## Gage Biros

CS-340

ReadMe

## About the Project/Project Title

The purpose of this software is to quickly organize information from animal shelters around the to populate a database that will help us find perfect candidates for our training program.

## Motivation

We are making this software so clients can quickly identify specific characteristics of animals from different shelters that would meet the requirements for search and rescue dogs. This software is critical since the criteria is specific, so we will ensure that the client is able achieve their needs with our product.

## Getting Started

This has all been completed in a linux shell, so if there are issues check to make sure everything is compatible before continuing. We must import the CSV file so that we can access the data within the file. Also, we must ensure that we are using a valid socket extension, or if there is a main socket, we use that and identify that in the import function. Once the document has been imported into mongo, then we can access this data in the mongo shell.

Text

Description automatically generated

## Installation

Linux

MongoDB

## Usage

We want the customer to search through the database to find the ideal candidate for their operation. The end user of this application should be able to use different functions to sort through the data table to maximize efficiency. We must initialize our code with the necessary items, every project is different so be sure to analyze what is needed for the project to reduce the file size and save the ambiguity of having some unused assets in the program

Text

Description automatically generated

We then must then import our CRUD class into the source code.



The next step would be to set up the data manipulation items, we will need our login for the specific account we want to use, the account we will be using in the example is “aacuser”, we want to also initialize the password, it may be easier to fill in the password at this point although you do not need to. We also want to initialize our cursor object so we can retrieve the information we are searching for.

Graphical user interface, text, application

Description automatically generated

In this program I initialized the data table to have some quick access functions for the columns, by clicking the arrows there is a quick sort feature. There is also a field in which the client will be able to just search for a key word that they want returned in that column.

Graphical user interface, text, email

Description automatically generated

One tool that this application has is the filter by Rescue type button selection. The three rescue types that we want to look for, are listed in the buttons located near the top of the data table. The rescue types are not listed in the CSV file, this was supplied by the customer which allowed me to code in the necessary conditions that would return the results we wanted from the csv file. The user can quickly filter the data by selecting the animal rescue type they want to search through to only show results that match that specific animal rescue type

Text

Description automatically generated with medium confidence

We also want to properly position the visuals on our GUI, so we must also modify the code to achieve this step.

Graphical user interface, text, application, chat or text message

Description automatically generated

**DATA INTERACTION**

At this point in the code, we have a real solid starting point, but now we will need to start interacting with the data through our definitions and our app callbacks to be able to actually manipulate the data table.

To sort through the animal rescue types we need to initiate conditions so we can get those results to actually populate the data table:

Text

Description automatically generated with medium confidence

**Helpful Extras**

We have everything we need to achieve the goal of finding the results we are looking for within our CSV file. Sometimes visuals can be helpful. Here are some examples of some other tools we can implement into our application to help the end user understand the data:

**Pie Graph**

**Graphical user interface, text

Description automatically generated**

**Mapped results**

**Graphical user interface, text, application

Description automatically generated**

I believe my biggest struggle with this project is doing this in the linux operating system. I was getting an error for a while that just kept telling me that my CRUD class didn’t exist. I was able to actually get the code to work, and then I was wrapping up and I ran into the same problems, I couldn’t get screen shots of my work to show that it works unfortunately. I think I would need to spend more time with linux and have a better understanding and I believe I could be an expert in MongoDB.