README

About AnimalShelter and the CRUD Module
The animalShelter program allows users to access a database of rescue animals at a local shelter. Here, users can create an account to create, read, update, or delete animal records in the database. These features make up the computer programming acronym, "CRUD". The CRUD Python module is an effective way to manage information in a client-server database environment.

Motivation
AnimalShelter is a project demonstrating my ability to work with databases and their information.

Completing the Project
1. Access Mongo and import the "aac_shelter_outcomes.csv" file with the **mongoimport** tool.
    a. Use the database name "AAC".
    b. Use the collection name "animals".
2. Create simple and complex indices to parse through data in the csv file named above.
3. Create accounts for the AAC database using the **createUser** tool.
    a. An "Admin" account for read/write permissions across all layers of the program, mainly backend.
    b. An "aacuser" account for read/write permissions within certain layers of the program, mainly frontend.
4. Verify user account credentials and permissions using the **getUser** tool.
5. Install Python and create the CRUD program in a .py file via Jupyter. Each feature in the CRUD program should have its own function; create(), read(), update(), and delete().
6. Develop tests for each of the CRUD functions in an .ipynb file via Jupyter.
7. Incorporate a working user interface with Grazioso Salvare's logo and data requirements/filters.

Tools
- MongoDB to access the database.
    - MongoDB allows the animal rescue to aggregate information, but also to pinpoint information within their database as needed. MongoDB can store any kinds of files at any kinds of sizes without the stack being affected. This is useful for scaling and ensures data integrity.
- The latest version of Python to support the .py and .ipynb files.
- The latest version of Jupyter to create and run the .py and .ipynb files.
- Dash framework.
    - The Dash framework is an open-source framework that helps visualize databases. This allows data from the AnimalShelter database to be viewed more easily compared to single-line JSON formats.
- Grazioso Animal Rescue logo.

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

Usage

Code Examples

This block of code in animalShelter.py demonstrates the **Create** feature. In this example, users can create simple or complex indices for the database.

```
# Complete this create method to implement the C in CRUD.
    def create(self, data):
        if data is not None:
            self.database.animals.insert_one(data)  # data should be dictionary
            return True
        else:
            raise Exception("Nothing to save, because data parameter is empty")
```

This block of code in animalShelter.py demonstrates the **Read** feature. In this example, users can search for animals based on specific fields and details.

```

# Create method to implement the R in CRUD.
    def read(self, data):
        if data is not None:
            data = self.database.animals.find(data, {"_id": False})
        else:
            data = self.database.animals.find({}, {"_id": False})
        return data

```

This block of code in animalShelter.py demonstrates the **Update** feature. In this example, users can update entries of animals based on specific fields and details.

```
# Create method to implement the U in CRUD.
    def update(self, initialData, changedData):
        if initialData is not None:
            if self.database.animals.count_documents(initialData, limit = 1) != 0:
                update_result = self.database.animals.update_many(initialData, {"$set": changedData})
                result = update_result.raw_result
            else:
                result = "Document could not be found."
            return result
        else:
            raise Exception("Nothing to update, because data parameter is empty.")
```

This block of code in animalShelter.py demonstrates the **Delete** feature. In this example, users can delete entries of animals based on specific fields and details.

```
# Create method to implement the D in CRUD.
    def delete(self, remove):
        if remove is not None:
            if self.database.animals.count_documents(remove, limit = 1) != 0:
                delete_result = self.database.animals.delete_many(remove)
                result = delete_result.raw_result
            else:
                result = "Document could not be found."
            return result
        else:
            raise Exception("Nothing to update, because data parameter is empty.")
```

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.

Tests

This was the test for the **Create** method. We created an “animal type” field with type “dog”.

```
In [3]: created = CRUD.create({"animal_type": "Dog"})
        print(created)

        True
```

