# CS 340 README

## About the Project/Zoo

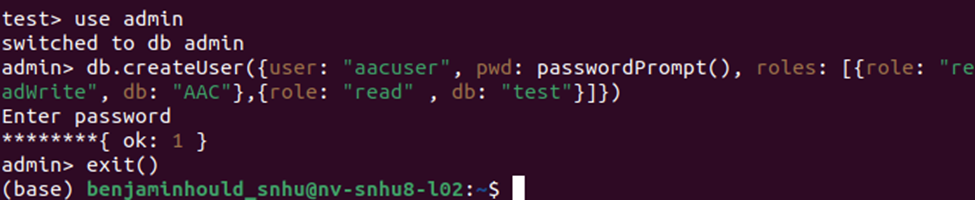
This project is about providing an easy access to animal data from five Austin, Texas Animal shelters. The data provided will be put into a MongoDB. Our custom code will interact with the database to provide a graphical user interface that can provide the functionality… Project Two stuff goes here

## Motivation*.*

Grazioso Salvare is a rescue-animal training company. They have a need to select the best candidates for their training program and have gathered data from five local animal shelters to do so. This project is to help make that information easy to navigate and access so employees at Grazioso Salvare can make more informed decisions about training candidates.

## Getting Started

1. Make sure you are working Linux as your operating system.
2. Import the necessary tools listed below in the Installation section.
3. Use the Linux shell to upload your data set into MongoDB by importing a CSV file using the mongo import command.
   1. Here is an example of this import command  
      A computer screen with white text

      Description automatically generated
   2. If you are struggling with this step, documentation on how to use the mongoimport command can be found at [mongoimport - MongoDB Database Tools](https://www.mongodb.com/docs/database-tools/mongoimport/?msockid=280e5ce7e397604705c84f5de26a611e)
4. Authenticate a new user. Run the command mongosh in the Linux shell to enter the mongo shell. Navage to the admin database by using command use admin. Once switched to admin authenticate a new user with the createUser command.
   1. 
   2. If you are struggling with this step, documentation on how to use the createUser command can be found at [db.createUser() - MongoDB Manual v8.0](https://www.mongodb.com/docs/manual/reference/method/db.createUser/?msockid=280e5ce7e397604705c84f5de26a611e)
5. To access the custom python code navigate in the shell to the folder with the zoo.py file and run the python command. This will allow you to run python commands right in the terminal.
6. Run the command import zoo
7. Create an Animal Shelter object by replacing the user and password with the username and password you used to import the database. my\_object = zoo.AnimalShelter("username", "password")

## Installation

MongoDB [MongoDB: The Developer Data Platform | MongoDB](https://www.mongodb.com/)

Python [Download Python | Python.org](https://www.python.org/downloads/)

## Usage

### Code Example

The custom python code can read and create objects from the MondoDB as shown below.

*A screenshot of a computer program

Description automatically generated*

*A screen shot of a computer

Description automatically generated*

### Tests

Test\_zoo.ipynb is the test used to verify the code in zoo.py. This can be ran in a browser and should provide outputs as shown below.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

*.*

*A screenshot of a computer

Description automatically generated*

*A screenshot of a map

Description automatically generated*

## Roadmap/Features (Optional)

* *Fix Bugs*

**Describe the required functionality.**

The user interface should display the Grazioso Salvare logo, a datatable with options to filter for certain types of candidates based on rescue type, as well as a map and chart showing information about these candidates. The filtering options will be Water, Mountain and Disaster Rescue and will filter for the proper breed, sex, and age of Dog best suited for training.

**Tools Used**

MongoDB is used as the database to store the data collected from the animal shelters.

Zoo is our custom mid layer build on using python and the MongoDB.py module.

Dash was the framework used to produce the application, that the user will interact with.

**Project Steps**

A mongo Data base was created and the data from the animal shelters around Austen was imported. We set up proper user accounts for security and control.

We created a mid layer application program interface so that we could make calls to the database indirectly.

Using the Dash framework we created a web application to view and filter information from the database to make it accessible.

**Challenges**

I spent about 20 hours trying to get the proper data to appear in the application and have it change depending on the selection and I could not figure it out. The template started with errors that I thought would get fixed as I filled in the missing pieces but never did. I couldn’t replicate the given example, or the examples in the Dash documentation, or even with the help of a LLM. From start to end I would get the same Callback error / TypeError. I never figured out why it had two names or why it was coming up. Without fixing this I couldn’t get a graph to populate or the table to change. I need someone who give it a look over and tell me what I’m missing. But sadly too late for that :(

## Contact

Benjamin Houldridge   
bhouldridge@snhu.edu