# CS 340 README - DASHBOARD

## About the Project/Project Title

*This project is a dashboard of Austin Animal Rescue Database, it consists of a table previewing data in the database, and the table has filtering options. The dashboard also employ a pie chart and a map, that are interactive with the data shown in the table.*

## Motivation

*Websites and services consists of a database and a front end HTML pages. In this project we use Dash as a framework to code the model, view and controller. The database used is MongoDB and it is an example of a real website.*

## Getting Started

*In order to get MongoDB running locally on the machine, we can run*

**

*This will instantiate a mongo deamon on the machine and launch a server. We will get a port number when the server is up.*

*Optionally we can access the database by running the following command with the respective credentials*

**

*Here as we can see we are authenticating with MyAdmin username and jadrehaoui as a password and we are entering as an admin role.*

*We need Jupyter Notebook, to run dash jupyter and a mongo client like pymongo.*

## Installation

*MongoDB*

*Python*

*JupyterDash*

*Pymongo*

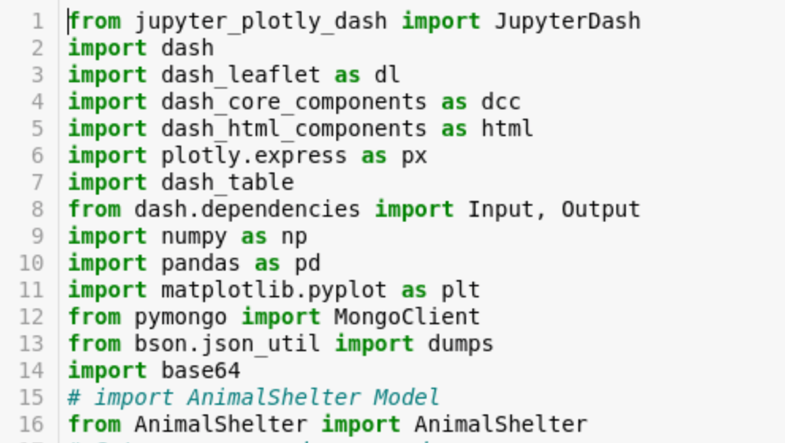
*Pandas*

*Numpy*

## Usage

*First lets start by importing the corresponding libraries for this project*

### Code Example



Here Animal Shelter is a custom module written by us, serves as a controller

*Then we set up the credentials in a username and password, in order to make the code cleaner and more flexible. Then we initiate the controller to connect to the database.*

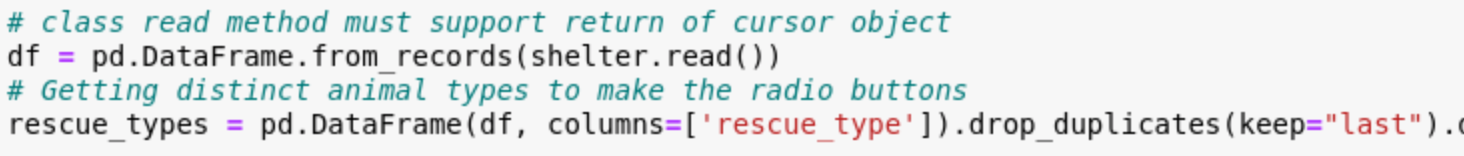
### Code Example

Text

Description automatically generated with medium confidence

*Next we query all the data in the animals collection in order to fill the data table. We declare a Data Frame by pandas. And then we filter from that data frame on rescue type in order to get distinct rescue types in the collection.*

### Code Example



*Next we prepare the image we need to show on the page and we encode it in order for the HTML to read it correctly and we declare the variable app to start implementing the layout.*

### Code Example

Text

Description automatically generated with medium confidence

*Now we are ready to start with the app layout, we start with a div and inside that div we can code what we want, for this project we need*

1. An image
2. A header
3. A row of radio button for custom filtering on Rescue type
4. A table
5. A pie chart
6. Map

### Code Example

A picture containing text

Description automatically generated

*The pie chart and the map will be next to each other in one div. ( refer to code)*

*Then we start implementing the functionality and the interactivity procedure using callbacks already found in the dash framework. Here’s how the radio buttons functionality goes.*

### Code Example

*Graphical user interface, text, application

Description automatically generated*

*Here we are listening to the radio button that has the id as radio-id, and getting the value out of it, and returning data to the data table with the id datatable-id.*

### Screenshots

**

*In this screenshot we can see that the image, the header, the radio buttons and the table are being rendered. We can see that the table has the data from the database. And the Rescue type is also computed using the same data got from the database.*

*To verify that the radio buttons work, here’s a screenshot of the Mountain or Wilderness Rescue type selected.*

**

*Below the table, we implemented a Pie chart and a map, that are interactive to the data being viewed, as the data is reset, we can see that the pie chart shows this output*

*Chart, pie chart

Description automatically generated*

*But when the data get filtered, we can see that the pie chart values differ.*

*Chart, bubble chart

Description automatically generated*

*Lastly we have the map, that is also interactive.*

*Map

Description automatically generated Map

Description automatically generated Map

Description automatically generated*

*Map

Description automatically generated*

## Contact

Jad Alrehaoui

**Code:**

<https://github.com/jadalrehaoui/Client-Server-Development.git>

Thank you for reading !