**Jess Dowd**

**Professor Sanford**

**CS 340**

**19 June 2025**

**7-2 Project Two Submission Readme**

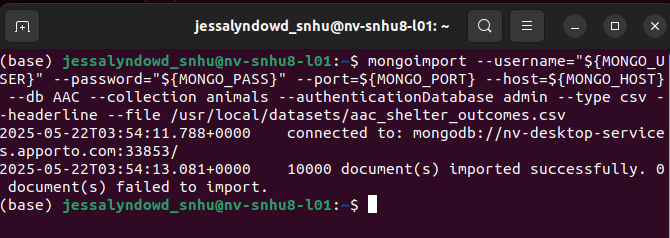
**About the Project / Project Title**  
This project is the final phase of a full-stack development assignment for CS 340. It focuses on building a complete dashboard in Python using the Dash framework connected to a MongoDB database containing data from the Austin Animal Center. The goal is to let Grazioso Salvare, an international animal rescue training organization, filter and visualize data to identify dogs best suited for different types of search-and-rescue training.

**Motivation**  
The goal was to apply client-side development skills by creating an interactive dashboard that connects to a backend MongoDB database. This project builds on the CRUD module I developed in Project One and provides Grazioso Salvare with a branded, intuitive dashboard to explore shelter animal data and make training decisions based on breed, age, and sex.

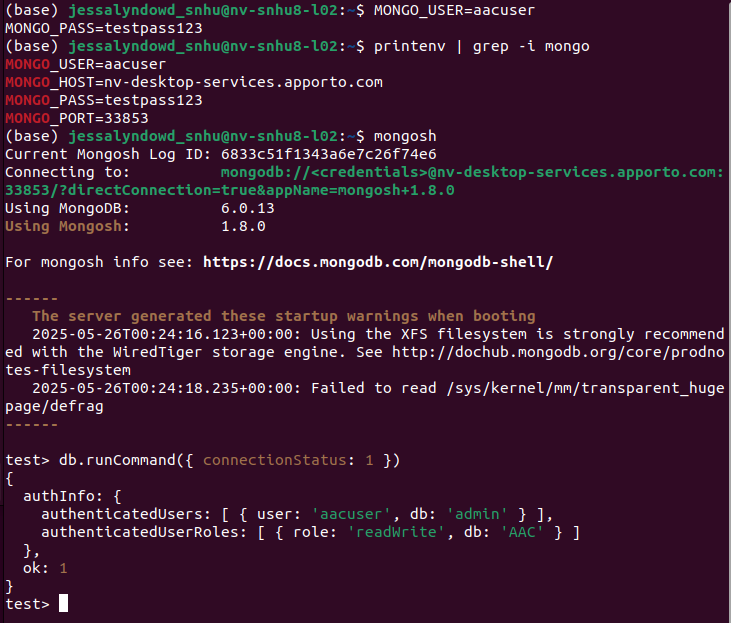
**Getting Started**

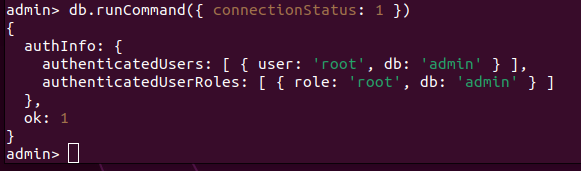
## I used the mongoimport tool in the Apporto terminal to load the aac\_shelter\_outcomes.csv file from the /usr/local/datasets/ directory. I imported this into a database called AAC and a collection called animals. I also set up a user called aacuser with readWrite access to that database.

## Then I created a Python file called animal\_shelter.py with a class called AnimalShelter, which connects to the database using the proper login credentials. In a separate Jupyter notebook called test\_crud.ipynb, I tested inserting, retrieving, updating, and deleting data.



