



CS 340 README

About the Project/Project Title

This software application is used to search for dogs from a database that can be used for search-and-rescue training. The user will be able to search for a specific characteristic such as age or breed. This application will work with an existing database with the options to add, delete, and update profiles.

Motivation

Grazioso Salvare, a rescue-animal training company, and a nonprofit agency that operates five animal shelters have teamed up to find potential dogs for search-and-rescue training. This will give a dog a new purpose and job that is for the greater good. These dogs will be trained for different types of rescues such as water rescue and locating humans after a disaster.

Getting Started

To set up a project locally, the user will need to import the database by using the Mongo import tool. Then the user will need to log in with a username and password. Depending on the users' permissions, they will be able to read, create, update, and delete documents. The create method takes the data that was submitted and inserts a document into the specified collection. The method will return true if the document was inserted successfully and false if it was not. The read method searches through the specified collection and will return the results. If the command was unsuccessful, it will return an empty list.

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](#).



Installation

To import your database, a user will use the mongoimport tool in the terminal. The default datatype is JSON and the user will need to specify that they are importing a CSV file. To create a new user, a user will need to use the admin database. The createUser command can be used to assign a username, password, and role. After importing the database, the user will need to log in using their username and password. Jupyter Notebooks was used to test our CRUD Python module. A IPYNB file was created in Jupyter Notebooks which called and tested the CRUD functionality. For the dashboard layout, we used the Dash framework. Dash is easy to use with pre-build components.

```
(base) /usr/local/datasets$ mongoimport --username="${MONGO_USER}" --password="${MONGO_PASS}" --port=${MONGO_PORT} --host=${MONGO_HOST} --db AAC --collection animals --aut
henticationDatabase admin --drop ./aac_shelter_outcomes.csv --type csv --headerline
2023-09-13T01:13:56.297+0000 connected to: mongodbi://nv-desktop-services.apporto.com:31567/
2023-09-13T01:13:56.299+0000 dropping: AAC.animals
2023-09-13T01:13:57.291+0000 10000 document(s) imported successfully. 0 document(s) failed to import.
(base) /usr/local/datasets$
```

```
admin> db.createUser (
... {
...
... user: "aacuser",
... pwd: "SNHU1234",
... roles: [{role: "readWrite", db: "aac"}]
... }
... )
{ ok: 1 }
admin>
```

```
(base) ~$ cd /usr/local/datasets
(base) /usr/local/datasets$ MONGO_USER=aacuser
(base) /usr/local/datasets$ MONGO_PASS=SNHU1234
(base) /usr/local/datasets$ printenv | grep -i m
MONGO
MONGO_USER=aacuser
MONGO_HOST=nv-desktop-services.apporto.com
MONGO_PASS=SNHU1234
MONGO_PORT=31567
```

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](#).



Steps

First, you will need to create a user that can access the database. Then the user can import the database and log into the database. Then you will need to create python module that will hold the functions to interact with the database such as read or create. Then you will need to create a IPYNB file in Jupyter Notebooks to create a dashboard.

Usage

Code Example

Python has great support for data manipulation and processing tasks which is why it was used for the CRUD module. First, I created the python file. The first function in the module is used to create a document. A user will set key/value pairs and if the creation is successful the function will return True. If there is an error, the function will return an error message and print False. The next function will query for documents from a specific MongoDB database and collection. A user will specify the key/value lookup pair and if the function was successful, it will return a list. If the function was unsuccessful, it will return an empty list. The update function will request a key/value lookup pair that the user query for documents from the specified database. Then the function will request what data the user wants to update. If it was successfully updated the function will return the number of objects that were modified. If not, it will return an error message. The last function is the delete method. The user will input a key/value lookup pair and the function will search and remove the documents from the specified database. The function will return the number of objects removed if it was successfully and it will return an error message if it failed.

