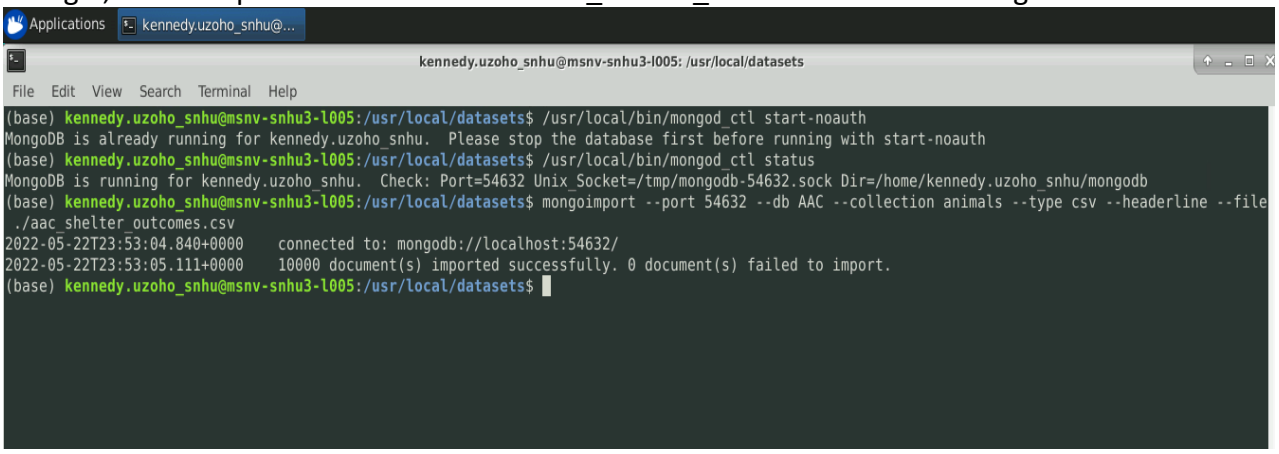
Kennedy Uzoho

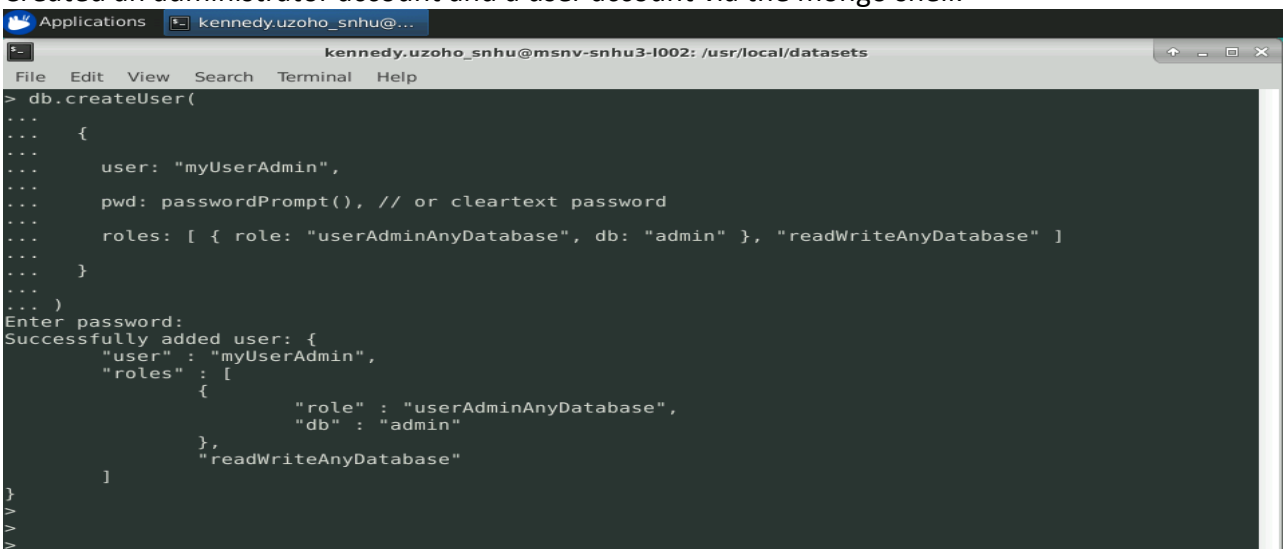
CS 340 Client/Server Development

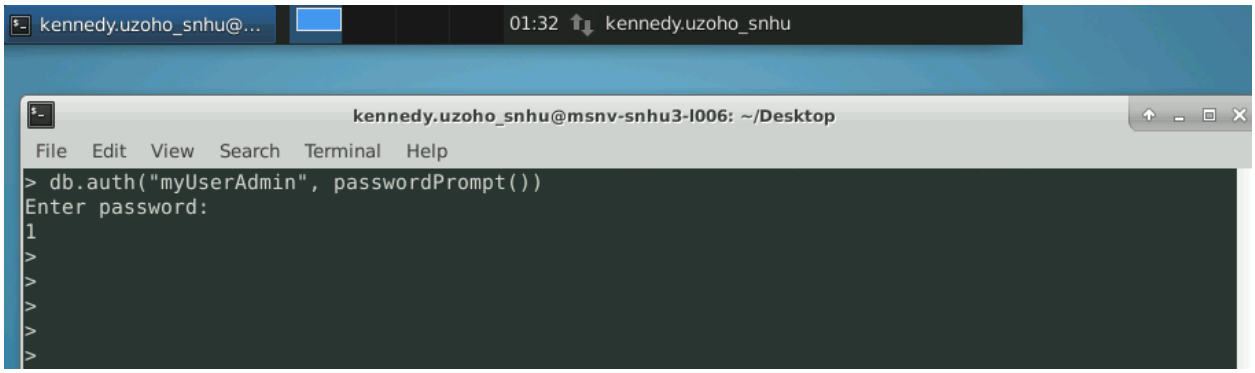
The purpose of this project is to create a database in MongoDB that can interact with client-side code. The full-stack project is to help create software that includes a database