# CS 340 README

## About the Project/Project Title

Our software team has been assigned to work on a system for an innovative international rescue-animal training company, Grazioso Salvare. Grazioso Salvare identifies dogs that are good candidates for search-and-rescue training. This is a system that will work with existing data from the animal shelters to identify and categorize available dogs to train for search and rescue. The application will include a database and a client-facing web application dashboard.

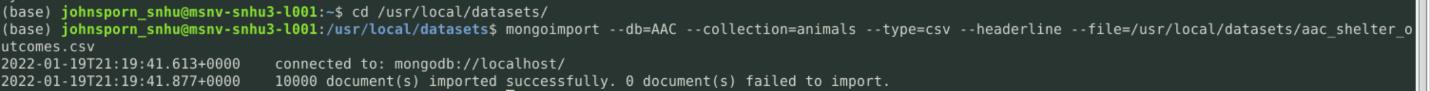
## Motivation

The system will be used to identify dogs through five animal shelters that are good candidates to be trained in search-and-rescue. Once trained the dogs will be used to help find and rescue humans and other animals in life threatening situations such as locating someone after a disaster has occurred. The system will help identify specific dog breeds for each type of rescue.

## Getting Started

Users will need to download the project files and open them in a python IDE.

They will also need to open their command prompt and import the animal data set CSV file using the mongoimport tool:



Then they will need to authenticate a user for the database by first creating an administrator account and then creating a user account for the AAC database:

Text

Description automatically generated

Text

Description automatically generated

Now the application is ready to go.

## Installation

The system requires a few tools to use the application. First, install Mongodb and the latest version of Python. Then download the following extensions through the command prompt:

* pip install pymongo
* pip install jupyter\_plotly\_dash
* pip install dash
* pip install pandas

Download the project files to your computer. This includes the AnimalShelter.py file and the ApplicationDashboard.ipynb file.

## Usage

This project will be used as a framework for users to store and process animal data for search and rescue training. The use of pymongo and jupyter\_plotly\_dash allowed us to create an easy-to-use user interface to make these processes easier for the user.

### Code Example

CRUD: AnimalShelter class

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

Front-end code snippets for dashboard:

Importing AnimalShelter.py CRUD class into the ApplicationDashboard.ipynb file

Text

Description automatically generated

Code for drop down menu above data table

Text

Description automatically generated

Queries for finding rescue specific breeds

Text

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

### Dashboard Screenshots

Default view of Dashboard

A picture containing diagram

Description automatically generated

Table

Description automatically generated with medium confidence

Map

Description automatically generated with medium confidence

Query for water rescue dogs

Graphical user interface, table

Description automatically generatedGraphical user interface

Description automatically generated with medium confidence

Query for mountain and wilderness rescue dogs

Graphical user interface, application, table

Description automatically generated

Graphical user interface, map

Description automatically generated

Query for disaster or individual tracking rescue dogs

Graphical user interface

Description automatically generated

Resetting data table to default

Graphical user interface, application, Excel

Description automatically generated

Geolocation change upon row selection

Graphical user interface, map

Description automatically generated

Graphical user interface, application

Description automatically generated

## Roadmap/Features

Future features will provide multiple marker locations to pop up on map. Adding the ability to select multiple rows and displaying each individual location of selected rows on map.

## Contact

Your name: John Sporn