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](#).



```
def create(self, data):
    try:
        result = self.collection.insert_one(data)
        return True if result.inserted_id else False
    except Exception as e:
        print("Error:", e)
        return False

def read(self, query):
    try:
        cursor = self.collection.find(query)
        return list(cursor)
    except Exception as e:
        print("Error:", e)
        return []

def update(self, query, update_data):
    try:
        result = self.collection.update_many(query, update_data)
        return result.modified_count
    except Exception as e:
        print("Error:", e)
        return 0

def delete(self, query):
    try:
        result = self.collection.delete_many(query)
        return result.deleted_count
    except Exception as e:
        print("Error:", e)
        return 0
```

Tests

To test the methods, I used the aacuser account. Then I created a class named animal_shelter and input the connection variables. Then I created a document using the create function. I set the type to dog and the name to Bob. The document was successfully created, and the results returned True. I tested the read function by using the type dog as my key/value lookup pair. A list was

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](#).



returned that included the correct key/value pair. After that, I tested the update function by searching for the type dog and updating the type to cat. The test was successful, and the method return 1 as the number of documents updated. The last test was the delete function. I used the key/value look up pair, type cat. Then the function searched the database using that key/value pair and deleted the documents. Then it returned how many objects were deleted.

```
animal_shelter = ANIMAL_SHELTER(username, password, host, port, db_name, collection_name)
```

```
#Test Create
create_data = {"name": "Bob", "type": "Dog"}
create_result = animal_shelter.create(create_data)
print("Create Result:", create_result)
```

Create Result: True

```
#Test Read
read_query = {"type": "Dog"}
read_result = animal_shelter.read(read_query)
print("Read Result:", read_result)
```

Read Result: [{'_id': ObjectId('650f3dd77a5ff1c180314be5'), 'name': 'Bob', 'type': 'Dog'}]

```
#Test Update
update_query = {"type": "Dog"}
update_data = {"$set": {"type": "Cat"}}
update_result = animal_shelter.update(update_query, update_data)
print("Update Result:", update_result)
```

Update Result: 1

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](#).



```
#Test Delete
delete_query = {"type": "Cat"}
delete_result = animal_shelter.delete(delete_query)
print("Detele Result:", delete_result)
```

Detele Result: 1

Dashboard

The dashboard has different features such as a list of the profiles of the dogs within the database.

When a user logs in they will see an unfiltered list of dogs. At the top of the page, a user can filter the list by selecting a button or reset the list. Towards the bottom of the page, there is a map of the location of the dog. The user can select the button on the left side of the list to find the location of the dog. The dashboard also includes a pie graph that displays the percentage of available dogs based on the list. If a filter is selected, then the pie graph will update based on the filtered list.

After the user logs in, the IPYNB file will connect to the database using the python module that was previously created. Then I designed how the dashboard will look. I created the buttons for the filters, the design of the database list, and the placement for the graphs. Then we created a function that will update the list based on the button that the user pushes. The next function will update the pie chart. The function update_styles will add a background color to a column that is selected. The last function will update the map when a user selects a certain profile.

```
className='row',
style={'display':'flex'},
children=[
    html.Button(id='button-one', n_clicks=0, children='Water Rescue'),
    html.Button(id='button-two', n_clicks=0, children='Mountain or Wilderness Rescue'),
    html.Button(id='button-three', n_clicks=0, children='Disaster Rescue or Individual Tracking'),
    html.Button(id='button-four', n_clicks=0, children='Reset'),
],
```

