Ericka Resendez
Southern New Hampshire University
CS-340 Client/ Server Development
February 21, 2025

Project Two README

About the Project/Project Title
This project is a data visualization dashboard developed for Grazioso Salvare, an animal rescue organization. Grazioso Salvare is searching for and identifying dogs that are good candidates for search and rescue training from five animal shelters in the region of Austin, TX. The dashboard provides interactive filtering and visualization of animal data stored in MongoDB using a pie chart and geolocation to aid in selecting suitable rescue dogs for various operations, such as water rescue, mountain/wilderness rescue, and disaster/individual tracking. The dashboard filters by gender, breed, and age to choose the best candidates.

Motivation
The motivation for this project is to develop an interactive dashboard for Grazioso Salvare. This interactive dashboard will allow the client to help identify dogs for search and rescue training that will help rescue animals and humans.

Getting Started
To get a local copy up and running, follow these simple steps:
1. Have MongoDB, Python, and Jupyter Notebook installed.
2. Update AnimalShelter class with appropriate credentials.
3. Import csv file aac_shelter_outcomes.csv, if necessary, using monogimport.
4. Create a Python file to begin enabling CRUD functionality for the database to get started.
5. Launch Jupyter Notebook to begin with interactive dashboard.

Installation
MongoDB:
1. Download MongoDB using the MongoDB Install page and can be found at https://www.mongodb.com/docs/manual/installation/

Python Libraries:
1. Install Python if not installed. This article is from Python.Land has instructions for installing Python on Windows, MacOS, and Linux: https://python.land/installing-python
2. Ensure pymongo is installed to interact with MongoDB using the pip install pymongo command.

JupyterDash:
1. Jupyter Notebook can be installed using the pip install notebook command.
2. The Jupyter Install page has further instructions and can be found at https://jupyter.org/install

Dash Leaflet:
1. Install Dash Leaflet to view the animal's location, it is an interactive map. You can do this by using the command pip install dash-leaflet. Further information can be found at: https://www.dash-leaflet.com/docs/getting_started

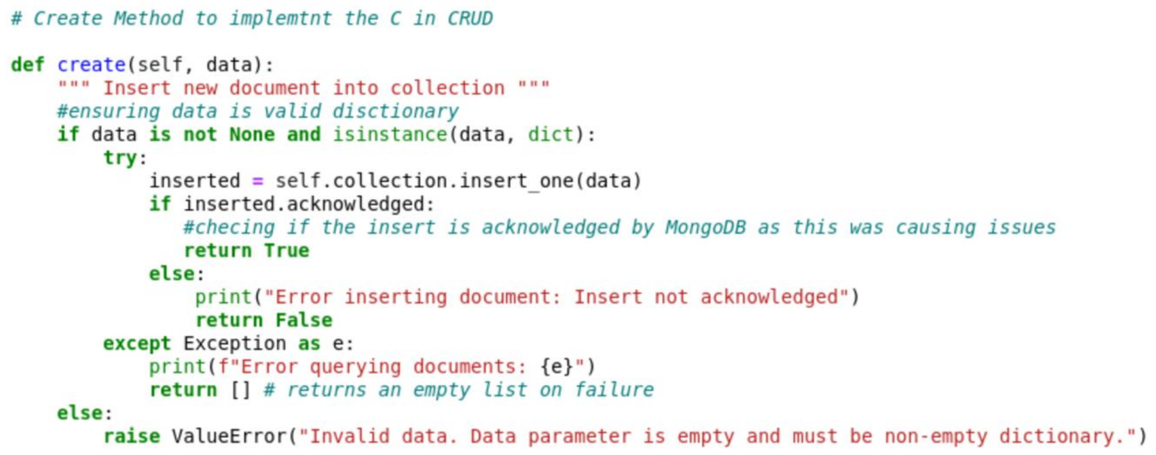
Note: This template has been adapted from the following sample templates: Make a README, Best README Template, and A Beginners Guide to Writing a Kickass README.

Usage

Code Example

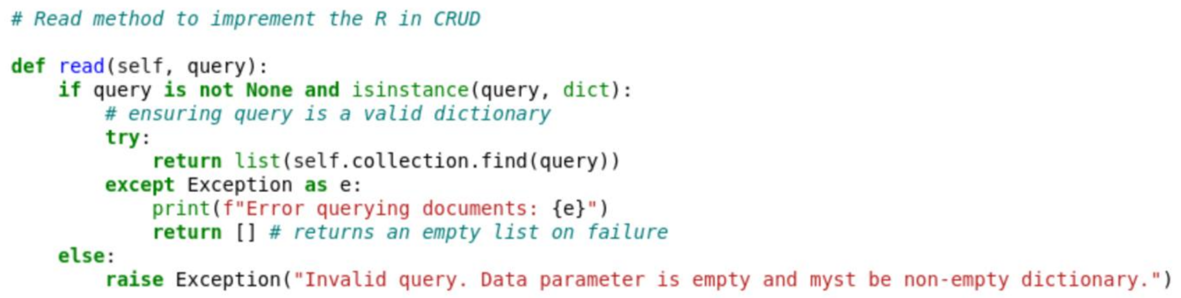
This project utilizes the aac_shelter_outcomes.csv and the CRUD functionality (create, read, update, and delete records). To ensure the dashboard is interactive, we use def update_dashboard(filter_type), def update_map(viewData, index), and def update_graphs(viewData) to dynamically respond to the filtering options. This project also makes use of radio items as interactive filtering options.

