# CS 340 Project Two README

## About the Project/Project Title

## The Grazioso Salvare Dashboard is an interactive web-based data visualization application developed using Python Dash and MongoDB. The dashboard was created for Grazioso Salvare, a company specializing in animal rescue training, to assist in identifying suitable animals from the Austin Animal Center dataset for various types of rescue training (Water, Mountain, and Disaster).

## The dashboard allows users to:

## Filter animals by rescue type.

## View detailed animal data from a connected MongoDB database.

## Visualize top breeds in a pie chart.

## Display the selected animal’s location on an interactive map.

## Motivation

The goal is to apply CRUD (Create, Read, Update, Delete) operations, integrate a backend database with a frontend dashboard, and demonstrate data-driven decision-making through visualization tools. The motivation behind the project is to help Grazioso Salvare quickly identify animals with the physical and behavioral traits suited for specific rescue training programs, improving the efficiency of their selection process.

## Getting Started

Make sure you have the following installed:

* **Python 3.8+**
* **JupyterLab or Jupyter Notebook**
* **MongoDB** (local or cloud instance such as MongoDB Atlas)

## Installation

Install the required Python packages using pip:

pip install dash jupyter-dash dash-leaflet plotly pandas numpy matplotlib pymongo

If using JupyterLab, also install:

pip install jupyterlab

Make sure your MongoDB server is running and that the credentials in your CRUD module (AnimalShelter.py) match your MongoDB connection information.

## Usage

### Code Example

Below is a brief excerpt showing how the database connection and data retrieval work:

from CRUD\_Python\_Module import AnimalShelter

import pandas as pd

# Connect to MongoDB

username = 'aacuser'

password = 'JeyAndMax2025'

db = AnimalShelter(username, password)

# Read all records from MongoDB

df = pd.DataFrame.from\_records(db.read({}))

# Remove MongoDB ObjectID for display

df.drop(columns=['\_id'], inplace=True)

### Tests

To verify database functionality, you can run simple CRUD tests in Python:

# Test database connection

print(db.read({})) # Should return a list of animal documents

# Example query test for Water Rescue animals

query = {

'animal\_type': 'Dog',

'breed': {'$in': ['Labrador Retriever Mix', 'Chesa Bay Retr Mix',

'Newfoundland Mix']},

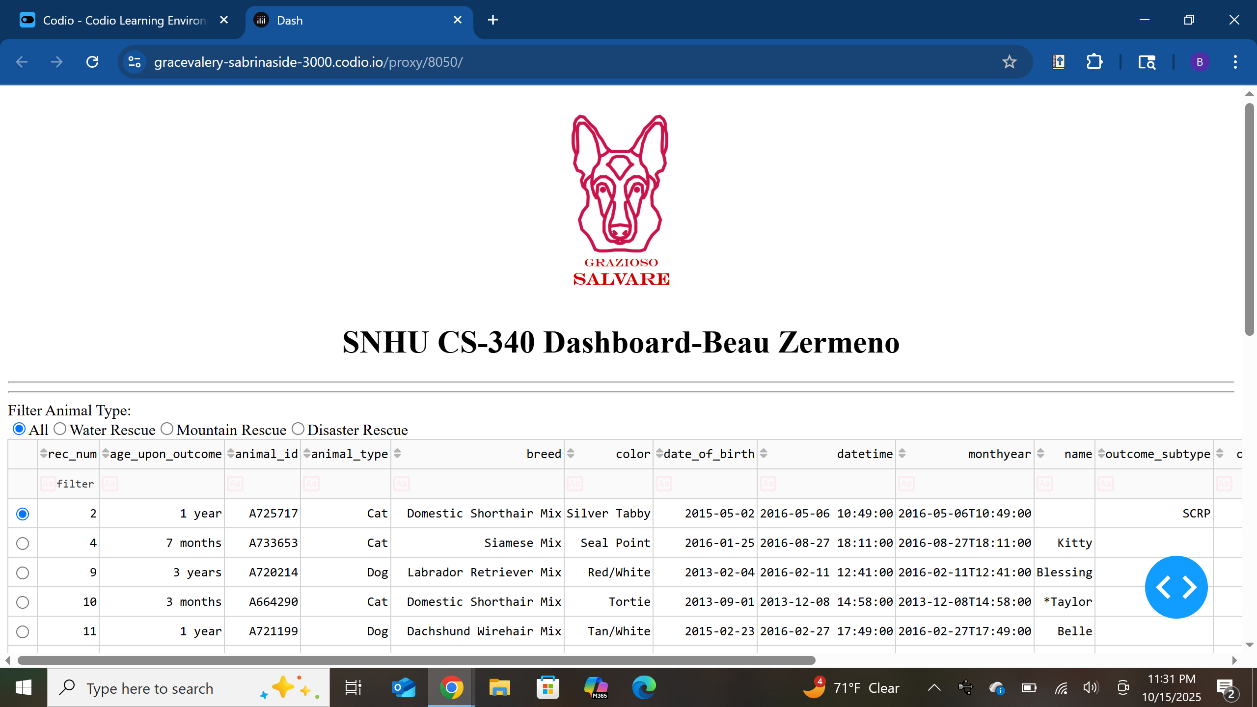
'sex\_upon\_outcome': 'Intact Female',

'age\_upon\_outcome\_in\_weeks': {'$gte': 26, '$lte': 156}

}

print(db.read(query))

### Screenshots



A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Roadmap/Features

Future improvements may include:

* Adding more interactive filters (e.g., gender, age range).
* Displaying outcome type statistics for each rescue category.
* Allowing users to export filtered datasets as CSV files.
* Implementing CRUD operations for adding or updating animal records directly in the dashboard.

## Contact

Your name: Beau Zermeno