Module 7-2 Project Two Submission

Arturo Santiago-Rivera

Prof. Tad Kellogg M.S.

CS-340-T3237 Client/Server Development 21EW3

Southern New Hampshire University

February 21, 2021

Revised - March 31, 2022

Salvare Search For Rescue Web App

build v2.0.0 python v3.9 mongoDB v4.4 python driver pymongo python framework dash license MIT

Table of Content

About the Project

The web application works with an existing animal shelter database based on Python, PyMongo driver, Dash framework, and MongoDB. It helps identify and categorize available dogs to train for different types of rescue, such as water rescue, mountain or wilderness rescue, locating humans after a disaster, or finding a specific human by tracking their scent. This application helps interact with and visualize individual dog profiles to train from a MongoDB database. Through a user-friendly, intuitive client-facing web application dashboard, the user reduces errors and training time. The software design pattern used for this multi-tier application is the Model View Controller (MVC).

Additionally, the RESTful protocol extends the HTTP protocol to give an application programming interface (API). The primary user interface is a dashboard created with different components in Python code and the Dash framework. Dashboard web applications lend well to the MVC design pattern, so the model is contained and accessed in MongoDB, and the views are Dash framework widgets. The controller uses a CRUD Python module for queries as part of the interaction between components.

Motivation

Grazioso Salvare, an innovative international rescue-animal training company, identifies dogs that are good candidates for search-and-rescue training. The company has noted specific types and breeds of dogs to train. For instance, search-and-rescue training is generally more helpful for dogs no more than two years old. Additionally, certain breeds of dogs are proficient at different types of rescue, such as water rescue, mountain or wilderness rescue, locating humans after a disaster, or finding a specific human by tracking their scent. These dogs can find and help rescue humans or other animals when trained, often in life-threatening conditions. To help identify dogs for training, the company agrees with non-profit agencies that operate animal shelters in some specific regions. These non-profit agencies provide the company with data from their shelters.

Getting Started

Before proceeding, you should have a good understanding of the Python programming language and MongoDB. Understanding the MongoDB database, Python driver PyMongo, and the Python Dash framework. To get a local copy up and running of the web app, make sure you have properly installed and running Python (along with PIP) and MongoDB:

- Python version 3.9 or up (https://www.python.org)
- MongoDB version 4.4 or up (https://www.mongodb.com)

Starting with MongoDB 4.4, the MongoDB Database Tools are a suite of command-line utilities for working with MongoDB. It is necessary to install the Database Tools on the Windows platform.

- MongoDB Database Tools (https://docs.mongodb.com/database-tools/installation/installation-windows/)

Installation

Salvare Search for Rescue App uses the Python driver PyMongo the officially supported Python driver by MongoDB, and the Python framework Dash. PyMongo (https://pymongo.readthedocs.io/en/stable/) is a python distribution that provides tools to work with MongoDB. To install PyMongo execute the following command in your Terminal CLI:

```
pip install pymongo
```

Dash (https://dash.plotly.com/) is a productive Python framework for building analytic web applications. Dash framework provides the view and controller structure for the web application. To install Dash libraries, execute the following command in your Terminal CLI:

```
pip install jupyter_plotly_dash dash_leaf panda
```

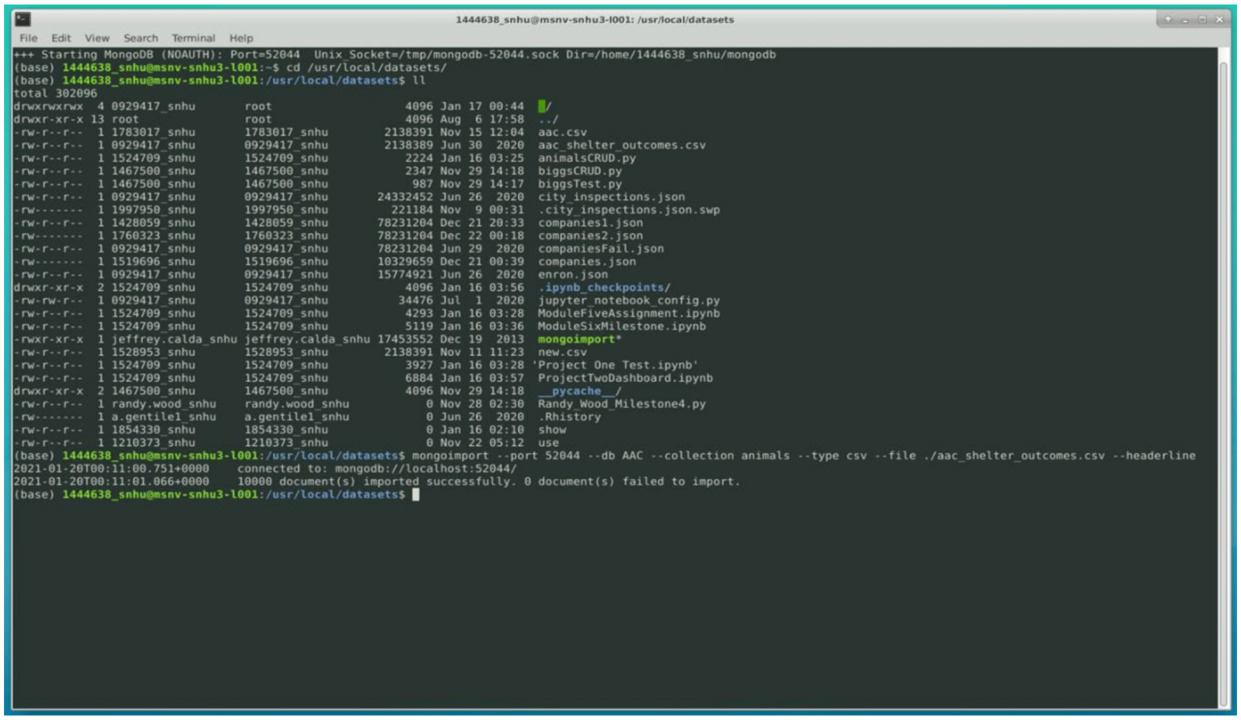
Having installed the python driver and framework, you need to upload the Austin Animal Center Outcomes data set into MongoDB by inserting the CSV file using the mongoimport tool in your Terminal CLI. Replace the "#####" with your MongoDB port number.

Linux:

```
mongoimport -–port ##### --db AAC –-collection animals –-type csv –-file ./aac_shelter_outcomes.csv –headerline
```

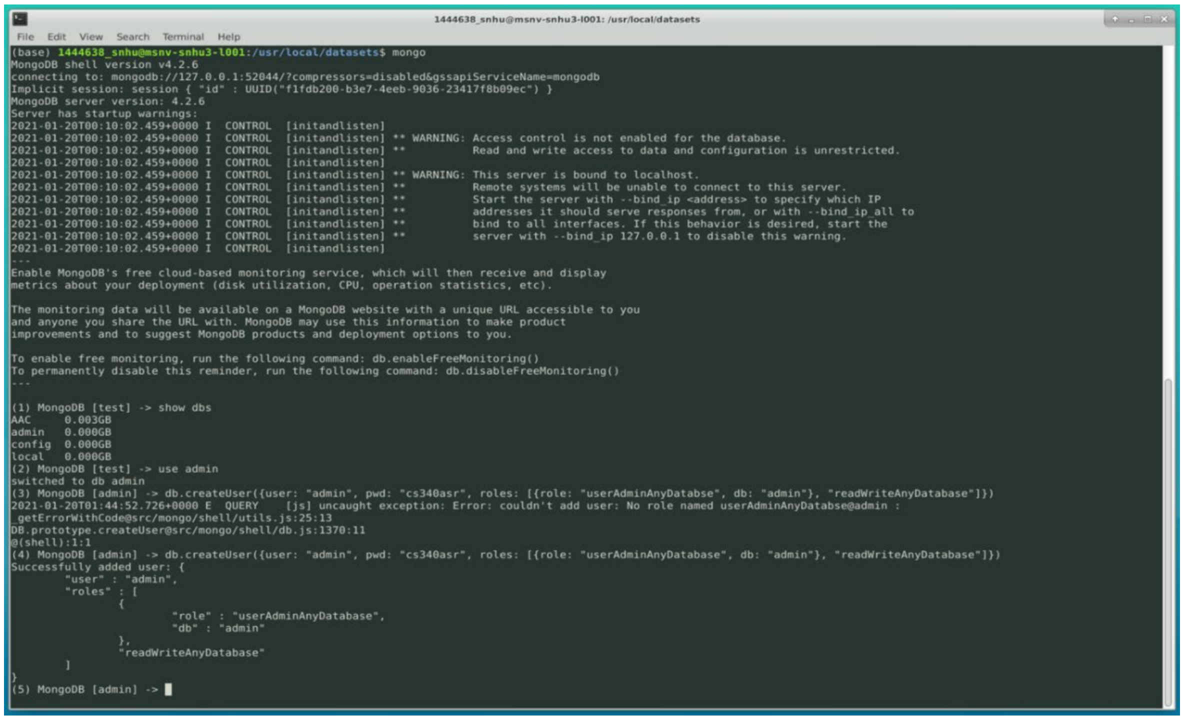
Windows:

```
mongoimport /port:##### /db:AAC /collection:animals /type:csv /file:.\aac_shelter_outcomes.csv /headerline
```



```
+++ Starting MongoDB (NOAUTH): Port=52044  Unix_Socket=/tmp/mongodb-52044.sock Dir=/home/1444638_snhu/mongodb
(base) 1444638_snhu@msnv-snhu3-l001:~$ cd /usr/local/datasets/
(base) 1444638_snhu@msnv-snhu3-l001:/usr/local/datasets$ ll
total 302096
drwxrwxrwx  4 0929417_snhu       root                   4096 Jan 17 00:44  ./
drwxr-xr-x 13 root               root                   4096 Aug  6 17:58  ../
-rw-r--r--  1 1783017_snhu       1783017_snhu        2138391 Nov 15 12:04  aac.csv
-rw-r--r--  1 0929417_snhu       0929417_snhu        2138389 Jun 30  2020  aac_shelter_outcomes.csv
-rw-r--r--  1 1524709_snhu       1524709_snhu           2224 Jan 16 03:25  animalsCRUD.py
-rw-r--r--  1 1467500_snhu       1467500_snhu           2347 Nov 29 14:18  biggsCRUD.py
-rw-r--r--  1 1467500_snhu       1467500_snhu            987 Nov 29 14:17  biggsTest.py
-rw-r--r--  1 0929417_snhu       0929417_snhu       24332452 Jun 26  2020  city_inspections.json
-rw-------  1 1997950_snhu       1997950_snhu         221184 Nov  9 00:31  .city_inspections.json.swp
-rw-r--r--  1 1428059_snhu       1428059_snhu       78231204 Dec 21 20:33  companies1.json
-rw-------  1 1760323_snhu       1760323_snhu       78231204 Dec 22 00:18  companies2.json
-rw-r--r--  1 0929417_snhu       0929417_snhu       78231204 Jun 29  2020  companiesFail.json
-rw-------  1 1519696_snhu       1519696_snhu       10329659 Dec 21 00:39  companies.json
-rw-r--r--  1 0929417_snhu       0929417_snhu       15774921 Jun 26  2020  enron.json
drwxr-xr-x  2 1524709_snhu       1524709_snhu           4096 Jan 16 03:56  .ipynb_checkpoints/
-rw-rw-r--  1 0929417_snhu       0929417_snhu          34476 Jul  1  2020  jupyter_notebook_config.py
-rw-r--r--  1 1524709_snhu       1524709_snhu           4293 Jan 16 03:28  ModuleFiveAssignment.ipynb
-rw-r--r--  1 1524709_snhu       1524709_snhu           5119 Jan 16 03:36  ModuleSixMilestone.ipynb
-rwxr-xr-x  1 jeffrey.calda_snhu jeffrey.calda_snhu 17453552 Dec 19  2013  mongoimport*
-rw-r--r--  1 1528953_snhu       1528953_snhu        2138391 Nov 11 11:23  new.csv
-rw-r--r--  1 1524709_snhu       1524709_snhu           3927 Jan 16 03:28 'Project One Test.ipynb'
-rw-r--r--  1 1524709_snhu       1524709_snhu           6884 Jan 16 03:57  ProjectTwoDashboard.ipynb
drwxr-xr-x  2 1467500_snhu       1467500_snhu           4096 Nov 29 14:18  __pycache__/
-rw-r--r--  1 randy.wood_snhu    randy.wood_snhu           0 Nov 28 02:30  Randy_Wood_Milestone4.py
-rw-------  1 a.gentile1_snhu    a.gentile1_snhu           0 Jun 26  2020  .Rhistory
-rw-r--r--  1 1854330_snhu       1854330_snhu              0 Jan 16 02:10  show
-rw-r--r--  1 1210373_snhu       1210373_snhu              0 Nov 22 05:12  use
(base) 1444638_snhu@msnv-snhu3-l001:/usr/local/datasets$ mongoimport --port 52044 --db AAC --collection animals --type csv --file ./aac_shelter_outcomes.csv --headerline
2021-01-20T00:11:00.751+0000    connected to: mongodb://localhost:52044/
2021-01-20T00:11:01.066+0000    10000 document(s) imported successfully. 0 document(s) failed to import.
(base) 1444638_snhu@msnv-snhu3-l001:/usr/local/datasets$ 
```

Create an administrator account in the mongo shell by following steps #2 to #5 of the MongoDB Manual Enable Access Control tutorial: SCRAM. Then exit the mongo shell, stop the mongo driver and start the driver again.

