# CS 340 Project One README

## About the Project/Project Title

This project is called Project Two, and focuses on using Jupyter Notebook to create a Python file and test situation. These files enabled the *create*, *read, update, and delete (CRUD)* functionalities in the MongoDB database, Austin Animal Center (AAC). Additionally, an interactive dashboard was essential in order to make this program user-friendly.

## Motivation

This project exists because I must learn how to use different interfaces to utilize databases in my future programming career. By completing this assignment, I am beginning to learn how MongoDB and Python can work together to provide a richer environment for exploring databases. I am also learning how to create a fully functional dashboard for a program.

## Getting Started

First, I accessed the Linux desktop in Apporto. I then started MongoDB, signing into my AAC admin account so that Python would be able to access the AAC database.

Second, I opened a Jupyter Notebook session and accessed two files: my CRUD file I completed in Module Five (animalsModFive.py) and the new Project Two Jupyter notebook file (ProjectTwoDashboard.ipynb).

Third, created code to initialize several things – a logo, my name, a dropdown menu for the user to select a particular set of animals, a pie chart, and a map. The pie chart and the map were dynamic, in that they would change when a different set of animals was chosen.

Fourth, I used a test I had created to call the CRUD functions that I had written in to the .py file. I then ran the tests, getting the desired outcomes from the CRUD operations.

## Installation

The tools I used were embedded in the Apporto virtual lab. I used the Linux desktop to access MongoDB, then used Jupyter Notebook to create my Python file and my notebook file and the test for the Python file.

## Usage

The following are examples of how the project works and how it can be used.

### MongoDB Code Examples

\*Importing AAC database

Text

Description automatically generated

\*Accessing Mongo and entering database using my username and password

Graphical user interface, text

Description automatically generated

\*Finding data from AAC database

Text

Description automatically generated

\*Connecting to the admin database using my username and password

Text

Description automatically generated

### Python Code Example

\*Creating CRUD functionality

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated with medium confidence

### Python Code Example

\*Creating dashboard functionality

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

## Graphical user interface, text, application Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Tests

### Jupyter Notebook Code Example

\*Testing the CRUD functionalities from the Python file

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text, letter

Description automatically generated

\*Testing the dashboard functionalities from the Jupyter notebook file

Grazioso Salvare Logo:

A picture containing text

Description automatically generated

Unique identifier and message:

Graphical user interface, text

Description automatically generated

Dropdown menu:

Graphical user interface, text, application, email

Description automatically generated

Filtered menu:

**Graphical user interface, text, application, email

Description automatically generated**

Pie chart/map using filtered data:

**Graphical user interface, application

Description automatically generated**

**Roadmap/Features**

\*It is important to note that when first testing my user-friendly dropdown menu, I had trouble accessing the different animals that were required to be included in the “Water Rescue” category. For some reason, only Labrador Retrievers were populating the list. Upon going to my Mongo shell and doing a db.animals.find({‘breed’ : ‘Chesapeake Bay Retriever’}), I realized that nothing happened. So I did a generic find and realized that Chesapeake Bay Retriever was listed as Chesa Bay Retr Mix, then went on to find that Newfoundland had several unique entries (I ended up using Newfoundland Mix for my query). I realized with this hiccup that I have to constantly test all of my output to be sure it includes all desired outcomes.

## Contact

Kelly Illescas

kelly.illescas@snhu.edu