**CS 340 Project Two README**

**Sarah Myers**

**2/28/2025**

**About the Project/Project Title**

This project focuses on creating a front-end application that interacts with a MongoDB database containing information about various animal breeds. The application allows users to filter and sort the database to identify breeds suited for different rescue missions, such as mountain and wilderness rescues or disaster tracking. Users can customize their searches based on criteria like size, temperament, and training level, facilitating better matches for specific rescue needs. The application also includes interactive graphs and maps visually representing the data, making it easier to grasp complex information and identify patterns. Overall, this design aims to enhance accessibility and understanding of the data, improving the efficiency of animal rescue operations.

**Getting Started:**

1.Enter the MongoDB terminal

2. Import the csv file aac\_shelter\_outvome.csv.

3. Create a simple and complex index to analyze the data within the document.

4. To authenticate a user, you need to create two accounts: an Admin account and a user account. The admin account allows you to manage the database, while the user account provides access to the necessary features.

5. You need to have access to Python and run the program outside of a notebook.

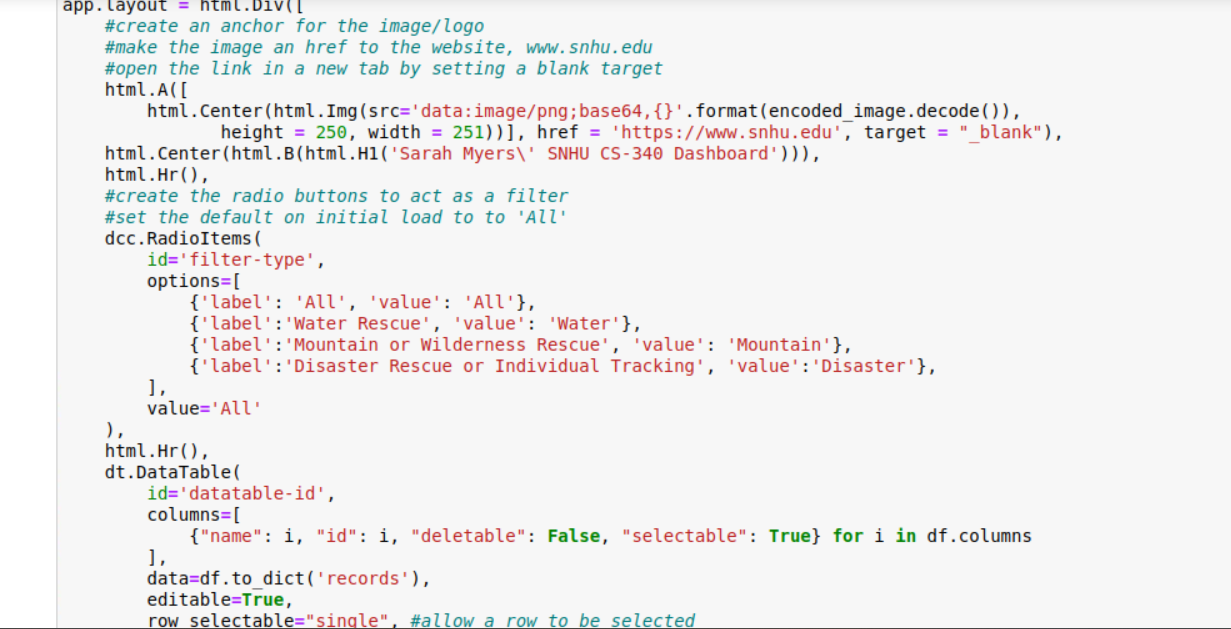
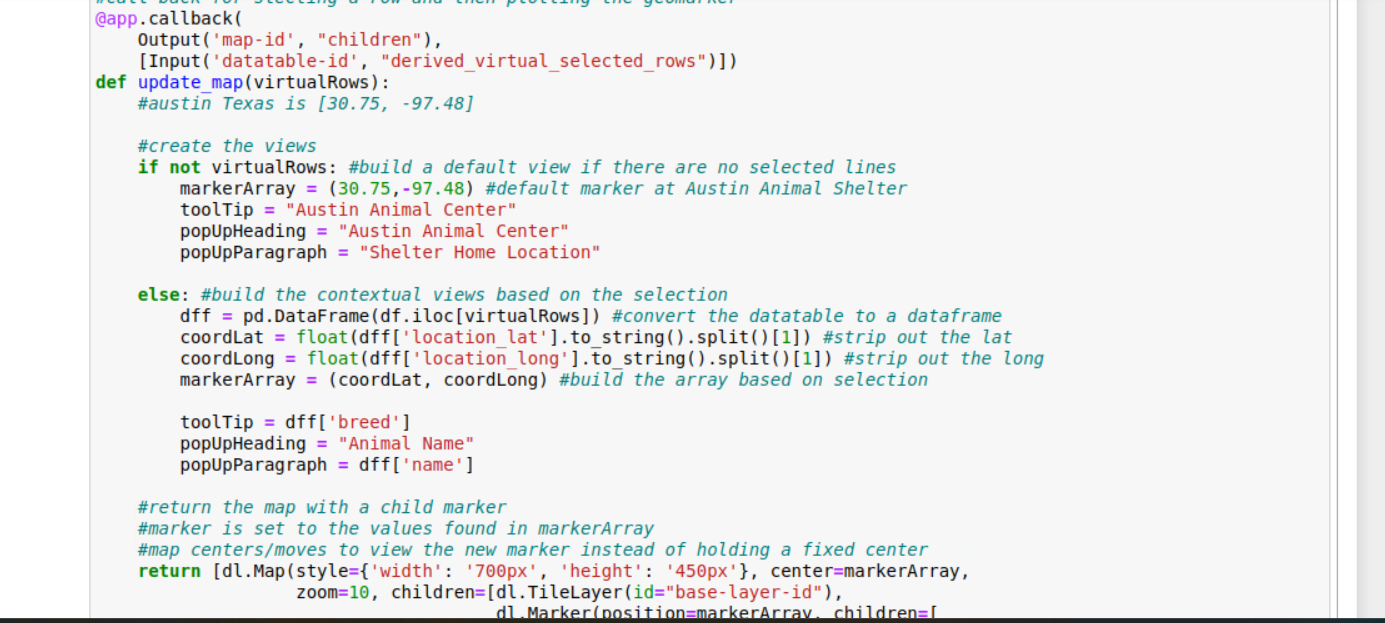
**Installation:**