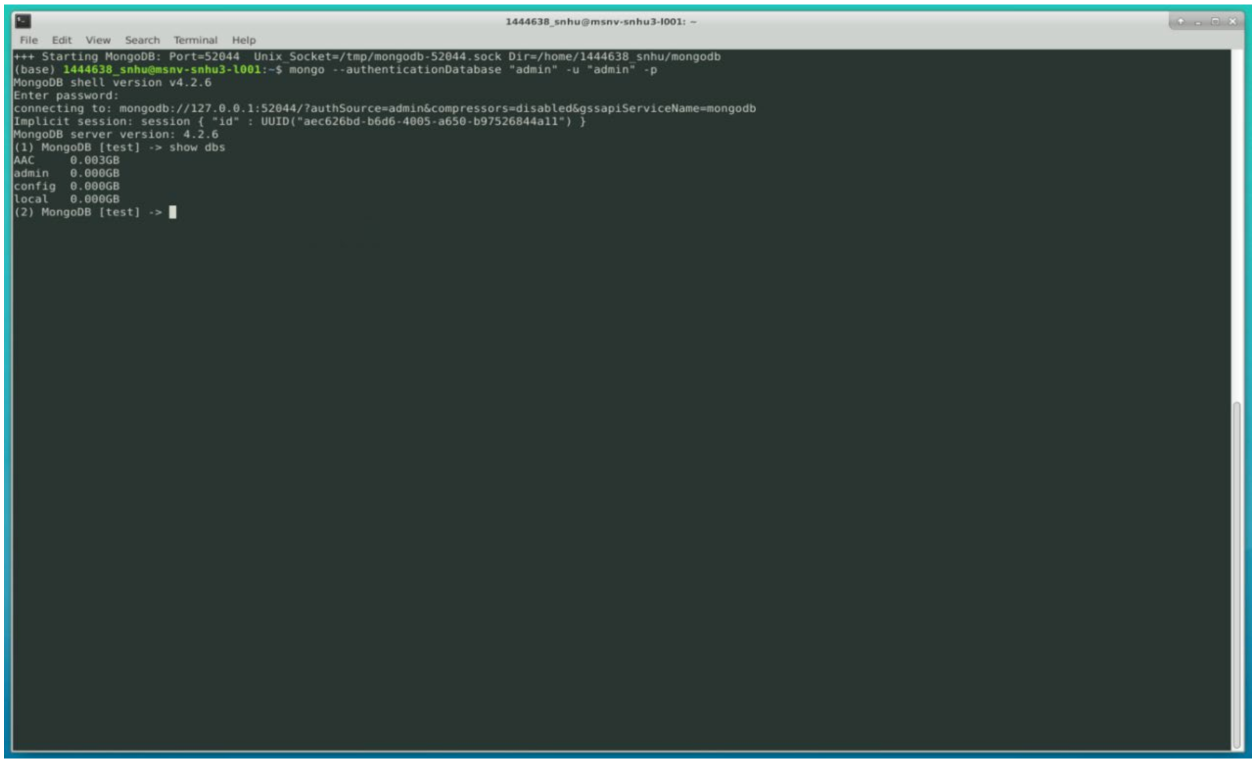


```
(base) 1444638_snhu@msnv-snhu3-l001:/usr/local/datasets$ mongo
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:52044/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("f1fdb200-b3e7-4eeb-9036-23417f8b09ec") }
MongoDB server version: 4.2.6
Server has startup warnings:
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten]
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten]
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2021-01-20T00:10:02.459+0000 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

(1) MongoDB [test] -> show dbs
AAC     0.003GB
admin   0.000GB
config  0.000GB
local   0.000GB
(2) MongoDB [test] -> use admin
switched to db admin
(3) MongoDB [admin] -> db.createUser({user: "admin", pwd: "cs340asr", roles: [{role: "userAdminAnyDatabse", db: "admin"}, "readWriteAnyDatabase"]})
2021-01-20T01:44:52.726+0000 E  QUERY    [js] uncaught exception: Error: couldn't add user: No role named userAdminAnyDatabse@admin :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype.createUser@src/mongo/shell/db.js:1370:11
@(shell):1:1
(4) MongoDB [admin] -> db.createUser({user: "admin", pwd: "cs340asr", roles: [{role: "userAdminAnyDatabase", db: "admin"}, "readWriteAnyDatabase"]})
Successfully added user: {
        "user" : "admin",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                },
                "readWriteAnyDatabase"
        ]
}
(5) MongoDB [admin] -> 
```

You can verify that you have enabled user authentication by accessing MongoDB with your new username/password. Type the following command into the Linux shell to start mongo:

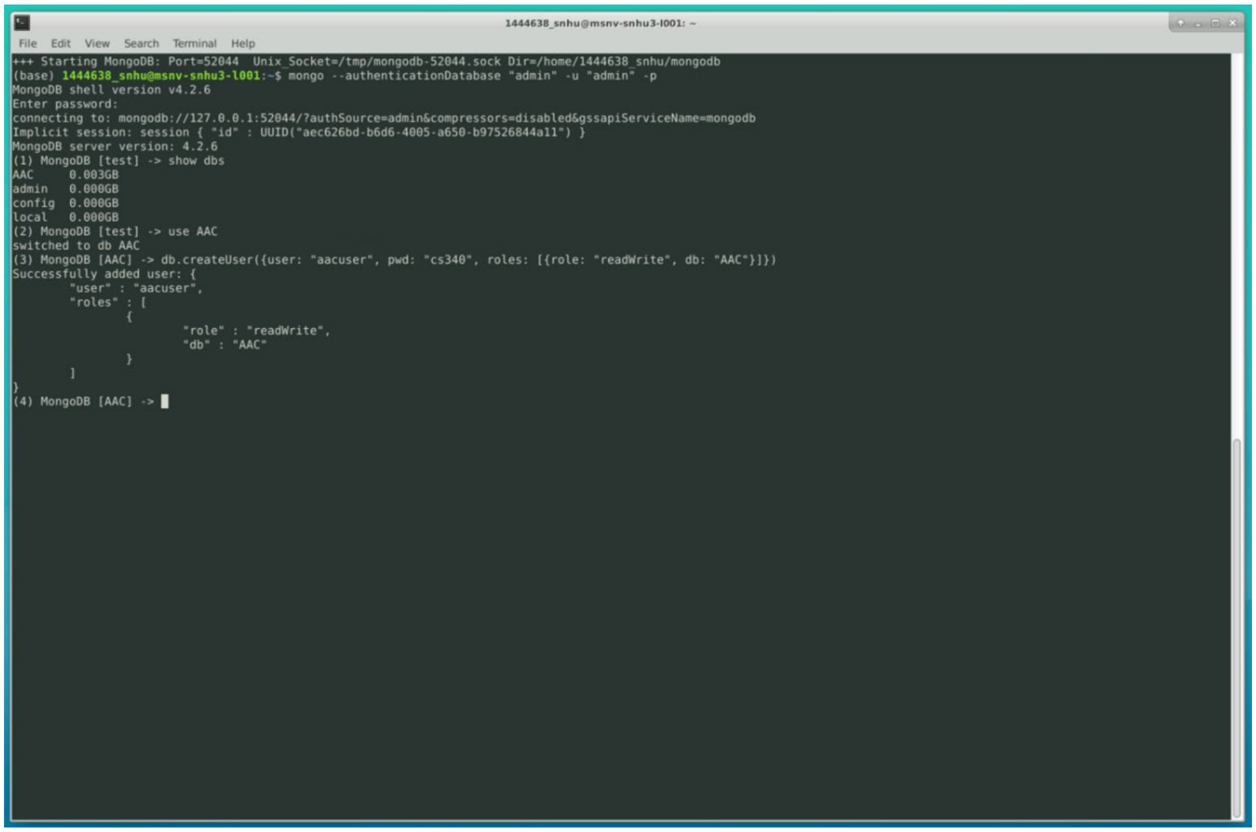
```
$ mongo --authenticationDatabase "admin" -u "username" -p
```

The above command prompt you to enter the username password. Then use the mongo shell command to show databases to verify that you have set up authentication correctly. If you are not logged in with your admin account, no databases will be viewable.


1444638_snhu@msnv-snhu3-l001: ~
File Edit View Search Terminal Help

```
+++ Starting MongoDB: Port=52044  Unix_Socket=/tmp/mongodb-52044.sock Dir=/home/1444638_snhu/mongodb
(base) 1444638_snhu@msnv-snhu3-l001:~$ mongo --authenticationDatabase "admin" -u "admin" -p
MongoDB shell version v4.2.6
Enter password:
connecting to: mongodb://127.0.0.1:52044/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("aec626bd-b6d6-4005-a650-b97526844a11") }
MongoDB server version: 4.2.6
(1) MongoDB [test] -> show dbs
AAC     0.003GB
admin   0.000GB
config  0.000GB
local   0.000GB
(2) MongoDB [test] ->
```

Create a new user account called "aacuser" with a password and role “readWrite” for the database AAC in the mongo shell. Refer to step #2 of the MongoDB Manual Enable Access Control tutorial: Create a User to help you with this task. You need to modify the commands so that the account name is "aacuser". Then exit the mongo shell, stop the mongo driver, and start the driver again.

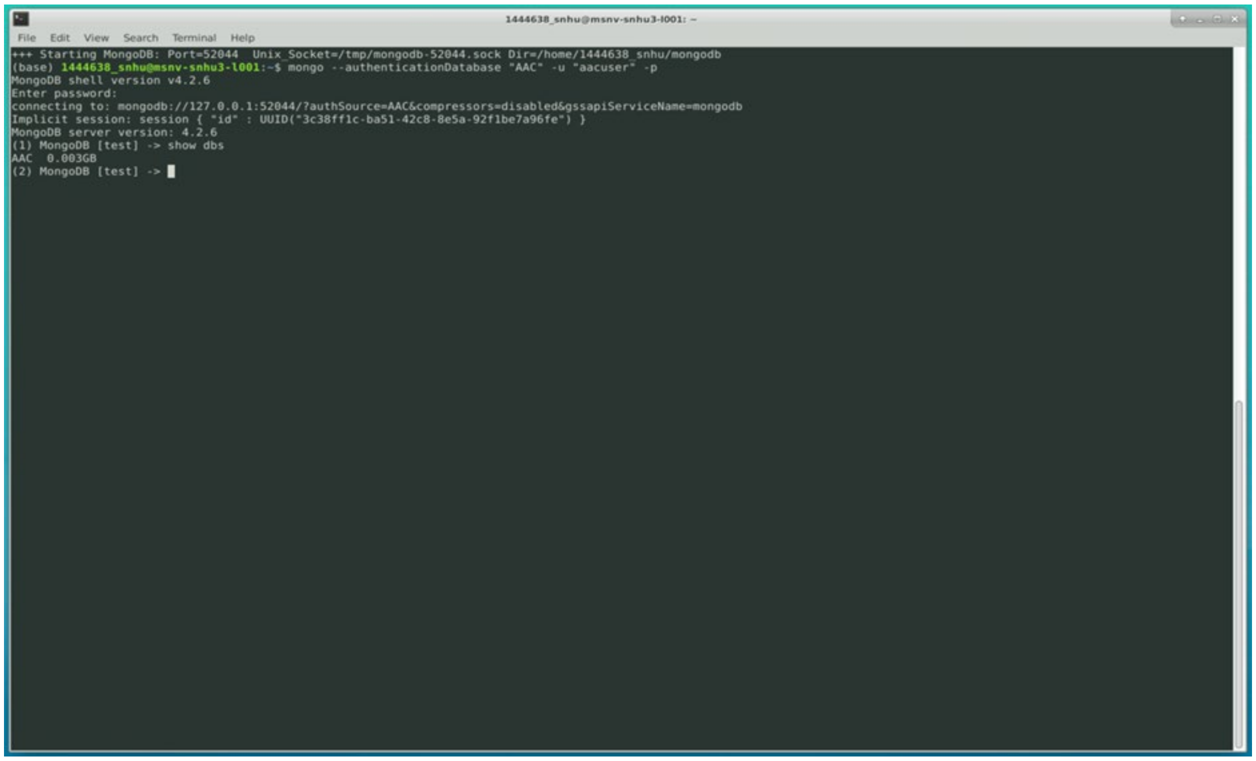

1444638_snhu@msnv-snhu3-l001: ~
File Edit View Search Terminal Help

```
+++ Starting MongoDB: Port=52044  Unix_Socket=/tmp/mongodb-52044.sock Dir=/home/1444638_snhu/mongodb
(base) 1444638_snhu@msnv-snhu3-l001:~$ mongo --authenticationDatabase "admin" -u "admin" -p
MongoDB shell version v4.2.6
Enter password:
connecting to: mongodb://127.0.0.1:52044/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("aec626bd-b6d6-4005-a650-b97526844a11") }
MongoDB server version: 4.2.6
(1) MongoDB [test] -> show dbs
AAC     0.003GB
admin   0.000GB
config  0.000GB
local   0.000GB
(2) MongoDB [test] -> use AAC
switched to db AAC
(3) MongoDB [AAC] -> db.createUser({user: "aacuser", pwd: "cs340", roles: [{role: "readWrite", db: "AAC"}]})
Successfully added user: {
        "user" : "aacuser",
        "roles" : [
                {
                        "role" : "readWrite",
                        "db" : "AAC"
                }
        ]
}
(4) MongoDB [AAC] ->
```

You can verify that you have enabled user authentication by accessing MongoDB with your new username/password. Type the following command into the terminal CLI to start mongo:

```
mongo --authenticationDatabase "AAC" -u "aacuser" -p
```

The command above will prompt you to enter the username password. Then use the mongo shell command to show databases to verify that you have set up authentication correctly. If you are not logged in with your admin account, no databases will be viewable.


1444638_snhu@msnv-snhu3-l001: ~
File Edit View Search Terminal Help

```
+++ Starting MongoDB: Port=52044  Unix_Socket=/tmp/mongodb-52044.sock Dir=/home/1444638_snhu/mongodb
(base) 1444638_snhu@msnv-snhu3-l001:~$ mongo --authenticationDatabase "AAC" -u "aacuser" -p
MongoDB shell version v4.2.6
Enter password:
connecting to: mongodb://127.0.0.1:52044/?authSource=AAC&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3c38ff1c-ba51-42c8-8e5a-92f1be7a96fe") }
MongoDB server version: 4.2.6
(1) MongoDB [test] -> show dbs
AAC  0.003GB
(2) MongoDB [test] ->
```

Usage

The software application is composed of a Python script module called **crud.py,** enabling the database's CRUD functionality and the **app.py** script file that generates the user dashboard in a browser based on the components of the Dash framework and is the link between the user and the database. To run the web app, enter the following command":

```
python app.py
```

Note the following points while running the app:

- Dash is running on http://127.0.0.1:8050/
- Serving Flask app 'Salvare Search for Rescue Wen App' (lazy loading)
- Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
- Debug mode: on

CRUD Module

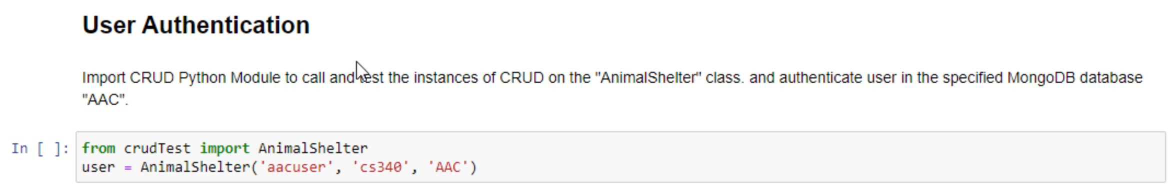
The **crud.py** supports code reusability by importing it as a module by other Python scripts. To test the script module, create a new ***app.py*** Python script file in which you add the following code statements to use the testing script module **crudTest.py**:

To import the test-crud.py module, copy and paste the following line of code:

*from **crudTest** import **AnimalShelter*** *// crud.py class name*

To authenticate the user in the database, add a line of code like:

*your_variable = **AnimalShelter**('username', 'password', 'database')*



User Authentication

Import CRUD Python Module to call and test the instances of CRUD on the "AnimalShelter" class. and authenticate user in the specified MongoDB database "AAC".

