# Alex Crosswhite

2/25/24

CS 340

SNHU

# Web Application Dashboard README

## About the Project/Project Title

This web application allows for the user to access databases and display the information in an organized manner.

## Motivation

Grazioso Salvare required a user interface that reads, sorts, and displays data charts to find dogs who may be eligible candidates for their search-and-rescue training program.

## Getting Started

To create a usable local version of this web application a few things are required. The desired database must be uploaded, a CRUD is necessary to Create, Read, Update, and Delete documents, and the web application itself must be run:

1. Upload the desired database in Mongodb and grant the desired user access.

**A screenshot of a computer program

Description automatically generated**

1. Log into the desired user account with access to the CRUD functions. (Be sure to use the [use “useraccount”] and [show dbs] commands in Mongo Shell to verify access has been granted to this account.)

**A computer screen shot of a computer program

Description automatically generated**

1. Save the .PY file with the python code from the CRUD Python Module. (be sure to update the connection variables to suit your needs.)
2. Save the testing .IPYNB file with the Jupyter Notebook Script in the same file as the .PY file.
3. Jupyter Notebook can be used for testing the code and ensuring the CRUD methods produce the intended outcomes.
4. Save the web application .IPYNB file in same folder as the previous files.
5. To run the web application, first launch the desired database.
6. Lastly, run the web application .IPYNB file in Jupyter Notebook,

## Installation

* Mongodb - <https://www.mongodb.com/try/download/community>
* A Python IDE
  + Pycharm - <https://www.jetbrains.com/pycharm/download/?section=windows>
  + Or your choice of IDE
* Jupyter Notebook - <https://www.jetbrains.com/dataspell/promo/?source=google&medium=cpc&campaign=AMER_en_US-PST+MST_DataSpell_Search&term=jupyterlab&content=680273770979&gclid=Cj0KCQiA5fetBhC9ARIsAP1UMgG_zRhRvmCcH9ZYfsycyhy4ZQVWRT5J6TiMq6DesHUDwwH0mR4C1GMaAhXqEALw_wcB>

## Usage

**CRUD:**

The .create() function can be called on to insert new data into an existing database.

### Code Example



The .read() function can be called on the find and read the chosen data.

A close up of a computer code

Description automatically generated

The .update() function can be called to update the data in an existing database.

A close up of a computer screen

Description automatically generated

The .delete() function can be called to remove data from an existing database.

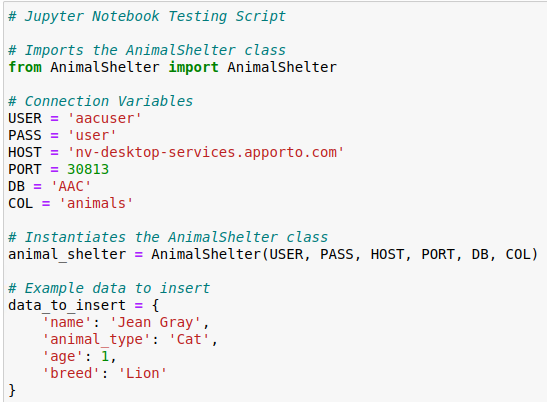
A computer code with red green and blue text

Description automatically generated

### Tests:

Tests can be constructed in Jupyter Notebook that can target each of the functions to ensure the methods perform properly.

### Screenshots



A screenshot of a computer

Description automatically generated

## Web Application:

The application can be used to sort the database in numerous ways that allow the user greater control of the data. A set of filters has been specially designed to aid in the search for specific rescue dogs. These filters are **Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking,** and a **Reset** option.

**Reset:**

A screenshot of a computer

Description automatically generated

**Water Rescue:**

**A screenshot of a computer

Description automatically generated**

## Mountain or Wilderness Rescue:

A screenshot of a computer

Description automatically generated

**Disaster or Individual Tracking:**

**A screenshot of a map

Description automatically generated**

## Roadmap/Features (Optional)

This is the final update for this web application.

## Contact

Your name: Alex Crosswhite