# CS 340 README

## About the Project/Project Title

The purpose of this project is to create a python program using object-oriented programming that can import a CSV file, incorporates CRUD functionality and includes a DASH framework to interact with this database. This is meant to create a dashboard for users to interact with the database and ensures access is restricted to users with the correct permissions. There will also be an interactive data table and geolocation chart available to sort animals by type of rescue capabilities.

## Motivation

This project exists as a means of storing information for a rescue-animal training company. The main purpose of this project is to identify dogs for training in search and rescuer operations. The application will take information from shelters that currently have dogs and strives to be able to search within for viable candidates for search-and-rescue animals. There will be a visual representation of available dogs, as well as a map for geolocation.

## Getting Started

To get a local copy up and running, follow these simple example steps:

Import the CSV file containing the shelter’s information on current animals.

Log into a user account on Mongo.

Use the Create method to add files

Use the Read method to find files

Use the Update method to update existing files

Use the Delete method to delete files as needed

Then incorporate a DASH framework with this Python module that includes interactive buttons and search functions for the three different types of animals users can search for: Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Rescue. There should also be a button to reset search parameters. This information will be used to display a map and a graph of available animals.

## Installation/Rationale of Choices

A database was initially created for use with MongoDB. Then a program was written to incorporate this database into a python program that is meant to have CRUD functionality. The program was written in Spyder in Python. Then testing functionality was done using Jupyter. MongoDB was chosen to due its scalability and smooth incorporation with Python code. MongoDB also includes official drivers and libraries for Python, allowing easy incorporation. The Python code in Jupyter incorporated a DASH framework that allowed visualization and interaction with the data using widgets and images. DASH was chosen because DASH is Python-based and also can incorporate plotly, allowing for the map functionality that Grazioso is looking for. Animal latitude and longitude coordinates were used in the creation of the interactive map. Below you will find proof of the functionality in the form of screenshots.

## Usage

*Use this space to show useful examples of how your project works and how it can be used. Be sure to include examples of your code, tests, and screenshots.*

### Code Example

*# Example data to insert*

*example\_data = {"age\_upon\_outcome" : "2 years", "animal\_id": "B746874", "animal\_type":"Dog", "breed" : "Pug", "date\_of\_birth" : "4/10/2014", "datetime" : "4/11/2017 9:00", "name" : "Archer", "outcome\_subtype": "Transfer", "sex\_upon\_outcome" : "Neutered Male", "location\_lat" : 30.50665787, "location\_long" : -97.34087807, "age\_upon\_outcome\_in\_weeks" : 145.949}*

*# Create a document*

*if crud.create(example\_data):*

*print("Document inserted successfully!")*

*else:*

*print("Failed to insert document.")*

### Tests

**Original State**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Mountain or Wilderness Rescue**

**A screenshot of a computer

Description automatically generated**

**Disaster or Individual Tracking**

**A screenshot of a map

Description automatically generated**

**Water Rescue**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Reset**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

## Roadmap/Features/Challenges

There were a few challenges encountered when working on this project. Unfortunately, I was not able to get the logo to load appropriately. Working o the project on a remote server like Apporto, then trying to import an image in the project proved to be challenging. I kept getting security warnings when running the application on a local server and checking the console with f12. I tried many ways to incorporate the image and could not end up getting it to load at all. I can see it in my files uploaded to Apporto. I checked the permissions to ensure it could be read. I checked the file path to ensure I had tried to load it in the application correctly. None of this seemed to work and the default browser on the remote desktop blocked it and cited security each attempt. In the future, having the image load correctly would be important.

## Contact

Mikaela Spence