A current version of Python is needed to run both .py and .ipynb files. MongoDB is required to access the database.

**Usage:**

I began by importing the necessary documents into MongoDB for Python to recognize them during CRUD (Create, Read, Update, Delete) operations. I then developed specific CRUD methods that facilitate these actions and allow efficient backend processing by MongoDB. To ensure secure access, I implemented user authentication with authorized credentials, allowing only authenticated users to perform CRUD operations. After setting up the backend, I focused on the front-end development, creating an intuitive user interface with functional buttons for easy database filtering. This streamlined approach enhanced both security and user experience.

**Screenshots**:



**Challenges**:

I have faced significant challenges in successfully connecting my MongoDB authentication to my Python code. Every attempt to execute CRUD (Create, Read, Update, Delete) operations results in multiple error messages indicating authentication failures. To resolve these issues, I have carefully rewritten the CRUD functions and thoroughly checked my credentials, including the username, password, database name, and port number. I’ve also confirmed that the user has the required permission to read and write to the database, which makes the ongoing errors all the more frustrating. Unfortunately, these authentication issues have prevented me from demonstrating any successful outcomes, so I cannot provide screenshots or evidence of functionality. Nevertheless, I have utilized all available resources to troubleshoot this problem as thoroughly as possible, despite facing these obstacles.

**Contact**:

Sarah Myers