Create method to insert a new animal into the database.



```python
# Create Method to implemtnt the C in CRUD

def create(self, data):
    """ Insert new document into collection """
    #ensuring data is valid disctionary
    if data is not None and isinstance(data, dict):
        try:
            inserted = self.collection.insert_one(data)
            if inserted.acknowledged:
               #checing if the insert is acknowledged by MongoDB as this was causing issues
               return True
            else:
                print("Error inserting document: Insert not acknowledged")
                return False
        except Exception as e:
            print(f"Error querying documents: {e}")
            return [] # returns an empty list on failure
    else:
        raise ValueError("Invalid data. Data parameter is empty and must be non-empty dictionary.")
```

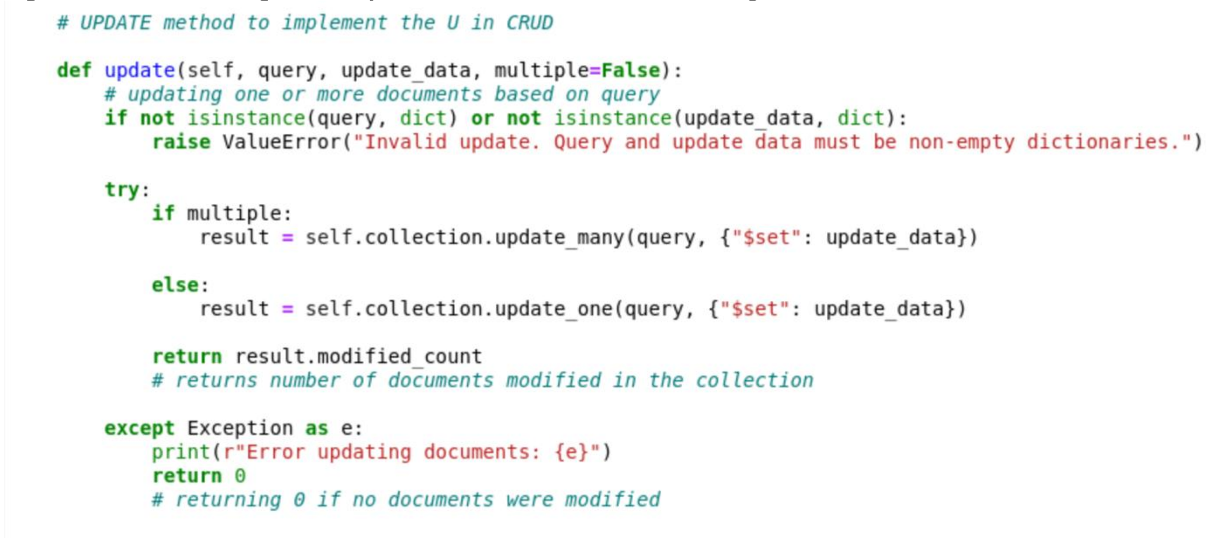
Read Method to find and display animal data.



```python
# Read method to imprement the R in CRUD

def read(self, query):
    if query is not None and isinstance(query, dict):
        # ensuring query is a valid dictionary
        try:
            return list(self.collection.find(query))
        except Exception as e:
            print(f"Error querying documents: {e}")
            return [] # returns an empty list on failure
    else:
        raise Exception("Invalid query. Data parameter is empty and myst be non-empty dictionary.")
```

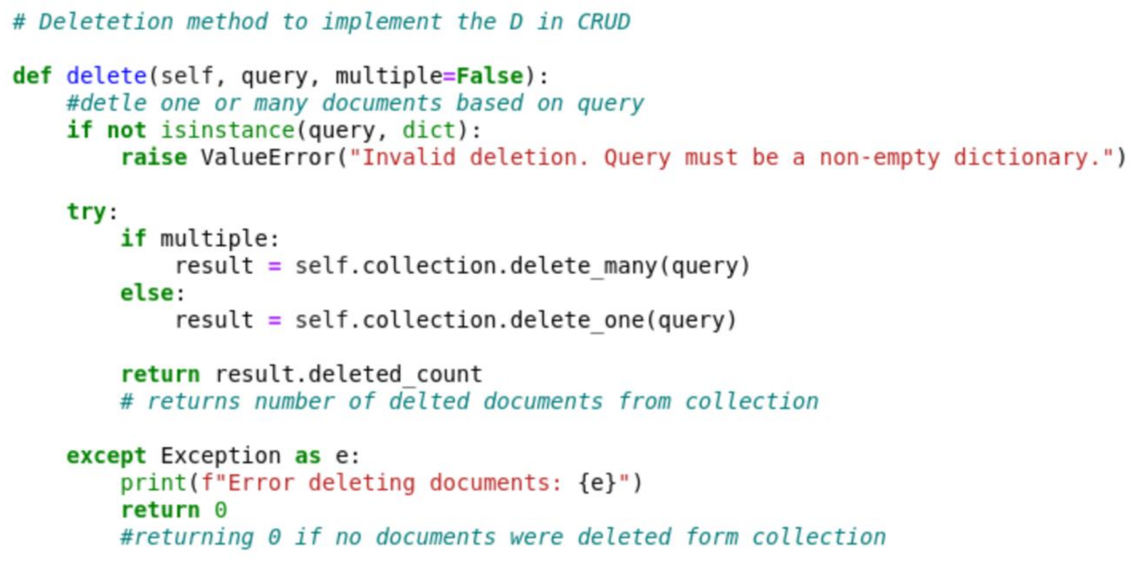
Note: This template has been adapted from the following sample templates: [Make a README](), [Best README Template](), and [A Beginners Guide to Writing a Kickass README]().