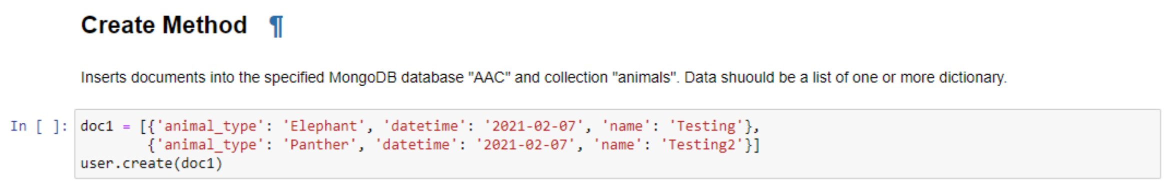
```
In [ ]: from crudTest import AnimalShelter
        user = AnimalShelter('aacuser', 'cs340', 'AAC')
```

CRUD functionalities in the crudTest.py module are:

- ***Create***

  create_variable = [{ <document 1> }, { <document 2>}, …]
  *your_variable.**create**(data_variable)*

  The create method's input requires a list of one or more dictionary. See the code and test example for the proper structure of the input statement. The create method returns the number of documents created and a list of the ObjectId of each document.



Create Method ¶

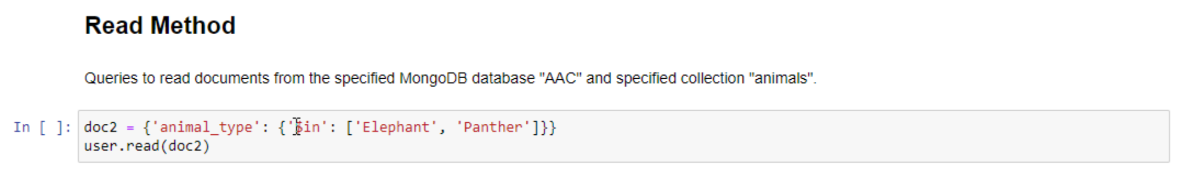
Inserts documents into the specified MongoDB database "AAC" and collection "animals". Data shuould be a list of one or more dictionary.

```
In [ ]: doc1 = [{'animal_type': 'Elephant', 'datetime': '2021-02-07', 'name': 'Testing'},
                {'animal_type': 'Panther', 'datetime': '2021-02-07', 'name': 'Testing2'}]
        user.create(doc1)
```

- ***Read***

  read_variable = { <query filter>, … }
  *your_variable.**read**(data_variable)*

The read method's input requires to be a dictionary with a query filter with or without operators. See the code and test example for the proper structure of the input statement. The read method returns the number of documents found and the list of documents in a readable field:value form.



Read Method

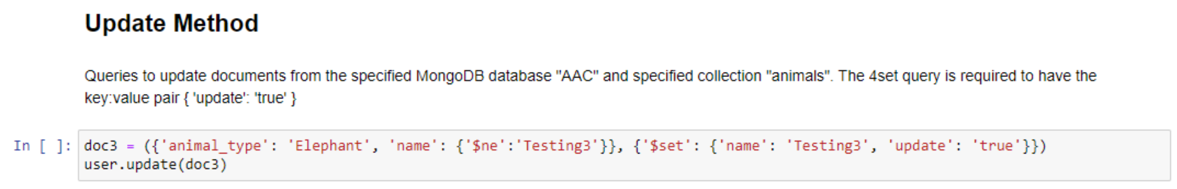
Queries to read documents from the specified MongoDB database "AAC" and specified collection "animals".

```
In [ ]: doc2 = {'animal_type': {'$in': ['Elephant', 'Panther']}}
        user.read(doc2)
```

- ***Update***

```
update_variable = (
        { <query filter> },                              // Query parameter
        {                                                // Update document
            <update operators>: { <field>:<value> },
            $set: { 'update': 'true' }                   // Required
        },
        { 'upsert': 'true' }                             // Options
)
your_variable.update(data_variable)
```

The update method's input requires a tuple of a dictionary with a query filter and update operators with field values to modify. The field value 'update':'true' must be part of a $set update operator. See the code and test example for the proper structure of the input statement. The update method returns the number of documents updated and a list of the modified documents in a readable field:value form.



Update Method

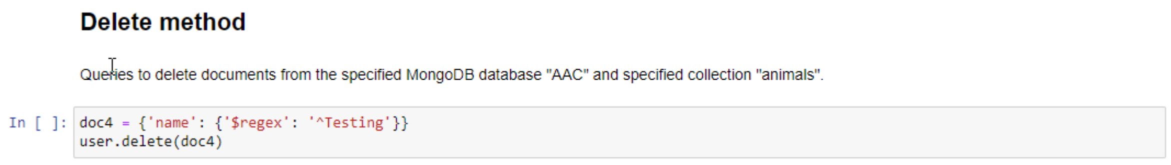
Queries to update documents from the specified MongoDB database "AAC" and specified collection "animals". The 4set query is required to have the key:value pair { 'update': 'true' }

```
In [ ]: doc3 = ({'animal_type': 'Elephant', 'name': {'$ne':'Testing3'}}, {'$set': {'name': 'Testing3', 'update': 'true'}})
        user.update(doc3)
```

- ***Delete***

```
delete_variable = { <query filter> }
your_variable.delete(data_variable)
```

The delete method's input requires a dictionary with a query filter with or without operators. See the code and test example for the proper structure of the input statement. The delete method returns the number of documents removed from the collection.



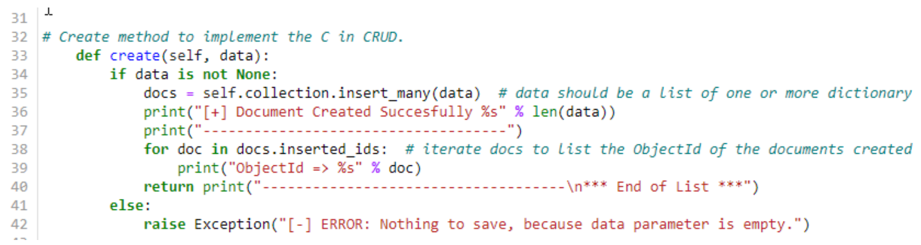
Delete method

Queries to delete documents from the specified MongoDB database "AAC" and specified collection "animals".

```
In [ ]: doc4 = {'name': {'$regex': '^Testing'}}
        user.delete(doc4)
```

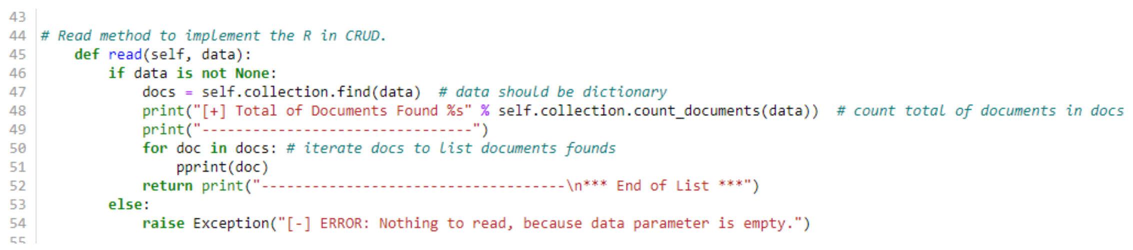
Code Example for crudTest.py

AnimalShelter class create method



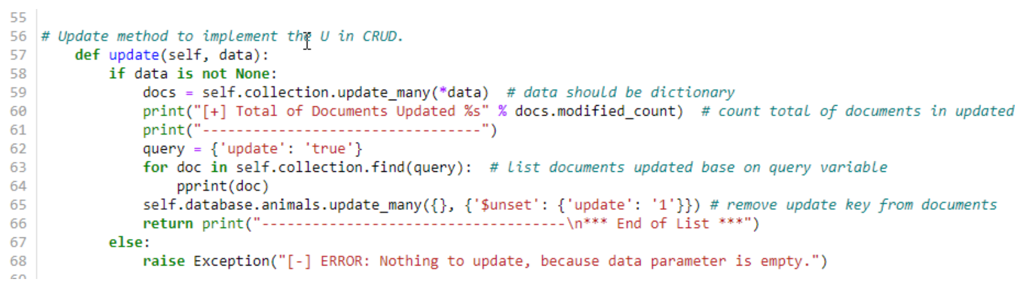
```
# Create method to implement the C in CRUD.
    def create(self, data):
        if data is not None:
            docs = self.collection.insert_many(data)  # data should be a list of one or more dictionary
            print("[+] Document Created Succesfully %s" % len(data))
            print("-----------------------------------")
            for doc in docs.inserted_ids:  # iterate docs to list the ObjectId of the documents created
                print("ObjectId => %s" % doc)
            return print("-----------------------------------\n*** End of List ***")
        else:
            raise Exception("[-] ERROR: Nothing to save, because data parameter is empty.")
```

AnimalShelter class read method



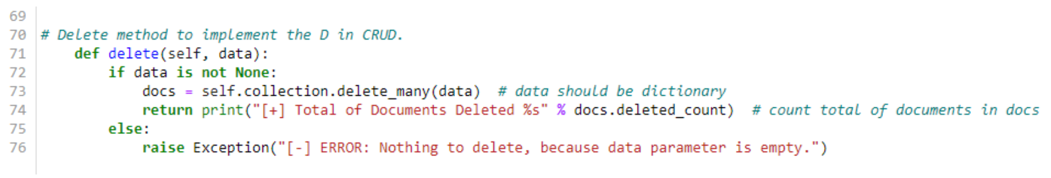
```
# Read method to implement the R in CRUD.
    def read(self, data):
        if data is not None:
            docs = self.collection.find(data)  # data should be dictionary
            print("[+] Total of Documents Found %s" % self.collection.count_documents(data))  # count total of documents in docs
            print("-------------------------------")
            for doc in docs: # iterate docs to list documents founds
                pprint(doc)
            return print("-----------------------------------\n*** End of List ***")
        else:
            raise Exception("[-] ERROR: Nothing to read, because data parameter is empty.")
```

AnimalShelter class update method



```
# Update method to implement the U in CRUD.
    def update(self, data):
        if data is not None:
            docs = self.collection.update_many(*data)  # data should be dictionary
            print("[+] Total of Documents Updated %s" % docs.modified_count)  # count total of documents in updated
            print("--------------------------------")
            query = {'update': 'true'}
            for doc in self.collection.find(query):  # list documents updated base on query variable
                pprint(doc)
            self.database.animals.update_many({}, {'$unset': {'update': '1'}}) # remove update key from documents
            return print("-----------------------------------\n*** End of List ***")
        else:
            raise Exception("[-] ERROR: Nothing to update, because data parameter is empty.")
```

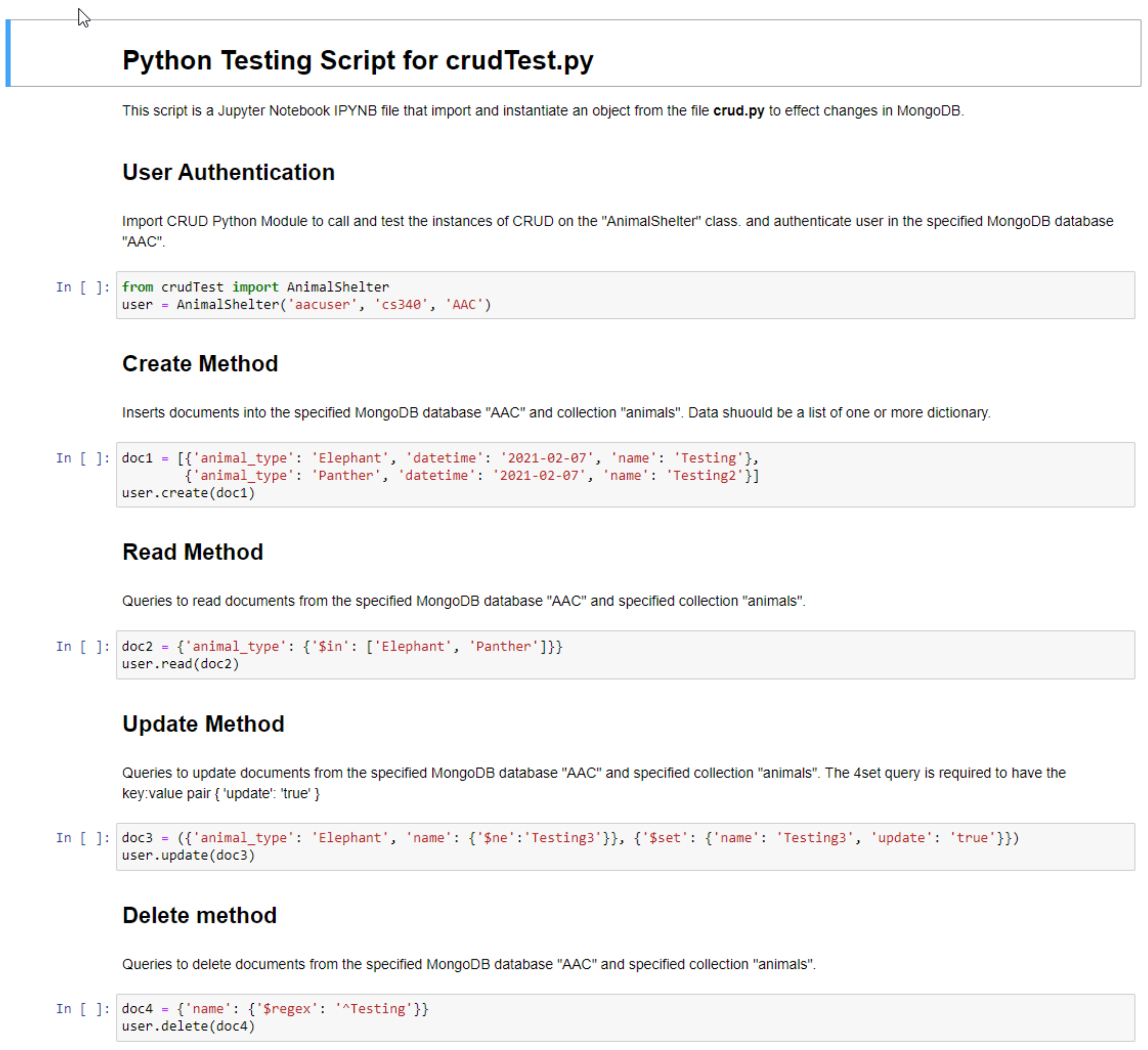
AnimalShelter class delete method



```

# Delete method to implement the D in CRUD.
    def delete(self, data):
        if data is not None:
            docs = self.collection.delete_many(data)  # data should be dictionary
            return print("[+] Total of Documents Deleted %s" % docs.deleted_count)  # count total of documents in docs
        else:
            raise Exception("[-] ERROR: Nothing to delete, because data parameter is empty.")
```

Tests

Using a Python test script in a Jupyter Notebook IPYNB file, **crud-test.ipynb**, you can import and instantiate an object from the Python module **crudTest.py** to effect changes in the MongoDB database. Remember to start the MongoDB server before the execution of the test script.



Python Testing Script for crudTest.py

This script is a Jupyter Notebook IPYNB file that import and instantiate an object from the file **crud.py** to effect changes in MongoDB.

