# AnimalShelter README

## About the Project/Project Title

This project is to develop software for Grazioso Salvare, which is an innovative rescue-animal training company. When trained, some dogs are able to find and help to rescue humans or other animals, often in life-threatening conditions. This project allows users to identify ideal candidates from five animal shelters in the region around Austin Texas.

This program works with existing data from the animal shelters to identify and categorize available dogs and present it in a user-friendly, intuitive interface that will reduce user errors and training time

## Motivation

The motivation behind the project is to increase and grow the company’s ability to find good candidate dogs for rescue training that are currently up for adoption. The project aims to reduce the training time and user errors in the rescue dog training process.

## Getting Started

To utilize this library, you must have the Austin Animal Center (AAC) CSV file imported into your MongoDB database and then you must enable authentication for the collection by following these steps:

1. Import the CSV file with the following command: mongimport –port ##### --db AAC –collection animals –type csv –headerline ./aac\_shelter\_outcomes.csv

NOTE: ##### should be replaced with the port of your MongoDB connection

1. Enable authentication for the AAC collection: mongo –authenticationDatabase “AAC” -u “aacuser” -p

## Installation

The module uses MongoClient from pymongo and ObjectId from bson.objectid. The database is populated from an Austin Animal Center (AAC) CSV file. MongoDB was chosen as the model component because of its ability to quickly search through and filter indexed documents and its ability to interface with Python via several tried and tested modules. The project uses the Dash framework to provide the view and controller structure for the application. Plotly was used to create the chart.

## Usage

### Code Example

To create an instance of AnimalShelter you must pass in a valid username and password:

*animals = AnimalShelter("username", "password")*

To create a new document, use the create function of the AnimalShelter library:

*animals.create ({*

*'age\_upon\_outcome': "3 years",*

*'animal\_id': "test",*

*'animal\_type': "Dog",*

*'breed': "Cocker Spaniel",*

*'color': "Brown",*

*'date\_of\_birth': "2019-05-01",*

*'datetime': "2022-05-29 05:43:00",*

*'monthyear': "2022-05-29T05:43:00",*

*'name': "Poppy Jr",*

*'outcome\_subtype': "Foster",*

*'outcome\_type': 'Adoption',*

*'sex\_upon\_outcome': "Female",*

*'location\_lat': 55.4434533,*

*'location\_long': 53.4324343,*

*'age\_upon\_outcome\_in\_weeks': 105.2})*

Successful document creation will return True and unsuccessful document creation will print “Error creating document” to the screen and return False.

To search documents, use the read function of the AnimalShelter class along with any number of search criteria in a dictionary such as:

*animals.read({“breed”: “Cocker-Spaniel”, “name”: “Poppy Jr”})*

To update documents, use the update function of the AnimalShelter class with any number of key/value pairs passed in to find the documents needing updating and any number of key/value pairs of entries to be updated in the found documents.

*animals.update({“breed”:”Cocker-Spaniel”, “name”:”Poppy Jr”}, {“name”:”Poppy Sr”}*

The update function returns the result.raw\_data for successful updates or error message for failed updates.

To delete documents, use the delete function of the AnimalShelter class with any number of key/value pairs passed in to find the documents that need to be deleted.

*Animals.delete({“breed”:”Cocker-Spaniel”, “name”:”Poppy Jr”})*

The delete function returns the result.raw\_data for successful deletions or an error message for failed deletes.

### Tests

To perform tests the application was run and each of the options required were selected and verified to return the correct data. Screenshots of these test are below.

### Screenshots

Installing have the Austin Animal Center (AAC) CSV file:

A screenshot of a computer

Description automatically generated

Enabling authorization of the AAC collection with user “aacuser”:

A computer screen capture

Description automatically generated with medium confidence

Testing the AnimalShelter module:

Text

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Tests of the application at initial startup:

Graphical user interface, table

Description automatically generated

Graphical user interface, chart, application

Description automatically generated

Screenshot showing the Water Rescue option selected:Graphical user interface, application, table

Description automatically generatedGraphical user interface, chart, application

Description automatically generated

Screenshot showing the Mountain Rescue option:

Graphical user interface, table

Description automatically generated

Chart, pie chart

Description automatically generated

Screenshot showing the Disaster Rescue option:

Graphical user interface, application, table

Description automatically generatedChart

Description automatically generated

Screenshot showing the reset option after another option previously:

Graphical user interface, table

Description automatically generatedGraphical user interface, chart, application

Description automatically generated

## Contact

Scott Vanderwilt

Scott.vanderwilt@snhu.edu