# CS 340 README Project Two

## About the Project/Project Title

*Grazioso Salvare has requested a user-friendly web application dashboard for identifying and locating potential dogs that could become candidates for their search and rescue training. This web dashboard utilizes the AnimalShelter python module which invokes CRUD functionality within the AAC database and the animals collection through the pymongo driver. The dashboard includes multiple filter options which correspond to preferred dog attributes; presents all animals within a dynamic and interactive data table; includes a geolocation map which identifies the location of a particular animal and also presents a histogram of all breed counts (dependent on the filter applied). Lastly, the dashboard incorporates the Grazioso Salvare logo which links to their website and contains a unique identifier for their dashboard.*

## Motivation

*This project was taken on by Global Rain to provide a full stack development of a software application that can work with existing data from Austin animal shelters to identify and categorize available dogs for Grazioso Salvare.*

## Getting Started

*Simply run the Grazioso Salvare dashboard application and open the link in a new tab to begin exploring the Austin animal shelters dataset. This dashboard is interactive, allowing users to manipulate data tables or filtering options to locate animals based on their needs and preferences.*

*Additionally, if developers wish to utilize this code or refactor it there are a few key ideas to keep in mind. Firstly, access to the database is hardcoded within the dashboard python file. This will most definitely need to be changed before going into production. Secondly, developers must have several libraries and frameworks installed and imported to recreate this dashboard. Particularly: pymongo, the crud python module, dash, dash-leaflet, plotly, os, etc. These can be found in the example code provided. Thirdly, there are several callback methods being utilized in order to dynamically modify the widgets on the dashboard in response to what the user selects either through the provided filter options or through their own use of sorting and selecting via the data table. These callbacks are extremely important for the functionality and viability of the dashboard.*

## Installation

*This module requires pymongo to be installed. Pymongo can be installed via pip. For example:*

*‘‘’bash*

*python -m pip install pymongo*

*‘’’*

*Pymongo is a necessary python driver for communicating with MongoDB databases. This allows developers to access MongoDB methods through python code making functionality more modular, readable, and reusable.*

*Additionally, developers and users alike will have to ensure they have database accessibility in order to utilize the dashboard to retrieve potential dogs.*

*Developers should also install dash (similar to the pip install of pymongo), dash-leaflet, and plotly. These libraries and frameworks are essential for creating and modifying the Grazioso Salvare dashboard.*

*MongoDB is a non-relational database which allows for unique and schemaless collections which is perfect for Grazioso Salvare, as their dataset consists of multiple animal shelters in Austin which may or may not have all of the animal attributes identified or labeled. This is achieved via MongoDB by allowing records to be stored as JSON-like documents that make it flexible for missing or additional attributes. MongoDB also has a powerful and developer-friendly driver which allows for seamless python integration. Consequently, this is advantageous for the Grazioso Salvare project as they wish for an intuitive dashboard to organize and locate dogs. The python module acts as the controller within this full stack development which allows for models from the MongoDB database to be transformed into views on the users’ browser. Through the use of python, MongoDB, dash, dash-leaflet, plotly, and the AnimalShelter module Global Rain and I were able to achieve a functioning and interactive dashboard that met the needs and goals of Grazioso Salvare.*

*A list of references that were used to create this application are as follows:*

[*https://dash.plotly.com/dash-core-components*](https://dash.plotly.com/dash-core-components)

[*https://dash.plotly.com/dash-html-components*](https://dash.plotly.com/dash-html-components)

[*https://dash.plotly.com/datatable*](https://dash.plotly.com/datatable)

[*https://dash-leaflet-docs.onrender.com/*](https://dash-leaflet-docs.onrender.com/)

[*https://plotly.com/python/histograms/*](https://plotly.com/python/histograms/)

[*https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=nlebk&AN=3366175&site=ehost-live&ebv=EK&ppid=Page-\_\_-87&custid=shapiro*](https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip,shib&db=nlebk&AN=3366175&site=ehost-live&ebv=EK&ppid=Page-__-87&custid=shapiro)

[*https://pymongo.readthedocs.io/en/stable/tutorial.html*](https://pymongo.readthedocs.io/en/stable/tutorial.html)

## Usage

*The AAC database is a list of animals sheltered in an Austin animal shelter. Users of the database must be given access by the database admin who will grant access by creating a user role and permissions for an individual. The web application dashboard will allow users to search and identify potential animals, either through filter options provided for specific needs or through manual selection and sorting via the data table. As intended, there are multiple filter options that correspond to Grazioso Salvare’s needs for search and rescue candidates. Additionally, there are widgets incorporated to provide insightful knowledge such as the histogram which counts the quantity of certain breeds from the table and a geolocation map which points to the location of a particular animal.*

### Screenshots

**Starting State**

*A screenshot of a computer

Description automatically generated*

***Water Rescue Filter***

*A screenshot of a computer

Description automatically generated*

***Mountain or Wilderness Rescue Filter***

*A screenshot of a computer

Description automatically generated*

***Disaster or Individual Tracking Filter***

***A screenshot of a computer

Description automatically generated***

***Reset/No Filter Applied***

*A screenshot of a computer

Description automatically generated*

## Roadmap/Features

*This project contains an interactive and functional web application dashboard for Grazioso Salvare to locate potential search and rescue dogs. Future features will include styling overhauls to the dashboard and more interactive widgets to assist in determining candidates. If you have any questions or ideas for future features, feel free to email me at* [*finnegan.thomas@snhu.edu*](mailto:finnegan.thomas@snhu.edu)*.*

***Notes***

*This application was constructed and finalized via a series of milestones which broke down the project into manageable chunks. I first imported a data set into a MongoDB database to hold all pertinent information. Then I granted access to the database and collection to a user and paired this access with a python module that I created for the purpose of CRUD functionality with the collection. Afterward, I experimented with and created a web dashboard with the dash framework for web applications. Following this dashboard, I learned about running a dashboard alongside creating and customizing the widgets. I incorporated an MVC modeled web application by tying in the dashboard with the python module and the MongoDB database. This full stack development met all expectations and needs of Grazioso Salvare. However, this project did not come without its difficulties. Firstly, the provided codebase was incorrect or overlooked which lead to some frustrating errors. However, I was able to overcome these by paying close attention to what the code was intended to do and began to develop my own code to meet the needs. Additionally, I found difficulty in utilizing the dash framework for certain aspects such as using the html wrappers. I found this style quite unintuitive and slow, however, through the use of documentation and consistent checks, I was able to overcome this obstacle.*

## Contact

Your name: Finnegan Thomas