User Authentication

Import CRUD Python Module to call and test the instances of CRUD on the "AnimalShelter" class. and authenticate user in the specified MongoDB database "AAC".

```
In [ ]: from crudTest import AnimalShelter
        user = AnimalShelter('aacuser', 'cs340', 'AAC')
```

Create Method

Inserts documents into the specified MongoDB database "AAC" and collection "animals". Data shuould be a list of one or more dictionary.

```
In [ ]: doc1 = [{'animal_type': 'Elephant', 'datetime': '2021-02-07', 'name': 'Testing'},
                {'animal_type': 'Panther', 'datetime': '2021-02-07', 'name': 'Testing2'}]
        user.create(doc1)
```

Read Method

Queries to read documents from the specified MongoDB database "AAC" and specified collection "animals".

```
In [ ]: doc2 = {'animal_type': {'$in': ['Elephant', 'Panther']}}
        user.read(doc2)
```

Update Method

Queries to update documents from the specified MongoDB database "AAC" and specified collection "animals". The 4set query is required to have the key:value pair { 'update': 'true' }

```
In [ ]: doc3 = ({'animal_type': 'Elephant', 'name': {'$ne':'Testing3'}}, {'$set': {'name': 'Testing3', 'update': 'true'}})
        user.update(doc3)
```

Delete method

Queries to delete documents from the specified MongoDB database "AAC" and specified collection "animals".

```
In [ ]: doc4 = {'name': {'$regex': '^Testing'}}
        user.delete(doc4)
```

The execution of the test script shows the required input and return of the crudTest.py features:

Python Testing Script for crudTest.py

This script is a Jupyter Notebook IPYNB file that import and instantiate an object from the file **crud.py** to effect changes in MongoDB.

User Authentication

Import CRUD Python Module to call and test the instances of CRUD on the "AnimalShelter" class. and authenticate user in the specified MongoDB database "AAC".

```
In [1]: from crudTest import AnimalShelter
        user = AnimalShelter('aacuser', 'cs340', 'AAC')
```

```
[+] User Authenticated in Database [ AAC ]

[+] List of collections
-----------------------
animals

[?] Enter collection to use: animals

[+] Collection to use < animals >

*** Authentication Complete ***
```

Create Method

Inserts documents into the specified MongoDB database "AAC" and collection "animals". Data shuould be a list of one or more dictionary.

```
In [2]: doc1 = [{'animal_type': 'Elephant', 'datetime': '2021-02-07', 'name': 'Testing'},
                {'animal_type': 'Panther', 'datetime': '2021-02-07', 'name': 'Testing2'}]
        user.create(doc1)
```

```
[+] Document Created Succesfully 2
------------------------------------
ObjectId => 6247074d6f0a7b88af2921d2
ObjectId => 6247074d6f0a7b88af2921d3
------------------------------------
*** End of List ***
```

Read Method

Queries to read documents from the specified MongoDB database "AAC" and specified collection "animals".

```
In [3]: doc2 = {'animal_type': {'$in': ['Elephant', 'Panther']}}
        user.read(doc2)
```

```
[+] Total of Documents Found 2
--------------------------------
{'_id': ObjectId('6247074d6f0a7b88af2921d2'),
 'animal_type': 'Elephant',
 'datetime': '2021-02-07',
 'name': 'Testing'}
{'_id': ObjectId('6247074d6f0a7b88af2921d3'),
 'animal_type': 'Panther',
 'datetime': '2021-02-07',
 'name': 'Testing2'}
------------------------------------
*** End of List ***
```

Update Method

Queries to update documents from the specified MongoDB database "AAC" and specified collection "animals". The 4set query is required to have the key:value pair { 'update': 'true' }

```
In [4]: doc3 = ({'animal_type': 'Elephant', 'name': {'$ne':'Testing3'}}, {'$set': {'name': 'Testing3', 'update': 'true'}})
        user.update(doc3)
```

```
[+] Total of Documents Updated 1
---------------------------------
{'_id': ObjectId('6247074d6f0a7b88af2921d2'),
 'animal_type': 'Elephant',
 'datetime': '2021-02-07',
 'name': 'Testing3',
 'update': 'true'}
------------------------------------
*** End of List ***
```

Delete method

Queries to delete documents from the specified MongoDB database "AAC" and specified collection "animals".

```
In [5]: doc4 = {'name': {'$regex': '^Testing'}}
        user.delete(doc4)
```

```
[+] Total of Documents Deleted 2
```

APP File

To launch the application dashboard, enter in the terminal CLI the following command line:

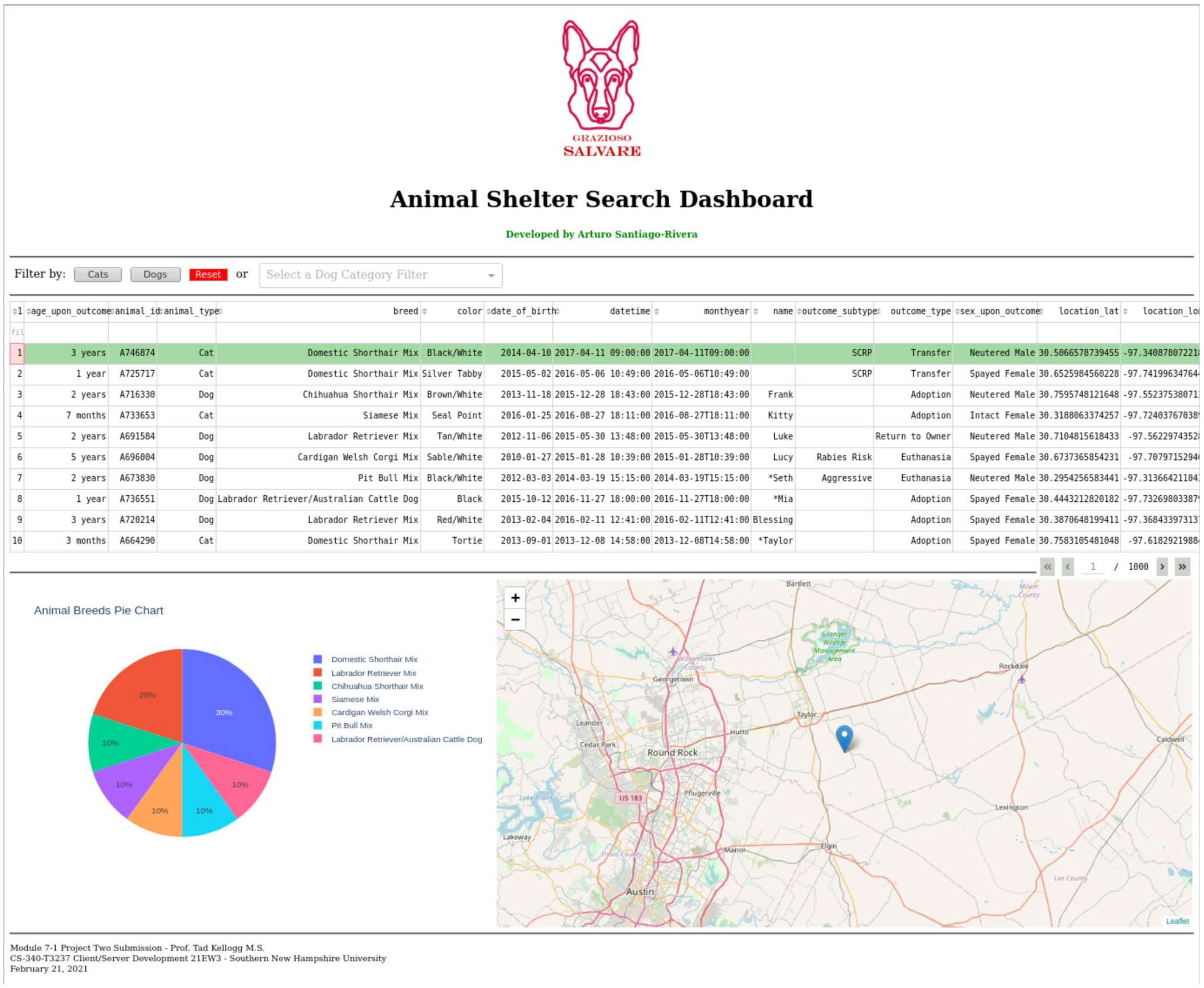
```
python app.py
```

Note the following points while running the app:

- Dash is running on http://127.0.0.1:8050/
- Serving Flask app 'Salvare Search for Rescue Wen App' (lazy loading)
- Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
- Debug mode: on

Open a browser with a new tab pointing to http://127.0.0.1:8050/. The browser starts loading and

generates the client-facing web application dashboard similar to the following screen:



Animal Shelter Search Dashboard

Developed by Arturo Santiago-Rivera

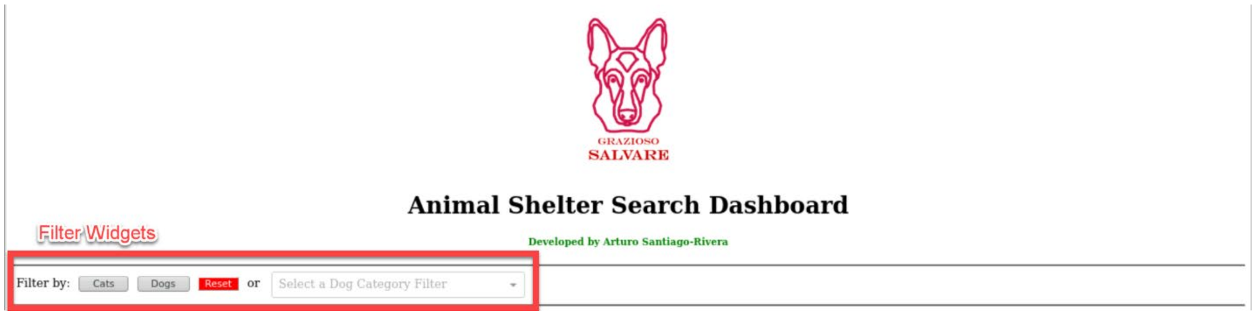
1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_lo
1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.34087807221
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	-97.74199634764
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	-97.55237538071
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	-97.72403767038
5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.5622974352
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	-97.7079715294
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	-97.31366421104
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	-97.73269803387
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	-97.36843397313
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	-97.6182921988

« ‹ 1 / 1000 › »

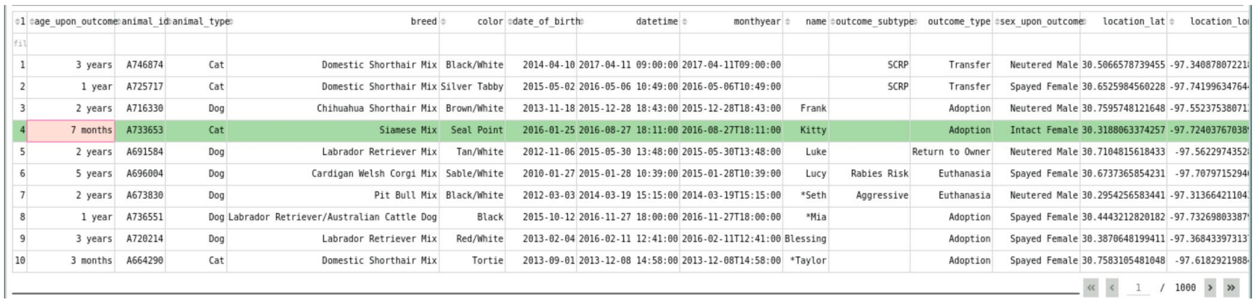
Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

The dashboard is composed of Grazioso Salvare's header branding and footer and the following widgets:

- Interactive filter options (buttons and dropdowns) to filter the shelter data set by:
    - Cat
    - Dogs
    - Dogs Rescue Categories:
        - Water Rescue
        - Mountain or Wilderness Rescue
        - Disaster Rescue or Individual Tracking
    - Reset (returns all widgets to their original, unfiltered state)

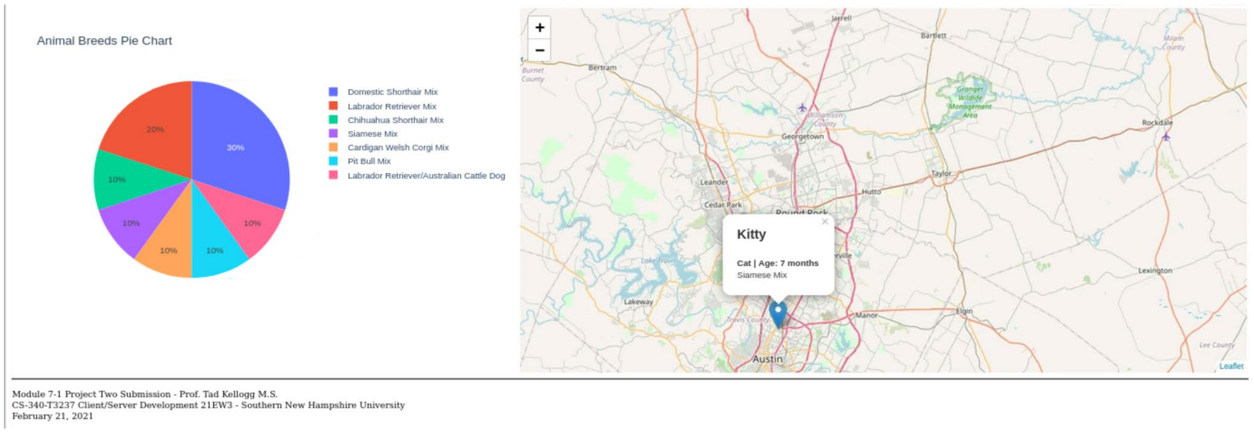

GRAZIOSO
SALVARE
Animal Shelter Search Dashboard
Developed by Arturo Santiago-Rivera
Filter Widgets
Filter by:
Cats
Dogs
Reset
or
Select a Dog Category Filter

- A data table that dynamically responds to the filtering options



1	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_lo
fil														
1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.3408780722
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	-97.7419963476
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	-97.5523753807
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	-97.7240376703
5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.562297435
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	-97.707971529
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	-97.3136642110
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	-97.7326980338
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	-97.3684339731
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	-97.618292198

- A geolocation chart and a pie chart that dynamically responds to the filtering options