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](#).



```
dash_table.DataTable(id='datatable-id',
    columns=[{"name": i, "id": i, "deletable": False, "selectable": True} for i in df.columns],
    data=df.to_dict('records'),
    style_table={'height': '400 px', 'overflowY': 'auto', 'overflowX': 'auto'},
    style_cell={'textAlign': 'left'},
    style_header={
        'backgroundColor': 'rgb(30, 30, 30)',
        'fontWeight': 'bold',
        'color': 'white'
    },
    style_data={
        'backgroundColor': 'rgb(50, 50, 50)',
        'color': 'white'
    },
    style_data_conditional=[
        {
            'if': {'row_index': 'odd'},
            'backgroundColor': 'rgb(86, 86, 86)',
        }
    ],

    #tooltips
    tooltip = {i: {
        'value': i,
        'use_with': 'both'
    } for i in df.columns},
    tooltip_delay=0,
    tooltip_duration=None,
    page_size=15,
    row_selectable="single",
    selected_rows=[0],
    selected_columns=[],
    column_selectable=False,
    row_deletable=False,
    editable=False,
    page_action="native",
    page_current=0,
```

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](#).



GRAZIOSO SALVARE

CS-340 Dashboard - Denise Kendall

CS-340 Dashboard - Denise Kendall

Water Rescue Mountain or Wilderness Rescue Disaster Rescue 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
	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	*Hia		Adoption	Spayed Female	30.4443212820182	-97.732680033879
	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	-97.368433973137
	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
	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	-97.724037670389
	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	30.7296272761146	-97.375332821613
	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
	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
	2	1 year		A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49:00		SNRP	Transfer	Spayed Female	30.6525984560228	-97.741996347644
	14	2 years		A742287	Dog	Boxer/Bulldog	Brown Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30:00	*Kawhi		Adoption	Neutered Male	30.4551148649096	-97.308778047397
	17	6 months		A668960	Dog	Pit Bull Mix	Blue/White	2013-06-12	2013-12-27 16:56:00	2013-12-27T16:56:00	*Gigi		Adoption	Spayed Female	30.5943410758588	-97.248993356951
	16	5 years		A723742	Dog	Miniature Schnauzer Mix	Black/White	2011-04-05	2016-04-10 17:27:00	2016-04-10T17:27:00	Gretchen		Adoption	Spayed Female	30.4792884863566	-97.408833158799
	15	3 years		A712638	Dog	Pit Bull Mix	Red/White	2012-09-26	2016-07-18 17:52:00	2016-07-18T17:52:00	Marcus	Partner	Transfer	Neutered Male	30.5798299207017	-97.558848793653
	3	2 years		A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7595748121648	-97.552375380713
	7	2 years		A673830	Dog	Pit Bull Mix	Black/White	2012-03-03	2014-03-19 15:15:00	2014-03-19T15:15:00	*Seth	Aggressive	Euthanasia	Neutered Male	30.2954256583441	-97.313664211043
	18	2 months		A693288	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-09-28	2014-12-09 18:36:00	2014-12-09T18:36:00			Adoption	Spayed Female	30.4527678292931	-97.462058716767

<<

<

1

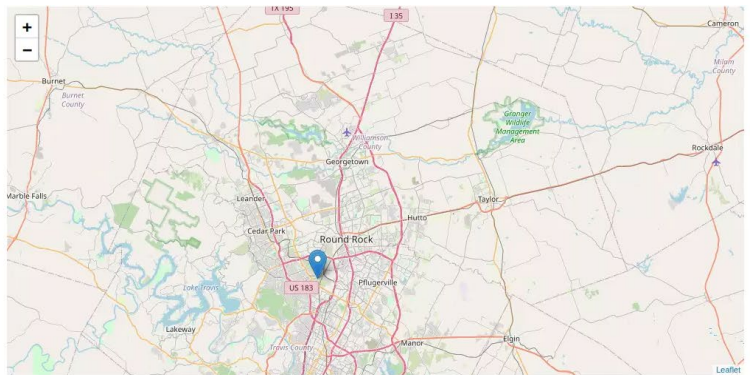
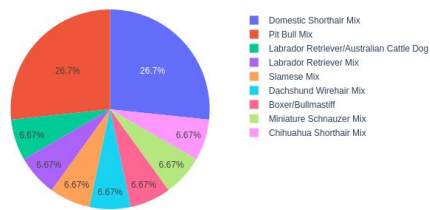
>