This was the test for the **Read** method. We searched for animals in the database with the name “Moose”.

```
In [4]: readResult = CRUD.read({"name":"Moose"})

        for r in readResult:
            print (r)
```


```
{'rec_num': 1647, 'age_upon_outcome': '6 years', 'animal_id': 'A586033', 'animal_type': 'Dog', 'breed': 'Beagle/Labr
ador Retriever', 'color': 'White', 'date_of_birth': '2010-01-02', 'datetime': '2016-03-06 17:56:00', 'monthyear': '2
016-03-06T17:56:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adoption', 'sex_upon_outcome': 'Neuter
ed Male', 'location_lat': 30.4481264417081, 'location_long': -97.4494227220167, 'age_upon_outcome_in_weeks': 322.249
603174603}
{'rec_num': 2213, 'age_upon_outcome': '6 years', 'animal_id': 'A586033', 'animal_type': 'Dog', 'breed': 'Beagle/Labr
ador Retriever', 'color': 'White', 'date_of_birth': '2010-01-02', 'datetime': '2016-02-21 19:17:00', 'monthyear': '2
016-02-21T19:17:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adoption', 'sex_upon_outcome': 'Neuter
ed Male', 'location_lat': 30.7607628821204, 'location_long': -97.540094408076, 'age_upon_outcome_in_weeks': 320.2576
38888889}
{'rec_num': 3460, 'age_upon_outcome': '3 months', 'animal_id': 'A681348', 'animal_type': 'Cat', 'breed': 'Domestic S
horthair Mix', 'color': 'Brown Tabby/White', 'date_of_birth': '2014-03-24', 'datetime': '2014-06-28 17:11:00', 'mont
hyear': '2014-06-28T17:11:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adoption', 'sex_upon_outcome
': 'Spayed Female', 'location_lat': 30.5605916759449, 'location_long': -97.6581172285255, 'age_upon_outcome_in_weeks
': 13.8165674603175}
{'rec_num': 4032, 'age_upon_outcome': '1 year', 'animal_id': 'A691463', 'animal_type': 'Dog', 'breed': 'German Sheph
erd Mix', 'color': 'Tan/Black', 'date_of_birth': '2013-11-04', 'datetime': '2014-11-05 16:13:00', 'monthyear': '2014
-11-05T16:13:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Return to Owner', 'sex_upon_outcome': 'Ne
utered Male', 'location_lat': 30.5863370809376, 'location_long': -97.7004659726529, 'age_upon_outcome_in_weeks': 52.
3822420634921}
{'rec_num': 5642, 'age_upon_outcome': '3 months', 'animal_id': 'A765387', 'animal_type': 'Dog', 'breed': 'Labrador R
etriever Mix', 'color': 'Chocolate/White', 'date_of_birth': '2017-10-04', 'datetime': '2018-01-22 19:01:00', 'monthy
ear': '2018-01-22T19:01:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adoption', 'sex_upon_outcome':
'Spayed Female', 'location_lat': 30.5580807764267, 'location_long': -97.6356438422187, 'age_upon_outcome_in_weeks':
15.8274801587302}
{'rec_num': 6048, 'age_upon_outcome': '2 months', 'animal_id': 'A680796', 'animal_type': 'Cat', 'breed': 'Domestic S
horthair Mix', 'color': 'Black/White', 'date_of_birth': '2014-04-20', 'datetime': '2014-07-14 18:38:00', 'monthyear
': '2014-07-14T18:38:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adoption', 'sex_upon_outcome': 'N
eutered Male', 'location_lat': 30.4663361702729, 'location_long': -97.7264277963487, 'age_upon_outcome_in_weeks': 1
2.253769841269 8}
{'rec_num': 8605, 'age_upon_outcome': '3 months', 'animal_id': 'A709750', 'animal_type': 'Cat', 'breed': 'Domestic M
edium Hair Mix', 'color': 'Orange Tabby/White', 'date_of_birth': '2015-07-21', 'datetime': '2015-11-10 15:58:00', 'm
onthyear': '2015-11-10T15:58:00', 'name': 'Moose', 'outcome_subtype': 'Foster', 'outcome_type': 'Adoption', 'sex_upo
n_outcome': 'Intact Male', 'location_lat': 30.4492151112687, 'location_long': -97.657196197171, 'age_upon_outcome_in
_weeks': 16.0950396825397}
```


