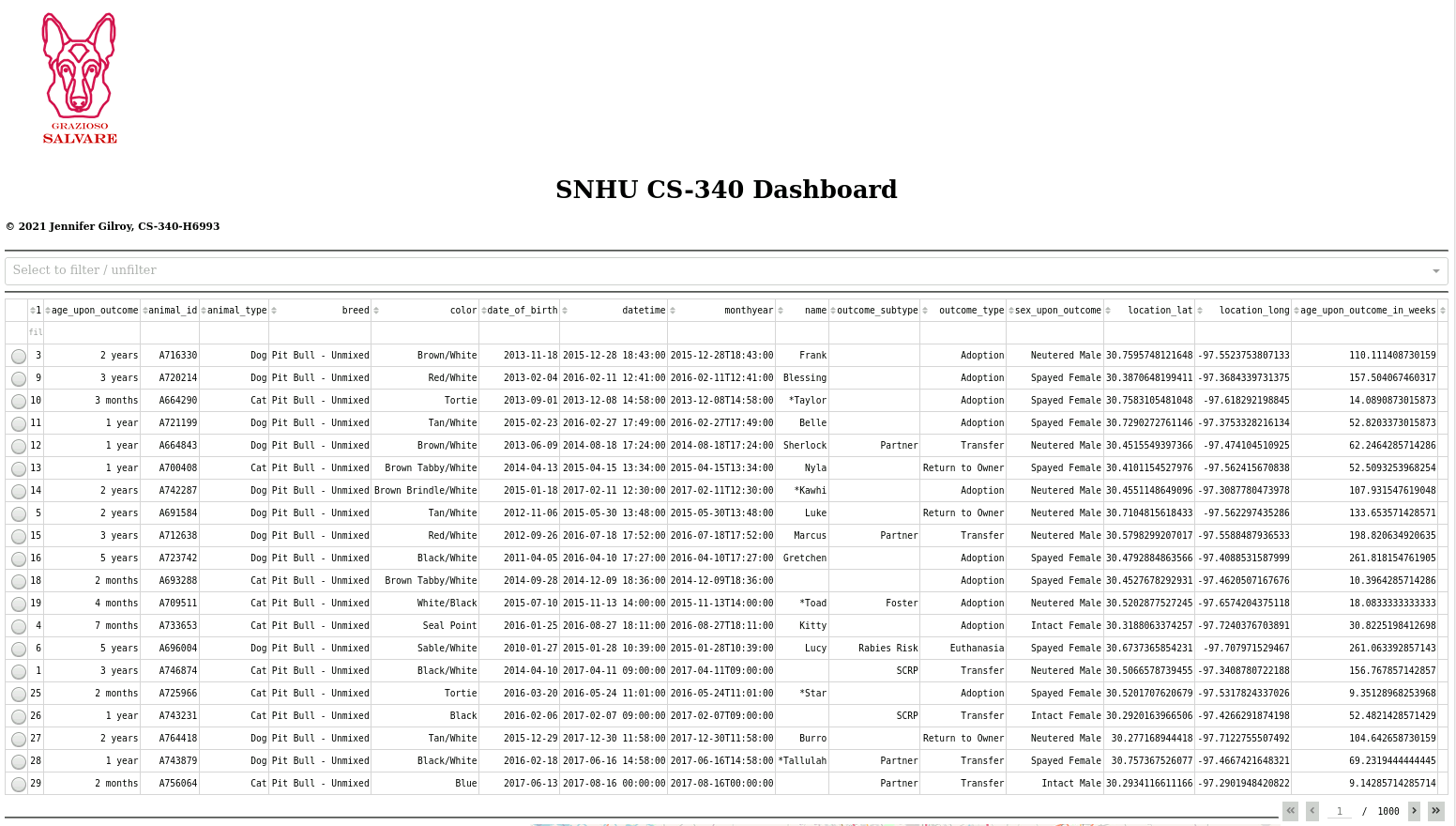




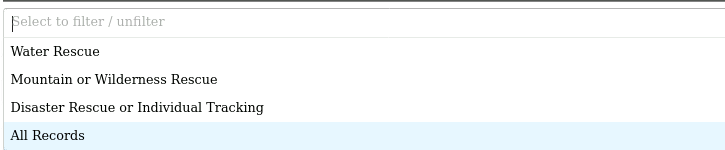
## Dashboard Screenshots

Here are screenshots of the dashboard highlighting the interactive filters. When you first run the dashboard, you will see the unfiltered data table that shows all the animals in the system. The Diagram will also reflect that. Clicking in the box “*Select to filter*” enables you to filter the list based on the criteria best suited for the type of rescue. The pie chart will update to reflect available breeds for that rescue type. Choosing the dropdown list enables you to select a different filter or view all animals.