>>

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](#).


```
html.Div(className='row',
        style={'display' : 'flex'},
        children=[
            html.Div(
                id='graph-id',
                className='col s12 m6',
            ),
            html.Div(
                id='map-id',
                className='col s12 m6',
            )
        ]
    )
```

Preferred Animals



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](#).

```
@app.callback(Output('datatable-id', 'data'),
               [Input('radio-item', 'value')])
def update_dashboard(radio_item):
    ## FIX ME Add code to filter interactive data table with MongoDB queries

    if radio_item == 'Water Rescue':
        records = db.read({
            '$and': [
                {'$or': [
                    {'breed': 'Labrador Retriever Mix'},
                    {'breed': 'Chesapeake Bay Retriever'},
                    {'breed': 'Newfoundland'}
                ]},
                {'sex_upon_outcome': 'Intact Female'},
                { "$and": [
                    {'age_upon_outcome_in_weeks': {"$gt": 26}},
                    {'age_upon_outcome_in_weeks': {"$lt": 156}}
                ]}
            ]}
        )

    elif radio_item == 'Mountain or Wilderness Rescue':
        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'},
                { "$and": [
                    {'age_upon_outcome_in_weeks': {"$gt": 26}},
                    {'age_upon_outcome_in_weeks': {"$lt": 156}}
                ]}
            ]}
        )
```

```
elif radio_item == 'Disaster Rescue or Individual Tracking':
    records = db.read({
        '$and': [
            {'$or': [
                {'breed': 'Doberman Pinscher'},
                {'breed': 'German Sheperd'},
                {'breed': 'Golden Retriever'},
                {'breed': 'Bloodhound'},
                {'breed': 'Rottweiler'}
            ]},
            {'sex_upon_outcome': 'Intact Male'},
            {"$and": [
                {'age_upon_outcome_in_weeks': {"$gt": 20}},
                {'age_upon_outcome_in_weeks': {"$lt": 300}}
            ]}
        ]}
    )

else:
    records = db.read({})

df = pd.DataFrame.from_records(records)
df.drop(columns=['_id'], inplace=True)
return df.to_dict('records')
```

○ Water Rescue ○ Mountain or Wilderness Rescue ○ Disaster Rescue 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
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	-97.732698033879
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	30.3870640199411	-97.368433973137
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	30.758105481048	-97.618292198845
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	30.3188863374257	-97.724037670389
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	location long	
13	1 year	A708408	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
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
2	1 year	A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49:00		SCRIP	Transfer	Spayed Female	30.6525984560228	-97.741996347644
14	2 years	A742287	Dog	Boxer/Bullmastiff	Brown Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30:00	*Kamhi		Adoption	Neutered Male	30.4551140649896	-97.308778047397
17	6 months	A668960	Dog	Pit Bull Mix	Blue/White	2013-06-12	2013-12-27 16:56:00	2013-12-27T16:56:00	*Gigi		Adoption	Spayed Female	30.5943410758588	-97.248993356951
16	5 years	A723742	Dog	Miniature Schnauzer Mix	Black/White	2011-04-05	2016-04-10 17:27:00	2016-04-10T17:27:00	Gretchen		Adoption	Spayed Female	30.4792884863566	-97.408853158799
15	3 years	A712638	Dog	Pit Bull Mix	Red/White	2012-09-26	2016-07-18 17:52:00	2016-07-18T17:52:00	Marcus	Partner	Transfer	Neutered Male	30.5798299207017	-97.558848793653
3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-20 18:43:00	2015-12-20T18:43:00	*Frank		Adoption	Neutered Male	30.7595740121648	-97.552375380713
7	2 years	A673830	Dog	Pit Bull Mix	Black/White	2012-03-03	2014-03-19 15:15:00	2014-03-19T15:15:00	*Seth	Aggressive	Euthanasia	Neutered Male	30.2954256583441	-97.313664211843
18	2 months	A693288	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-09-28	2014-12-09 18:36:00	2014-12-09T18:36:00			Adoption	Spayed Female	30.4527678292931	-97.462050716767

