# CS 340 README Animal Shelter

## Project Two: Animal Shelter

*This application allows a user access to a database of animals in CS 340 Austin Animal Center (AAC) to search and filter the database. The user would search using a filter for animals with Grazioso Salvarethe requester of this project to create a dashboard. Additionally this project allows a user to benefit from geolocation mapping, user interaction, and charts to locate and utilize animals for search and rescue.*

## Motivation

*This program was designed to test my skill working with databases and manipulating the data within. Python was chosen as the language to drive for mongodb. Python is easily written, manipulated and works well with mongodb. Additionally, Python can be compiled quickly, with a built in compiler using Jupyter Notebook.*

## Getting Started

*To get this program started you would first,  
1. Enter and Mongo and import the csv file aac\_shelter\_outvome.csv.*

*2. next one would want to create a simple and a complex index to parse the data stored within the document.*

*3. Now to authenticate a user would want to create both an Admin account and a aacuser account to access the database.*

*4. next a user would need to have access or install python and run the program out of a notebook.*

## Installation

*A current version of Python to run both the .py and the .ipynb files*

*MongoDB - to access the database.*

## 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

*The code allows a user to test, add, edit, read, and delete animals in a shelter. To test this a user would use. After starting mongo and loading the python files necessary to run the program a user could enter print(animals.create (STRING\_TYPE) to add animals with the program throwing a boolean if it is successfully added or an error if not added.*

* **A Create method that inserts a document into a specified MongoDB database and collection**
  + Input -> argument to function will be a set of key/value pairs in the data type acceptable to the MongoDB driver insert API call.
  + Return -> “True” if successful insert, else “False”.

Graphical user interface, text, application, email

Description automatically generated

* **A Read method that queries for document(s) from a specified MongoDB database and specified collection**
  + Input -> arguments to function should be the key/value lookup pair to use with the MongoDB driver find API call.
  + Return -> result in cursor if successful, else MongoDB returned error message.

Graphical user interface, text, application, email

Description automatically generated

* **An Update method that queries for and changes document(s) from a specified MongoDB database and specified collection**
  + Input -> arguments to function should be the key/value lookup pair to use with the MongoDB driver find API call. Last argument to function will be a set of key/value pairs in the data type acceptable to the MongoDB driver insert API call.
  + Return -> result in JSON format if successful, else MongoDB returned error message.

Graphical user interface, text

Description automatically generated

* **A Delete method that queries for and removes document(s) from a specified MongoDB database and specified collection**
  + Input -> arguments to function should be the key/value lookup pair to use with the MongoDB driver find API call.
  + Return -> result in JSON format if successful, else MongoDB returned error message.

Graphical user interface, text

Description automatically generated

### Tests

*This code was tested using an invalid statement of print(animals.create(0:0)) making an invalid argument as it tries to create an invalid data type.*

*To search for your added animal and ensure it was added one could use query = animals.read({”name”: “NAME”})*

### Screenshots

Importation of AAC Animal database:

Graphical user interface, text, application

Description automatically generated

user authentication:

Text

Description automatically generatedGraphical user interface, text

Description automatically generated

Authentication:

Text

Description automatically generated

New entry creation with boolean outcome:  
Text, letter

Description automatically generated

Invalid creation script with outcome:

Text

Description automatically generated

Query to locate my created animal:  
Text, letter

Description automatically generated

In valid query to search for animal:  
Graphical user interface, text, application, email

Description automatically generated  
  
Animal Documentation update:  
Graphical user interface

Description automatically generated  
Text, letter

Description automatically generated  
Invalid documentation update:  
Graphical user interface, text, application, email

Description automatically generated  
Valid documentation deletion:  
Graphical user interface, text, application, chat or text message

Description automatically generated

Invalid deletion:  
Text

Description automatically generated

## Contact

Your name: Ethan D.