

Grazioso Salvare Animal Rescue Dashboard README

Created by: Corey Arnold

Date: August 24, 2025

What's This Project About?

This dashboard is like a trusty sidekick for Grazioso Salvare, a company that trains dogs to be search-and-rescue heroes. It connects to a MongoDB database filled with Austin Animal Center shelter data, letting users filter dogs by rescue type—think Water Rescue, Mountain or Wilderness Rescue, or Disaster or Individual Tracking. With simple radio buttons, an interactive table, a geolocation map, and a pie chart showing breed breakdowns, it's built to be easy to use, cutting down on confusion and training time. Plus, it's open-source, so other rescue groups can tweak and use it too!

Why We Built It

We wanted to make life easier for Grazioso Salvare to find the perfect dogs for their life-saving training programs. This dashboard pulls it off with clear filters and visuals that update instantly, saving time and reducing errors. By sharing the code openly, we're helping other animal rescue teams benefit from the same tool, spreading the love for pups and people alike.

How to Get Started

Ready to try the dashboard? Here's the game plan:

1. Grab the code from GitHub.
2. Get Python 3.x and MongoDB set up.
3. Install the Python libraries you'll need.
4. Load the shelter data into MongoDB.
5. Fire up the dashboard and start exploring!

Setting It Up

1. Get Python 3.x: Head to <https://www.python.org/downloads/> and download it.
2. Install MongoDB: Download the Community Server from <https://www.mongodb.com/try/download/community>.
3. Clone the Code:
`git clone https://github.com/your-username/grazioso-salvare-dashboard.git`
4. Move to the Project Folder:
`cd grazioso-salvare-dashboard`
5. Install Python Libraries:
`pip install jupyter-dash pandas dash dash-leaflet plotly pymongo`
6. Set Up MongoDB:
 - Start MongoDB: `mongod`
 - Import the dataset (Austin_Animal_Center_Outcomes.csv):
`mongoimport --type csv -d aac -c animals --headerline --file Austin_Animal_Center_Outcomes.csv`
 - Create a MongoDB user:

```
use aac
```

```
db.createUser({ user: "aacuser", pwd: "Bocephus", roles: [{ role: "readWrite", db: "aac" }] })
```

7. Add the Logo: Drop `grazioso_logo.png` into an `assets/` folder in the project directory.

How to Use It

Launch the dashboard:

jupyter notebook ProjectTwoDashboard.ipynb

- Filter Dogs: Click radio buttons to pick a rescue type (Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, or Reset).
- Check the Table: See filtered dogs in a table you can sort, page through, or select rows from.
- Explore Visuals: The map pinpoints the selected dog's location, and the pie chart shows the top breeds, both updating as you filter.
- Look for the Grazioso Salvare logo (linked to <https://www.snhu.edu>) and my name: "Developed by Corey Arnold".

Code Example

Here's a peek at how we query MongoDB for Water Rescue dogs:

```
from AnimalShelter import AnimalShelter
db = AnimalShelter("aacuser", "Bocephus")
query = {
    "animal_type": "Dog",
    "breed": {"$in": ["Labrador Retriever Mix", "Chesapeake Bay Retriever", "Newfoundland"]},
    "sex_upon_outcome": "Intact Female",
    "age_upon_outcome_in_weeks": {"$gte": 26, "$lte": 156}
}
records = db.read(query)
df = pd.DataFrame.from_records(records)
```