This was the test for the **Update** method. We updated adoption statuses to “Adopted” for animals in the database with the name “Moose”.

```
In [6]: # View updated entries from Test U in CRUD

        readResult = CRUD.read({"name":"Moose"})

        for r in readResult:
            print (r)
```


```
{'rec_num': 1647, 'age_upon_outcome': '6 years', 'animal_id': 'A586033', 'animal_type': 'Dog', 'breed': 'Beagle/Labr
ador Retriever', 'color': 'White', 'date_of_birth': '2010-01-02', 'datetime': '2016-03-06 17:56:00', 'monthyear': '2
016-03-06T17:56:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adopted', 'sex_upon_outcome': 'Neutere
d Male', 'location_lat': 30.4481264417081, 'location_long': -97.4494227220167, 'age_upon_outcome_in_weeks': 322.2496
03174603}
{'rec_num': 2213, 'age_upon_outcome': '6 years', 'animal_id': 'A586033', 'animal_type': 'Dog', 'breed': 'Beagle/Labr
ador Retriever', 'color': 'White', 'date_of_birth': '2010-01-02', 'datetime': '2016-02-21 19:17:00', 'monthyear': '2
016-02-21T19:17:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adopted', 'sex_upon_outcome': 'Neutere
d Male', 'location_lat': 30.7607628821204, 'location_long': -97.540094408076, 'age_upon_outcome_in_weeks': 320.25763
8888889}
{'rec_num': 3460, 'age_upon_outcome': '3 months', 'animal_id': 'A681348', 'animal_type': 'Cat', 'breed': 'Domestic S
horthair Mix', 'color': 'Brown Tabby/White', 'date_of_birth': '2014-03-24', 'datetime': '2014-06-28 17:11:00', 'mont
hyear': '2014-06-28T17:11:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adopted', 'sex_upon_outcome
': 'Spayed Female', 'location_lat': 30.5605916759449, 'location_long': -97.6581172285255, 'age_upon_outcome_in_weeks
': 13.8165674603175}
{'rec_num': 4032, 'age_upon_outcome': '1 year', 'animal_id': 'A691463', 'animal_type': 'Dog', 'breed': 'German Sheph
erd Mix', 'color': 'Tan/Black', 'date_of_birth': '2013-11-04', 'datetime': '2014-11-05 16:13:00', 'monthyear': '2014
-11-05T16:13:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adopted', 'sex_upon_outcome': 'Neutered M
ale', 'location_lat': 30.5863370809376, 'location_long': -97.7004659726529, 'age_upon_outcome_in_weeks': 52.38224206
34921}
{'rec_num': 5642, 'age_upon_outcome': '3 months', 'animal_id': 'A765387', 'animal_type': 'Dog', 'breed': 'Labrador R
etriever Mix', 'color': 'Chocolate/White', 'date_of_birth': '2017-10-04', 'datetime': '2018-01-22 19:01:00', 'monthy
ear': '2018-01-22T19:01:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adopted', 'sex_upon_outcome':
'Spayed Female', 'location_lat': 30.5580807764267, 'location_long': -97.6356438422187, 'age_upon_outcome_in_weeks':
15.8274801587302}
{'rec_num': 6048, 'age_upon_outcome': '2 months', 'animal_id': 'A680796', 'animal_type': 'Cat', 'breed': 'Domestic S
horthair Mix', 'color': 'Black/White', 'date_of_birth': '2014-04-20', 'datetime': '2014-07-14 18:38:00', 'monthyear
': '2014-07-14T18:38:00', 'name': 'Moose', 'outcome_subtype': '', 'outcome_type': 'Adopted', 'sex_upon_outcome': 'Ne
utered Male', 'location_lat': 30.4663361702729, 'location_long': -97.7264277963487, 'age_upon_outcome_in_weeks': 12.
2537698412698}
{'rec_num': 8605, 'age_upon_outcome': '3 months', 'animal_id': 'A709750', 'animal_type': 'Cat', 'breed': 'Domestic M
edium Hair Mix', 'color': 'Orange Tabby/White', 'date_of_birth': '2015-07-21', 'datetime': '2015-11-10 15:58:00', 'm
onthyear': '2015-11-10T15:58:00', 'name': 'Moose', 'outcome_subtype': 'Foster', 'outcome_type': 'Adopted', 'sex_upon
_outcome': 'Intact Male', 'location_lat': 30.4492151112687, 'location_long': -97.657196197171, 'age_upon_outcome_in_
weeks': 16.0950396825397}
```


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.

This was the test for the **Delete** method. We deleted entries animals in the database with the name “Moose”.

```
In [7]: # Test D in CRUD

        deleteAnimal = CRUD.delete({"name": "Moose"})
        print(deleteAnimal)

        {'n': 7, 'ok': 1.0}
```

```
In [8]: # View updated entries from Test D in CRUD

        readResult = CRUD.read({"name":"Moose"})

        for r in readResult:
            print (r)
```

