# Grazioso Salvare README

## About the Project/Project Title

Our client Grazioso Salvare would like to be able to identify proper candidates for their training regime. They have identified the best combination of breed, sex, and age for their different rescue types. Grazioso Salvare is in need of a dashboard that helps them filter through their rescue dog database in a easy and friendly manner.

### Process

I decided to start with the simplest part first which is both the image and the module name change for AnimalShelter. For the project I first implemented the image centered and used a href tag to add the www.snhu.edu website. I also modified the username and password when I called the AnimalShelter() function. I then added the four buttons that are going to be used for the filters, these buttons are each named accordingly. I added features that were done in module six for making the table used friendly. I then modified the update\_dashboard to take in the four buttons as input and and used triggered\_id to check for button press. After pressing the button, the filter is then applied. I finally added the map function from module 6 which was functioning flawlessly.

## Installation

* Python v3.6.9
* PyMongo v4.3.3
* MongoDB v4.2.6
* Jupyter Notebook v6.0.1
  + Python v3.6.9
  + GCC v7.3.0
  + IPython v7.8.0
* jupyter\_plotly\_dash
* base64
* dash
* dash\_leaflet
* dash\_core\_components
* dash\_html\_components
* plotly.express
* dash\_table
* dash.dependencies
  + Input
  + Output
  + State
* os
* numpy
* pandas
* pymongo
  + MongoClient
* bson.json\_util
  + dumps
* AnimalShelter
  + AnimalShelter

For the creation of the dashboard the use MongoDB alongside Python is necessary for achieving an easy CRUD implementation. Python has a module specifically made for interacting with MongoDB named PyMongo which makes the CRUD implementation easy to make. In the project there are various dash modules used for interpreting the html that is written in python. Some of the modules help with graphing and some help with tables.

### Screenshots

In the following images we showcase the functionality of the program. There are four filters on the top each with their dedicated recue type which filters optimal rescue animals for that role. When clicked the table refreshes alongside the graph and map.*Graphical user interface, text

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generatedText

Description automatically generated*

### Challenges

During development I ran into many issues. The biggest issue I had was with the virtual machine which was slow and kept refreshing every 10 to 15 seconds which made the already laggy experience an absolute nightmare. MongoDB during the middle of development was having issue and mixing up the socket file that is being used which made MongoDB refuse access to the connection and because I was in a virtual machine I couldn’t use sudo access to delete the files or use the kill command to restart MongoDB which it did not want to stop after using the stop command on the mongod\_ctl file. I also had issues with getting Jupyter Notebook to render the website eventually I figured out that using the restart kernel and run all cells button would work pretty good but if it didn’t work I had to clear the cookies and cache to fix the issue.

## Contact

Jehu Domenech Ortiz