# CS 340 README Template

## About the Project/Project Title

*Austin Animal Center (AAC) Database is a database that contains all animals in the shelter based on a variety of variables including when they were entered into the shelter, breed, name, and age. We are utilizing filters to improve search functions for name, breed, age, gender, species, and outcome. Geolocation is factored in to see a real time location for a specific animal.*

## Motivation

*I began implementation to help with organization for AAC. They have thousands of animals of varying breeds and keeping handwritten documentation would be a nightmare. This simplifies many steps to make the user experience as friendly as possible. Grazio Salvare wants to implement more filters to specify between Water Rescue, Mountain or Wilderness Rescue, Disaster Rescue, or Individual Tracking. Certain breeds are better suited for each of these roles and the dashboard is meant to have this functionality included. I chose to use MongoDB because setting up a database is fairly quick to do, and it has workability with Python. Python is fairly forgiving when writing code, but the indentation matters greatly. Using the DASH application, it was simple to see results in real time and changes could be made without too much time in between.*

## Getting Started

*First gain access to the database as seen below. Data will be imported if you have already downloaded the csv file.*

*A screenshot of a computer

Description automatically generated*

*You will now set your username and password for the databse. Then you will find your port number and host after following the below steps.*

*A screenshot of a computer screen

Description automatically generated*

## Installation

You will need Jupyter Notebooks, Python and MongoDB for everything to run smoothly. The installation of each is detailed explained below.

**Jupyter Notebooks**: Jupyter can be installed from the command line in any major operating system. <https://jupyter.org/install>.

**Python**: Detailed installation instructions for Python are available here: <https://realpython.com/installing-python/>. Once you have Python installed, you should be able to use this program from the Terminal on Mac or Linux or from the Command Prompt for Windows.

**MongoDB**: MongoDB comes in Community or Enterprise editions. <https://docs.mongodb.com/manual/installation/>.

**Plotly**

Plotly is a charting tool for Python applications and can be imported directly into your Python module from your Jupyter notebook. It is necessary for the rendering of charts. <https://plotly.com/python/getting-started/>

**Dash**

Dash is a framework used to build web applications. You can import the Dash Core Components into your Jupyter notebook.

<https://dash.plotly.com/installation>

**Pandas**

Pandas is used in this web application as well. Pandas is a tool for Python that creates data frames.

<https://pandas.pydata.org/docs/getting_started/install.html>

### Usage

The dashboard has the unfiltered data presented with a unique header and the Grazio Salvare Logo as requested.

A screenshot of a computer

Description automatically generated

There is a load screen when first loading due to the sheer number of items in the file.

A computer screen shot of a computer screen

Description automatically generated

Specific filter based on each filter list will show results only for that range.

A screenshot of a computer

Description automatically generated

Geolocation pulls individual search results in real time.

Pie chart is pulling data from the entire csv file.

A screenshot of a computer

Description automatically generated

Each individual “rescue type” filter is currently non functioning with callback errors.

*A screenshot of a computer

Description automatically generated*

### Tests

Create, Read, Update, & Delete Functions are demonstrated below.

A computer screen shot of a computer screen

Description automatically generatedA computer screen shot of a computer screen

Description automatically generated

Screenshots & Code Example

*Screenshots from the dashboard application code.*

*Setting imports*

*A screenshot of a computer

Description automatically generated*

*Display for the header of the application.*

*A computer screen shot of a computer

Description automatically generated*

*Setting the specifications of the rows and presentation.*

*A screenshot of a computer

Description automatically generated*

*Sets up pie chart next to geolocation.*

*A screenshot of a computer

Description automatically generated*

*Sets callback for the Water, Mountain & Wilderness, & Disaster Rescues & Individual Tracking methods*

*These callbacks are not functional at this time and return a Schema error.*

*A computer screen shot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated*

*Sets the data for the pie chart and loading screen.*

*A screenshot of a computer

Description automatically generated*

*Sets up geolocation.*

*A computer screen with text

Description automatically generated*

*Challenges:*

*I had a tough time getting the dashboard to read from the database at first but then was helped to use a better document list. I am still unsure of how the callbacks for the filtered searches are not functioning.*

## Contact

Robert Lulashi