# CS 340 README Template

## About the Project/Project Title

The client, Grazioso Salvare, is an international rescue-animal training company that are seeking to implement a software application that is compatible with their current data to identify and categorize their search-and-rescue animals. Since the CRUD (create, read, update, and delete) functionalities have already been implemented, the client wants to expand and create a dashboard that is user-friendly interface to identify and categorize available dogs from their database. The dashboard requires a MongoDB layer that will contain all of the client’s data to create a visual when producing the html dashboard.

## Motivation

The project is to demonstrate how to implement a dashboard that is user-friendly but also be used to help other similar organizations adapt to similar methods.

## Getting Started

* Users must have an account that includes the read/write permissions. (See appendix for more information)
* The file “animalShelter.py“ includes all of the CRUD functionalities that will enable the user to modify as needed. The CRUD operations assist in handling all of the setter and getter methods. (See appendix for more information)
  + Create
  + Read
  + Update
  + Delete
* The file “ProjectTwoDashboard.ipynb” consists of built functions to import the information and transcribe it into a visual, interactive web-based dashboard.
  + Branding for the client logo
  + An interactive data table to filter information imported from Austin Animal Shelter’s data.
  + Pie chart that organizes and categorizes available breeds based on selected filter.
  + Custom map shows the location of the selected animal by using a pin tooltip.
* Directions

1. Ensure user access is granted, if not, please visit the appendix for more information.
2. Log onto the Jupyter Software and upload “ProjectTwoDashboard.ipynb” file.
3. Run the test file to open the link for the dashboard to few the datachart, pie chart, and map location of Austin Animal Center (AAC) breed that is selected.

(Users will be able to see the entire CSV file or select an interactive filter.)

**Installation - Software**

* MongoDB – Installation guide can be found [here](https://www.mongodb.com/docs/manual/installation/).
  + MongoDB is a cloud-based system that will be used to distribute the AAC database across multiple servers. MongoDB will also provide scalability when expanding storage capacity and other data-driven applications.
* PyMongo – Installation guide can be found [here](https://pypi.org/project/pymongo/).
  + PyMongo, with its set of tools, is used to build the CRUD commands and communicate with the MongoDB server.
  + Jupyter Notebook is an open-source web application for developers to create, share, and edit the code. Jupyter Notebook also has a testing station for the code workflow making it easier for developers to compile all portions of the data project in one place.
* Spyder – Installation guide can be found [here](https://docs.spyder-ide.org/3/installation.html).
  + Spyder is an IDE that uses the Python language that allows developers to create a project that is user-friendly by including syntax errors, installation packages, and console testing.

**Installation – Libraries**

* from jupyter\_dash import JupyterDash
  + Used to create simple and effective dashboards.
* import dash\_leaflet as dl

from dash import dcc

from dash import html

* + Used for interactive maps.
* import plotly.express as px

from dash import dash\_table

from dash.dependencies import Input, Output

* + Library to create high API in graph data.
* numpy as np (Numerical Python)
  + Enables mathematical and logical operations on arrays.
* pandas as pd
  + Used for data manipulation purposes.

**Usage – Internal Codes**

* Creating a header (Displays a header with text and image)



* Interactive filter option code



* Features of the interactive data table.

A screen shot of a computer code

Description automatically generated

* Rescue type and preferred dog breeds based on filtered options.



A screen shot of a computer code

Description automatically generated

## Pie Chart

A screen shot of a computer code

Description automatically generated

* Map Location

A screenshot of a computer code

Description automatically generated

**Display Modules**

* The standard setting is to show all available animals with the pie chart and map.

However, once a filter is selected, the pie chart and map will change based on the selected animal. (examples shown below for each filter)

* + Filter, data table, pie chart and map

A screenshot of a computer

Description automatically generated

A screenshot of a map

Description automatically generated

* + Water Selection

A screenshot of a computer

Description automatically generated

* + Mountain and Wilderness Rescue

A screenshot of a computer

Description automatically generated

* + Disaster Rescue and Individual Tracking

A screenshot of a computer screen

Description automatically generated

**Appendix**

Setting up account permission

1. Using MongoDB, import the CSV File. Note: only an admin user can create and remove use from a database. (Removing users should be done with extreme caution!)

For more guidelines, visit the Enable Access Control link [here](https://www.mongodb.com/docs/v6.0/tutorial/enable-authentication/).

A screenshot of a computer program

Description automatically generated

1. Have users log on the created account and ensure all access rights are granted properly with the following procedure:
2. Log onto linux terminal and input mongo user and password information.
3. Log onto mongo terminal and test all access is granted properly.

A computer screen with white text and green text

Description automatically generated

A screenshot of a computer program

Description automatically generated

CRUD operations

The CRUD operations will enable developers to create, read, update, and delete operations to have a smooth transaction with modifying a database. Images below will include the “animalShelter.py” code for a better understanding, including the test station.

* **Code to Initialize the Client** A screenshot of a computer program

  Description automatically generated
* **Code to Create & Read Within the Index** A computer screen shot of a program

  Description automatically generated
* **Code to Update & Delete Within the Index**A screen shot of a computer program

  Description automatically generated

### Tests

* **Ensuring log in was successful.**
  + When the log in is successful, there will be a notification of its success.

A screenshot of a computer program

Description automatically generated

* **Testing the “create” operation.**

A screenshot of a computer

Description automatically generated

* **Testing the “read’ operation.**

**A screenshot of a computer

Description automatically generated**A screenshot of a computer

Description automatically generated

* **Testing the “update” operation.**

**A screenshot of a computer program

Description automatically generated**

* **Testing the “delete” operation.**

**A screenshot of a computer code

Description automatically generated**

## Contact

Your name: Winnie Kwong