and a client-facing web application dashboard that can work with existing data from animal shelters to identify and categorize available dogs.

1. To begin, I have imported a CSV file called aac_shelter_outcomes.csv into MongoDB



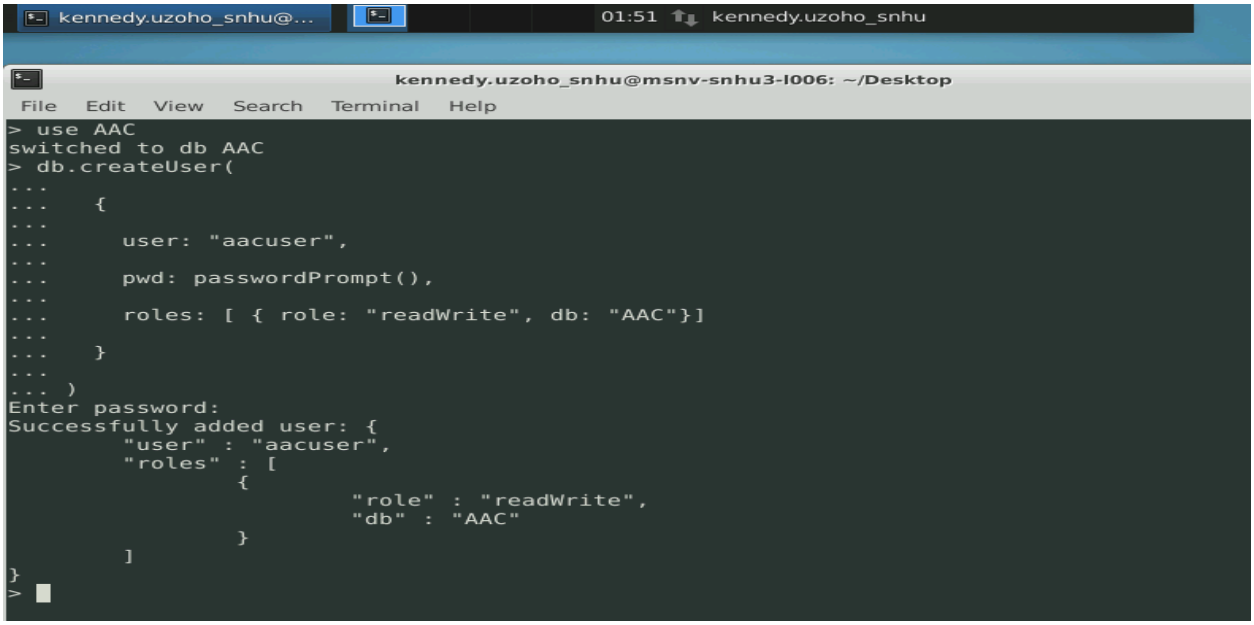
2. Created an administrator account and a user account via the mongo shell.