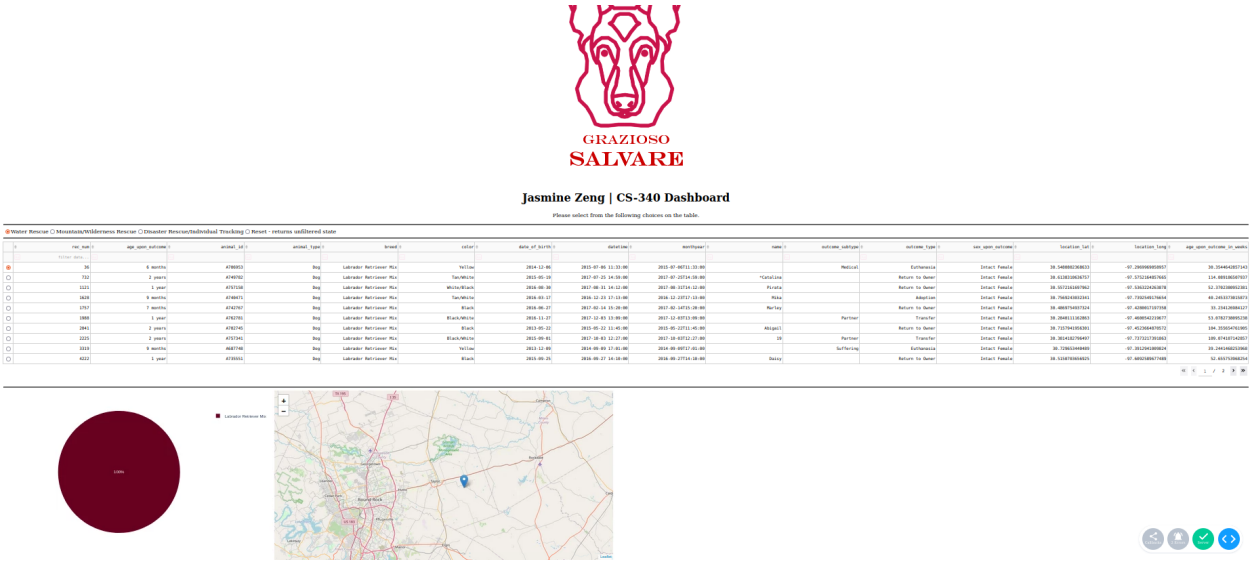
Required Functionality

The deliverable requires the following:

- Grazioso Salvare logo and unique identifier.
- Database filters for the following kinds of rescues:
    - Water Rescue.
    - Mountain or Wilderness Rescue.
    - Disaster or Individual Tracking.
    - Reset.
- Correlating pie chart for each rescue filter.
- Correlating map for each rescue filter.

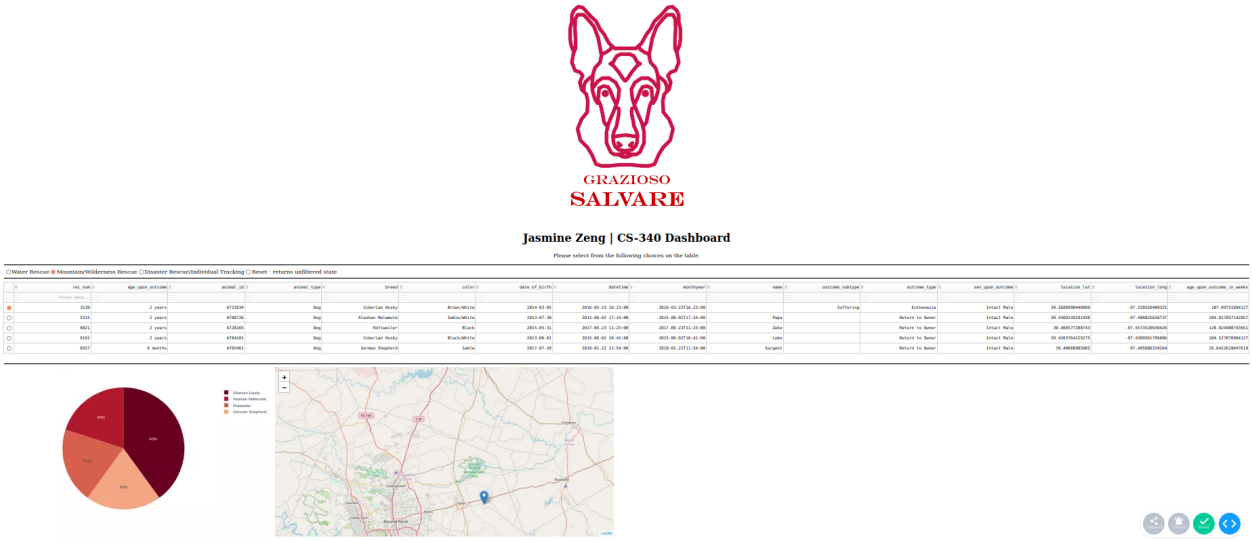
The screenshots below represent each rescue filter and their correlating data in the respective order.