○ Water Rescue ○ Mountain or Wilderness Rescue ○ Disaster Rescue 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
36	6 months	A706953	Dog	Labrador Retriever Mix	Yellow	2014-12-06	2015-07-06 11:33:00	2015-07-06T11:33:00		Medical	Euthanasia	Intact Female	30.5480802368633	-97.2969969058957	30.3544642857143
732	2 years	A749782	Dog	Labrador Retriever Mix	Tan/White	2015-05-19	2017-07-25 14:59:00	2017-07-25T14:59:00	*Catalina		Return to Owner	Intact Female	30.6138310636757	-97.5752164857665	11.089186587937
1121	1 year	A757158	Dog	Labrador Retriever Mix	White/Black	2016-08-30	2017-08-31 14:12:00	2017-08-31T14:12:00	Pirata		Return to Owner	Intact Female	30.5572161697962	-97.5363224263878	52.3702380952281
1628	9 months	A740471	Dog	Labrador Retriever Mix	Tan/White	2016-03-17	2016-12-23 17:13:00	2016-12-23T17:13:00	Mika		Adoption	Intact Female	30.7569243032341	-97.7392549176654	48.2453373015873
1757	7 months	A742767	Dog	Labrador Retriever Mix	Black	2016-06-27	2017-02-14 15:20:00	2017-02-14T15:20:00	Marley		Return to Owner	Intact Female	30.4869754937324	-97.4280017197358	33.234126984127
1988	1 year	A762781	Dog	Labrador Retriever Mix	Black/White	2016-11-27	2017-12-03 13:09:00	2017-12-03T13:09:00		Partner	Transfer	Intact Female	30.2840111162063	-97.4680542219677	53.0782738095238
2041	2 years	A702745	Dog	Labrador Retriever Mix	Black	2013-05-22	2015-05-22 11:45:00	2015-05-22T11:45:00	Abigail		Return to Owner	Intact Female	30.7157941956301	-97.4523664870572	184.355654761985
2225	2 years	A757341	Dog	Labrador Retriever Mix	Black/White	2015-09-01	2017-10-03 12:27:00	2017-10-03T12:27:00	19	Partner	Transfer	Intact Female	30.3814182796497	-97.73732171391863	109.074107142857
3319	9 months	A687748	Dog	Labrador Retriever Mix	Yellow	2013-12-09	2014-09-09 17:01:00	2014-09-09T17:01:00		Suffering	Euthanasia	Intact Female	30.729533449489	-97.3912941009824	39.2441468253968
4222	1 year	A735551	Dog	Labrador Retriever Mix	Black	2015-09-25	2016-09-27 14:10:00	2016-09-27T14:10:00	Daisy		Return to Owner	Intact Female	30.5150703656925	-97.6892580677489	52.655753968254
4391	2 years	A739950	Dog	Labrador Retriever Mix	Black/White	2014-12-09	2016-12-10 13:47:00	2016-12-10T13:47:00	*Leah	Partner	Transfer	Intact Female	30.2749778566093	-97.7052228203318	184.653472222222
4846	6 months	A739549	Dog	Labrador Retriever Mix	Tan	2016-06-04	2016-12-04 15:59:00	2016-12-04T15:59:00	Penny		Return to Owner	Intact Female	30.7472967860317	-97.6152946055269	26.237996031746
4956	1 year	A736948	Dog	Labrador Retriever Mix	Black/White	2015-10-19	2016-10-23 00:00:00	2016-10-23T00:00:00		Partner	Transfer	Intact Female	30.601325440225	-97.2533625326776	52.8571428571429
6237	1 year	A747734	Dog	Labrador Retriever Mix	Chocolate/White	2016-04-24	2017-04-24 16:03:00	2017-04-24T16:03:00	Ella		Return to Owner	Intact Female	30.3400609542736	-97.6350634308579	52.2383928571429
7583	2 years	A757347	Dog	Labrador Retriever Mix	Black	2015-09-01	2017-10-09 19:34:00	2017-10-09T19:34:00	22	Partner	Transfer	Intact Female	30.6322306158419	-97.2747762759927	100.973611111111

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](#).