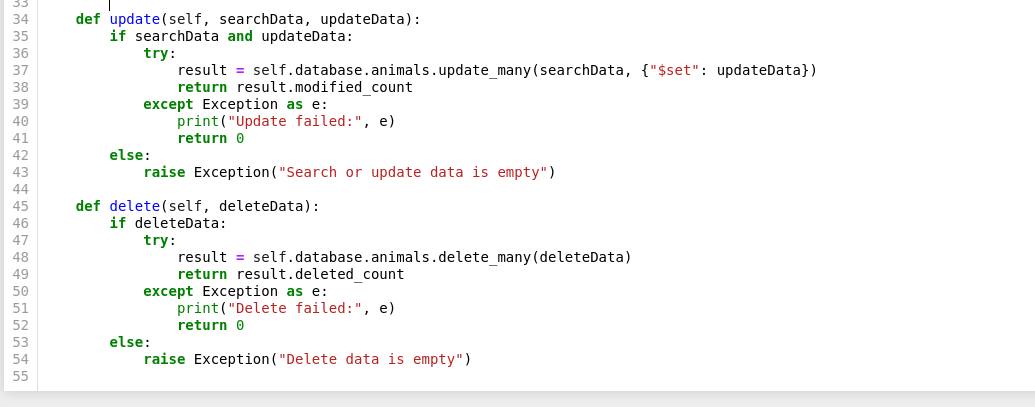
1. Uploaded the Austin Animal Center (AAC) Outcomes data set into MongoDB by importing a CSV file using the appropriate MongoDB import tool.





1. My login process to MongoDB using the mongo shell. Making I can access MongoDB and list the databases using both the admin and aacuser accounts.





3. I created a user with readWrite access and built a Python class AnimalShelter to connect and perform CRUD operations using PyMongo.

1. I used the provided ProjectTwoDashboard.ipynb as the base file and included my CRUD module to load and interact with the MongoDB data.

**Installation / Tools Used**

These are the tools used and why:

**Python** – used to build the module

**PyMongo** – Python driver for MongoDB, used to connect and run database operations

**MongoDB** – stores the Austin Animal Center data

**Jupyter Notebook** – to run and test code interactively

**Apporto/Anaconda** – provided environment from SNHU for running everything

**Dash by Plotly**: Used for building the interactive web dashboard inside Jupyter notebooks.

**JupyterDash**: Allows Dash apps to run inline in Jupyter Notebooks.

**Pandas**: Used for manipulating MongoDB data and displaying it in tables and charts.

**Dash Leaflet**: Used for displaying dynamic maps with animal locations.

**Plotly Express**: Used to create dynamic visualizations such as pie charts.

**Why MongoDB Was Used**  
MongoDB's flexibility with document storage, dynamic schemas, and native JSON compatibility made it good for a project like this. Its can store complex variable animal records and support efficient querying based on breed, age, and sex. That allows for dynamic interaction with the dashboard.

**Why Dash Was Used**  
Dash is a good way to build interactive web apps directly in Python. It supports real-time updates between widgets and backend data, making it good for a data-focused dashboard like this one. It also runs easily in Jupyter Notebooks with JupyterDash.

