# CS 340 README Template

## About the Project/Project Title

Welcome to the Grazioso Salvare Rescue Dog Training Application!

This application was developed by Global Rain, a software engineering company that specializes in custom software design and development. The application was created for Grazioso Salvare, an innovative international rescue-animal training company that trains dogs for search-and-rescue missions.

The purpose of this application is to identify and categorize dogs that are good candidates for search-and-rescue training. To do this, the application works with data from animal shelters in the Austin, Texas region, provided by a non-profit agency. The application enables users at Grazioso Salvare to access and manipulate this data through a user-friendly, intuitive dashboard.

The code for this project is open source and available on GitHub, so that it may be used and adapted by similar organizations. If you would like to reproduce this project, please see the instructions in the repository.

We hope that this application will be a valuable resource for Grazioso Salvare as they work to train rescue dogs and save lives. Thank you for choosing Global Rain for your software development needs.

## Motivation

This project exists to make it easier to upload records into a MongoDB animal shelter database. Most of these records and queries could be performed manually, but the purpose of this library is to automate the process and make it simpler for the user to add, search, delete, and update data within the DB.

## Getting Started

To get started with the Grazioso Salvare Rescue Dog Training Application, you will need to follow the steps below:

1. Install the necessary dependencies for the application. You can install these packages by running `pip install [package name]` in your command line.
2. Ensure that you have a MongoDB database set up and running. If you do not already have a MongoDB database, you can follow the instructions [here](https://www.mongodb.com/docs/manual/installation/) to set one up.
3. In the code, update the username and password variables with your MongoDB username and password.
4. Replace the AnimalShelter class with your own CRUD Python module that has been designed to work with your MongoDB database.
5. Run the application by executing the code in your Python environment. The application should launch in your web browser.
6. Use the dashboard to view and manipulate the data in your MongoDB database. You can use the radio buttons to filter the data by rescue type, and the interactive data table to view and select individual records. The chart and map on the bottom of the dashboard will update based on your selections.

## Installation

*Python -* [*https://www.python.org/downloads/*](https://www.python.org/downloads/)

*MongoDB - https://www.mongodb.com/docs/manual/installation/*

*Jupyter Notebook - https://jupyter.org/install*

## Usage

To use the Grazioso Salvare Rescue Dog Training Application, follow the steps below:

1. Navigate to the dashboard in your web browser.
2. Use the radio buttons at the top of the dashboard to filter the data by rescue type. Selecting a rescue type will display only those records in the data table that match the selected type. Selecting the "Reset" option will remove the filter and display all records.
3. Use the interactive data table to view and select individual records. You can use the pagination controls at the bottom of the table to navigate between pages of data, and you can use the search box to find specific records. You can also sort the table by clicking on the column headers.
4. When you select a record in the data table, the chart and map on the bottom of the dashboard will update to show relevant information. The chart displays the distribution of rescue dog breeds for the selected record, and the map shows the location of the animal shelter where the selected dog is available for adoption.

### Code Example

# Define callback function to update chart and map when a row in the data table is selected

@app.callback([Output('graph-id','children'),

Output('map-id','children')],

[Input('datatable-id','selected\_rows'),

Input('hidden-div','children')])

def update\_graph\_map(selected\_rows, hidden):

if selected\_rows is None:

selected\_rows = [0]

df\_new = df.iloc[selected\_rows]

graph = dcc.Graph(id='pie', figure=px.pie(df\_new, values='Count', names='Breed'))

center = [df\_new['Latitude'].mean(), df\_new['Longitude'].mean()]

map\_ = dl.Map(id='my-map', center=center, zoom=4)

return graph

This code defines a callback function that is triggered when a row in the data table is selected. The function takes the selected row indices as input, and uses them to extract the corresponding record from the data frame. It then creates a pie chart showing the distribution of rescue dog breeds for the selected record, and a map showing the location of the animal shelter where the selected dog is available for adoption. The chart and map are returned as output, and are used

## Contact

Your name: Caleb VanDerMaas