# CS 340 README Template

## About the Project/Project Title

The project Rescue Mission was made for the international rescue-animal training company, Grazioso Salvare. Grazioso Salvare’s mission is to identify dogs that make good candidates for their save-and-rescue training programs. After these dogs are trained, they can rescue both humans and dogs from life-threatening conditions. Grazioso Salvare has partnered with animal shelters around Austin, Texas to select dogs for their training program. However, Grazioso Salvare has a certain profile they look for in dogs to train. This program is designed to help GS work with existing data from the animal shelters in Austin to identify and categorize available dogs.

Grazioso Salvare requested a dashboard with geolocation mapping, user interaction, and charts that helped find dogs that were good candidates for their mission. *Rescue Mission* uses Austin Animal Center (AAC) database and prior to the creation of *Rescue Mission*, animal\_shelter.csv was imported into the AAC database. An admin and user account were created to access the AAC database to find and/or modify any information within the AAC database. To support all functions, animal\_shelter.py was created to support the CRUD functions for the database.

## Motivation

Dogs are a man’s best friend. This software assists in the process of Grazioso Salvare’s mission and allows dogs to be used for a greater purpose. It gives dogs in shelters a second chance at serving a greater purpose.

## Getting Started

1. Enter Mongo and import the provided aac\_shelter\_outcome.csv file into the AAC database and local animals’ collection within your local port by using the mongoimport tool.

*Text

Description automatically generated*

1. Create a simple and compound index for the imported documents within the AAC database.

*Text

Description automatically generated*

1. Create an admin and user account to allow the user to readWrite within the documents in AAC.animals.

*Text

Description automatically generated*

*Text

Description automatically generated*

1. Have a mongoDB database with the appropriate user authentication (username and password) to access the data.

Text

Description automatically generated

1. Check to ensure that any newly created log in information has the ability to manipulate the AAC database.
2. Create animal\_shelter.ipynb and animal\_shelter.py files to be used in the Project2 files.
3. Update the port number on local host of the animal\_shelter.py file because it is essential when importing the file.
4. Update the “aacuser” and “password” that was created.

Text

Description automatically generated

1. Create CRUD functions.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Create a Dash web application with the desired HTML/CSS layout for the dataframe, map, and chart.

Text

Description automatically generated

Graphical user interface, text

Description automatically generatedText

Description automatically generated

1. Create an app callback to populate the data frame with the data.

Text

Description automatically generated

1. Create a callback to update the map with map markers based on user selection and filters.

Text

Description automatically generated

1. Create a pie chart from the displayed data.
2. Create a callback that updates the pie chart based on filtered results.

Graphical user interface, text, application

Description automatically generated

## Tools

*MongoDB* will need to be installed to access the entire project.

*Jupyter Notebook* is used to write and create ipynb files and run files to check the output of the files.

*PyMongo/ Leaflet* are libraries that need to be accessible because they are the libraries that are used to create the functions within the dashboard. Some of the functionality of these libraries create tables, data frames, charts (pie charts and bar charts), and the geolocation map.

*Ploty* must be imported because this is a charting tool for Python applications that will allow us to generate the proper charts needed. In this case, we needed the pie chart.

*Dash/Pandas* is the frameworks used to build web applications. Pandas is used for Python to create data frames.

**Known issues (Dashboard):**

The animal\_shelter.py and ProjectTwo.ipynb do not want to work seamlessly together. The code has been completed however, I keep getting a local host error that connection is refused.

There were many challenges with this assignment. In the very beginning, my port in MongoDB was refusing to close. This did not allow me to make changes on the aacuser and admin user for the database. After a couple weeks of troubleshooting, I was able to get this resolved.

When trying to build the dashboard, I kept getting an error that the connection with the port was refused. I am assuming that this was coming from my previous troubles with closing and opening the port. I looked for many different ways to fix this but the bugs were still present in the final build, however all coding is correct.

## Contact

Your name: Hanah Deering

Hanah.deering@snhu.edu