**Project Two: README**

Elizabeth Marticello

Southern New Hampshire University

CS 340-15239-M01 Client/Server Development 2025 C-4

Professor Philip Barello

Due Date: August 24th, 2025

# CS 340 README

## About the Project

In short this application is an open-source project to identify good candidates for search-and-rescue training. This app uses Linux, Mongo, Dash, Jupyter Notebook, and Python modules to read potential candidates from a csv file and offers an interactive web app to filter them.

CRUD (Create, Read, Update, Delete) functionality is fundamental for interacting with the database. In a past project CRUD commands were practiced within the Mongo shell. The current project builds on that foundation by developing a portable Python module designed to handle CRUD operations for an Animal Shelter with future intentions to be open source and accessible on GitHub for other shelters to use as well.

## Motivation

The purpose of the project is to build a reusable and module Python component that simplifies database interaction for a web-based Animal Shelter. Implementing CRUD allowing for more efficient data management, seamless integration, and scalable development with the end goal of an interactive web app or dashboard anyone can use.

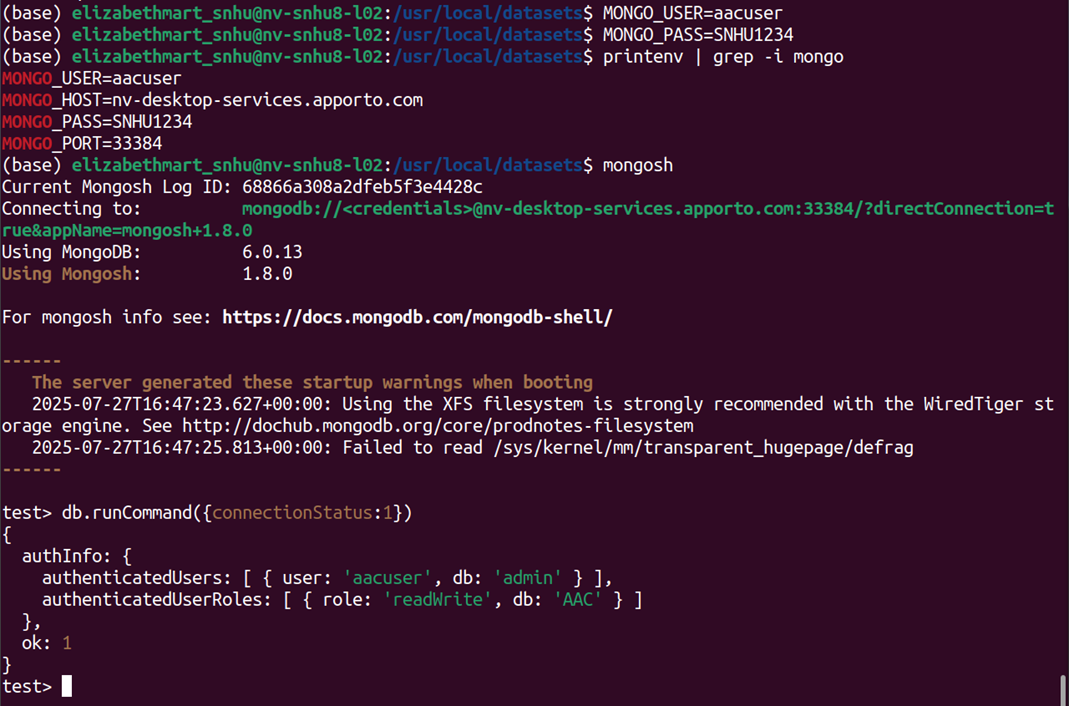
## Getting Started

A previous project was mentioned in the About This Project Section, let’s go into more detail.

Database Indexing and Authentication are essential to start the project. Utilize the prebuilt admin account to create a user account called “aacuser”, short for “Austin Animal Center User”.

Here’s some screenshots from within the Linux Window.

1. A computer screen shot of a program

   AI-generated content may be incorrect.This first picture shows me opening the Mongo Shell, logging into the prebuilt admin profile and creating the new user.
2. **The second picture shows me checking the connected users, verifying that this new user was created properly.

***Now That You’re Caught Up*** we can talk about the Python section. Let’s shift focus on building the Create (C ), Read (R ), Update (U), and Delete (D) portions of the Python module.

To implement these features, I…

* Used the provided template as a starting point and designed a function to insert new documents into the MongoDB database.
* Built a read function to retrieve the filter data, allowing flexibility in searching shelter records.

See the **Usage** section to see some examples.

**Python Code for Dashboard**

The following screenshots show a few different functions implemented in the Python code.

The first figure shows some of the code required for the dashboard. Here you can personalize the headers, logo, and embedded link. Further down you’ll see that the dashboard shows an interactive data table utilizing radio buttons to help filter the table.

A screenshot of a computer program

AI-generated content may be incorrect.**Additional Features**

The data table is not the only feature with this project. There is a geolocation chart which will help the client visualize the data, and a pie chart that shows the percentage of different breeds within which filter is toggled. Both charts dynamically respond to the filtering options. Below is some of the Python code used to implement these charts.