Update Method to update any information from an existing animal in the database.



```
# UPDATE method to implement the U in CRUD

def update(self, query, update_data, multiple=False):
    # updating one or more documents based on query
    if not isinstance(query, dict) or not isinstance(update_data, dict):
        raise ValueError("Invalid update. Query and update data must be non-empty dictionaries.")

    try:
        if multiple:
            result = self.collection.update_many(query, {"$set": update_data})

        else:
            result = self.collection.update_one(query, {"$set": update_data})

        return result.modified_count
        # returns number of documents modified in the collection

    except Exception as e:
        print(r"Error updating documents: {e}")
        return 0
        # returning 0 if no documents were modified
```

Deletion Method to find and delete an existing animal, or many animals from the database.

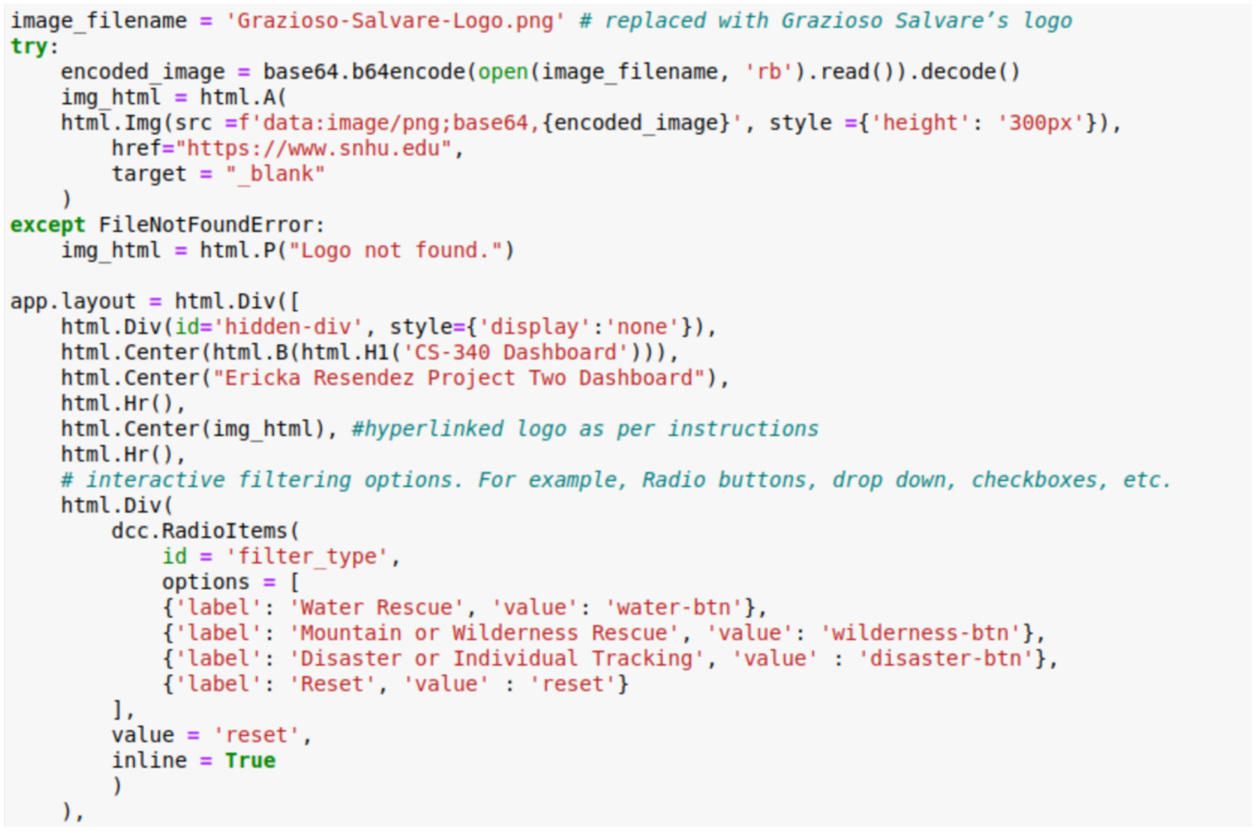


```
# Deletetion method to implement the D in CRUD

def delete(self, query, multiple=False):
    #detle one or many documents based on query
    if not isinstance(query, dict):
        raise ValueError("Invalid deletion. Query must be a non-empty dictionary.")

    try:
        if multiple:
            result = self.collection.delete_many(query)
        else:
            result = self.collection.delete_one(query)

        return result.deleted_count
        # returns number of delted documents from collection

    except Exception as e:
        print(f"Error deleting documents: {e}")
        return 0
        #returning 0 if no documents were deleted form collection
```

Note: This template has been adapted from the following sample templates: Make a README, Best README Template, and A Beginners Guide to Writing a Kickass README.

