# CS 340 README Project 2

## About the Project/Dashboard Application

The purpose of this project is to allow users to access an animal database, as well as create a dashboard to visualize the data for users. Currently, users are only able to create and read animals from the stored data in the database, and there is no medium that displays the data. We will create a client-facing dashboard application that users can access to view and filter data on rescue animals.

## Motivation

The desire for this project is to allow users to create and search animals in the animal database. There is no client side visualization for the rescue animal data.

## Getting Started

1. Enter the MongoDB terminal
2. Import the csv file “aac\_shelter\_outcome.csv”

A computer screen with text

Description automatically generated

1. Enter your user credentials for authenticationA screenshot of a computer screen

   Description automatically generated
2. Use Python to use CRUD functionality
3. Use Python to Create Dashboard Display

## Installation

In order to use the CRUD functionalities, you must have a current version of Python is needed to run py files. We will also establish the CRUD methods that we will use in order to manipulate and read our data. Here is a sample of some code I used:

A screenshot of a computer code

Description automatically generated

The C in CRUD (Create)

A computer screen shot of a computer code

Description automatically generated

The R in CRUD (Read)

A computer code with green and white text

Description automatically generated

The U in CRUD (Update)

A computer code on a white background

Description automatically generated

Finally, the D in CRUD (Delete)

A computer code with black text

Description automatically generated with medium confidence

We start the dashboard code by importing necessary libraries, initializing the dash application, setting up the connection to our MongoDB inputting the chosen credentials.

A screen shot of a computer program

Description automatically generated

We configure the dash to retrieve data from our database, as well as creating the layout of our dash using html.

A screen shot of a computer program

Description automatically generated

Here we create a callback that will update table style upon selection and we create a function that will update the mapping based on the selection of table

A screen shot of a computer code

Description automatically generated

Finally, we create a call back for the geolocation chart that will call the leaflet function update\_map.

A screen shot of a computer code

Description automatically generated

## 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.*

**Tests**

Now that we have established our methods lets see if they work, starting with animals.create()…

A screenshot of a computer code

Description automatically generated

Next, we will look to read a query from our database using animals.read()…

A close-up of a computer screen

Description automatically generated

Now, lets test the update functionality with animals.update()…

A screenshot of a computer

Description automatically generated

Finally, we will look to delete a query using animals.delete()…

A close-up of a computer screen

Description automatically generated

Running the two python files alongside each other we verified the interactive data table displayed the AAC data set correctly. We also verified that the geolocation chart updates based on the selected row in the data table.

## Contact

Your name: William Tunstall