# CS 340 Final Project README

## About the Project/Project Title

The Grazioso Salvare company has hired Global Rain to create a dashboard that aggregates animal shelter data across multiple states to search for dogs that make good training candidates. Grazioso Salvare trains animals for multiple training scenarios, and the company needs an easy way to identify potential candidates quickly and efficiently.

## Motivation

The motivation for this project was to allow a relatively non-technical user to search for and access profile information on potential candidates and metadata from those potential trainings. This process prior to this dashboard was highly manual, and required contacting animal shelters individual to check for candidates, and no common location was kept for any notes.

## Getting Started

Please note that your MongoDB database must be set up to use authentication. Authentication can be set up by starting Mongo DB in authentication mode, creating an access user, and supplying those credentials and the port to the AnimalShelter object instantiation above. An example of this creation can be seen below:

db.createUser(

{

user: "aacuser",

pwd: "aac123",

roles: [ { role: "readWrite", db: "AAC" } ]

}

)

Finally, the create function is defined with a returned Boolean to indicate the result of the insert. It will bubble-up the failure if one is present. The read function will return the cursor if data is found but does not indicate its success or failure.

## Installation

To utilize the module and dashboard, you will need the following items:

* Python >3.6
* Pymongo >1.4 (Package)
* ObjectId (Package)

## Usage / Functionality

The required functionality for this dashboard was to include Grazioso Salvare branding on the dashboard. The dashboard needed to contain a grid control to show data, and two data widgets. In my example, this was a scatterplot and map showing where the animal was located. Additionally, all controls needed to be driven by an interactive filter which in my example was a Radio Item group containing the 4 filter profiles; Water Rescue, Mountain Rescue, Disaster Rescue and Unfiltered (or Reset). Under the hood, we defined each mode via a specific subset of categories; specific age and breed limits as well as neutering requirements.

**Toolset**

For this project, we made extensive used of the DASH framework in order to setup our dashboard and its widgets. The DASH framework provides a set of functionality via callbacks that allow data to flow in the client side code between widgets based on events (and this exposed via decorators within the code). This is a pretty effective model for something as light weight as this dashboard, because it doesn’t require heavy server calls to a back end api in order to data manipulation and feed data to controls on the page.

MongoDB was used as the backend for this project, primarily due to its flexibility; we are able to easily define collections without needing to run extensive set up scripts, and the PYMongo drivers allow for easy querying from those tables.

### Challenges

During this project, a number of small challenges came up. First off, we had to think about logically how each widget on the page interacted with each other. We came to a design decision that the grid will query and update the dataset first, and each other widget will get that data from the grid. This was done to ensure that no item on the page had the wrong data as we could get into a race condition if all widgets were trying to get and refresh there data directly from the data frame.

Once this issue was resolved, I had another small issue with the ploty express library where I could not get the pie chart to display data, and the library was not emitting any errors. I decided to use a different chart altogether to step side this issue entirely.