Ensuring to use the Grazioso Salvare logo and include a URL anchor tag to the client's home page.

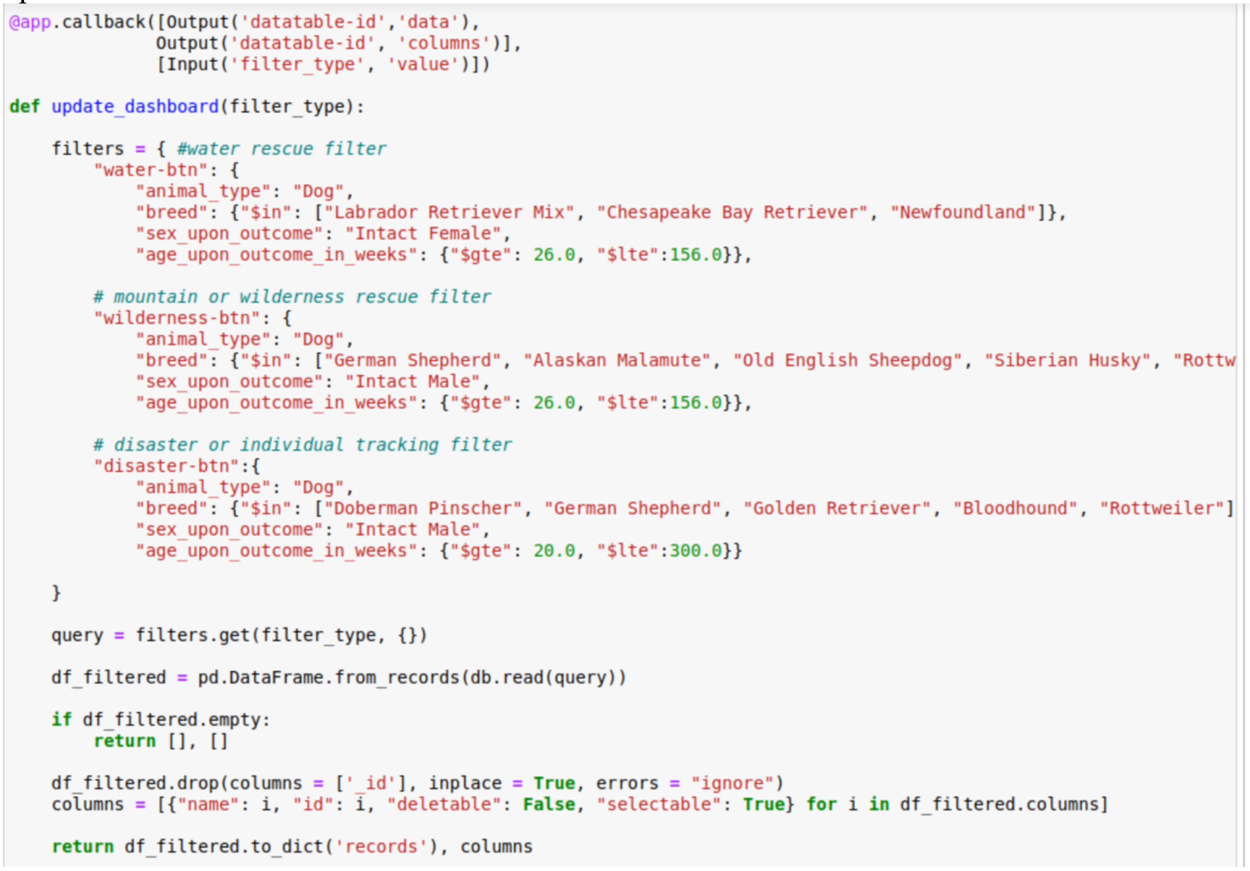
Using Radio Items to create interactive filtering options for dogs in water rescue, mountain or wilderness rescue, disaster rescue or individual tracking, and a reset button.



```
image_filename = 'Grazioso-Salvare-Logo.png' # replaced with Grazioso Salvare's logo
try:
    encoded_image = base64.b64encode(open(image_filename, 'rb').read()).decode()
    img_html = html.A(
    html.Img(src =f'data:image/png;base64,{encoded_image}', style ={'height': '300px'}),
        href="https://www.snhu.edu",
        target = "_blank"
    )
except FileNotFoundError:
    img_html = html.P("Logo not found.")

app.layout = html.Div([
    html.Div(id='hidden-div', style={'display':'none'}),
    html.Center(html.B(html.H1('CS-340 Dashboard'))),
    html.Center("Ericka Resendez Project Two Dashboard"),
    html.Hr(),
    html.Center(img_html), #hyperlinked logo as per instructions
    html.Hr(),
    # interactive filtering options. For example, Radio buttons, drop down, checkboxes, etc.
    html.Div(
        dcc.RadioItems(
            id = 'filter_type',
            options = [
            {'label': 'Water Rescue', 'value': 'water-btn'},
            {'label': 'Mountain or Wilderness Rescue', 'value': 'wilderness-btn'},
            {'label': 'Disaster or Individual Tracking', 'value' : 'disaster-btn'},
            {'label': 'Reset', 'value' : 'reset'}
        ],
        value = 'reset',
        inline = True
        )
    ),
```