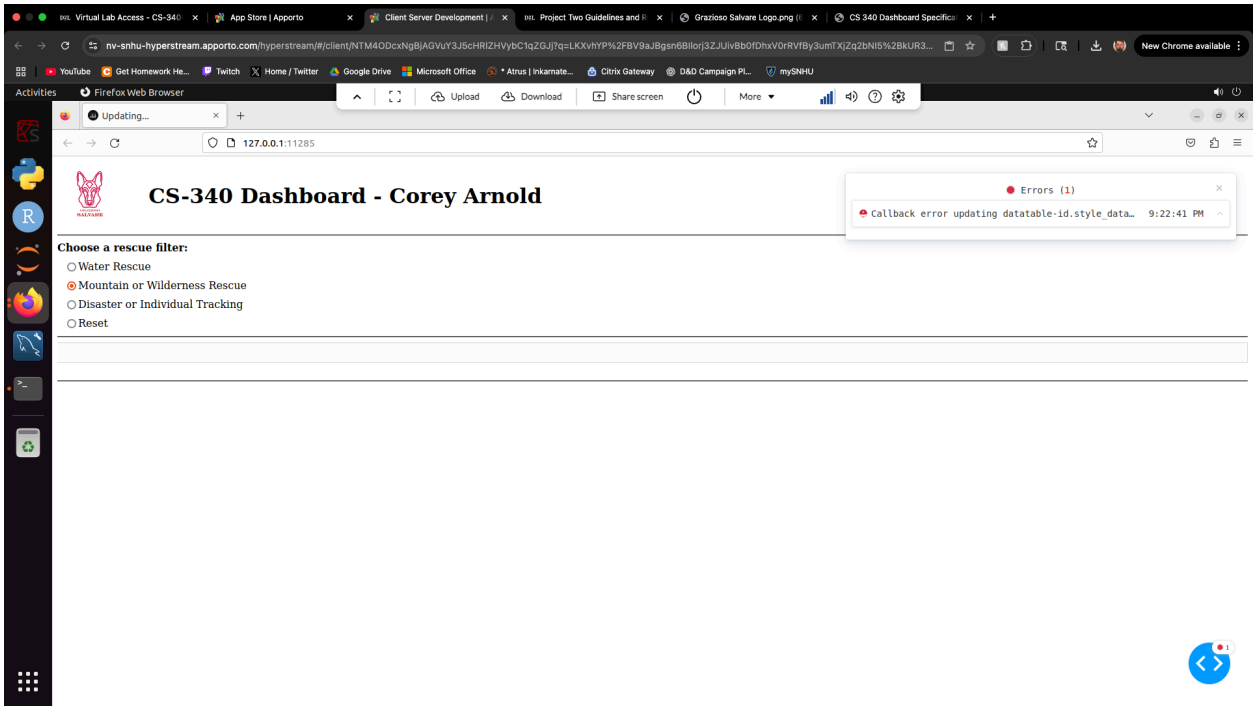
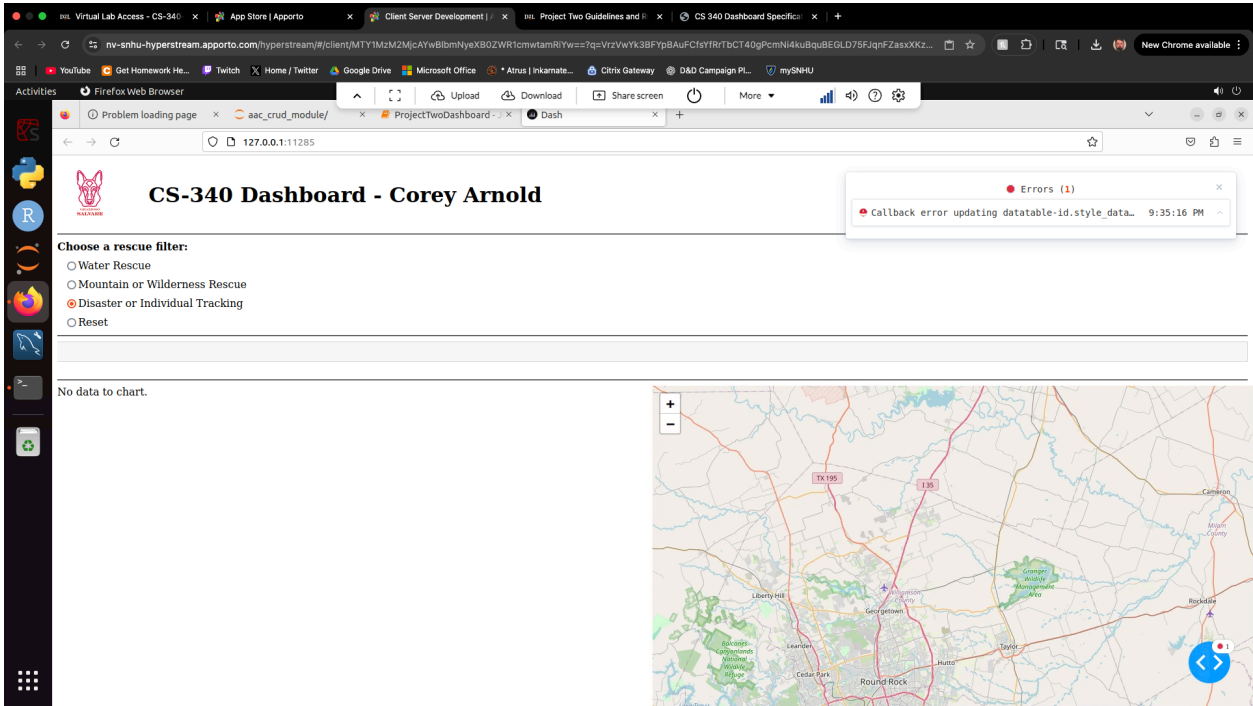
Testing It Out

