# CS 340 README

## About the Project/Project Title

*This project is for interacting with data from the Austin Animal Shelter. There is a database with all of the animals and data about them. The database will be fully functional and have multiple users with different privilege levels.*

## Motivation

*The motivation behind this is to help the Austin Animal Shelter run more efficiently. This way all of the data surrounding the animals and their care will be in one simple to use place. There will also be a hierarchy of privileges so that there is less chance of accidental corruption of the data.*

## Getting Started

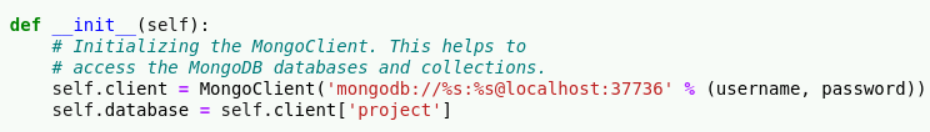
*To get started will depend on your position. If you are just a user with read privileges then getting started will be quite easy. It will just require a username and password to be set up by the administrator and then you will need to use some simple commands to pull up the desired information. If you have read/write privileges for the animal collection then it will be important to familiarize yourself with the documentation of that collection as well as specific commands and protocols. If you are the administrator then it will be important to know how to set up new users, maintain the database, edit and update commands, as well as be able to explain processes and updates to the rest of the organization.*

## Installation

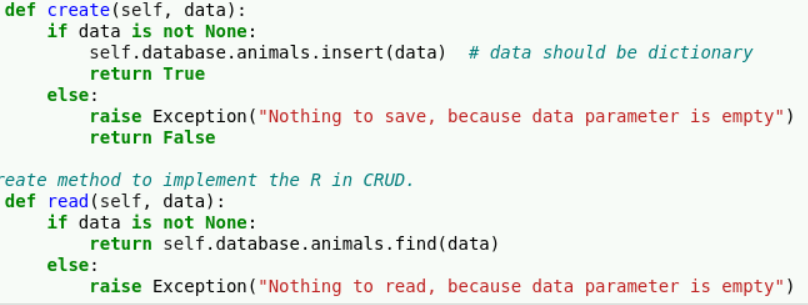
*Installation will depend. If the database is eventually hosted in a cloud with remote access then it will be different than one that is locally installed. At the moment the data is stored locally and imported from a csv file. Therefore, there would be nothing to install because it will have already been installed when being worked with. It may be helpful to understand the basics of Jupyter Notebook and Mongo.*

## Usage

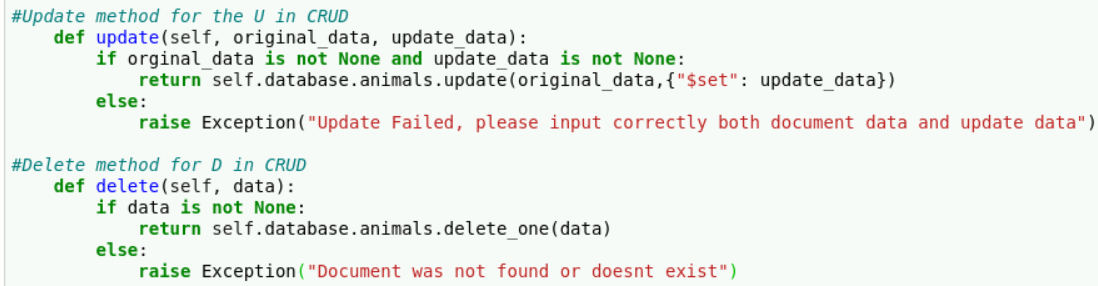
### Code Example

**

Here we can see the initial connection to the database via authentication.



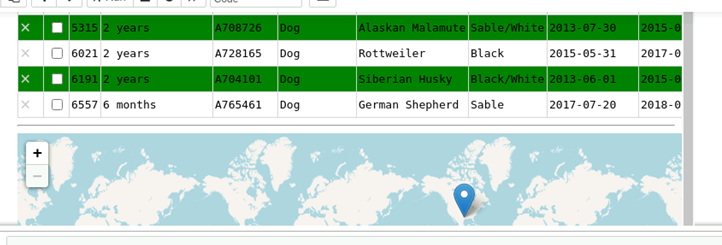
Here we can see the create (C) and read (R) parts for the CRUD and how exception handling is done.

**

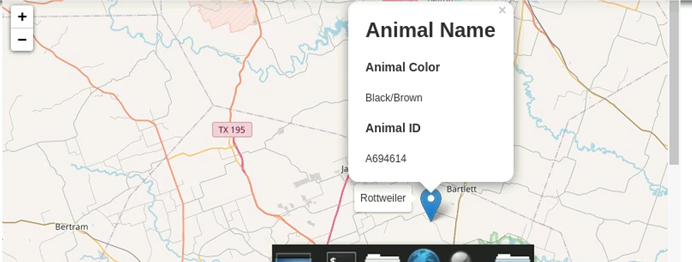
Here we can see the update (U) and delete (D) of the CRUD functionality. Update takes two parameters along with the self. “original\_data” is used to find the document one desires to update. The “update\_data” is the data that will replace the old one. The delete method just searches for the document and then removes it

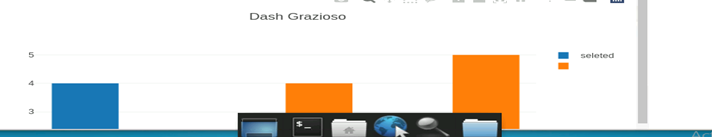
**Dash**

The dash is quite interactive, it allows for searching all the animals as well as specific requirements for Water Rescue, Wilderness Rescue and Disaster Rescue. It also contains a data table which filters those options. As well as a geolocation chart for the animal and bar charts demonstrating data points of the animals. Screenshots are below:









### Tests

*There has not been much testing of the program done yet but there has been a small amount to check if the create and read work properly. Here it is:*

**

### Screenshots

*Here is the output*

*There was also testing done on the Update and Delete portion, here it is:*

**

**Steps taken**

First we set up a database containing all of the animals and accompanying data. Then we implemented the CRUD method in python. We used Mongodb and python because they work very well together and are simplistic. Mongodb works well for this purpose because the non-relational documents Mongodb uses makes much more sense than a relational one due to the data. Python both works well with Mongodb as well as being simple to use and secure. After we set up the skeleton with the database and CRUD methods, we then made the dash. This was done to resolve the challenge of working with databases for the users. The dash was used to make it easy for non-technical users to interact with the database. This way users do not have to actually write queries or understand any code to use it.

## Contact

Your name: Christopher Eldred