# CS 340 README Template

Brianna De La Riva

Dr. Penmatsa

Client/Server Development (CS-340)

25 February 2024

## About the Project/Project Title

The project entails creating the AnimalShelter Python class, which makes it easier to do standard CRUD (Create, Read, Update, and Delete) actions on a MongoDB database. With an emphasis on managing animal data in particular, the class is made to offer a reusable and modular approach for interfacing with MongoDB collections. It integrates with MongoDB via the Pymongo package and includes techniques for generating and reading documents. After providing the required MongoDB connection information, the user can construct an AnimalShelter object and use its Create and Read methods to interface with the underlying database, enabling easy alteration of entries pertaining to animals. The ultimate objective is to improve code reusability and expedite database operations in Python scripts and applications that need to integrate with MongoDB.

**Module 7 additions:**

The additions to this README relating to Module 7 include creating a functional dashboard to navigate through the data in a user-friendly manner. This included creating radio buttons for filtering data showcasing appropriate groups of water rescue dogs, disaster rescue, mountain rescue, and overall without filtered information at the users request. It included the company logo for Grazioso Salvare, a geolocation chart, and a pie chart showcasing breed distribution for each category. Finally, this dashboard also displayed a unique identifier of my name, Brianna De La Riva.

## Motivation

The project was started in response to the demand for a reusable and easier method of carrying out CRUD operations on MongoDB databases, with an emphasis on the management of data pertaining to animals. The purpose of the AnimalShelter class is to provide a standardized and easy-to-use interface for MongoDB developers so they can easily input and retrieve animal entries. The goal is to simplify code, cut down on duplication, and offer a methodical way for Python scripts and applications to interact with MongoDB collections. The project's goal is to streamline database interactions so that developers can more easily include MongoDB features into their work while also encouraging code organization and maintainability.

**Module 7 additions:**

The motivation for creating this interactive dashboard was to enhance user-friendliness and ability to sort through data. This helped create a visual for the data table and aid users in finding specific information in an efficient manner.

## Getting Started

A user can start by opening a terminal and going to the project director in order to clone the project repository to their own computer. Use the `cd` command to navigate to the project directory, for example, cd animal-shelter-project. Installing any dependencies and making sure Python is installed on the system come next. The following commands should be used to install the necessary dependencies: pip install pymongo

It's crucial to configure the MongoDB connection after that. In the `animal\_shelter.py` file, edit the AnimalShelter class and add your MongoDB connection information (password, host, port, database, and collection). The user "aacuser" was made and used for the project's sample. The AnimalShelter class may then be seen running by using this command in a Python environment: `{{ python example\_script.py {{{

Following these procedures, the user will be able to set up and operate an AnimalShelter project, enabling them to explore its features, create and read content, and integrate it with other Python projects.

## Installation

It is vital to ensure a user has Python installed by downloading it from the official website before using the AnimalShelter software through Mongo. There is the option to clone the project repository from GitHub using Git. The steps to install MongoDB Community Server may be found on the MongoDB website. Moreover, use the terminal line {pip install pymongo` to install the Python pymongo library. After installing Python, MongoDB, and pymongo, modify the AnimalShelter class's MongoDB connection information. The user may now easily incorporate the AnimalShelter program into their Python applications. These softwares and installations are key for being able to successfully access, create, read, and more within this project.

## Usage

### Code Example

A screenshot of a computer

Description automatically generated

The ability to input documents and various csv files is possible within the database. This screenshot above shows successfully importing the aac\_shelter\_outcomes.csv using the command below:

mongoimport --username="${MONGO\_USER}"     --password="${MONGO\_PASS}" --port=${MONGO\_PORT}     --host=${MONGO\_HOST} --db AAC --collection animals --type csv --headerline --authenticationDatabase admin --file=aac\_shelter\_outcomes.csv

### Tests

There were testing scripts created to help ensure there was the ability to add in new documents/information into the program. There were also additions to ensure there was the ability for U – Updates and D – Deletes. The testing scripts and screenshots are included below.

### Screenshots

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Additionally, there was recent revising’s that allowed for the use of U – Update, and D – Delete to follow with the CRUD acronym. Below is the screenshots showcasing these changes being successfully done.

A screenshot of a computer program

Description automatically generated

Above is the screenshot of data for animal\_id A696004 before an update, below is after for secondName and the current seconds. A screenshot of a computer

Description automatically generated

A computer screen shot of a program

Description automatically generated

Above is the screenshot of the Jupyter code that successfully created the **U-Update** name changes.

A screenshot of a computer program

Description automatically generated

The above screenshot showcases an animal document for animal\_id of A742287. This is shown within the database prior to **D - Delete** / deletion is successful.

A screenshot of a computer

Description automatically generated

The screenshot above showcases that the python testing script was successful in deleting the document.

A screenshot of a computer program

Description automatically generated

Above is a screenshot showcasing that the deleted document for animal\_id A742287 is no longer within the database and was successfully deleted.

**Module 7 Additions**

Below are photos showcasing the dashboard upon first view. There is additional screenshots displaying each radio button filter successfully working, showcasing the geolocation, as well as the pie chart being fully functional.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*A screenshot of a computer

Description automatically generated*

The screenshot above showcases the successful functionality of the Water Rescue filter. The following screenshots below will display the successful usage of the Mountain Rescue, Disaster Rescue, and Reset/Unfiltered buttons below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Below are screenshots showcasing the geolocation chart and the pie chart functioning as well:

*A screenshot of a computer

Description automatically generated*

## Contact

Your name: Brianna De La Riva