○ Water Rescue ○ Mountain or Wilderness Rescue ○ Disaster Rescue or Individual Tracking ○ Reset

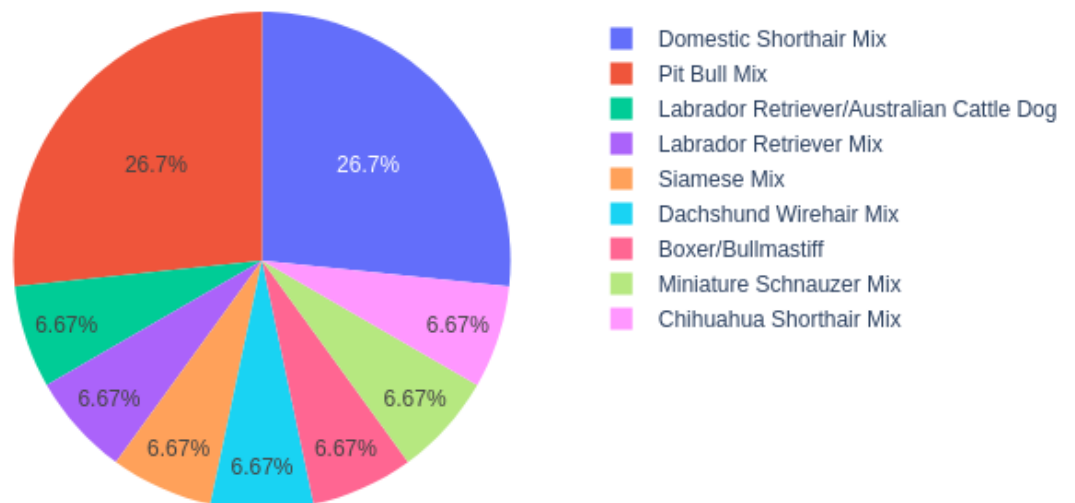
rec_num	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
3138	2 years	A721834	Dog	Siberian Husky	Brown/White	2014-03-05	2016-03-23 16:23:00	2016-03-23T16:23:00		Suffering	Euthanasia	Intact Male	30.5680998448899	-97.320550480325	107.09751984127
5315	2 years	A708726	Dog	Alaskan Malamute	Sable/White	2013-07-30	2015-08-02 17:24:00	2015-08-02T17:24:00	Papa		Return to Owner	Intact Male	30.4389339291938	-97.488825836737	104.817857142857
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	-97.5573520939426	120.924980793651
6191	2 years	A704101	Dog	Siberian Husky	Black/White	2013-06-01	2015-06-02 16:41:00	2015-06-02T16:41:00	Lobo		Return to Owner	Intact Male	30.4263764229275	-97.4369581796886	104.527876984127
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	30.48668985885	-97.48568034264	26.6422619847619

○ Water Rescue ○ Mountain or Wilderness Rescue ○ Disaster Rescue 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
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	-97.5492968838582	208.800899206349
5767	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	209.093551587382
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	-97.5573520939426	120.924980793651

