# CS 340 README Project Two

## About the Project/Project Title

This project was developed for the Grazioso Salvare to create a Dashboard that allows users to take the existing data from an animal shelter to identify and categorize available dogs for their search-and-rescue training.

## Motivation

The motivation is to create a widely usable dashboard that can help Grazioso Salvare to identify which dogs at shelters would be good for their search-and-rescue training in order to give dogs in shelters a new life that helps others. It can save Grazioso Salvare a lot of time and manpower to utilize this software.

## Getting Started

To get a local copy up and running you would need to modify the crud file to access your specific database and ensure the correct credentials are entered. You will also need to run it with Jupyter notebook and MongoDB.

## Installation

You will need pymongo and the mongodb installed for this to function correctly. I utilized Python 3.9.12 and MongoDB 6.0.13.

## Usage

*Use this space to show useful examples of how your project works and how it can be used. Be sure to include examples of your code, tests, and screenshots.*

### Code Example

Below is the code for filtering the table and returning the filtered table.

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

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

def update\_dashboard(filter\_type):

df = pd.DataFrame.from\_records(db.read({}))

if filter\_type == 'WR':

df = pd.DataFrame.from\_records(db.read({'$and': [

{'$or': [

{'breed': 'Labrador Retriever Mix'},

{'breed': 'Chesapeake Bay Retriever'},

{'breed': 'Newfoundland'},

]},

{'sex\_upon\_outcome': 'Intact Female'},

{'age\_upon\_outcome\_in\_weeks': {'$gte': 26}},

{'age\_upon\_outcome\_in\_weeks': {'$lte': 156}}

]}))

elif filter\_type == 'MWR':

df = pd.DataFrame.from\_records(db.read({'$and': [

{'$or': [

{'breed': 'German Shepherd'},

{'breed': 'Alaskan Malamute'},

{'breed': 'Old English Sheepdog'},

{'breed': 'Siberian Husky'},

{'breed': 'Rottweiler'}

]},

{'sex\_upon\_outcome': 'Intact Male'},

{'age\_upon\_outcome\_in\_weeks': {'$gte': 26}},

{'age\_upon\_outcome\_in\_weeks': {'$lte': 156}}

]}))

elif filter\_type == 'DRIT':

df = pd.DataFrame.from\_records(db.read({'$and': [

{'$or': [

{'breed': 'Doberman Pinscher'},

{'breed': 'German Shepherd'},

{'breed': 'Golden Retriever'},

{'breed': 'Bloodhound'},

{'breed': 'Rottweiler'}

]},

{'sex\_upon\_outcome': 'Intact Male'},

{'age\_upon\_outcome\_in\_weeks': {'$gte': 20}},

{'age\_upon\_outcome\_in\_weeks': {'$lte': 300}}

]}))

elif filter\_type == 'RESET':

df = pd.DataFrame.from\_records(db.read({}))

columns = [{"name": i, "id": i, "deletable": False, "selectable": True} for i in df.columns]

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

df.drop(columns=['\_id'],inplace=True)

return data

### Tests

The test is extremely easy as you run the Dashboard and then just click on the appropriate radio button to filter the data.

### Screenshots

*A screenshot of a computer

AI-generated content may be incorrect.*

*A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.*

*A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.*

## Roadmap/Features (Optional)

The issue I keep encountering is an error with the data table populating after the initialization.

## Contact

Your name: Ruby Bingham