# CS 340 Project Two- README

## About Project Two

This project was built for our client Grazioso Salvare, an animal rescue training company. Using CRUD functionality, a dashboard was developed to display sheltered dogs. From the dashboard, Grazioso Salvare will be able to select dogs to train as rescue dogs. Filters are provided to filter out the breed, sex, and training age. Using the charts, a user can see the geolocation of the first dog, as well as a pie chart listing the breeds and number of dogs provided.

## Motivation

There were three main guidelines for this project. The guidelines are as follows:

* Grazioso Salvare’s logo and a unique identifier including my name (Madison Lopert) must be visible on the dashboard.
* Dashboard filter options must be provided. Radio buttons were used in this project. The data filters can be seen below.
  + Water Rescue
  + Mountain or Wilderness Rescue
  + Disaster or Individual Tracking
  + Reset (returns all widgets to their original, unfiltered state)
* A geolocation map and a widget of my choice should be included on the dashboard. A pie chart was utilized in this project.

Further details can be found in this specification document. This resource was followed as a guideline during development.: [https://learn.snhu.edu/content/enforced/1272228-CS-340](https://learn.snhu.edu/content/enforced/1272228-CS-340-T4224-OL-TRAD-UG.23EW4/course_documents/CS%20340%20Dashboard%20Specifications%20Document.pdf?_&d2lSessionVal=NxqDMPfYIvryfdH3H9hQNNVOA&ou=1272228)

## Getting Started

The Austin Animal Center (AAC) data set needs to be downloaded using the Mongoimport tool. The provided notebook (.ipynb) starter code file should be downloaded and opened in Jupyter. Once downloaded, the code can be modified according to the client’s requirements. The user credentials used were created for the AAC database. The user was given read and write permissions to execute insert and find functions. The CRUD python file we created earlier in the term was imported and used to connect to the database and return query values.

## Installation

The software needed for this project is MongoDB and Jupyter. The Python file must use the PyMongo driver to have a synchronous connection to MongoDB. This is necessary for real-time CRUD functionality. If not already installed, Anaconda will need to be installed as well. MongoDB was a reliable resource to host our database information. Using values pulled from our imported AAC database, we were able to create a dashboard that displays those values. This dashboard was given multiple attributes to become a more user-friendly interface. Filtering, selection, and chart placement aids in user navigation while prioritizing space. Installation instructions and download files for software applications used can be found online on their respective websites:

MongoDB: https://www.mongodb.com/docs/manual/installation/

Anaconda(Contains Jupyter): <https://www.anaconda.com/>

For the AAC database import the following command lines can be utilized in MongoDB: (Please replace the user credentials and port numbers as needed)

/usr/local/bin/mongod\_ctl start

mongo --port <yourport#> --authenticationDatabase "admin" -u "myUserAdmin" -p

mongoimport --port ##### --db AAC --collection animals --type=csv --headerline ./aac\_shelter\_outcomes.csv

The CSV file for the AAC import can be found here: [https://learn.snhu.edu/content/enforced/AACCSV](https://learn.snhu.edu/content/enforced/1272228-CS-340-T4224-OL-TRAD-UG.23EW4/course_documents/aac_shelter_outcomes.csv?_&d2lSessionVal=NxqDMPfYIvryfdH3H9hQNNVOA&ou=1272228)

## Usage

### Challenges

Several challenges came up during the development of this project. Some of these included a base64 error, chart/map sizing, and a lack of data.

One of the errors I received was "NameError: name 'base64' is not defined". This confused me as the base64 line of code in the project was pre-provided in our starter code. After looking at online resources, I realized the program could not find base64 to pull from so I had to import it.

After I created the charts, I noticed that they did not want to display once I attempted to view the dashboard using the endpoint link. After I adjusted the pie chart's width and height, they ended up showing side-by-side as intended.

The filter conditions added through radio buttons show different queried results than what is set at default. For conditions such as “water rescue”, it only showed one breed of dog when I ran the code. I thought this might be an error for me to fix, but after testing in my client I realized there were no dogs with breeds “Newfoundland Mix” and “Chesa Bay Retr Mix” that fit the other sex and training age requirements as well. For further context, I chose to use these mixed breeds based on what breeds I saw available in the AAC CSV file.

### Screenshots

**Water Rescue:**

*Table

Description automatically generated*

***Mountain and Wilderness Rescue:***

***Table

Description automatically generated***

***Disaster or Individual Tracking:***

***Table

Description automatically generated with low confidence***

***Reset:***

***Table

Description automatically generated***

***Close up of Logo and Charts for Visibility:***

***Logo

Description automatically generated with medium confidence***

***Graphical user interface, map

Description automatically generated***

## Contact

Madison Lopert