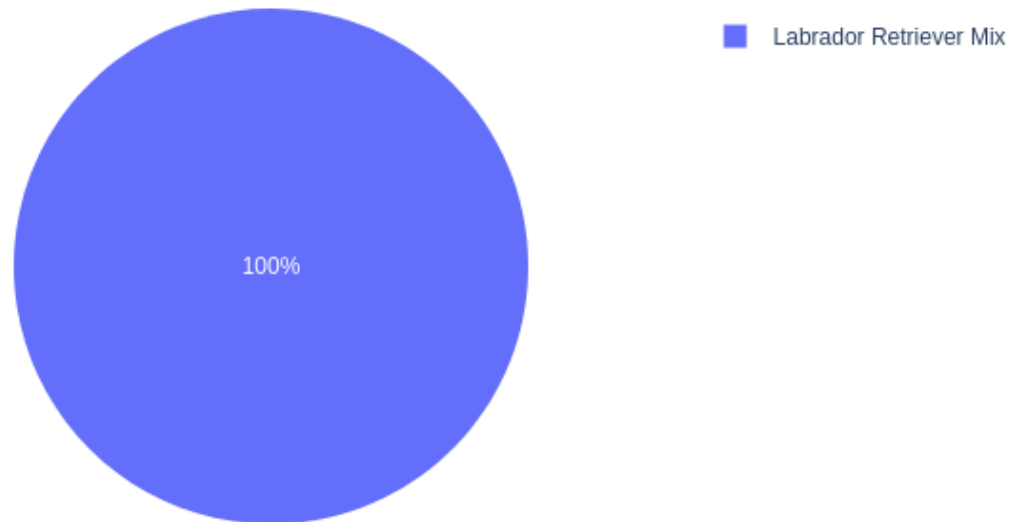
```
@app.callback(  
    Output('graph-id', "children"),  
    [Input('datatable-id', "derived_viewport_data")])  
def update_graphs(viewData, **kwargs):  
    ###FIX ME ###  
    # add code for chart of your choice (e.g. pie chart)  
  
    df = pd.DataFrame.from_dict(viewData)  
  
    return [  
        dcc.Graph(  
            figure = px.pie(df, names=df['breed'], title='Preferred Animals')  
        )  
    ]
```

Preferred Animals

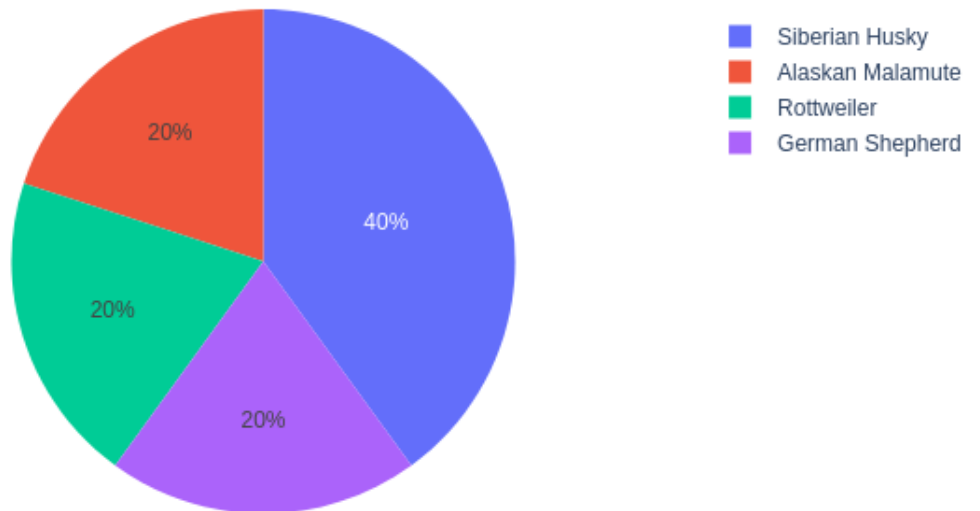


Note: This template has been adapted from the following sample templates: [Make a README](#), [Best README Template](#), and [A Beginners Guide to Writing a Kickass README](#).