Note: This template has been adapted from the following sample templates: [Make a README](), [Best README Template](), and [A Beginners Guide to Writing a Kickass README]().

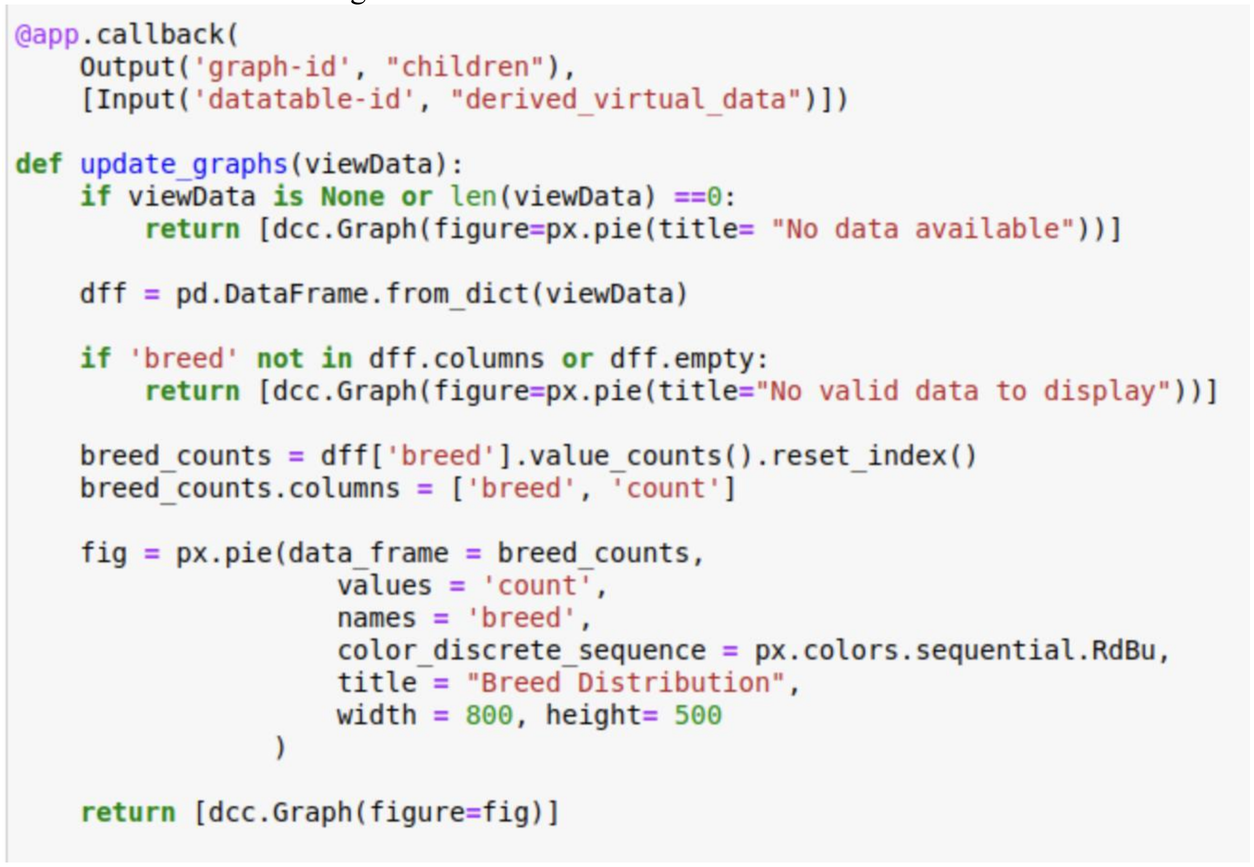
Utilizing the radio item filtering options, ensure the preferred dog breeds, age, and gender for different rescue types are specified and will be displayed correctly when the different filtering options are chosen.



```
@app.callback([Output('datatable-id','data'),
              Output('datatable-id', 'columns')],
              [Input('filter_type', 'value')])

def update_dashboard(filter_type):

    filters = { #water rescue filter
        "water-btn": {
            "animal_type": "Dog",
            "breed": {"$in": ["Labrador Retriever Mix", "Chesapeake Bay Retriever", "Newfoundland"]},
            "sex_upon_outcome": "Intact Female",
            "age_upon_outcome_in_weeks": {"$gte": 26.0, "$lte":156.0}},

        # mountain or wilderness rescue filter
        "wilderness-btn": {
            "animal_type": "Dog",
            "breed": {"$in": ["German Shepherd", "Alaskan Malamute", "Old English Sheepdog", "Siberian Husky", "Rottw
            "sex_upon_outcome": "Intact Male",
            "age_upon_outcome_in_weeks": {"$gte": 26.0, "$lte":156.0}},

        # disaster or individual tracking filter
        "disaster-btn":{
            "animal_type": "Dog",
            "breed": {"$in": ["Doberman Pinscher", "German Shepherd", "Golden Retriever", "Bloodhound", "Rottweiler"]
            "sex_upon_outcome": "Intact Male",
            "age_upon_outcome_in_weeks": {"$gte": 20.0, "$lte":300.0}}

    }

    query = filters.get(filter_type, {})

    df_filtered = pd.DataFrame.from_records(db.read(query))

    if df_filtered.empty:
        return [], []

    df_filtered.drop(columns = ['_id'], inplace = True, errors = "ignore")
    columns = [{"name": i, "id": i, "deletable": False, "selectable": True} for i in df_filtered.columns]

    return df_filtered.to_dict('records'), columns
```