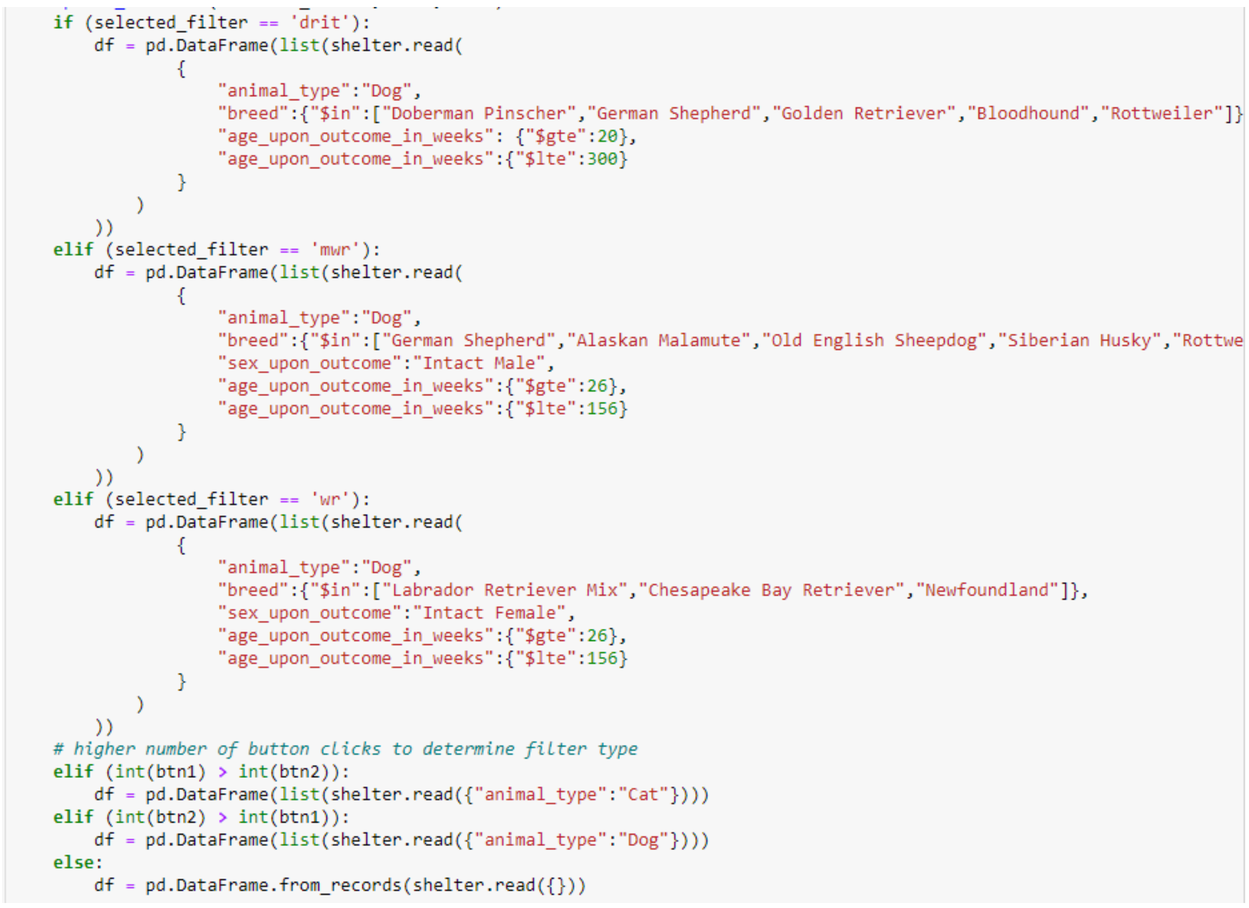


Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

Rescue Type and Preferred Dog Breeds Table

The dog's rescue categories filer is based on the research and experience with training rescue dogs. The table was a guide to the write queries for the interactive option functionality.

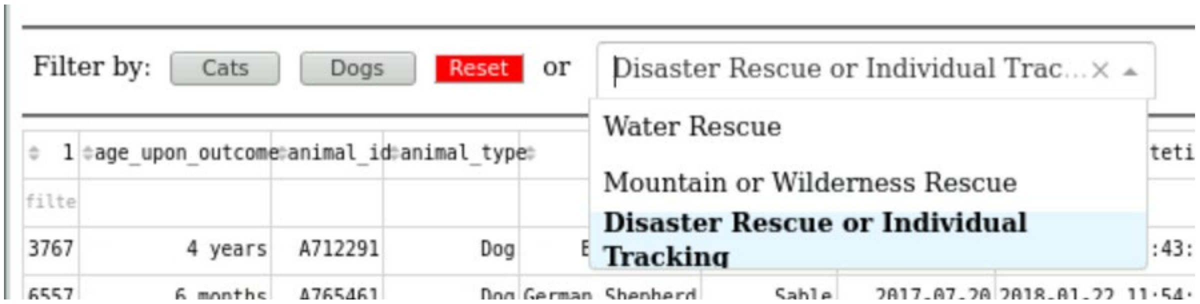
Rescue Type	Preferred Breeds	Preferred Sex	Training Age*
Water	Labrador Retriever Mix, Chesapeake Bay Retriever, Newfoundland	Intact Female	26 weeks to 156 weeks
Mountain or Wilderness	German Shepherd, Alaskan Malamute, Old English Sheepdog, Siberian Husky, Rottweiler	Intact Male	26 weeks to 156 weeks
Disaster or Individual Tracking	Doberman Pinscher, German Shepherd, Golden Retriever, Bloodhound, Rottweiler	Intact Male	20 weeks to 300 weeks



```
    if (selected_filter == 'drit'):
        df = pd.DataFrame(list(shelter.read(
                {
                    "animal_type":"Dog",
                    "breed":{"$in":["Doberman Pinscher","German Shepherd","Golden Retriever","Bloodhound","Rottweiler"]}
                    "age_upon_outcome_in_weeks": {"$gte":20},
                    "age_upon_outcome_in_weeks":{"$lte":300}
                }
            )
        ))
    elif (selected_filter == 'mwr'):
        df = pd.DataFrame(list(shelter.read(
                {
                    "animal_type":"Dog",
                    "breed":{"$in":["German Shepherd","Alaskan Malamute","Old English Sheepdog","Siberian Husky","Rottwe
                    "sex_upon_outcome":"Intact Male",
                    "age_upon_outcome_in_weeks":{"$gte":26},
                    "age_upon_outcome_in_weeks":{"$lte":156}
                }
            )
        ))
    elif (selected_filter == 'wr'):
        df = pd.DataFrame(list(shelter.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},
                    "age_upon_outcome_in_weeks":{"$lte":156}
                }
            )
        ))
    # higher number of button clicks to determine filter type
    elif (int(btn1) > int(btn2)):
        df = pd.DataFrame(list(shelter.read({"animal_type":"Cat"})))
    elif (int(btn2) > int(btn1)):
        df = pd.DataFrame(list(shelter.read({"animal_type":"Dog"})))
    else:
        df = pd.DataFrame.from_records(shelter.read({}))
```

Interactive Filter Options:

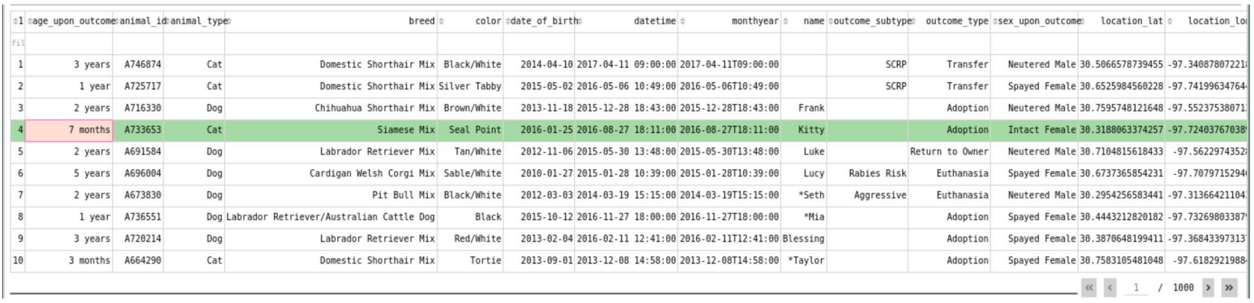
The ability to filter the data gives instantaneous interactive options to run the database queries to gather the required data.


Filter by:
Cats
Dogs
Reset
or
Disaster Rescue or Individual Trac...
Water Rescue
Mountain or Wilderness Rescue
Disaster Rescue or Individual Tracking
1
age_upon_outcome
animal_id
animal_type
teti
filte
3767 4 years A712291 Dog E :43:
6557 6 months A765461 Dog German Shepherd Sable 2017-07-20 2018-01-22 11:54:

The current five interactive options allow the dashboard user to retrieve data related to dog rescue types and classify per type of animal, Cat, and Dog. The dog rescue categories are based on the above table queries and grouped in a dropdown menu as:

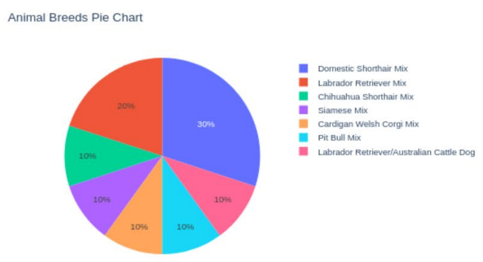
- Water Rescue
- Mountain or Wilderness Rescue
- Disaster or Individual Tracking

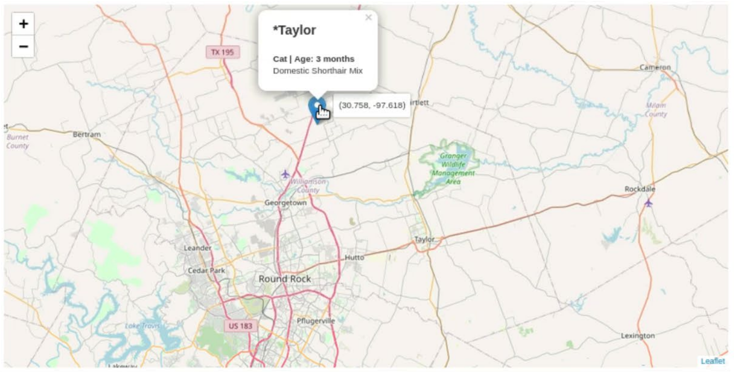
The filter widget buttons can be reset using the reset button, and the dropdown menu filter can be reset by clicking on the "x" adjacent to the selected filter category. When one of the interactive options is selected, the interactive data table and pie chart update show the selected filter records and statistics. Each page view list 10 rows of records.



1	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_lo
fil														
1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.3408780722
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	-97.7419963476
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	-97.5523753807
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	-97.7240376703
5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.5622974352
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	-97.7079715294
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	-97.3136642110
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	-97.7326980338
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	-97.3684339731
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	-97.6182921988

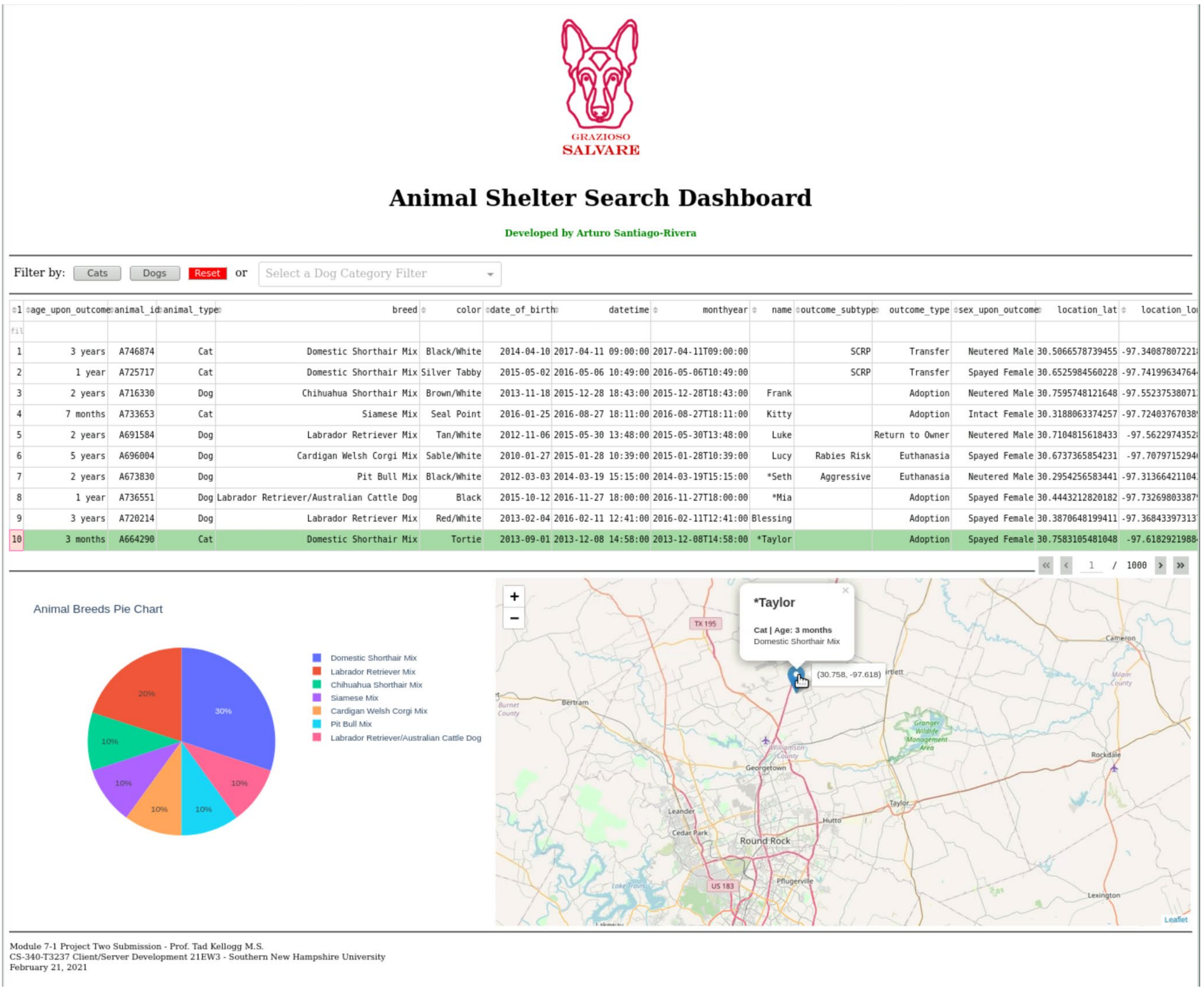
The user can click on a cell in the table, and the entire row of the cell is highlighted. Automatically the geolocation chart is updated to show the location of the animal shelter. The pie chart shows the percentage of animals per breed, and the geolocation chart shows a map pin with the location coordinates of the shelter where the animal is located. If the user clicks on the map-pin, a tooltip opens showing the animal's name, type of animal, age, and breed.





Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

In the following screenshot, you can see how the table and pie chart have been updated based on the list of ten records unfiltered on the first page, and the geolocation map shows the animal base's information on the chosen row. This mockup is the standard dashboard when the software application is initiated.



