# CS 340 Project Two README

## About the Project/Project Title

This project will allow users to access information regarding animals at an animal shelter through a unified, streamlined dashboard. The dashboard will use CRUD functionality (Create, Read, Update, and Delete) on an uploaded database to display information to the user.

## Motivation

This project exists because our client, Grazioso Salvare, wanted a software application that can work with existing data from animal shelters to identify and categorize available dogs for training/sear-and-rescue purposes.

## Getting Started

To get a local copy of this project up and running, simply change the “username” and “password” variables of the CRUD object to reflect your MongoDB environment needs. Be sure you have an Admin user account so you can properly access your database.

You will also need an up-to-date version of Python with a Dash Framework implementation.

## Installation

The following tools were used to get this software application up and running:

* **MongoDB –** MongoDB can be installed from <https://www.mongodb.com/docs/manual/installation/>. MongoDB will be used to host the database and includes a Python driver. This makes interactions between the animal shelter database and our Python program seamless.
* **Python –** Python works with both MongoDB and Dash, and an up-to-date version can be installed from <https://www.python.org/downloads/>. For this project, I used Jupyter notebook as a Python IDE.
* **Dash Framework –** The Dash framework is an open-source framework that is used for building data visualization interfaces. In this project, it will work with our Python program (controller) to “display” information (view). Information regarding how to properly implement the Dash frameworks can be found at <https://dash.plotly.com/>.

## Usage

After instantiating your CRUD object, you will need to create functions to filter and display the desired information from the animal shelter database. For this project, I used separate functions to filter information, display information, and generate a styled pie chart. I also used a function to generate and display an interactive map to show the location for any animal selected by the user.

As for the overall layout of the app, stylizing is created using html elements and the dash framework.

### Code Example

### Method to Filter Rescue and Breed Types

### *A screenshot of a computer AI-generated content may be incorrect.*

### Method to Generate Pie Chart

### *A screenshot of a computer code AI-generated content may be incorrect.*

### Method to Generate and Display Geolocation Map

### *A screenshot of a computer code AI-generated content may be incorrect.*

### *A computer code with text AI-generated content may be incorrect.*

### Screenshots

The following screenshots are images of the application after execution:

**Default Screen**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Filtering to Water Rescue**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Filtering to Mountain/Wilderness Rescue**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Filtering to Disaster/Individual Tracking**

**A screenshot of a map

AI-generated content may be incorrect.**

## Contact

Grace Fletcher – [grace.fletcher@snhu.edu](mailto:grace.fletcher@snhu.edu)