# CS 340 README

***Grazioso Salvare Dashboard***

## About the Project/Project Title

## *The Grazioso Salvare Dashboard is an interactive web application designed to visualize and analyze animal shelter data. It performs read operations on a MongoDB database containing information from the Austin Animal Center (AAC). The dashboard allows users to filter data based on specific rescue animal categories, view the data in a tabular format, visualize breed distributions through a pie chart, and see individual animal locations on a map*

## *.*Motivation

## The motivation behind this project is to provide Grazioso Salvare, a company specializing in training rescue animals, with an efficient tool to identify potential rescue animal candidates. By implementing this dashboard, users can easily retrieve, filter, and visualize data from the AAC database. This functionality is essential for streamlining the process of selecting suitable animals for rescue and training programs, enabling data-driven decision-making and improving the efficiency of Grazioso Salvare's operations.

## Getting Started

## To get a local copy up and running, follow these simple steps:

## Ensure that you have Python 3.x installed on your machine.

## Ensure that you have MongoDB installed and running on your machine.

## Install the required Python libraries by running pip install dash dash-leaflet plotly pandas pymongo.

## Clone this repository to your local machine.

**Installation**

Python 3.x: The primary programming language used for this project.

MongoDB: A NoSQL database used to store the AAC dataset.

Dash: A Python framework for building analytical web applications.

PyMongo: A Python driver for MongoDB to interact with the database.

Plotly: A graphing library for creating interactive plots.

Pandas: A data manipulation library for Python.

pip install dash dash-leaflet plotly pandas pymongo

To install these tools:

Download and install Python from python.org.

Download and install MongoDB from mongodb.com.

Install the required Python libraries by running the command:

## Usage

*The CRUD Python module provides methods for creating and reading documents in a MongoDB collection. Below are examples of how to use these methods:*

### Code Example

from dash import Dash

import pandas as pd

from pymongo import MongoClient

# Create a connection to MongoDB

client = MongoClient('mongodb://localhost:27017')

db = client['aac\_shelter']

collection = db['animals']

# Load data into a pandas DataFrame

df = pd.DataFrame(list(collection.find()))

# Initialize the Dash app

app = Dash(\_\_name\_\_)

# Define the layout and callbacks here (omitted for brevity)

# Run the app

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(debug=True)

### Tests

### To test the dashboard functionality, follow these steps:

### Ensure that MongoDB is running and populated with the AAC dataset.

### Run the dashboard application using the provided Python script.

### Open a web browser and navigate to the URL where the dashboard is being served (typically http://127.0.0.1:8050).

### test the following functionalities:

### Click on each rescue category button and verify that the data table updates accordingly.

### Check that the pie chart updates to reflect the current data selection.

### Select a row in the data table and verify that the map updates to show the correct location.

### Verify that all components of the dashboard (data table, pie chart, and map) are rendering correctly.

### Screenshots

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated