# CS 340 README V.2

## About the Project/Project Title

Grazioso Salvare rescue-animal training database. As that title implies, this project involves the database of animals for a rescue-animal training company.

## Motivation

The intent of this project is to provide an interface that will allow someone with access to the provided animal shelter database to easily search for an animal using any available field, to add new entries to the collection, to update any entry, or to delete entries when necessary.

## Getting Started

To use the created code, simply download and install the corresponding .py file. Once that is done, you should be able to search the collection or create a new entry (update and delete functions to be added in a future version).

## Installation

All that is required to use this code on your personal computer is MongoDB. From the Mongo interface, type the word import, followed by the .py file’s name.

Mongo is being used for this database because it was chosen for me. Having now used it, I can see why it has become among the most popular modern day database software. The code is simple to understand and implement, and there is no shortage of resources and community support for it.

## Usage

### Code Examples

# Method to implement the C in CRUD.

def create(self, data):

if data is not None:

self.database.animals.insert(data) # data should be dictionary

else: error message

# Method to implement the R in CRUD.

def read(self,data):

return self.database.animals.find\_one(data) #returns exactly one document as a python directory

# Method to implement the D in CRUD

def delete(self, data):

if data is not None:

self.database.animals.remove(data) #deletes document

else: error message

# Method to implement the U in CRUD

def update(self, filter, data):

if data is not None:

self.database.animals.update(filter, data) #updates document

else: error message

#code for interactive filtering Radio buttons

dcc.RadioItems(

['Water Rescue', 'Wilderness Rescue', 'Tracking', 'Reset'],

'Reset',

#inline=True

)

#pie chart code

dcc.Graph(

figure = px.pie(df, values='percentages', names='breed')

)

### Tests

The code’s CRUD functionality can be utilized by creating an AnimalShelter instance, then using the line:

data = {“valid key identifier”: ”corresponding info for new entry”, “optional second key”: “etc”}

After that line, call the desired function (either shelter.create(data), or shelter.delete(data))

Calling the “Update” function requires an additional “filter” field.

The interface can be tested simply by clicking any part of it. The four radio buttons can be clicked to filter data, and there are scroll bars to find any specific cell of information.

### Screenshots

*Graphical user interface, text, application

Description automatically generated*

*(“create” functionality shown above)*

*Graphical user interface, application

Description automatically generated*

*(“delete” functionality shown above)*

*Graphical user interface, text

Description automatically generated*

*(“update” functionality shown above)*

## Roadmap/Features

V1: As mentioned, future versions of this project should add “update” and “delete” functionality. In addition, ease of use improvements and visual tweaks are being considered for the interface.

V2: “delete” functionality added. “update” functionality added.

V3:

-added return values to Delete and Update methods

-added radio buttons to interface for 4 different filters on rescue animals

-added company logo to interface

-began work on adding a pie chart to interface

## Contact

Created by: Travis Garlick