Check the database queries:

python test_animal_shelter.py

Example test for Water Rescue dogs:

```
def test_read_water_rescue():
    db = AnimalShelter("aacuser", "Bocephus")
    query = {
        "animal_type": "Dog",
        "breed": {"$in": ["Labrador Retriever Mix", "Chesapeake Bay Retriever",
"Newfoundland"]},
        "sex_upon_outcome": "Intact Female",
        "age_upon_outcome_in_weeks": {"$gte": 26, "$lte": 156}
    }
    result = db.read(query)
    assert len(result) > 0, "No Water Rescue dogs found"
```



Screenshots listed above.

Tools We Used and Why

- MongoDB: Perfect for the shelter's messy, unstructured data. Its document setup makes filtering by breed or age a breeze, and it plays nicely with Python's pymongo for smooth database work.
- Dash Framework: Dash (from Plotly) is our go-to for building interactive web dashboards in Python. It handles the visuals (like the table and charts) and the logic (like filter updates), making everything click together.
- Resources:
 - <https://doi.org/10.26000/025.000001> (Austin Animal Center Data)
 - <https://dash.plotly.com/> (Dash Docs)
 - <https://docs.mongodb.com/> (MongoDB Docs)
 - <https://www.python.org/doc/> (Python Docs)
 - <https://dash-leaflet.herokuapp.com/> (Dash Leaflet Docs)

How We Built It

1. Got MongoDB running and loaded the Austin Animal Center data.
2. Wrote AnimalShelter.py to handle MongoDB queries.
3. Built the dashboard in ProjectTwoDashboard.ipynb with Jupyter Dash.
4. Added a table with sorting, pagination, and row selection.
5. Created queries for rescue types based on the project's breed table.
6. Set up radio buttons for easy filtering.
7. Added a map and pie chart that update with filters.
8. Included the Grazioso Salvare logo (linked to <https://www.snhu.edu>) and my name.
9. Tested everything and took screenshots for each filter.
10. Wrote this README to share the journey.

Bumps in the Road and How We Fixed Them

- Challenge: The map was sluggish with tons of data.
 - Fix: Showed just the selected dog's location to keep things speedy.
- Challenge: Syncing the table, map, and pie chart with Dash was tricky.
 - Fix: Used print statements to debug and made sure queries returned clean data.
- Challenge: MongoDB connections were finicky across setups.
 - Fix: Added backup host/port options and error handling to stay robust.

Get in Touch

Your Name: Corey Arnold