Note: This template has been adapted from the following sample templates: Make a README, Best README Template, and A Beginners Guide to Writing a Kickass README.

The update graph section specifies how the graph should look and what it should contain when different interactive filtering actions are chosen.



```python
@app.callback(
    Output('graph-id', "children"),
    [Input('datatable-id', "derived_virtual_data")])

def update_graphs(viewData):
    if viewData is None or len(viewData) ==0:
        return [dcc.Graph(figure=px.pie(title= "No data available"))]

    dff = pd.DataFrame.from_dict(viewData)

    if 'breed' not in dff.columns or dff.empty:
        return [dcc.Graph(figure=px.pie(title="No valid data to display"))]

    breed_counts = dff['breed'].value_counts().reset_index()
    breed_counts.columns = ['breed', 'count']

    fig = px.pie(data_frame = breed_counts,
                 values = 'count',
                 names = 'breed',
                 color_discrete_sequence = px.colors.sequential.RdBu,
                 title = "Breed Distribution",
                 width = 800, height= 500
             )

    return [dcc.Graph(figure=fig)]
```

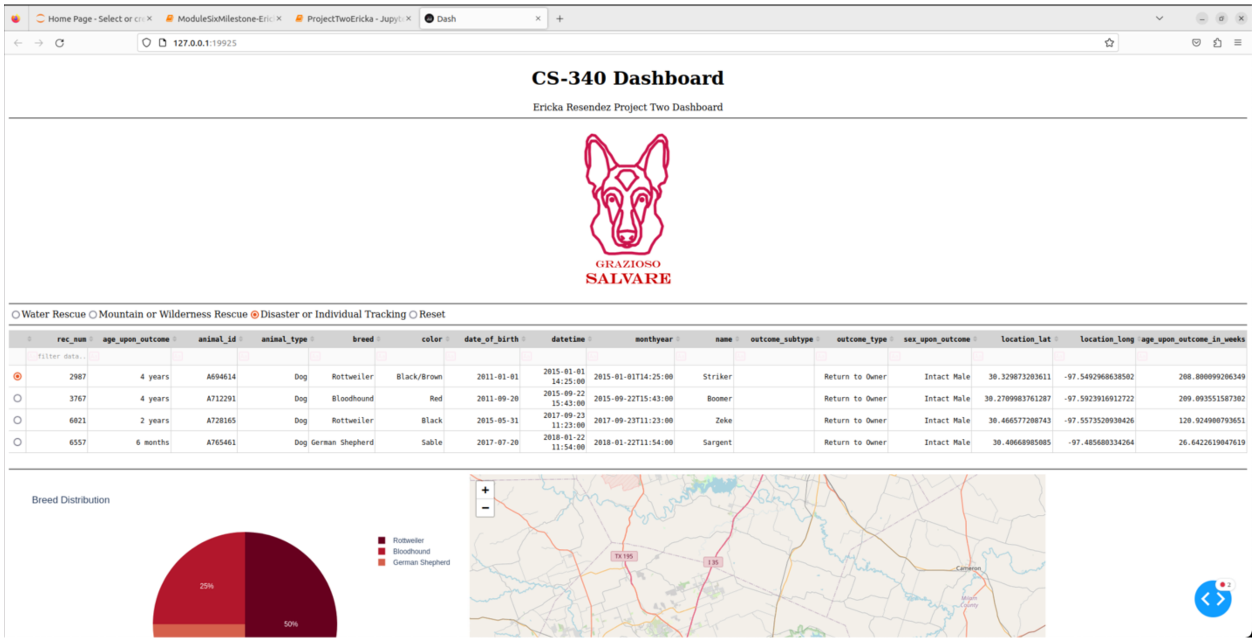
Note: This template has been adapted from the following sample templates: [Make a README](), [Best README Template](), and [A Beginners Guide to Writing a Kickass README]().

