## **Grazioso Salvare Animal Dashboard - CS340 Final Project**

**Author:** Shoh Janobilov  
**Course:** SNHU CS-340 – Client/Server Development  
**Date:** June 2025

### **Project Overview**

This dashboard is a full-stack web application built for **Grazioso Salvare**, an international rescue-animal training organization. The purpose of this project is to help them identify ideal shelter dogs for various types of rescue training, using data sourced from the **Austin Animal Center**.

It features:

* Real-time filtering by rescue type
* Interactive data table
* Live updating geolocation map
* Dynamic pie chart visualizing breed distribution
* Intuitive, user-friendly layout with brand identity
* Powered by MongoDB and the Dash framework

### **Tools & Technologies Used**

| **Tool** | **Purpose** |
| --- | --- |
| **MongoDB** | NoSQL database to store and query shelter data |
| **Dash** | Python framework for building interactive dashboards |
| **Pandas** | Data manipulation and cleaning |
| **Dash Leaflet** | For rendering geolocation maps |
| **Plotly** | For rendering dynamic pie chart visualizations |
| **VSCode / Jupyter** | Development and testing environment |

### **Why MongoDB?**

MongoDB is schema-flexible and handles JSON-like documents, making it ideal for dynamic animal shelter data. Its ability to natively integrate with Python (via pymongo) enables seamless backend integration with Dash.

### **Why Dash?**

Dash provides:

* Native integration with Plotly charts and tables
* MVC-like design: models via MongoDB, views via Dash Core Components, controllers via Python callbacks
* Rapid development with a clean layout and interactivity

### **Functionality & Features**

Below are screenshots (or screencast links) demonstrating the functionality of each dashboard state. You can substitute these descriptions with actual screenshots in your final document.

1. **Default Dashboard View** (Unfiltered)  
   * Displays the full animal dataset
   * Default map marker and pie chart
2. **Water Rescue Filter Applied**
   * Shows only dogs under 2 years of age from water-rescue-approved breeds
   * Pie chart updates by breed
   * Map centers on selected dog
3. **Mountain or Wilderness Rescue Filter Applied**
   * Shows filtered data and visuals for applicable breeds
4. **Disaster or Individual Tracking Filter Applied**
   * Tailored filtering and dynamic visuals for scent-tracking breeds
5. **Reset**
   * Returns the view to the full unfiltered state

Each view includes:

* Map with marker and tooltip/popup
* Pie chart showing breed distribution
* Interactive table for row selection

### **🧪 How to Run the Project**

1. Clone the repository or download all files into the same directory:  
   * ProjectTwoDashboard.ipynb (main dashboard code)
   * animal\_shelter.py (CRUD Mongo module)
   * aac\_shelter\_outcomes (1).csv (CSV data if testing without MongoDB)
2. Open **Jupyter Notebook** or **VSCode with Jupyter Extension**
3. If testing without Mongo:  
   * Replace the MongoDB read with pd.read\_csv("aac\_shelter\_outcomes (1).csv")
4. Run all cells in ProjectTwoDashboard.ipynb
5. Click the http://127.0.0.1:XXXX/ link in the output to open the dashboard

### **MongoDB Credentials**

python

CopyEdit

username = "aacuser"

password = "SNHUaac1234"

### **Challenges & Resolutions**

| **Challenge** | **Resolution** |
| --- | --- |
| Layout error due to missing data | Implemented fallback CSV mode for local testing |
| Map not rendering due to bad coordinates | Added default lat/lon and validation checks |
| Dash crashing with large dataset | Limited rows shown in table with pagination |
| ZMQ binding conflict | Restarted kernel and avoided duplicate runs |

### **Folder Structure**

pgsql

CopyEdit

Project Folder

├── ProjectTwoDashboard.ipynb

├── animal\_shelter.py

├── aac\_shelter\_outcomes (1).csv

├── README.md

└── screenshots/

├── initial.png

├── water.png

├── mountain.png

├── disaster.png

└── reset.png

### **References**

* Dash Documentation
* [MongoDB Docs](https://www.mongodb.com/docs/)
* Austin Animal Center Data
* SNHU CS340 Module Resources