# CS 340 README Template

## About the Project/Grazioso Salvare Web Application Dashboard

The purpose of this project is to provide a user-friendly web application that allows users to sort through the AAC database for specific breeds, ages, and sex based on a selection on the drop-down menu.

## Motivation

This project exists to aid Grazioso Salvare in looking for specific characteristics in dogs in animal shelters in Texas to rescue and train to complete rescue work. This application will not only save many dog’s lives but also save many human lives.

## Getting Started

To get a local copy up and running, the first step is to create/import the AAC shelter outcomes csv file into MongoDB.

A computer screen shot of a program

Description automatically generated

The next step is to create an account with read-and-write access to the AAC database.

A screenshot of a computer screen

Description automatically generated

Next, download a copy of the “animalShelter.py” file and the “ProjectTwoDashboard.ipynb” file and import the files into Jupyter Notebook. You will also need to import the image into Jupyter notebook to display the dashboard properly.

## Installation

The tools required for this project include MongoDB, Python, and Jupyter Notebook. Each of the software listed can be easily installed from their websites online and instructions are provided on the sites for multiple different systems.

## Usage

This application can be used to simplify access to the Animal Shelter Outcomes csv and provides four different filtering options. The filtering options include “Water”, “Mountain”, “Disaster”, and “All.” When the drop-down for water is selected, dogs matching the requirements for breed, age, and sex should populate the data table. The same effect should happen for both the mountain and disaster drop-downs. The all drop-down basically resets the filtering options and displays all animals in the database. The pie chart beneath the data table displays preferred animals based on breed and their respective percentages. The map is updated with the location of the animal selected in the data table.

### Code Example

*A screen shot of a computer

Description automatically generated*

The example of code pictured above is extremely important to access the files in the database. This method was used multiple times throughout the creation of the web application to list the animals based on the filtering options.

*A screen shot of a computer code

Description automatically generated*

The second example of code pictured above displays the creation of the title of the web application, the insertion and placement of the logo, and the drop-down menu options.

### Screenshots

The screenshot below displays the web application upon first running it, it is set to automatically display all animals in the database.

A red line drawing of a dog

Description automatically generated

*A screenshot of a computer

Description automatically generated*

The image below depicts the available filtering options for users.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The image above displays how the map locates any selected animal from the data table.

## Contact

Crystal Berkhan