The update map section ensures the client can see the exact location of the animals.

```python
@app.callback(
    Output('map-id', "children"),
    [Input('datatable-id', "derived_virtual_data"),
     Input('datatable-id', "derived_virtual_selected_rows")])
def update_map(viewData, index):
    if viewData is None:
        return
    elif index is None:
        return

    dff = pd.DataFrame.from_dict(viewData)
    # Because we only allow single row selection, the list can be converted to a row index here
    if index is None:
        row = 0
    else:
        row = index[0]

    # Austin TX is at [30.75,-97.48]
    return [
        dl.Map(style={'width': '1000px', 'height': '500px'}, center=[30.75,-97.48], zoom=10, children=[
            dl.TileLayer(id="base-layer-id"),
            # Marker with tool tip and popup
            # Column 13 and 14 define the grid-coordinates for the map
            # Column 4 defines the breed for the animal
            # Column 9 defines the name of the animal
            dl.Marker(position=[dff.iloc[row,13],dff.iloc[row,14]], children=[
                dl.Tooltip(dff.iloc[row,4]),
                dl.Popup([
                    html.H1("Animal Name"),
                    html.P(dff.iloc[row,9])
                ])
            ])
        ])
    ]
```

Screenshots

Results:



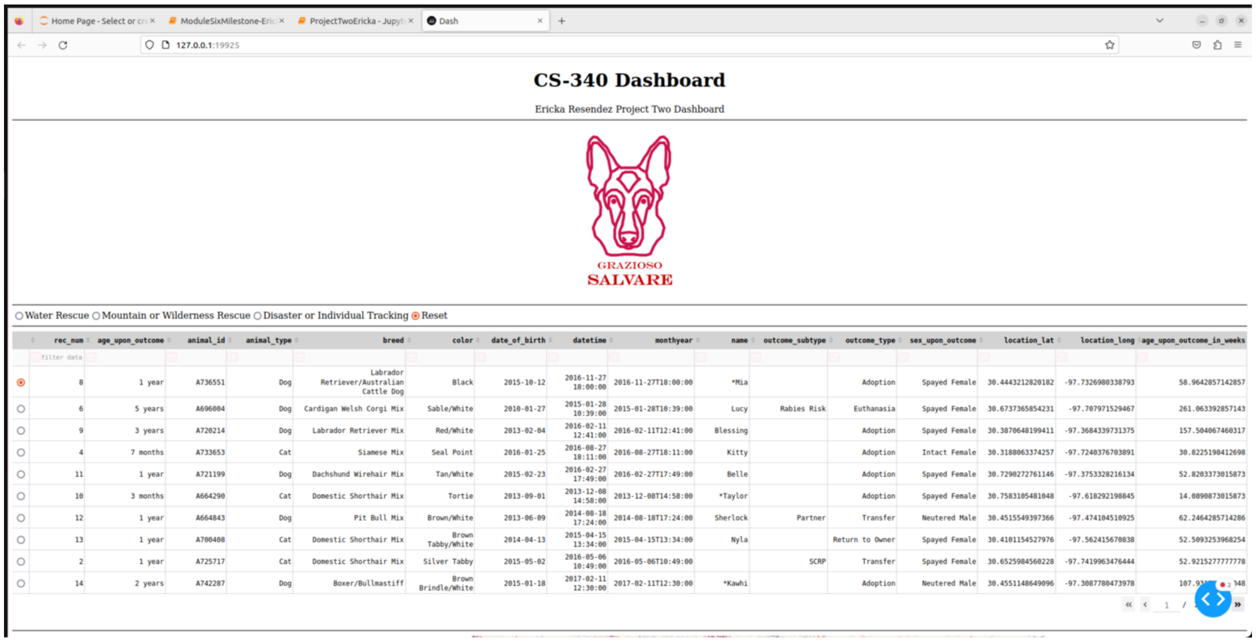
CS-340 Dashboard

Ericka Resendez Project Two Dashboard

Water Rescue Mountain or Wilderness Rescue [✓ Disaster or Individual Tracking] Reset

	rec_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
	filter data...															
●	2987	4 years	A694614	Dog	Rottweiler	Black/Brown	2011-01-01	2015-01-01 14:25:00	2015-01-01T14:25:00	Striker		Return to Owner	Intact Male	30.329873203611	-97.5492968638502	208.800099206349
○	3767	4 years	A712291	Dog	Bloodhound	Red	2011-09-20	2015-09-22 15:43:00	2015-09-22T15:43:00	Boomer		Return to Owner	Intact Male	30.2709983761287	-97.5923916912722	209.093551587302
○	6021	2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Return to Owner	Intact Male	30.466577208743	-97.5573520930426	120.924900793651
○	6557	6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Return to Owner	Intact Male	30.40668985085	-97.485680334264	26.6422619047619

Note: This template has been adapted from the following sample templates: Make a README, Best README Template, and A Beginners Guide to Writing a Kickass README.