Animal Shelter Search Dashboard

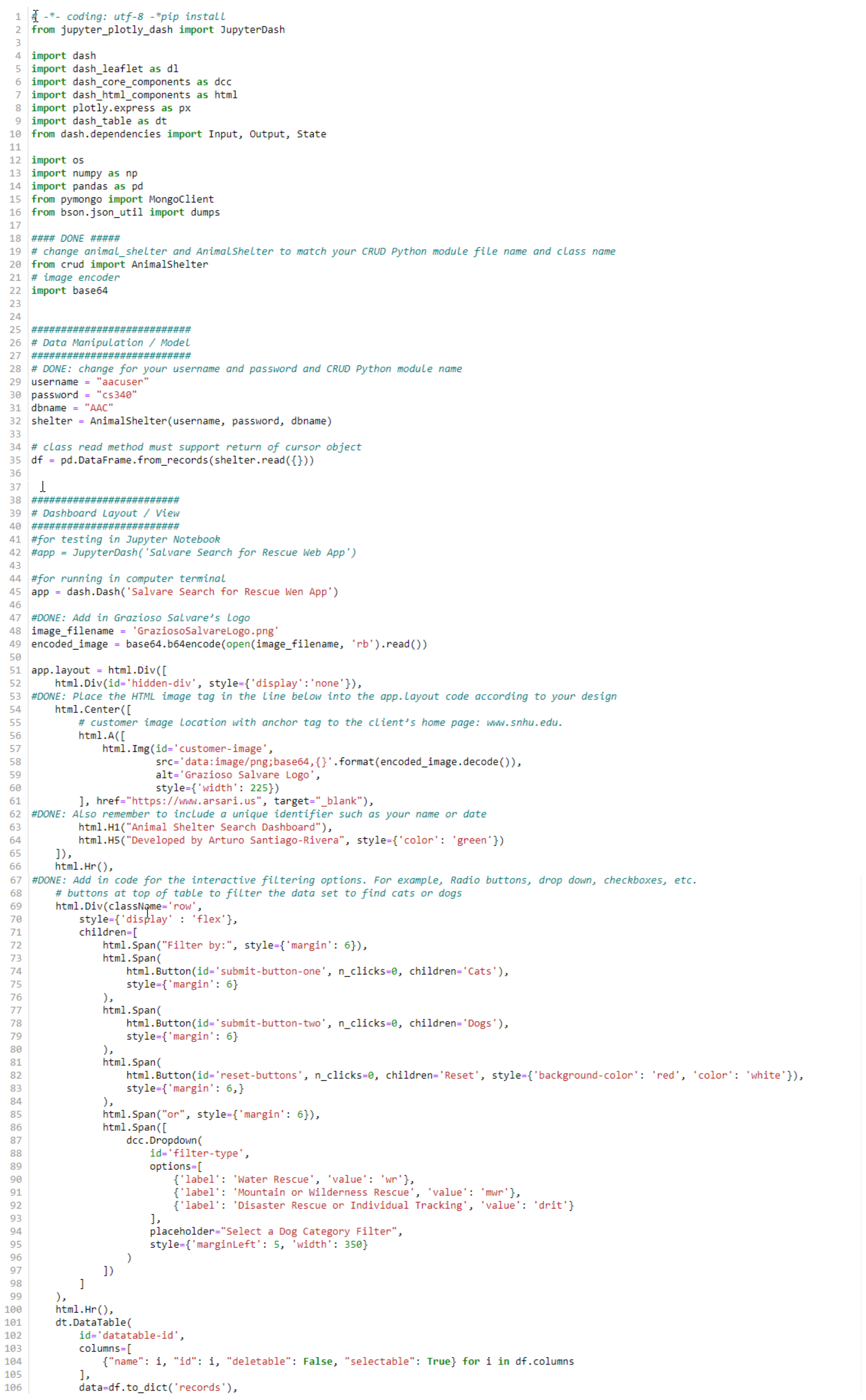
Developed by Arturo Santiago-Rivera

1	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_lo
1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.34087807221
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	-97.74199634764
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	-97.55237538071
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	-97.72403767038
5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.5622974352
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	-97.7079715294
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	-97.31366421104
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	-97.73269803387
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	-97.36843397313
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	-97.6182921988

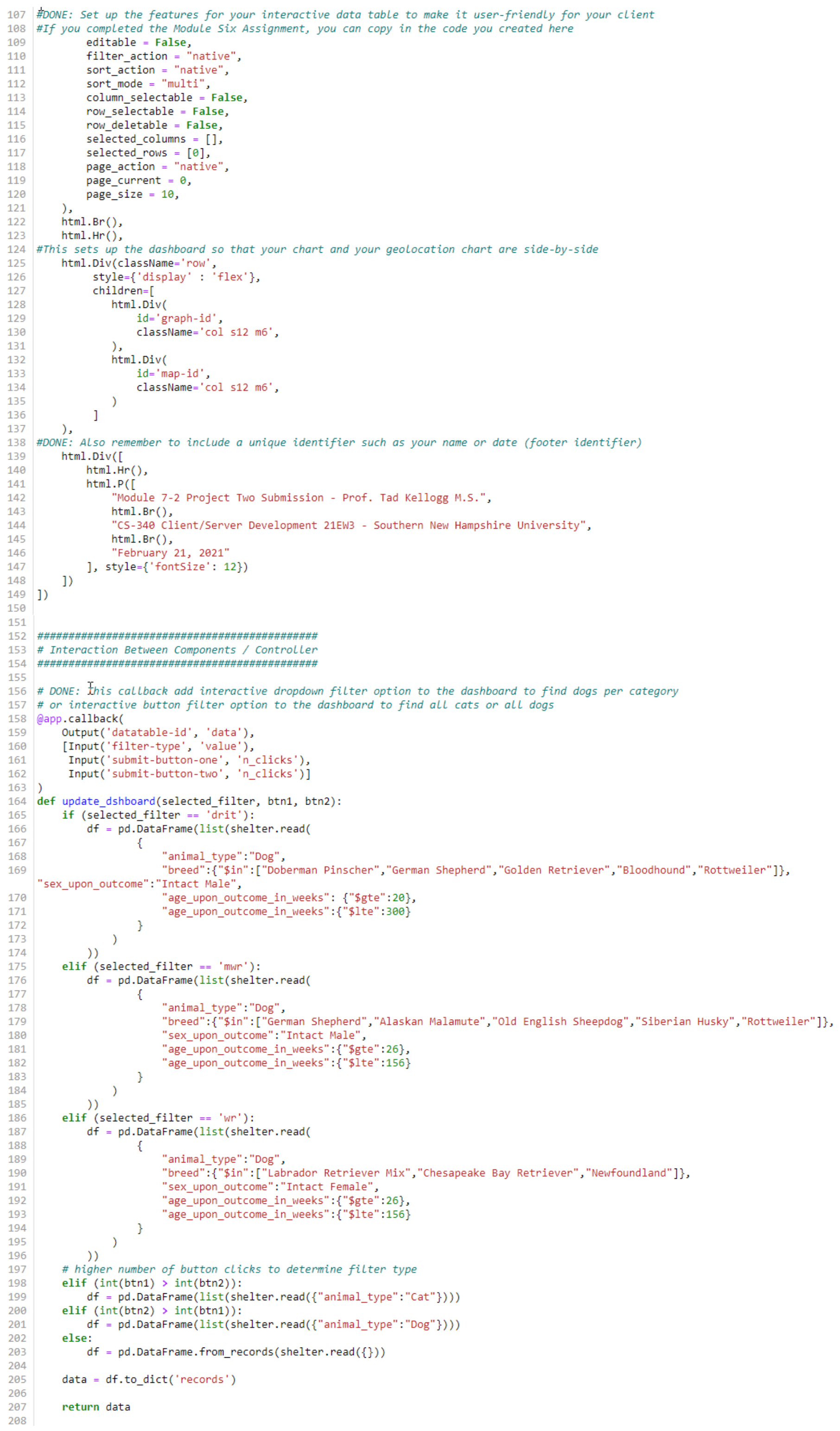
Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

Code Example

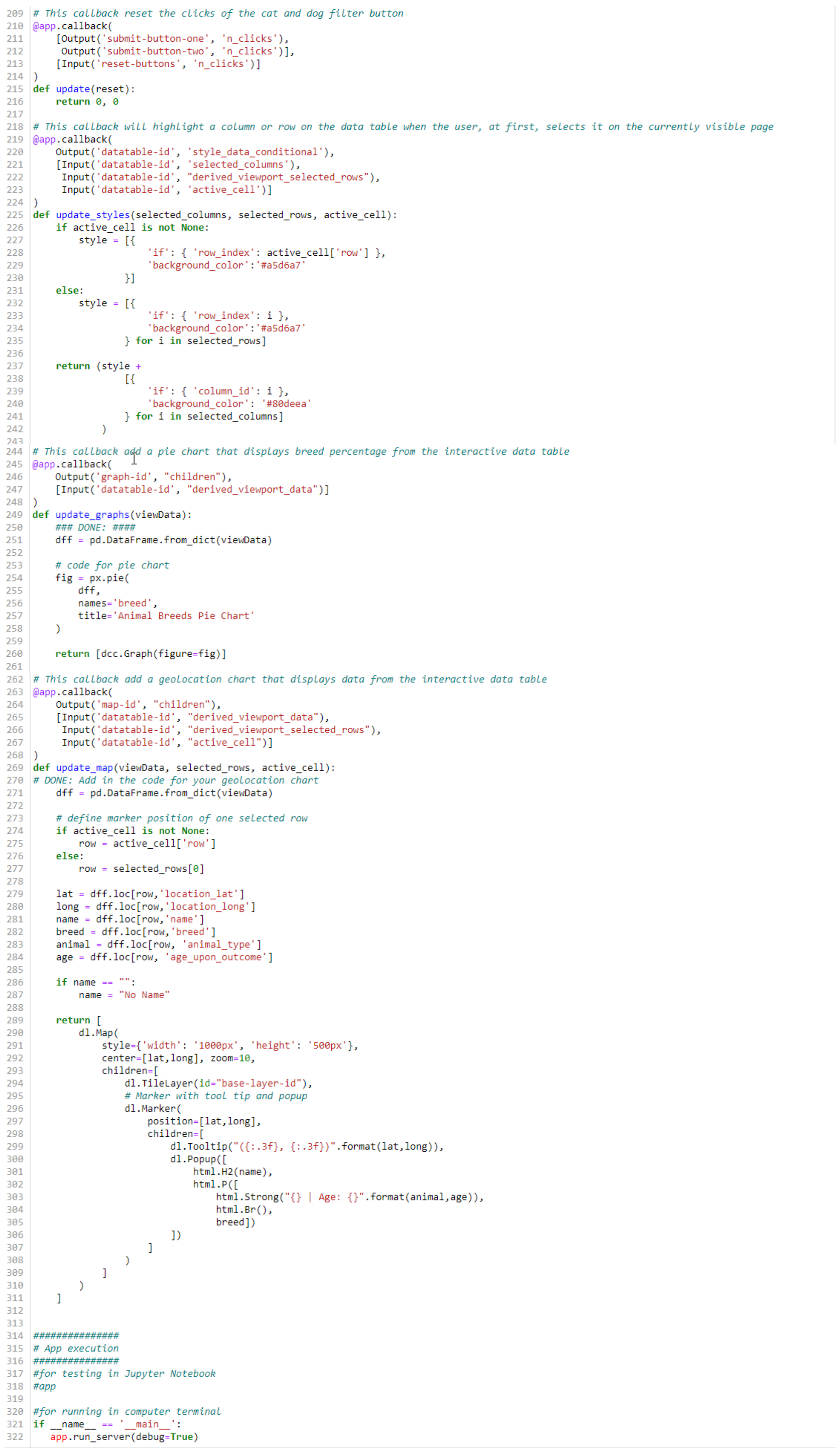
The following screenshots show the application source code base on dash framework components, callbacks, and functions.



```python
# -*- coding: utf-8 -*pip install
from jupyter_plotly_dash import JupyterDash

import dash
import dash_leaflet as dl
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
import dash_table as dt
from dash.dependencies import Input, Output, State

import os
import numpy as np
import pandas as pd
from pymongo import MongoClient
from bson.json_util import dumps

#### DONE #####
# change animal_shelter and AnimalShelter to match your CRUD Python module file name and class name
from crud import AnimalShelter
# image encoder
import base64


###########################
# Data Manipulation / Model
###########################
# DONE: change for your username and password and CRUD Python module name
username = "aacuser"
password = "cs340"
dbname = "AAC"
shelter = AnimalShelter(username, password, dbname)

# class read method must support return of cursor object
df = pd.DataFrame.from_records(shelter.read({}))


#########################
# Dashboard Layout / View
#########################
#for testing in Jupyter Notebook
#app = JupyterDash('Salvare Search for Rescue Web App')

#for running in computer terminal
app = dash.Dash('Salvare Search for Rescue Wen App')

#DONE: Add in Grazioso Salvare's logo
image_filename = 'GraziosoSalvareLogo.png'
encoded_image = base64.b64encode(open(image_filename, 'rb').read())

app.layout = html.Div([
    html.Div(id='hidden-div', style={'display':'none'}),
#DONE: Place the HTML image tag in the line below into the app.layout code according to your design
    html.Center([
        # customer image location with anchor tag to the client's home page: www.snhu.edu.
        html.A([
            html.Img(id='customer-image',
                     src='data:image/png;base64,{}'.format(encoded_image.decode()),
                     alt='Grazioso Salvare Logo',
                     style={'width': 225})
        ], href="https://www.arsari.us", target="_blank"),
#DONE: Also remember to include a unique identifier such as your name or date
        html.H1("Animal Shelter Search Dashboard"),
        html.H5("Developed by Arturo Santiago-Rivera", style={'color': 'green'})
    ]),
    html.Hr(),
#DONE: Add in code for the interactive filtering options. For example, Radio buttons, drop down, checkboxes, etc.
    # buttons at top of table to filter the data set to find cats or dogs
    html.Div(className='row',
        style={'display' : 'flex'},
        children=[
            html.Span("Filter by:", style={'margin': 6}),
            html.Span(
                html.Button(id='submit-button-one', n_clicks=0, children='Cats'),
                style={'margin': 6}
            ),
            html.Span(
                html.Button(id='submit-button-two', n_clicks=0, children='Dogs'),
                style={'margin': 6}
            ),
            html.Span(
                html.Button(id='reset-buttons', n_clicks=0, children='Reset', style={'background-color': 'red', 'color': 'white'}),
                style={'margin': 6,}
            ),
            html.Span("or", style={'margin': 6}),
            html.Span([
                dcc.Dropdown(
                    id='filter-type',
                    options=[
                        {'label': 'Water Rescue', 'value': 'wr'},
                        {'label': 'Mountain or Wilderness Rescue', 'value': 'mwr'},
                        {'label': 'Disaster Rescue or Individual Tracking', 'value': 'drit'}
                    ],
                    placeholder="Select a Dog Category Filter",
                    style={'marginLeft': 5, 'width': 350}
                )
            ])
        ]
    ),
    html.Hr(),
    dt.DataTable(
        id='datatable-id',
        columns=[
            {"name": i, "id": i, "deletable": False, "selectable": True} for i in df.columns
        ],
        data=df.to_dict('records'),
```