Figure 1 shows some of the Python code for the geolocation chart.

Figure 2 shows some of the Python code the pie chart.

*A screenshot of a computer code

AI-generated content may be incorrect.Figure 1*

*Figure 2*A screen shot of a computer code

AI-generated content may be incorrect.

***Challenges Faced***

This project requires the developer balancing many different tools. One of these tools is Jupyter Notebook, discussed more in the section below. If you’re like me and new to the IPYNB file format you may have trouble while working in this environment. The platform feels almost out-of-date, and the navigation required for the files and cells can be hard to comprehend. My advice would be to consider the cells as blocks for different functions you want to run separately. A lot of the errors I received while building this project were related to indentation, so be sure you’re spacing and indenting correctly while you work.

**Tools – Mongo and Dash**MongoDB was chosen as the model for this project because it provides a flexible, document-orientated database that integrates with Python. The animal records from the Austin Animal Center dataset are stored as documents in a JSON format which maps directly into Python dictionaries. This makes it easy to use MongoDB queries and immediately work with the results in Python without extra conversions. With the pymongo drier, query results can be loaded directly into pandas DataFrames, enabling smooth data manipulation, analysis, and visualization.

Dash was selected to provide both the view and controller because it allows Python developers to build interactive. Browser=based dashboards without needing separate front-end frameworks. As the view, Dash renders interactive tables, graphs, and maps directly from Python objects using components like dash\_table. Dash also used callback functions that connect user inputs, like radio button, to automatically update the interface.

## Installation

To be successful with the creation of this project you must have access to multiple tools.

1. Python. Having an IDE that can edit python is a must, I prefer to use [PyCharm](https://www.jetbrains.com/help/pycharm/installation-guide.html), I know others prefer [Microsoft Visual Studio](https://visualstudio.microsoft.com/vs/). Either is fine for this project.
   1. If you haven’t used either, I would try PyCharm as it’s more user friendly with its editing.
   2. A screenshot of a computer

      AI-generated content may be incorrect.Click the IDEs for a link to the website to download them! If you’re clicking on PyCharm, be sure to change your “Shortcut” before clicking “Get PyCharm”
2. Jupyter Notebook. Having access to this is a must. If you’re a SNHU student (or professor) you have easy access to the Virtual Environment (Apporto) to run this program.
3. Linux Window. Accessing the Mongo Shell is done by using the Linux Window. Again, as SNHU students (and professors) we have access to Apporto to easily access these platforms.

TIP. If you’re having trouble finding your virtual environment check out the Resources section in Learning Modules under Course Menu. Your professor will have uploaded extra help here.

## Usage

This project will support the Create, Read, Update, and Delete operations. These operations are supported by the Python module and tested through a separate Jupyter Notebook script. These scripts are a separate (IPYNB) file and instantiate an object from the CRUD library to effect changes in the MongoDB.

1. Create a Test Animal.

Here is an example of my cat, Captain. Listed are all the keys that an entry could have

1. Create (Insert) a Record

Use the create() function to add a new document to the database. The function accepts a dictionary of key-value pairs to save the animal.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer code

AI-generated content may be incorrect.

1. Read (Retrieve) Records

Use the read() function to query the database. You can provide a filter or leave it empty to return all the records.

## A screenshot of a computer code AI-generated content may be incorrect.A screenshot of a computer code AI-generated content may be incorrect.

1. Update the Test Animal

Call the python function to perform the update. Notice our test animal has received a promotion.

A screenshot of a computer code

AI-generated content may be incorrect.

## Delete the freshly updated entry.

A screenshot of a computer code

AI-generated content may be incorrect.

**Results**

A screenshot of a map

AI-generated content may be incorrect.*Starting State of Dashboard – or Filtered with Reset State*

*Dashboard After Filtering*

**A screenshot of a map

AI-generated content may be incorrect.****Water Rescue.**

**A screenshot of a map

AI-generated content may be incorrect.Mountain or Wilderness Rescue.**

A screenshot of a map

AI-generated content may be incorrect.**Disaster or Individual Tracking.**

**Resources**

Barry, P. (2016). *Head First Python, 2nd Edition*. O'Reilly Media, Inc. <https://learning.oreilly.com/library/view/head-first-python/9781491919521/?sso_link=yes&sso_link_from=SNHU>

SNHU (n.d.). *CS 340 Dashboard Sample Walkthrough* [PDF]. <https://learn.snhu.edu/content/enforced/1969359-CS-340-15239.202556-1/course_documents/CS%20340%20Module%20Six%20Dashboard%20Sample%20Walkthrough.pdf?isCourseFile=true&ou=1969359>

(n.d.). *Dash Core Components*. Plotly. <https://dash.plotly.com/dash-core-components>

(n.d.). *Dash DataTable*. Ploty. <https://dash.plotly.com/datatable>

(n.d.). *Pie Charts in Python*. Ploty. <https://plotly.com/python/pie-charts/>

## Contact

Your name: Elizabeth Marticello