### Screenshots

Table

Description automatically generated

Figure Initial load-up of application showing unfiltered grid

A picture containing map

Description automatically generated

Figure Initial load-up of application showing scatter widget and map

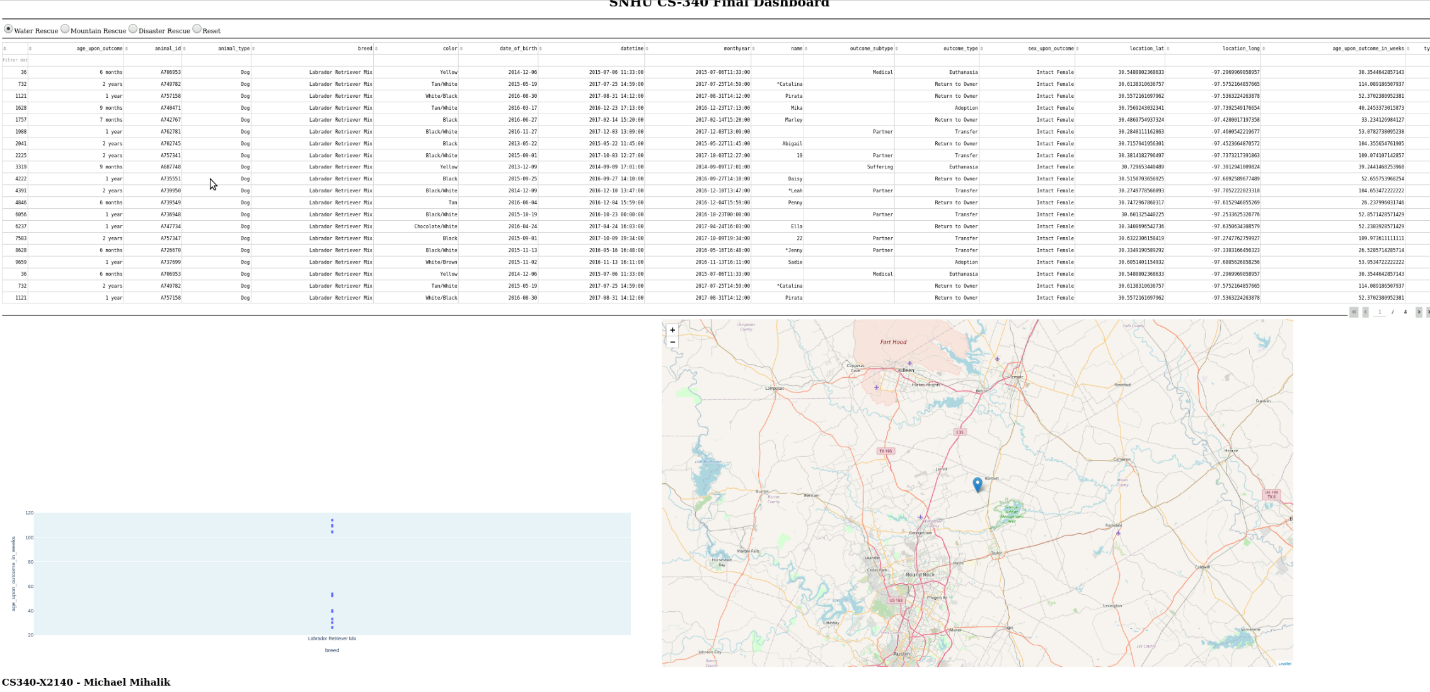


Figure Water Rescue profile selected