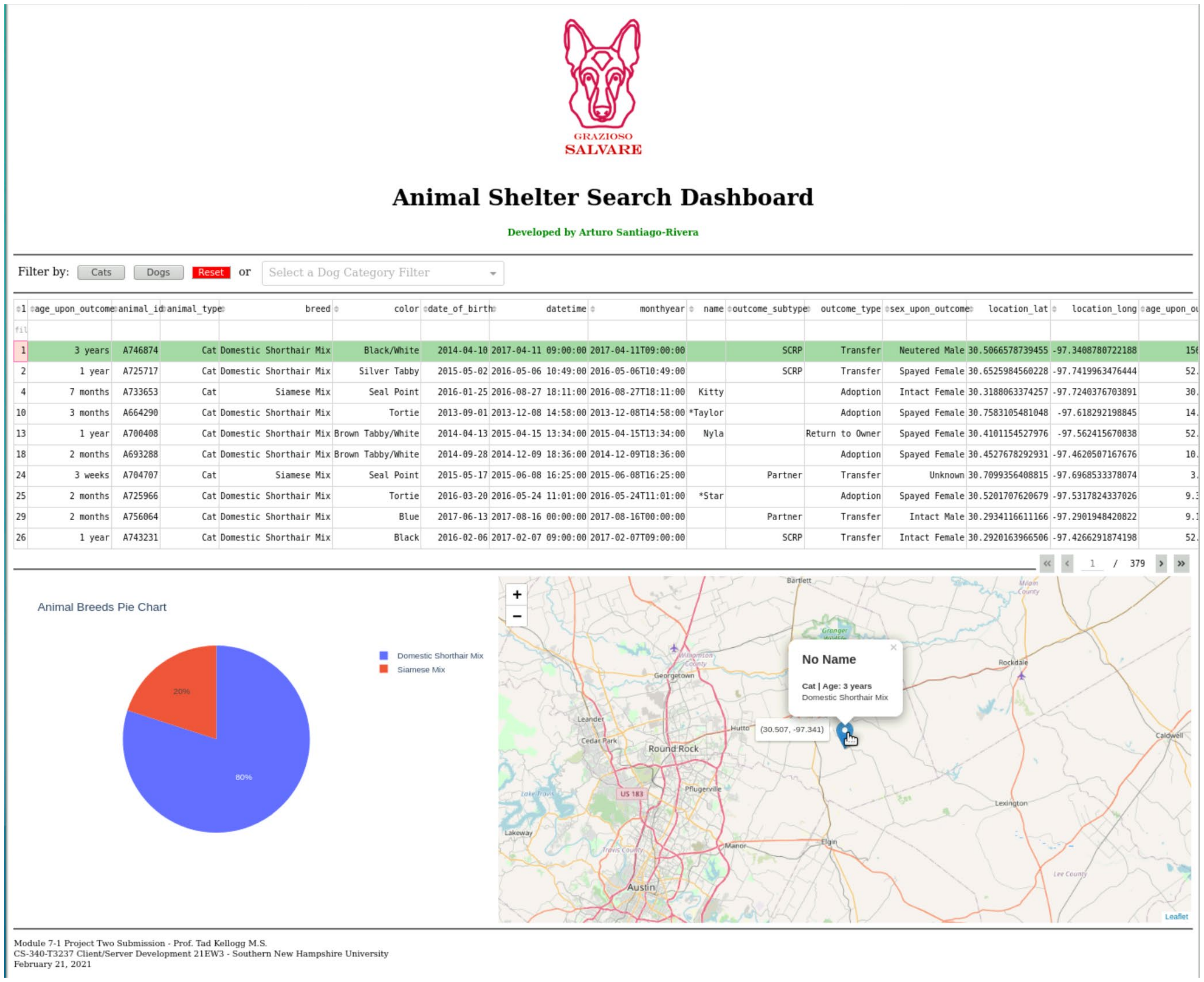
```python
#DONE: Set up the features for your interactive data table to make it user-friendly for your client
#If you completed the Module Six Assignment, you can copy in the code you created here
        editable = False,
        filter_action = "native",
        sort_action = "native",
        sort_mode = "multi",
        column_selectable = False,
        row_selectable = False,
        row_deletable = False,
        selected_columns = [],
        selected_rows = [0],
        page_action = "native",
        page_current = 0,
        page_size = 10,
    ),
    html.Br(),
    html.Hr(),
#This sets up the dashboard so that your chart and your geolocation chart are side-by-side
    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',
            )
         ]
    ),
#DONE: Also remember to include a unique identifier such as your name or date (footer identifier)
    html.Div([
        html.Hr(),
        html.P([
            "Module 7-2 Project Two Submission - Prof. Tad Kellogg M.S.",
            html.Br(),
            "CS-340 Client/Server Development 21EW3 - Southern New Hampshire University",
            html.Br(),
            "February 21, 2021"
        ], style={'fontSize': 12})
    ])
])


#############################################
# Interaction Between Components / Controller
#############################################

# DONE: This callback add interactive dropdown filter option to the dashboard to find dogs per category
# or interactive button filter option to the dashboard to find all cats or all dogs
@app.callback(
    Output('datatable-id', 'data'),
    [Input('filter-type', 'value'),
     Input('submit-button-one', 'n_clicks'),
     Input('submit-button-two', 'n_clicks')]
)
def update_dshboard(selected_filter, btn1, btn2):
    if (selected_filter == 'drit'):
        df = pd.DataFrame(list(shelter.read(
                {
                    "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},
                    "age_upon_outcome_in_weeks":{"$lte":300}
                }
            )
        ))
    elif (selected_filter == 'mwr'):
        df = pd.DataFrame(list(shelter.read(
                {
                    "animal_type":"Dog",
                    "breed":{"$in":["German Shepherd","Alaskan Malamute","Old English Sheepdog","Siberian Husky","Rottweiler"]},
                    "sex_upon_outcome":"Intact Male",
                    "age_upon_outcome_in_weeks":{"$gte":26},
                    "age_upon_outcome_in_weeks":{"$lte":156}
                }
            )
        ))
    elif (selected_filter == 'wr'):
        df = pd.DataFrame(list(shelter.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},
                    "age_upon_outcome_in_weeks":{"$lte":156}
                }
            )
        ))
    # higher number of button clicks to determine filter type
    elif (int(btn1) > int(btn2)):
        df = pd.DataFrame(list(shelter.read({"animal_type":"Cat"})))
    elif (int(btn2) > int(btn1)):
        df = pd.DataFrame(list(shelter.read({"animal_type":"Dog"})))
    else:
        df = pd.DataFrame.from_records(shelter.read({}))

    data = df.to_dict('records')

    return data
```



```python
# This callback reset the clicks of the cat and dog filter button
@app.callback(
    [Output('submit-button-one', 'n_clicks'),
     Output('submit-button-two', 'n_clicks')],
    [Input('reset-buttons', 'n_clicks')]
)
def update(reset):
    return 0, 0

# This callback will highlight a column or row on the data table when the user, at first, selects it on the currently visible page
@app.callback(
    Output('datatable-id', 'style_data_conditional'),
    [Input('datatable-id', 'selected_columns'),
     Input('datatable-id', "derived_viewport_selected_rows"),
     Input('datatable-id', 'active_cell')]
)
def update_styles(selected_columns, selected_rows, active_cell):
    if active_cell is not None:
        style = [{
                    'if': { 'row_index': active_cell['row'] },
                    'background_color':'#a5d6a7'
                }]
    else:
        style = [{
                    'if': { 'row_index': i },
                    'background_color':'#a5d6a7'
                } for i in selected_rows]

    return (style +
                [{
                    'if': { 'column_id': i },
                    'background_color': '#80deea'
                } for i in selected_columns]
            )

# This callback add a pie chart that displays breed percentage from the interactive data table
@app.callback(
    Output('graph-id', "children"),
    [Input('datatable-id', "derived_viewport_data")]
)
def update_graphs(viewData):
    ### DONE: ####
    dff = pd.DataFrame.from_dict(viewData)

    # code for pie chart
    fig = px.pie(
        dff,
        names='breed',
        title='Animal Breeds Pie Chart'
    )

    return [dcc.Graph(figure=fig)]

# This callback add a geolocation chart that displays data from the interactive data table
@app.callback(
    Output('map-id', "children"),
    [Input('datatable-id', "derived_viewport_data"),
     Input('datatable-id', "derived_viewport_selected_rows"),
     Input('datatable-id', "active_cell")]
)
def update_map(viewData, selected_rows, active_cell):
# DONE: Add in the code for your geolocation chart
    dff = pd.DataFrame.from_dict(viewData)

    # define marker position of one selected row
    if active_cell is not None:
        row = active_cell['row']
    else:
        row = selected_rows[0]

    lat = dff.loc[row,'location_lat']
    long = dff.loc[row,'location_long']
    name = dff.loc[row,'name']
    breed = dff.loc[row,'breed']
    animal = dff.loc[row, 'animal_type']
    age = dff.loc[row, 'age_upon_outcome']

    if name == "":
        name = "No Name"

    return [
        dl.Map(
            style={'width': '1000px', 'height': '500px'},
            center=[lat,long], zoom=10,
            children=[
                dl.TileLayer(id="base-layer-id"),
                # Marker with tool tip and popup
                dl.Marker(
                    position=[lat,long],
                    children=[
                        dl.Tooltip("({:.3f}, {:.3f})".format(lat,long)),
                        dl.Popup([
                            html.H2(name),
                            html.P([
                                html.Strong("{} | Age: {}".format(animal,age)),
                                html.Br(),
                                breed])
                        ])
                    ]
                )
            ]
        )
    ]


###############
# App execution
###############
#for testing in Jupyter Notebook
#app

#for running in computer terminal
if __name__ == '__main__':
   app.run_server(debug=True)
```

Tests

Using a Python test script in a Jupyter Notebook IPYNB file, **CS340-M7-2_DashboardCode.ipynb**, the application code was executed to mockup and show each widget integrated into the dashboard. It is important to remember that the MongoDB server should be initiated and running for the execution of the app.py code script. The mockup displays buttons and a dropdown menu as the interactive filtering options. The setup of the dashboard and interactive filtering options are intuitive to navigate.



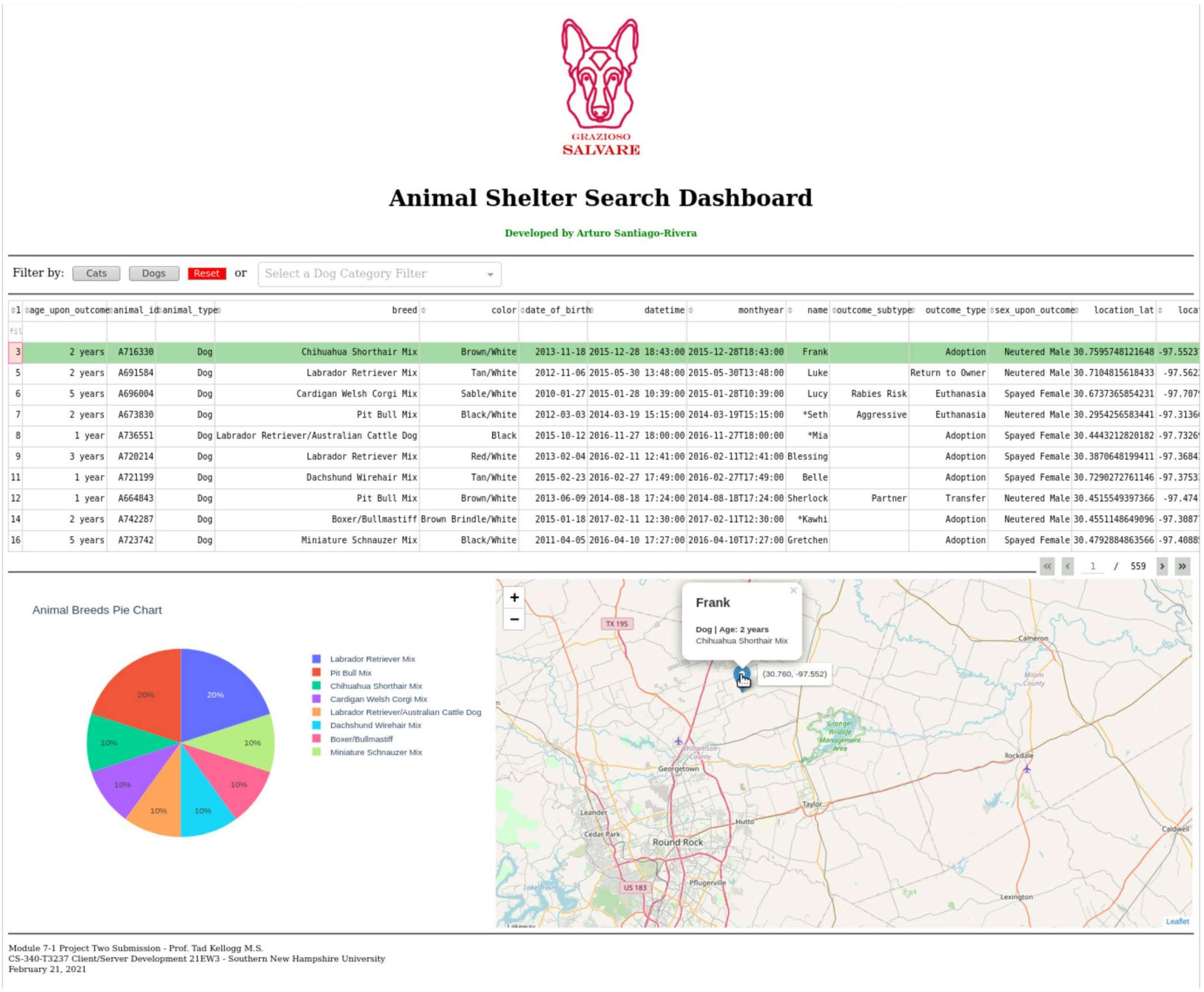
Animal Shelter Search Dashboard

Developed by Arturo Santiago-Rivera

1	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_ou
1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.3408780722188	15
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.
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.
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.
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.
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	-97.4620507167676	10.
24	3 weeks	A704707	Cat	Siamese Mix	Seal Point	2015-05-17	2015-06-08 16:25:00	2015-06-08T16:25:00		Partner	Transfer	Unknown	30.7099356408815	-97.6968533378074	3.
25	2 months	A725966	Cat	Domestic Shorthair Mix	Tortie	2016-03-20	2016-05-24 11:01:00	2016-05-24T11:01:00	*Star		Adoption	Spayed Female	30.5201707620679	-97.5317824337026	9.3
29	2 months	A756064	Cat	Domestic Shorthair Mix	Blue	2017-06-13	2017-08-16 00:00:00	2017-08-16T00:00:00		Partner	Transfer	Intact Male	30.2934116611166	-97.2901948420822	9.1
26	1 year	A743231	Cat	Domestic Shorthair Mix	Black	2016-02-06	2017-02-07 09:00:00	2017-02-07T09:00:00		SCRP	Transfer	Intact Female	30.2920163966506	-97.4266291874198	52.

Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

Mockup of dashboard Filtered by animal type, **CAT**.