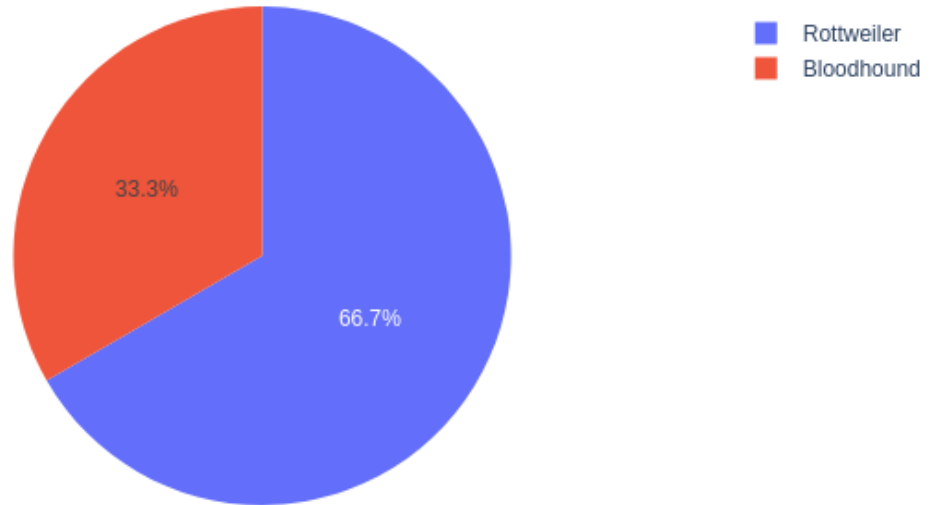
Preferred Animals



Preferred Animals



Preferred Animals



```
@app.callback(
    Output('datatable-id', 'style_data_conditional'),
    [Input('datatable-id', 'selected_columns')]
)
def update_styles(selected_columns):
    return [{
        'if': { 'column_id': i },
        'background_color': '#D2F3FF'
    } for i in selected_columns]
```

Domestic Shorthair Mix
Pit Bull Mix
Domestic Shorthair Mix

```
@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

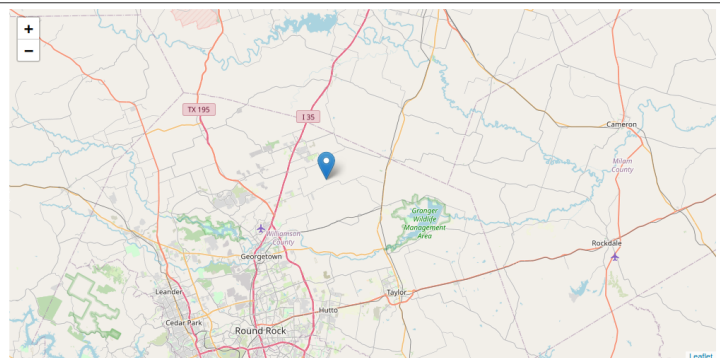
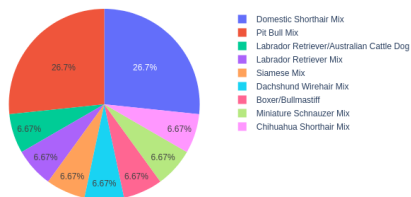
    dff = pd.DataFrame.from_dict(viewData)
    # Because we only allow single row selection, the list can be converted to a row index here
    if index is None:
        row = 0
    else:
        row = index[0]

    # 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])
                ])
            ])
        ])
    ]
```

3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7595748121648	-97.55
7	2 years	A673830	Dog	Pit Bull Mix	Black/White	2012-03-03	2014-03-19 15:15:00	2014-03-19T15:15:00	Seth	Aggressive	Euthanasia	Neutered Male	30.2954256583441	-97.31
18	2 months	A693288	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-09-28	2014-12-09 18:36:00	2014-12-09T18:36:00			Adoption	Spayed Female	30.4527678292931	-97.46

1 / 66

Preferred Animals



Contact

Denise Kendall

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](#).