**Resources Used**

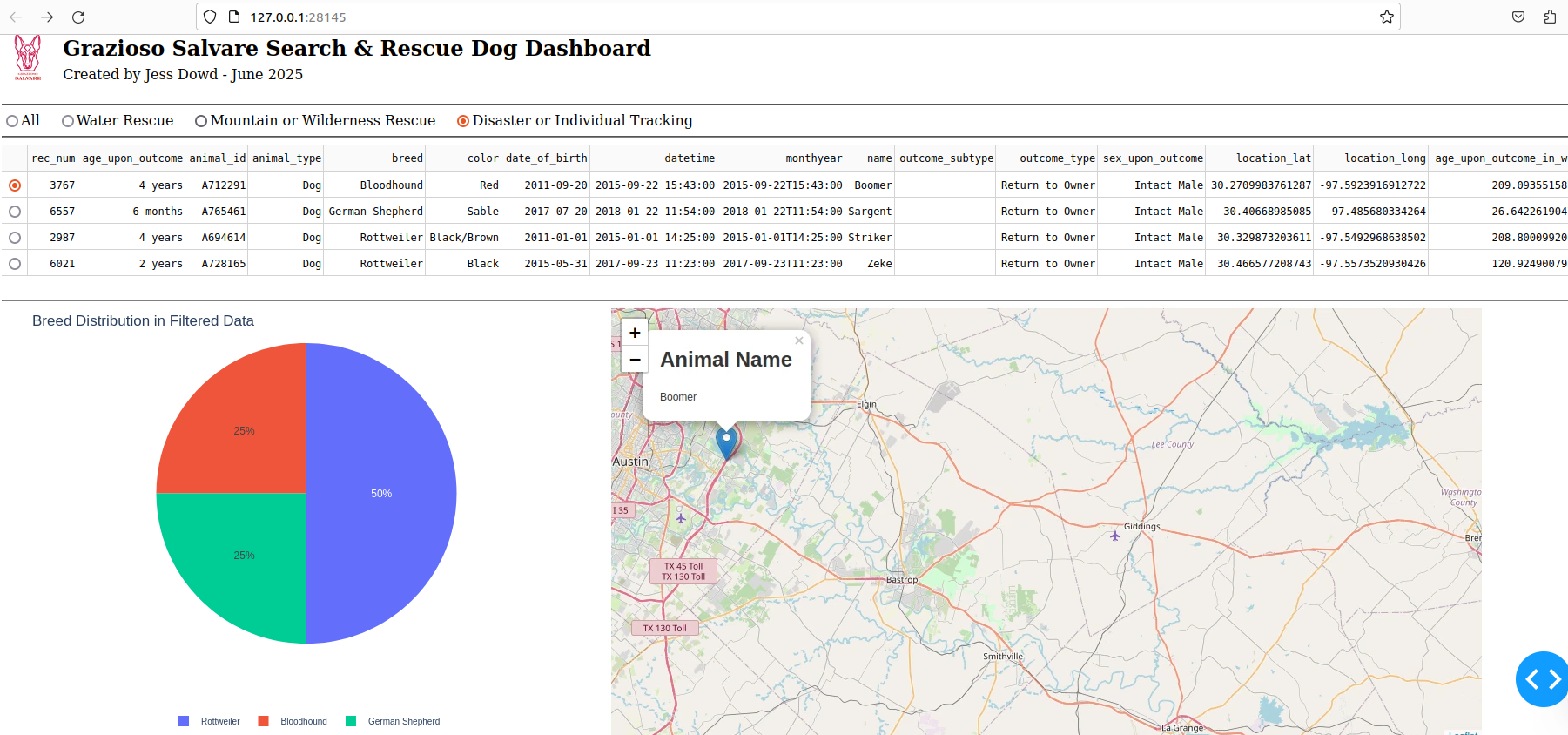
* CS 340 Dashboard Specifications PDF
* Dash Core Components Documentation
* [MongoDB Documentation](https://www.mongodb.com/docs/manual/)
* [PyMongo Documentation](https://pymongo.readthedocs.io/en/stable/)
* Plotly Express
* SNHU Virtual Lab (Apporto)
* aac\_shelter\_outcomes.csv from /usr/local/datasets/

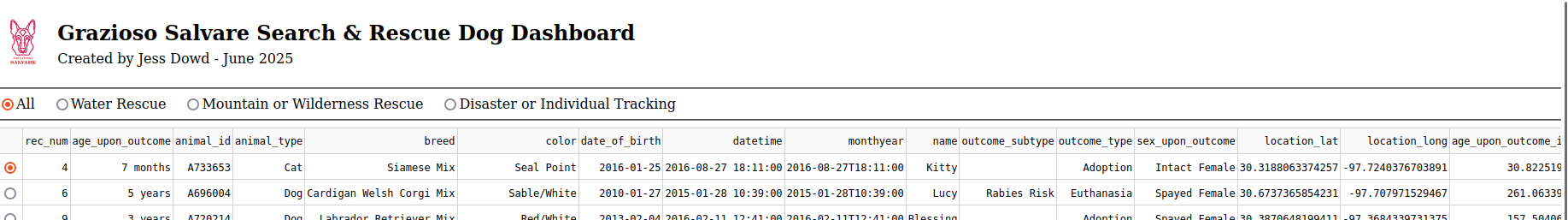
**Usage**  
The dashboard successfully implements:

