# Module 7 Project Two Dashboard README Curtis Felsher

## About the Project/Project Title

This project was designed for Grazioso Salvare to help identify dogs that are good candidates for search and rescue missions. This interactive dashboard allows users to navigate a database that contains animal data in an easy and graphical way. The application visualizes data from the dataset via a table, pie chart, and a geolocation map. The dashboard utilizes MongoDB for the database, Dash for the web framework, Plotly for the interactive visuals, and Python as the logic.

## Motivation

The motivation behind this project was to provide an easy-to-use graphical interface to scrape through animal data contained within a database. Users can use this dashboard to apply filters to view animals in different categories that are best suited for the task based off pre-applied filters within the drop-down menu. Users can select individual animals to view details about the animal such as their name and location on the geolocation map.

## Getting Started

To get this project working on your machine, follow these steps.

1.Copy the files to your local machine and ensure all files are in the same location, such as a folder.

2. Ensure all necessary python libraries needed for the project are installed (Installation section will go more into detail)

3. Check connection to MongoDB database to ensure you have access to the instance needed for the animal data.

4. Ensure Jupyter Notebook is installed to run the .ipynb.

5. Navigate to the directory that contains “animalShelter.py” as well as “ProjectTwoDashboard.ipynb”. This should be in the same directory

6. Open “ProjectTwoDashboard.ipynb” in Jupyter Notebook and run the cells. This will provide a link to click at the bottom that will take the user to the dashboard via webpage.

## Installation

Before running the application, you will need the following python libraries if they are not already added:

Pandas

Dash

Jupyter\_dash

Dash\_leaflet

Plotly

Pymongo

These packages can be installed easily via “pip install pandas dash jupyter\_dash dash\_leaflet plotly pymongo” if needed.

You will also need MongoDB. Follow the instructions specific to your OS from the MongoDB documentation from their website: <https://www.mongodb.com/docs/manual/installation/>

Jupyter notebook is also needed for this project and be installed using “pip install notebook”.

To launch Jupyter notebook, simply type “jupyter notebook” in your terminal where a web browser will then open, allowing you to navigate to the directory of the project.

## Usage

After running the cells in Jupyter Notebook that contains the dashboard code, navigate to the localhost provided at the bottom of the cells to access the dashboard. The dashboard should display a data table that contains all fields within the MongoDB database, a pie chart, geolocation data for an animal once selected, and a drop-down menu at the top that provides pre-made filters to show animals most capable for the task selected. These filters will influence all fields of data.

### Code Example

@app.callback(Output('datatable-id','data'),

[Input('filter-type', 'value')])

def update\_dashboard(filter\_type):

if filter\_type == 'Water':

df\_filtered = pd.DataFrame.from\_records(db.read({"animal\_type": "Dog", "breed": {"$in": ["Labrador Retriever Mix", "Chesapeake Bay Retriever", "Newfoundland"]}, "sex\_upon\_outcome": "Intact Female", "age\_upon\_outcome\_in\_weeks": {"$gte": 26, "$lte": 156}}))

# Code for other filter types

else: # Reset

df\_filtered = df

data = df\_filtered.to\_dict('records')

return data

This code example takes the filter contained in the drop-down menu and applies the filter to the data that is read from the MongoDB database. The filtered data is then displayed within the data table to show only relevant results.

### Tests

To test the functionality of the dashboard, the user can do a few different things.

1. Once the dashboard has loaded, ensure the data table for the database, pie chart, and map are all displaying correctly. This will ensure that the data is being passed through properly.
2. Apply each filter from the top to ensure that each visual element updates according to the filtered data that is displayed.
3. Select different animals within the table to ensure the location and data updates correctly within the geolocation map,

### Screenshots ( Sideways Screen shots will be provided at the end to maximize the size of the screenshot)

Starting State of Dashboard

A screenshot of a map

Description automatically generated with medium confidence

Water Rescue Filter

A screenshot of a map

Description automatically generated with medium confidence

Mountain or Wilderness Rescue Filter

A screenshot of a map

Description automatically generated with medium confidence

Disaster or Individual Tracking Filter

A screenshot of a map

Description automatically generated with medium confidence

Reset Filter

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a map

Description automatically generated with medium confidence

A screenshot of a map

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a map

Description automatically generated with medium confidenceA screenshot of a map

Description automatically generated with medium confidence

## Contact

Your name: Curtis Felsher