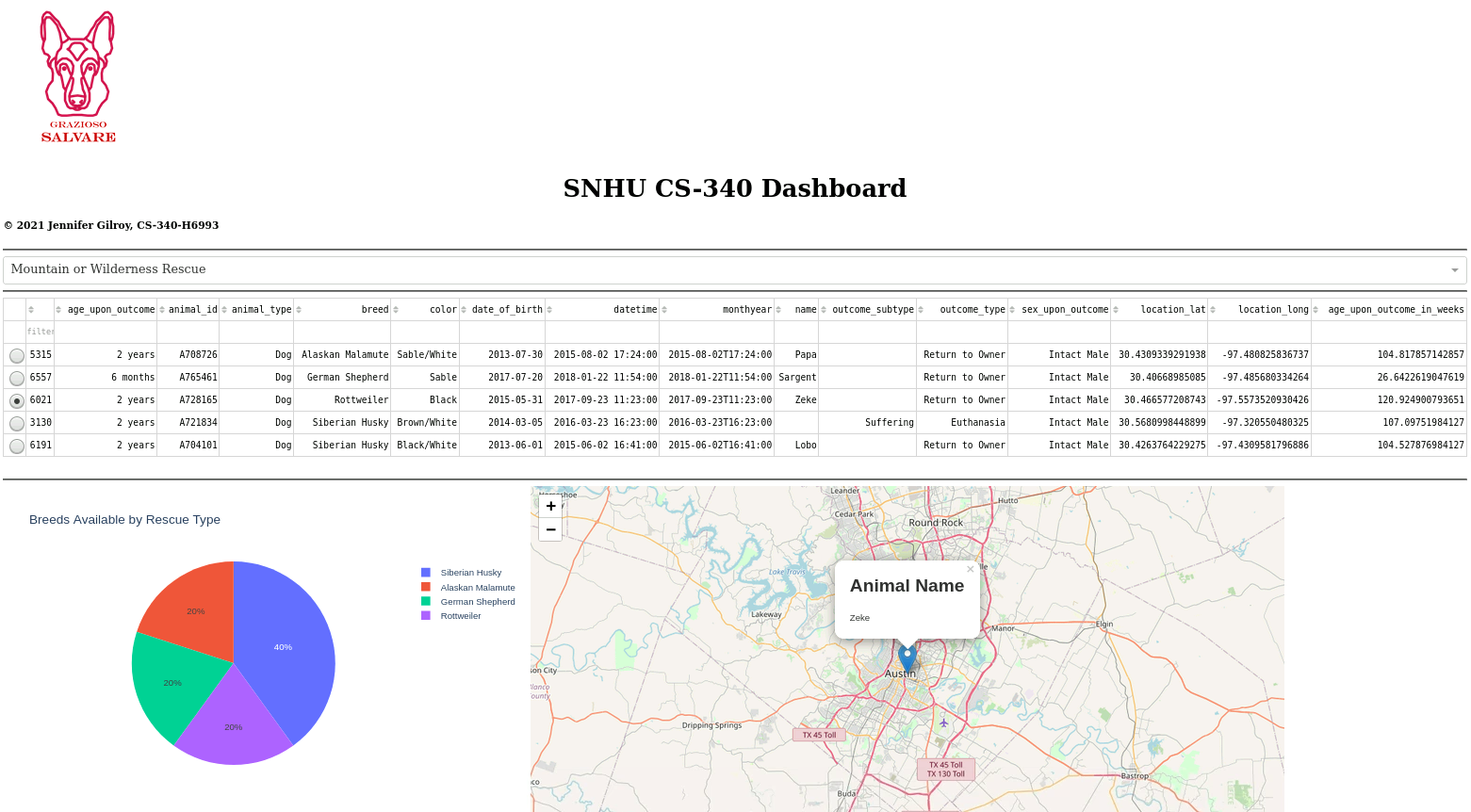
**Unfiltered Data Table:**

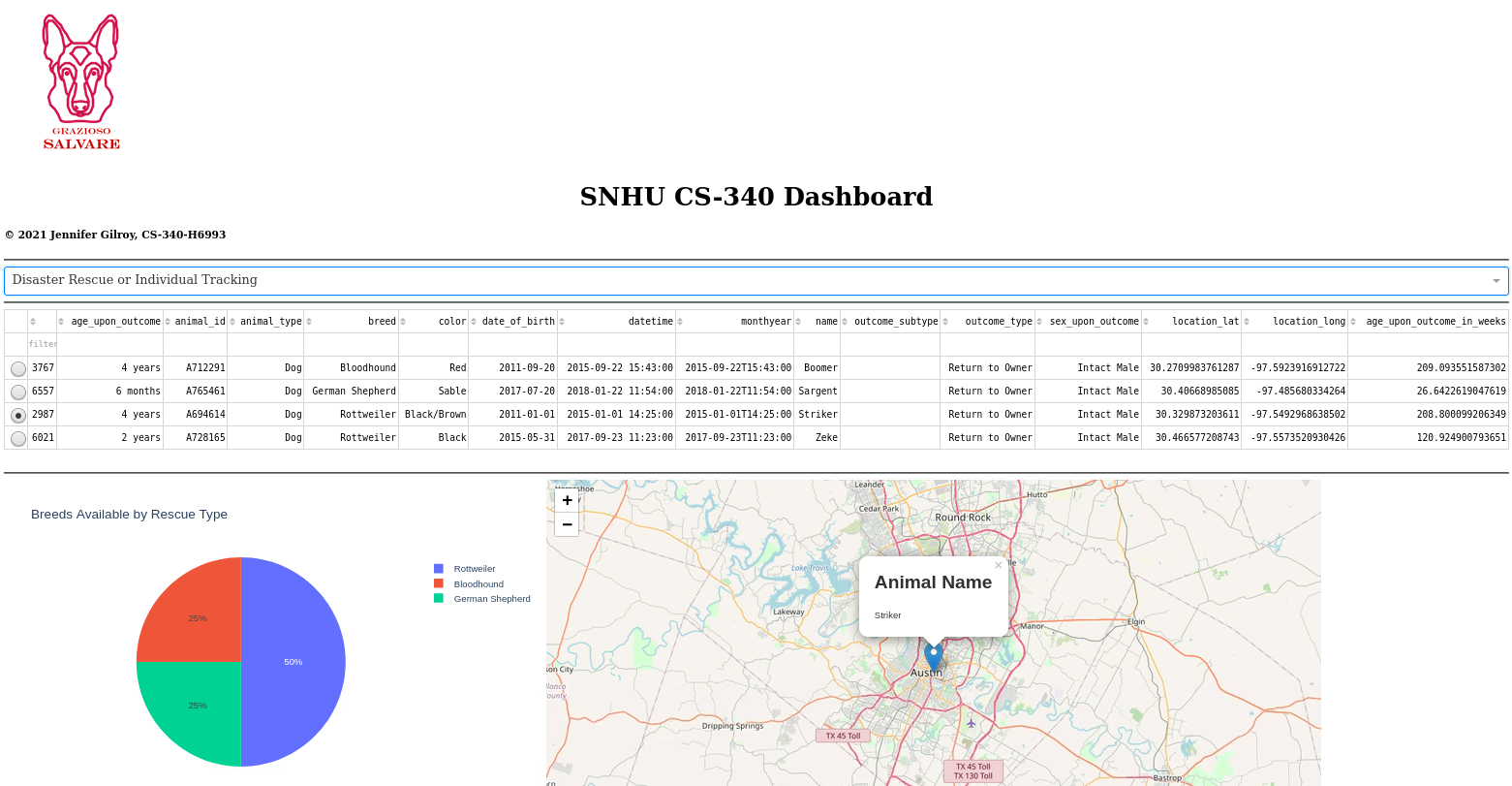
****

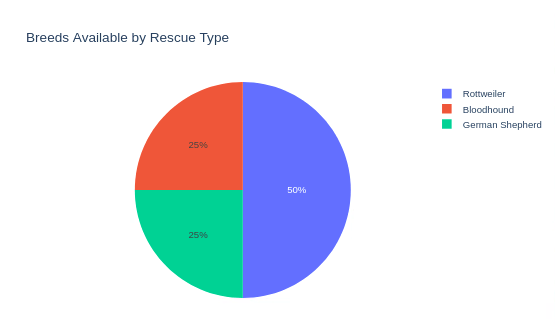
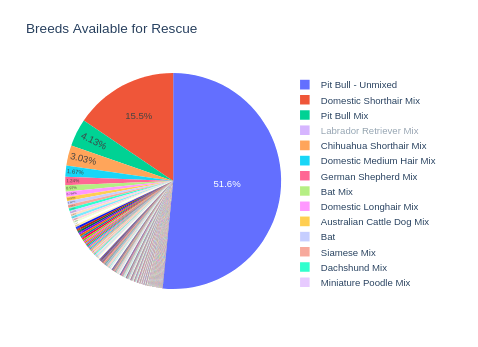
**Drop-down filtering:**



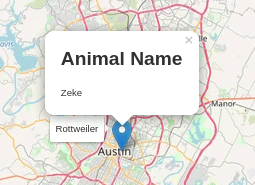
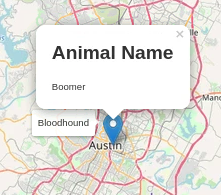
