# CS 340 README Template

## About the Project/Project Title

This project focuses on developing the backend and frontend functionality required to connect a user from an interactive dashboard to a MongoDB database for an animal shelter application. The goal is to support efficient and reusable access to animal records, with a focus for this scenario on rescuable animals for an organization like Grazioso Salvare. Using Python, we first built a function CRUD module (Create,Read,Update,Delete) following object-oriented programming principles.

To make sure access to the module was user-friendly we developed an interactive dashboard using Dash with a Jupyter Notebook. The dashboard provides filtering tools, data visualization through a map and breed distribution pie chart and streamlined access to the shelter’s data.

The overall goal and hope of this project are to empower organizations like Grazioso Salvare to easily be able to locate suitable rescue animals and ultimately support their mission of saving lives and serving communities’ needs.

## Motivation

Motivation for this project was to complete the assignment for class, as well as learn valuable information that will help us later. This information can be transferred to a variety of other projects that need similar backend functionality. Completing this also taught me about Jupyter which I was unfamiliar with prior. This also taught me a lot about documenting and properly organizing a project. Overall, I would say this is the most comprehensive project I have ever worked on. Hopefully programs like this can help people and organizations do good. And the reusable module can be used at other shelters for similar purposes.

## Getting Started

This project provides a python backend for interacting with MongoDB specifically for CRUD functions. It also provides an interactive dashboard built in Jupyter Notebook that utilize Dash and Plotly. To get started with this module, you need to ensure you have a MongoDB database up and running. The sign in information currently is already established when the object is created. It will need to be changed based on specific needs such as user/password/host/port/database/collection. I created the CRUD portions of the python module by utilizing the PyMongo Module commands. The interface for the dashboard is created on Jupyter and simply needs to be opened from there and run. It provides an easy to use interface. Once it is loaded via Jupyter, the user can filter rescue animal types, or view where the animals are located. Also a user can see a breakdown of what breeds are available in the filtered list, and sort the columns by ascending or descending order.

## Installation

Tools Required:

Python

The programming language used for building the module. The rational for this was that it was the language we were told to use, as well being an user friendly and fun language to work with. It also has easy MongoDB interactions with the PyMongo module.  
Download this from: <https://www.python.org/downloads/>

MongoDB

A NoSQL database used to store and organize data. Again the rational was that this was the database were told to use but also NoSQL based databases like this one utilize JSON file structure to easily store and retrieve information.

Download this from: <https://www.mongodb.com/try/download/community>

PyMongo

The Python Module to interact with MongoDB. This was used because it is necessary to communicate with MongoDB through python and that is not a built in function.

Command line: ‘pip install pymongo’

Spyder

This IDE was used to run the program the module itself. This was used because it full featured and available in the virtual Lab

Download From: <https://www.spyder-ide.org/>

**Jupyter Notebook**  
A development environment used for testing and demonstrating the module. The rational for using this was because it is perfect for testing python, and was the required software to use.  
Command line: ‘pip install notebook’

## Usage

**Dashboard Functionality**

**A screenshot of a computer

AI-generated content may be incorrect.**

In the above example we see the dashboard in its default state this is also the ‘Reset State’. The Logo is present and links to SNHU.edu, as well as the default text next to it, and my personal identifier “Alex Hitchens – Dashboard Project”. Below is the data frame with options to sort by a column, or filter results. Below are the pie chart that has all the breeds represented above 1% and the map showing the location of the animal according to the records.

A screenshot of a map

AI-generated content may be incorrect.

In the above picture we have clicked on the filter option for the Disaster Rescue animals. The list is filter to the what was specified in the CS340 Dashboard Specifications Document as well as the dog breed pie chart being updated. The Map was also updated to the new animal location.

A screenshot of a computer

AI-generated content may be incorrect.

In the above image, Mountain Rescue has been selected and again has been filtered according to the CS340 Dashboard Specifications Document. Once again the Pie chart and the map are updated to match.

A screenshot of a computer

AI-generated content may be incorrect.

With the final filter selected the last water rescue dogs are search for IAW the CS340 Dashboard Specifications Document and we see that only Labrador Retrievers are available in this category. The pie chart and the map are updated to match the new data frame.

**Documenting the Project**

**Challenges Faces:**

Challenges faced in the project were rather extreme. This project from start to finish was the first time I had worked with many of the systems. For instance, I had never worked with Jupyter before, or Dash, or MangoDB. It was a learning curve that I had to overcome by doing a lot of reading, seeking help where needed from friends, and a lot of trial and error. A specific example of this was trying to figure out why I got an array error when I switched filtered animal types. My solution in the end was to just add an entirely extra callback to reset the selected row every time a filter was changed. This prevented having something selected that went out of range. Another big challenge I remember while working on this was figuring out the whole system flow for Dash. I understood the concepts of callback and how to work with HTML a little but was very shaky on it until the last week or so when the pieces started coming together clearer on it. Overcoming this challenge came down again to a lot of reading and experimenting with the code to learn. In the end I learned a lot about several different topics.

**Why pymongo was chosen:**

Our module, animal\_shelter.py is designed to facilitate CRUD (Create, Read, Update, Delete) operations on a MongoDB database using Python. It employs the pymongo driver, which is MongoDB’s officially supported Python library, and was chosen for its reliability, easy to use nature, and direct support for Python’s native data types, such as dictionaries. It is also the driver that we were instructed to use for the class so it was an easy choice.

The AnimalShelter object establishes a connection to a specific database (aac) and collection (animals) and implements methods for each CRUD operation. The create() method uses insert\_one() to add new documents, ensuring data is a valid dictionary. The find() method retrieves documents that match given criteria using find(), returning the results as a list. The update() method uses update\_many() with the $set operator to modify fields of all matching documents, and the delete() method removes documents that match the provided filter via delete\_many(). Each method includes error handling and input validation to ensure safe and secure interactions with the MongoDB backend.

## Roadmap/Features (Optional)

This project currently supports all the CRUD functionality. Because we are utilizing Object Oriented Principles, we can easily add other features later. Other potential features could include specifying the connection information at initialization of the connection and addition error checking. Other features could be added to the Jupyter Notebook as well, such as the ability to customize search results or filters.

## Contact

Alex Hitchens