Animal Shelter Search Dashboard

Developed by Arturo Santiago-Rivera

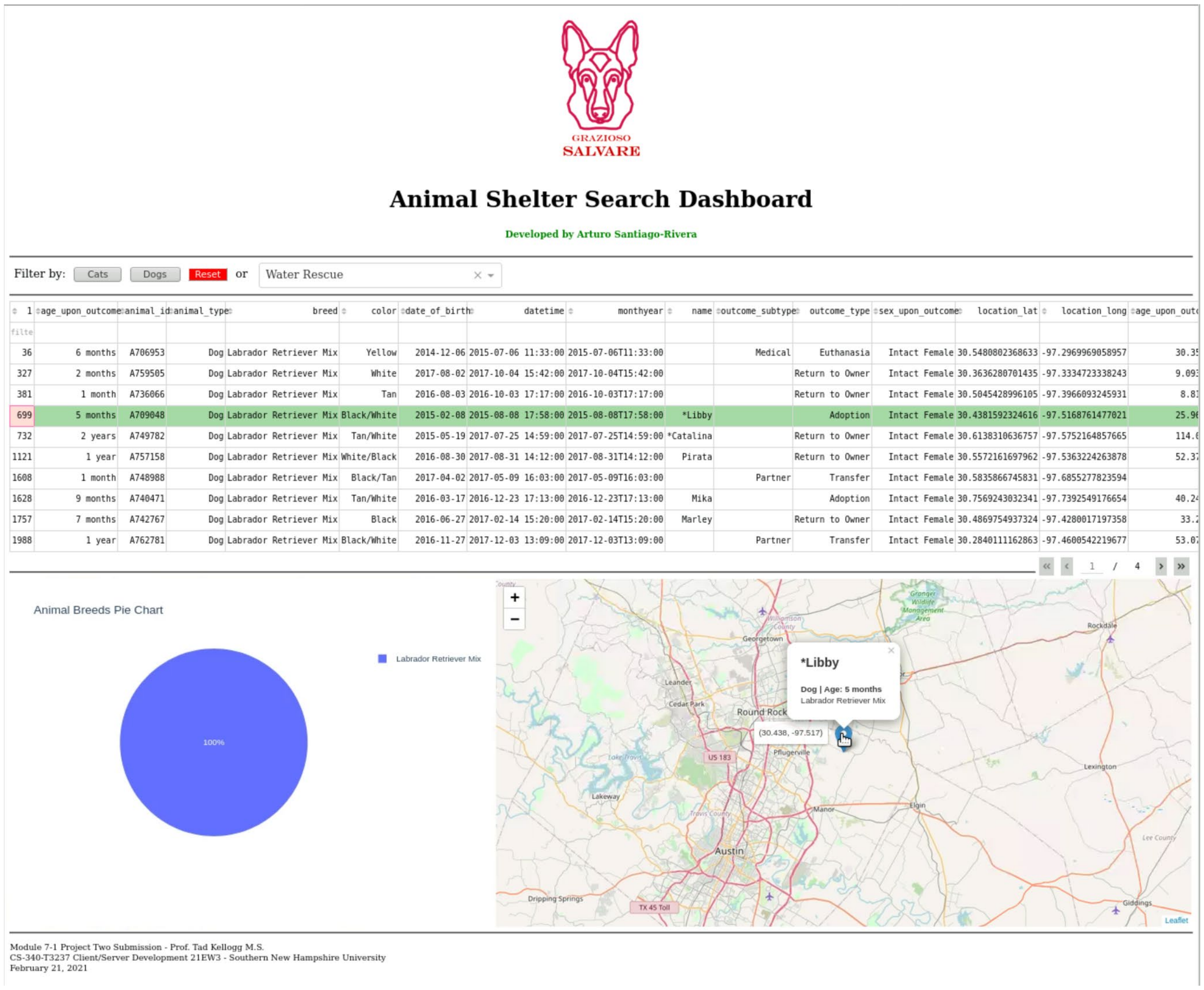
Filter by: Cats Dogs Reset or Select a Dog Category Filter

1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	loca
fil														
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	-97.5523
5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.562
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	-97.707
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	-97.3136
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	-97.7326
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	-97.3684
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	-97.3753
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	-97.474
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	-97.3087
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	-97.4088

« ‹ 1 / 559 › »

Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

Mockup of dashboard filtered by animal type, **DOG**.



GRAZIOSO SALVARE

Animal Shelter Search Dashboard

Developed by Arturo Santiago-Rivera

Filter by: Cats Dogs Reset or Water Rescue

1	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_out
filte															
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	30.3
327	2 months	A759505	Dog	Labrador Retriever Mix	White	2017-08-02	2017-10-04 15:42:00	2017-10-04T15:42:00			Return to Owner	Intact Female	30.3636280701435	-97.3334723338243	9.09
381	1 month	A736066	Dog	Labrador Retriever Mix	Tan	2016-08-03	2016-10-03 17:17:00	2016-10-03T17:17:00			Return to Owner	Intact Female	30.5045428996105	-97.3966093245931	8.8
699	5 months	A709048	Dog	Labrador Retriever Mix	Black/White	2015-02-08	2015-08-08 17:58:00	2015-08-08T17:58:00	*Libby		Adoption	Intact Female	30.4381592324616	-97.5168761477021	25.9
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	114.
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	52.3
1608	1 month	A748988	Dog	Labrador Retriever Mix	Black/Tan	2017-04-02	2017-05-09 16:03:00	2017-05-09T16:03:00		Partner	Transfer	Intact Female	30.5835866745831	-97.6855277823594	
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	40.2
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.
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	30.2840111162863	-97.4600542219677	53.0

1 / 4

Animal Breeds Pie Chart

🟦 Labrador Retriever Mix

100%

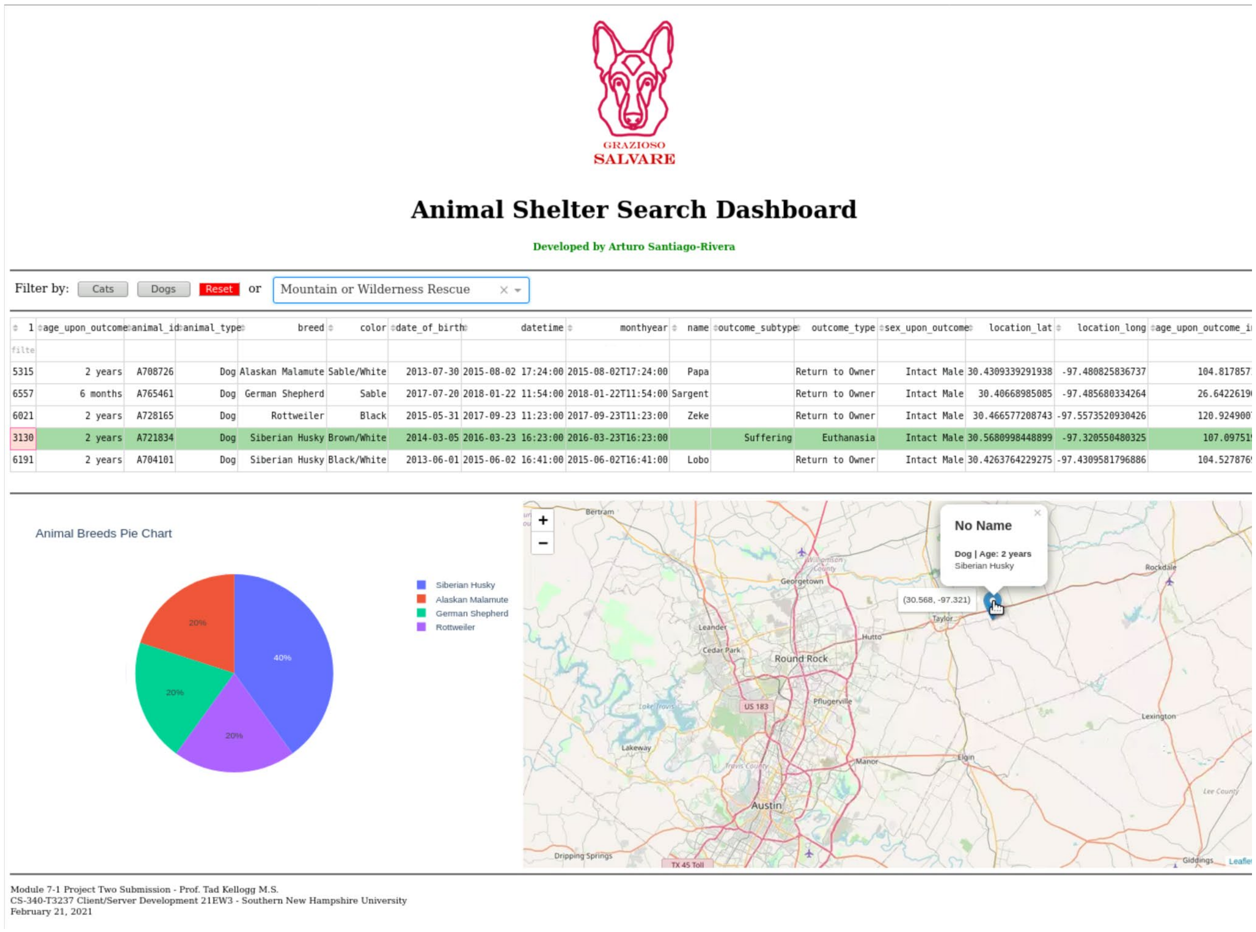
*Libby

Dog | Age: 5 months
Labrador Retriever Mix

(30.438, -97.517)

Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

Mockup of dashboard filtered by dog rescue category, **WATER RESCUE**.



Animal Shelter Search Dashboard

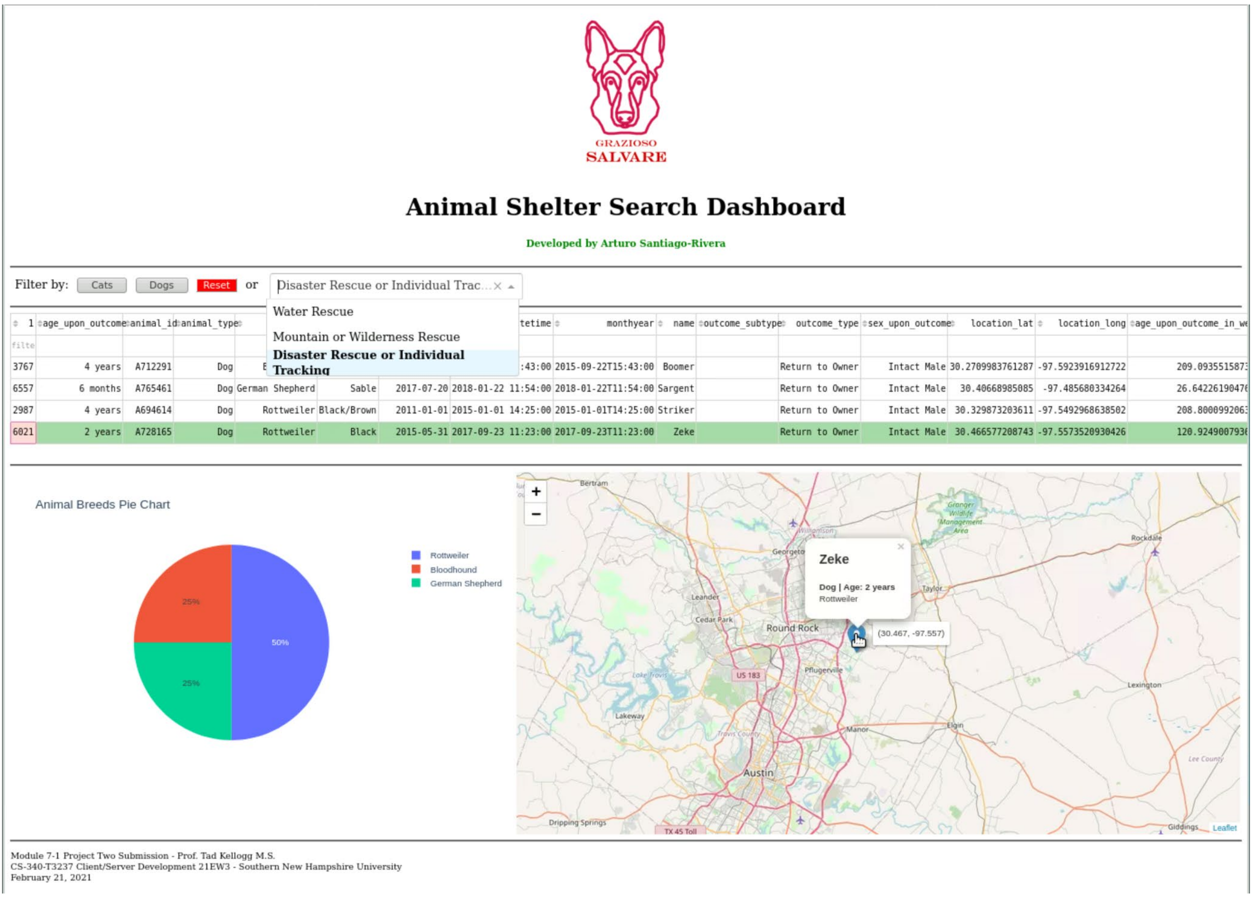
Developed by Arturo Santiago-Rivera

Filter by: Cats Dogs Reset or Mountain or Wilderness Rescue

1	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_i
filte															
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	30.4309339291938	-97.480825836737	104.817857
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	26.6422619
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	120.924900
3130	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	107.09751
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	-97.4309581796886	104.527876

Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

Mockup of dashboard filtered by dog rescue category, **MOUNTAIN OR WILDERNESS RESCUE**.


GRAZIOSO SALVARE

Animal Shelter Search Dashboard

Developed by Arturo Santiago-Rivera

Filter by: Cats Dogs Reset or Disaster Rescue or Individual Trac...

Water Rescue
Mountain or Wilderness Rescue
Disaster Rescue or Individual Tracking

1	age_upon_outcome	animal_id	animal_type				tetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_we
filte															
3767	4 years	A712291	Dog	E			:43:00	2015-09-22T15:43:00	Boomer		Return to Owner	Intact Male	30.2709983761287	-97.5923916912722	209.0935515873
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	26.6422619047
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	208.800099206
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	120.924900793

Module 7-1 Project Two Submission - Prof. Tad Kellogg M.S.
CS-340-T3237 Client/Server Development 21EW3 - Southern New Hampshire University
February 21, 2021

Mockup of dashboard filtered by dog rescue category, **DISASTER RESCUE OR INDIVIDUAL TRACKING**.

Challenges

There are identified challenges that need to be attended to for the proper development of the web application dashboard. Because of the facility that brings MongoDB to manage a significant amount of data, the software application's CRUD can be simple and be transparent. However, the dashboard development using the Dash framework could be more time-consuming. Understanding how the dash core, HTML components, and callbacks work to produce an efficient and straightforward coding structure. There is a lot behind the framework, and their libraries are under active development, so installation and upgrade frequently are necessary. However, because the Dash apps are rendered in the web browser, you deploy your app to servers and share them through URLs. Since Dash apps are viewed in the web browser, Dash is inherently cross-platform and mobile-ready.

Roadmap/Features

- Input fields for the user to enter the credentials (username, password, and database) to authenticate in a specified database
- Improvement to the dashboard for better user experience and user interaction

Contact

For questions or suggestions that can improve the app, please email Arturo Santiago-Rivera (arturo.santiago-rivera@snhu.edu)

License