Water Rescue

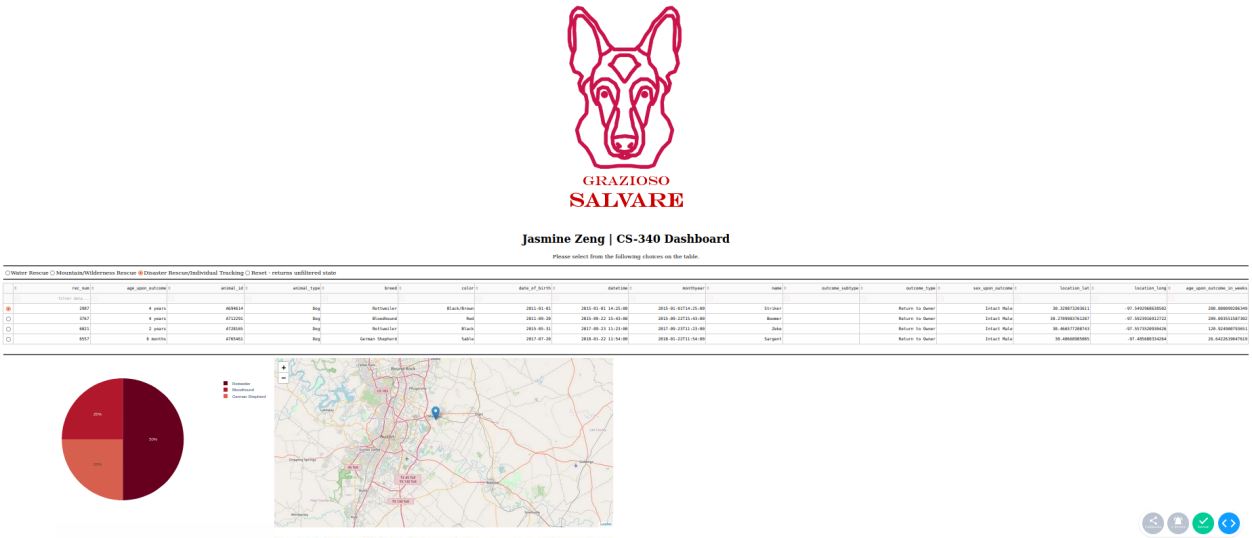


Note: This template has been adapted from the following sample templates: [Make a README](), [Best README Template](), and [A Beginners Guide to Writing a Kickass README]().

Mountain or Wilderness Rescue

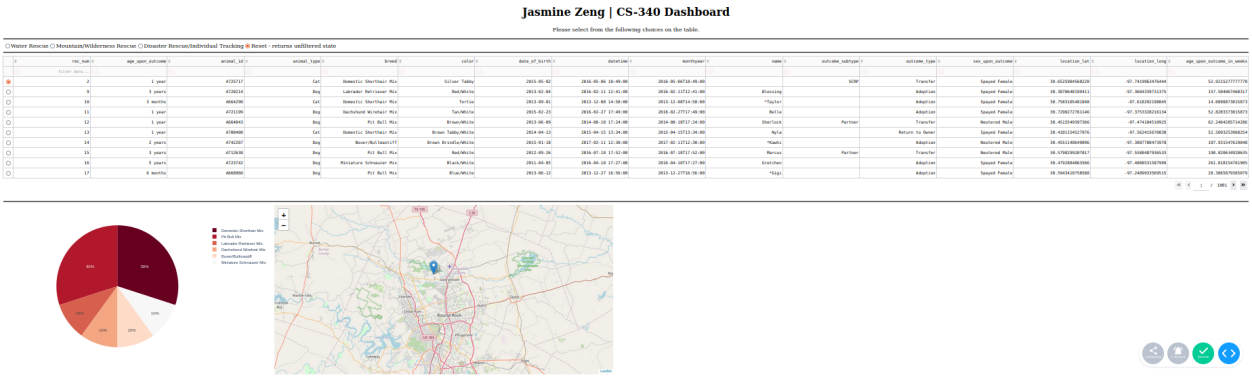


Disaster or Individual Tracking



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.

Reset



Challenges

One major challenge of the Animal Shelter project was ensuring the connection between the interface and the server. We addressed this by using MongoDB to provide stable and secure access to the databases. Paying attention to Python syntax is essential as well. To simplify the programming experience, it is recommended that developers assign account credentials and port information to variables in order to avoid code redundancy.

Contact

Name: Jasmine Zeng
Email: jasmine.zeng@snhu.edu

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.