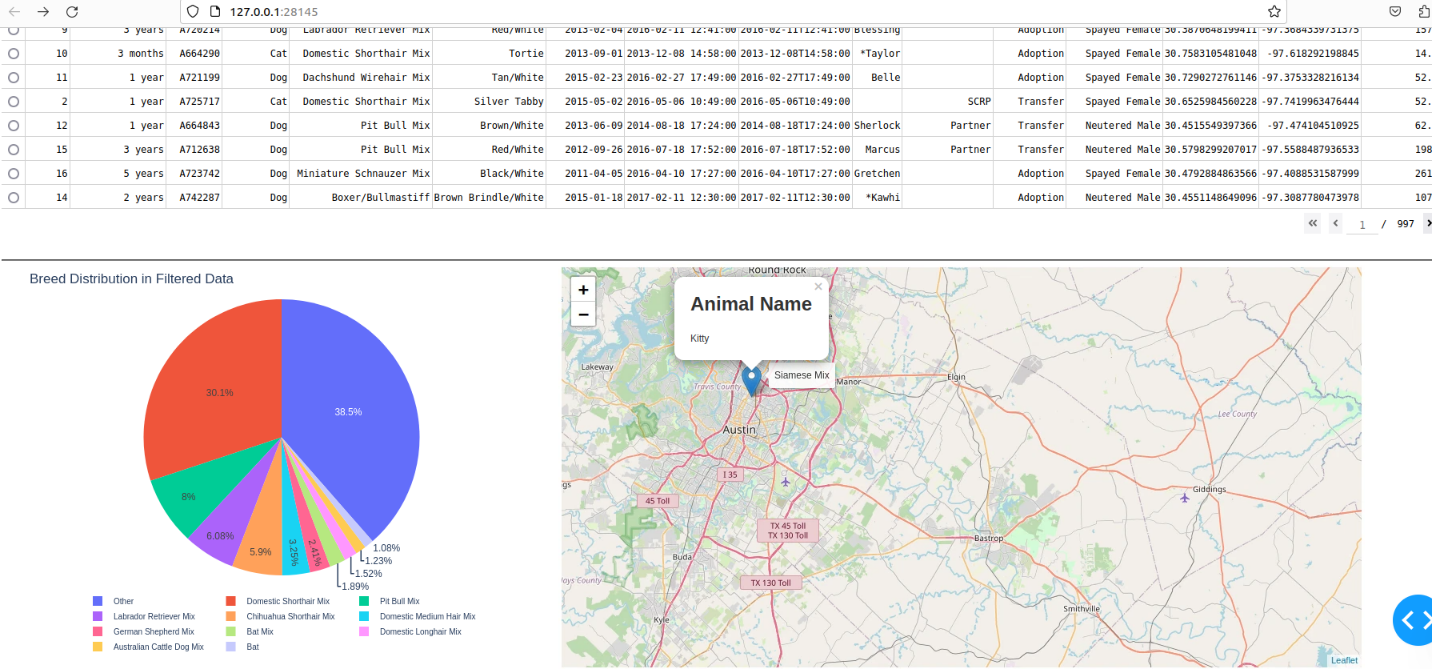
* **Interactive Filtering** using radio buttons:
  + Water Rescue
  + Mountain or Wilderness Rescue
  + Disaster or Individual Tracking
  + Reset (All)
* **Data Table** that dynamically updates with filter selections.
* **Breed Distribution Pie Chart** that updates based on filtered data.
* **Geolocation Chart** showing animal locations on an interactive map along with animal name tag.
* **Dashboard Branding**:
  + Grazioso Salvare logo linked to SNHU homepage.
  + Text credit with my name and date included.

**Screenshots of Functionality**  
📸 These screenshots demonstrate successful functionality as required:

**Disaster and Tracking filtered Dashboard with All Widgets Displayed**



**Unfiltered Dashboard**



**Steps Taken to Complete the Project**

1. Created and tested the AnimalShelter class from Project One.
2. Loaded data into MongoDB and confirmed connection using shell and Python.
3. Built a basic dashboard layout in Dash using JupyterDash.
4. Added the Grazioso Salvare logo and my name for branding.
5. Implemented filter logic using radio buttons.
6. Connected filter logic to update data table using MongoDB queries.
7. Built a pie chart for breed distribution that responds to filters.
8. Built a map to show animal geolocation based on selection.
9. Styled and tested the dashboard to make sure all features work.
10. Captured screenshots of all functionality states.

**Challenges and How I Overcame Them**

Since I have worked with Mongo before most of the issues I ran into were small bugs in the code caused by things like incorrect syntax, typos, or forgetting to add certain parts of the code. These kinds of bugs caused minor errors like the data table not displaying or the charts not updating properly. I fixed them by carefully going through the code, testing each feature as I added it, and debugging where necessary. I also used print statements and checked the shape and content of the DataFrames to make sure everything was working at each step. This made it easier to spot issues early and keep everything running smoothly.

**Contact**  
Jess Dowd  
Project submitted for CS 340 – Southern New Hampshire University