# CS 340 README-Hillary Loyd

## About the Project/Project Title

This project is a web application that can work with the existing data from animal shelters. This interface will sort through the database, through MongoDB, to return the information that is needed to determine if the animal is a good candidate for training. This project is made up of a database, API, and a interactive dashboard.

## Motivation

Mr. Salvare would like to identify three factors for the animals in the shelters. The first being that they are a dog. The second is the breed of dog. The last identifier is that they must be less than 3 years of age.

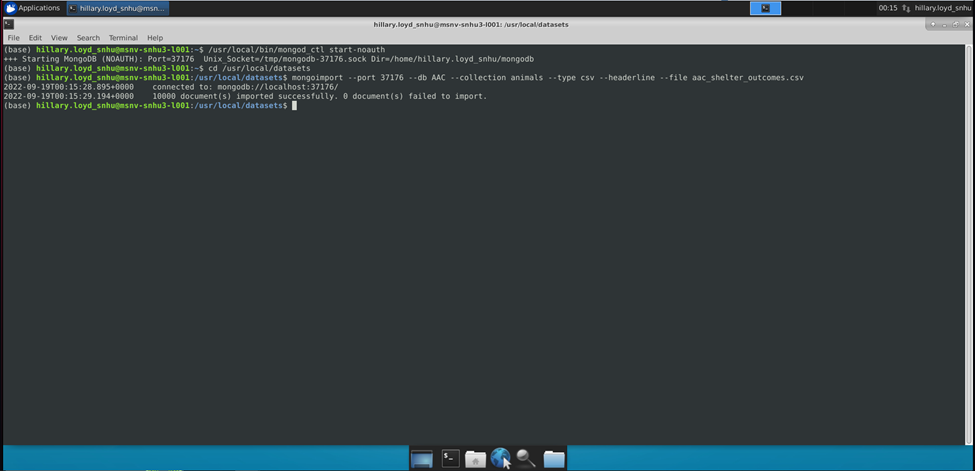
MongoDB is used because its a quick setup of the database from a csv document, like we have used, as well as a Python friendly interface. It is also a free and open source which is what the client has asked for the project to be. MongoDB can support large scale projects with rapid development timelines. Python was also used because it is a flexible and well-known programming language. Used in many applications today and very easy to learn and use. It pairs well with data analysis and can communicate with MongoDB very well. For MongoDB and Python to work with one another you need PyMongo.

Dash was a great choice for building the dashboard due how dynamic it is in nature. Dash is a react JavaScript tool that provides a responsive framework. It involves html Dash tags that control outputs to segments. Next it will update to any of the target inputs that have been specified in the callback process. This is based on the instructions that have been programmed in the Python module.

## Getting Started

There is a series of steps that need to be taken to get started.

1. Import the Austin Animal Care database or “AAC” and create an index.



Text

Description automatically generated

1. A new user must be made in the AAC database. This is done with a write privilege so the python code can access the database.

Graphical user interface, text, application

Description automatically generated

1. Make sure that the created user can log in by checking with authenticationDatabase.

Text

Description automatically generated

1. Update the port number to match your localhost, as well as adding in the username and password.

Text

Description automatically generated

## Installation

The tools needed to run this dashboard is Jupyter Notebook, Python for command line, Plotly, Dash, Pandas, and MongoDB.

Jupyter Notebook:

Jupyter Notebook can be installed through the command line using the instructions located on the install page of the website located here: <https://jupyter.org/install>. For proxy server detailed instructions you can located these here: <https://jupyterlab.readthedocs.io/en/stable/getting_started/installation.html>.

Python:

The location of the detailed instructions for installation of Python can be found here: <https://realpython.com/installing-python/>. Once this is installed you can use the command terminal for Mac and Linux and the command prompt for Windows.

MongoDB:

There are two editions for MongoDB and that is the Community and Enterprise Editions. The detailed instructions for downloading and installing MongoDB can be found here: <https://docs.mongodb.com/manual/installation/>.

Plotly:

To see the charts for the dashboard you will need to install Plotly to be able to make them proper. This is a charting tool for Python applications and can be directly imported into the Py module from the Jupyter Notebook. For information on the install of Plotly can be found here: <https://plotly.com/python/getting-started/#installation>

Dash:

Dash is the framework for the base of this web application. You can import the Dash Core Components into the notebook itself. For information on the install of Dash can be found here: <https://pypi.org/project/dash/#files>

Pandas:

Pandas is used to create the data frames for this web application. For the files to download and install the files can be located here:

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

## Usage

This application has a few functions. The first that is used is the Radio items that allow you to sort the data by training. This will sort the animals by dog, breed, age, and sex. The specifications are determined by Mr. Salvare. When clicking on any of these items it will allow the table to sort through the specifications and display them on the table. The last radio item is the reset. This will allow the user to reset the table back to the default. The next function is the pie chart. This will sort the data given by breed and output to the pie chart of the current data frame. The last function is the interactive map. The user can select up to three animals from the table and the map will show the location of the animal. When clicking on the blue arrow on the map the user can see the tool tip that gives the name, breed, sex, and age.

### Dashboard

**Water Rescue**

Graphical user interface, application, table

Description automatically generated

**Mountain/Wilderness Rescue**

Graphical user interface, application, table

Description automatically generated

**Disaster Rescue/Individual Tracking**

Graphical user interface, application

Description automatically generated

**RESET**

Graphical user interface

Description automatically generated with medium confidence

### Code

Dog Specifications and reset

Text, timeline

Description automatically generated

Map Markers/ToolTips

Text

Description automatically generated

Text

Description automatically generated with low confidence

Table

Description automatically generated with medium confidence

**Pie chart**

Graphical user interface, text, application

Description automatically generated

**CRUD:**

**Create**

Graphical user interface, text, application

Description automatically generated

**Read**

Graphical user interface, text, application

Description automatically generated

**Update**

Graphical user interface, text, application

Description automatically generated

**Delete**

Graphical user interface, text, application

Description automatically generated

**TESTING :**

**Create**

Text

Description automatically generated

**Read**

Graphical user interface, text, application, email

Description automatically generated

**Update**

Graphical user interface, text, application

Description automatically generated

**Delete**

Text

Description automatically generated

**Results**

Graphical user interface, text, application, email

Description automatically generated

## Challenges/Issues

One of the issues is that when working in Jupyter Notebook you can only see one error at a time. When I was working with the Project Two code and the code for some of the other assignments in this class it took me awhile to figure out how to read the error and what was happening. After you see that it is one error, you fix it, and then the next populates. While I do like this feature, it was not something that I was used to. I did not have any issues with the Dashboard itself other than figuring out what row was what when it came to displaying the information to the tooltip. I enjoyed watching the entire web application come together.

## Contact

Your name: Hillary Loyd