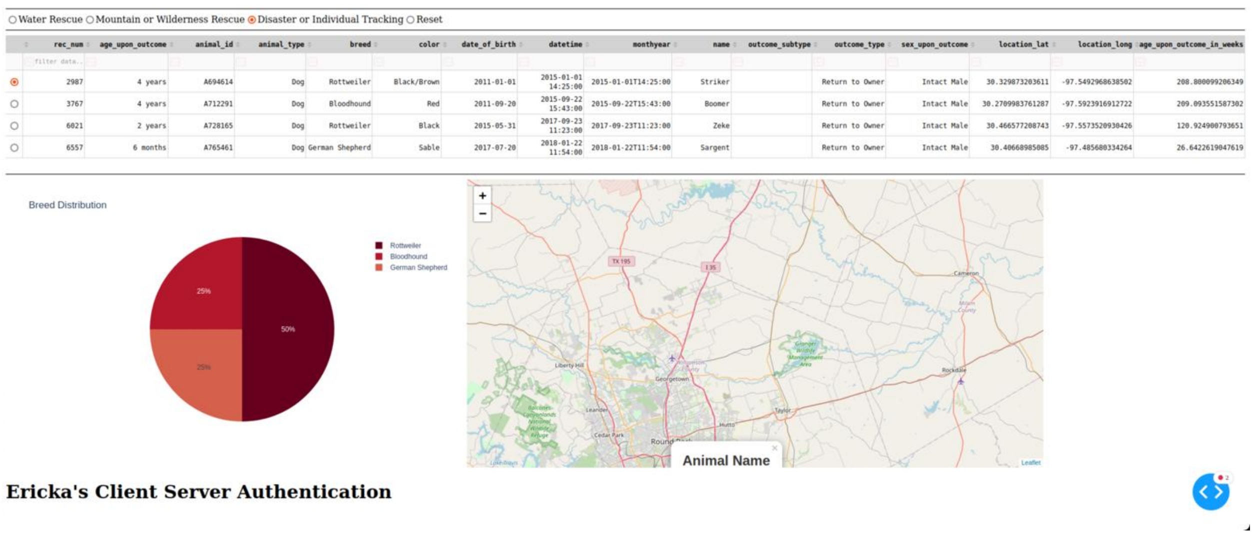
CS-340 Dashboard

Ericka Resendez Project Two Dashboard

Water Rescue Mountain or Wilderness Rescue Disaster or Individual Tracking [✓ Reset]

	rec_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
	filter data															
●	8	1 year	A736551	Dog	Labrador Retriever/Australian Cattle Dog	Black	2015-10-12	2016-11-27 18:00:00	2016-11-27T18:00:00	*Mia		Adoption	Spayed Female	30.4443212820182	-97.7326980338793	58.9642857142857
○	6	5 years	A696004	Dog	Cardigan Welsh Corgi Mix	Sable/White	2010-01-27	2015-01-28 10:39:00	2015-01-28T10:39:00	Lucy	Rabies Risk	Euthanasia	Spayed Female	30.6737365854231	-97.707971529467	261.063392857143
○	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3870648199411	-97.3684339731375	157.504067460317
○	4	7 months	A733653	Cat	Siamese Mix	Seal Point	2016-01-25	2016-08-27 18:11:00	2016-08-27T18:11:00	Kitty		Adoption	Intact Female	30.3188063374257	-97.7240376703891	30.8225198412698
○	11	1 year	A721199	Dog	Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49:00	Belle		Adoption	Spayed Female	30.7290272761146	-97.3753328216134	52.8203373015873
○	10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01	2013-12-08 14:58:00	2013-12-08T14:58:00	*Taylor		Adoption	Spayed Female	30.7583105481048	-97.618292198845	14.0890873015873
○	12	1 year	A664843	Dog	Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	30.4515549397366	-97.474104510925	62.2464285714286
○	13	1 year	A700408	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-04-13	2015-04-15 13:34:00	2015-04-15T13:34:00	Nyla		Return to Owner	Spayed Female	30.4101154527976	-97.562415670838	52.5093253968254
○	2	1 year	A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49:00		SCRP	Transfer	Spayed Female	30.6525984560228	-97.7419963476444	52.9215277777778
○	14	2 years	A742287	Dog	Boxer/Bullmastiff	Brown Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30:00	*Kawhi		Adoption	Neutered Male	30.4551148649096	-97.3087780473978	107.9[illegible]948

« ‹ 1 / »



Water Rescue Mountain or Wilderness Rescue [✓ Disaster or Individual Tracking] Reset

	rec_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
	filter data...															
●	2987	4 years	A694614	Dog	Rottweiler	Black/Brown	2011-01-01	2015-01-01 14:25:00	2015-01-01T14:25:00	Striker		Return to Owner	Intact Male	30.329873203611	-97.5492968638502	208.800099206349
○	3767	4 years	A712291	Dog	Bloodhound	Red	2011-09-20	2015-09-22 15:43:00	2015-09-22T15:43:00	Boomer		Return to Owner	Intact Male	30.2709983761287	-97.5923916912722	209.093551587302
○	6021	2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Return to Owner	Intact Male	30.466577208743	-97.5573520930426	120.924900793651
○	6557	6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Return to Owner	Intact Male	30.40668985085	-97.485680334264	26.6422619047619

Ericka's Client Server Authentication

Contact

Your name: Ericka Resendez

Note: This template has been adapted from the following sample templates: Make a README, Best README Template, and A Beginners Guide to Writing a Kickass README.