**Table and charts update based on the filter and row selected:**



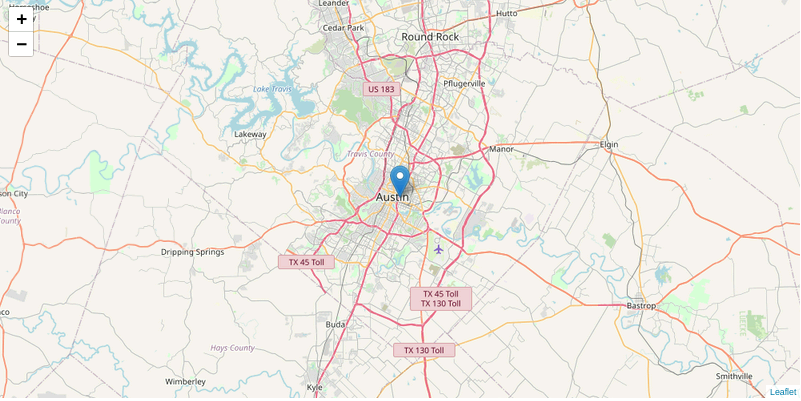


**Pie Chart displays the available breeds which changes based on the rescue type selected:**

**Selecting a row in the table will update the breed and animal’s name in the map:**

**Map is centered at Austin, Texas. The map can be controlled using normal drag and zoom functions:**



## Roadmap/Features

There is a non-bug in the application that does not update the map marker based on the animal’s location. This will be fixed in the next release.

## Contact

Jennifer Gilroy

[Jennifer.Gilroy@SNHU.edu](mailto:jennifer.gilroy@snhu.edu?subject=AAC%20Project)