Graphical user interface, application

Description automatically generated

Figure Mountain Rescue profile selected

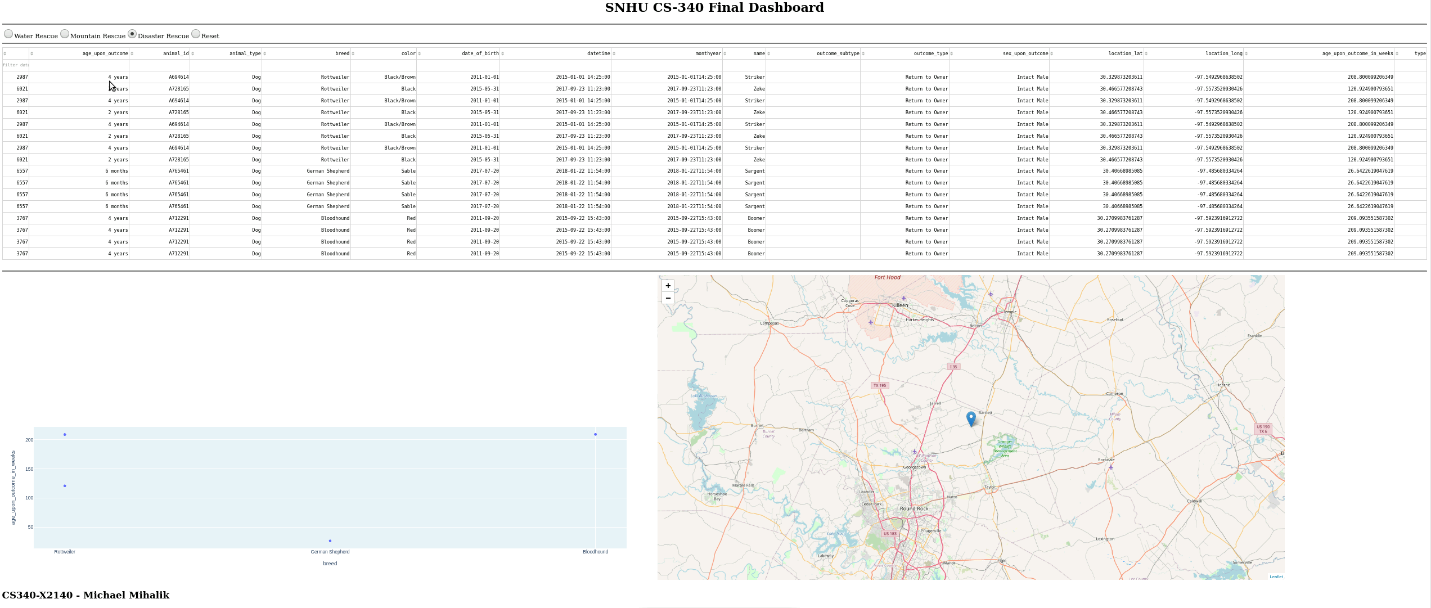


Figure Disaster Rescue profile selected

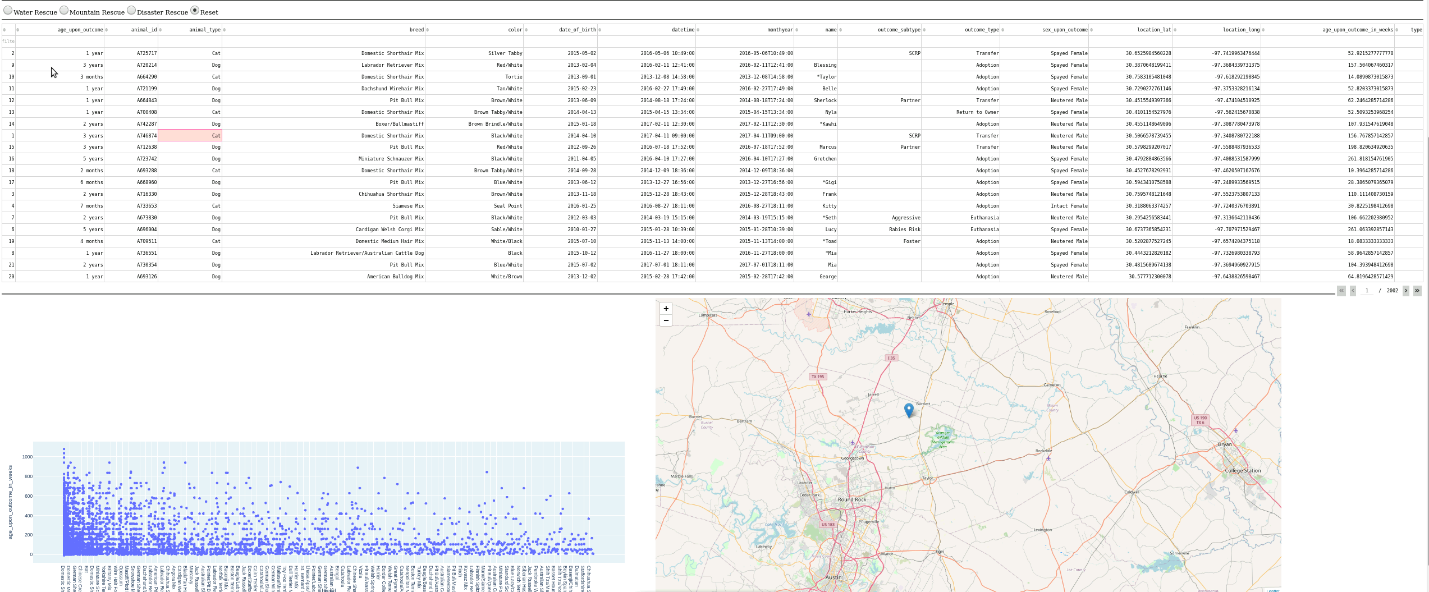


Figure Reset option selected

## Contact

Your name: Michael Mihalik