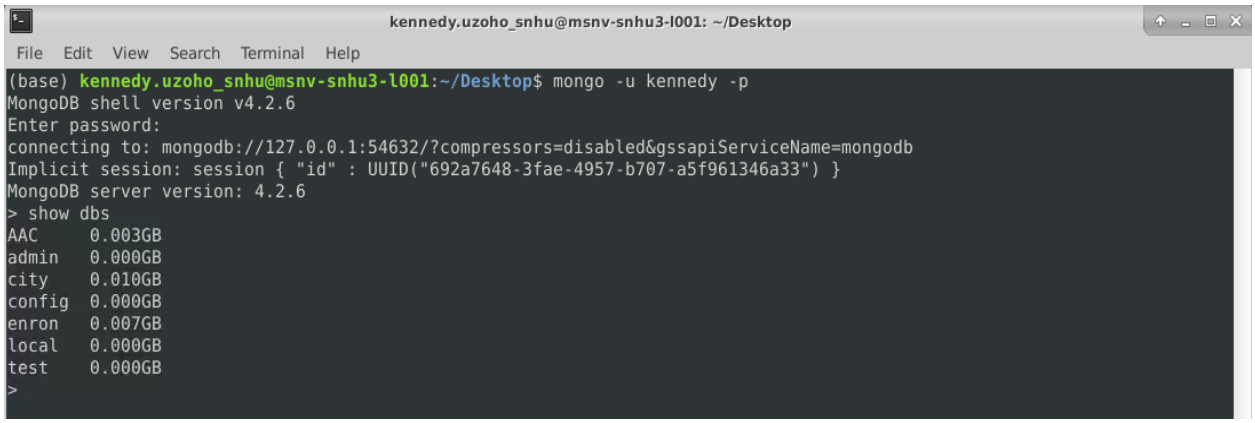
Note: I later changed my Admin username from “myUserAdmin” to “kennedy”

Created a new user account called “aacuser” for the database AAC via the mongo shell.



Log in screenshots.

Admin, “kennedy”



A user account, “aacuser"

3. Here is the PyMongo class that uses object-oriented programming methods to enable CRUD functionality in the database. This module supports code reusability, so other Python scripts can import this Python code as a module.

Animal_shelter.py

```python
from pymongo import MongoClient
from bson.objectid import ObjectId


class AnimalShelter(object):
    """ CRUD operations for Animal collection in MongoDB """

    def __init__(self, username, password):
        # Initializing the MongoClient. This helps to
        # access the MongoDB databases and collections.
        self.client = MongoClient(
            'mongodb://%s:%s@127.0.0.1:54632/AAC' % (username, password)
        )
        self.database = self.client["AAC"]

    # method to implement the C in CRUD
    def create(self, data):
        # Checks to see if the data is None or empty and returns False in either case
        if data is not None:
            if data:
                self.database.animals.insert_one(data)
                return True
        else:
            return False

    # method to implement the R in CRUD
    def read(self, search):
        # Checks to see if the data is None or empty and returns an exception in either case
        if search is not None:
            if search:
                search_result = self.database.animals.find(search)
                return search_result
        else:
            exception = "Nothing to search, because the search parameter is empty."
            return exception

    def update(self, query, new_data):
        # Method to implement the U in CRUD
        if query is not None and new_data is not None:
            if query and new_data:
                self.database.animals.update_one(query, {"$set": new_data})
                return True
            else:
                return False

    def delete(self, query):
        # Method to implement the D in CRUD
        if query is not None:
            if query:
                self.database.animals.delete_one(query)
                return True
            else:
                return False


```

4. Created a Python test script that imports the above CRUD Python module (animal_shelter.py) to call and test all instances of CRUD functionality.

test_script

In [9]:

```python
# import AnimalShelter class from animal_shelter
from animal_shelter import AnimalShelter

# define data
data = {"age_upon_outcome": "2 year",
        "animal_id": "A634647",
        "animal_type": "Dog",
        "breed": "Pitbull",
        "color": "Black",
        "date_of_birth": "2019-09-04",
        "datetime": "2019-09-04 10:49:00",
        "monthyear": "2019-09-04T10:49:00",
        "name": "Sari",
        "outcome_subtype": "SCRP",
        "outcome_type": "Transfer",
        "sex_upon_outcome": "Female",
        "location_lat": 30.6525984560228,
        "location_long": -97.7419963476444,
        "age_upon_outcome_in_weeks": 52.9215277777778}

# define search criteria
search = {"animal_id": "A634647"}

# instantiate an object of AnimalShelter class
assignment = AnimalShelter('aacuser', 'password')

# call the create method
success = assignment.create(data)
print(success)

# call the read method
results = assignment.read(search)
print(results)

# call the update method
new_data = {"name": "Sari 2.0"}
update_success = assignment.update(search, new_data)
print(update_success)

# call the read method again to check the updates
updated_results = assignment.read(search)
print(updated_results)


```

```
True
<pymongo.cursor.Cursor object at 0x7f89300a7da0>
True
<pymongo.cursor.Cursor object at 0x7f8934be7470>
```

In [10]:
```
# call the read method
results = assignment.read(search)

# convert the cursor to a list
results_list = list(results)

# loop through the list to access individual documents
for result in results_list:
    print(result)
```


```
{'_id': ObjectId('63d7f2f03a6c8ff61ee8fd64'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari 2.0', 'outcome_subtype': 'SCRP', 'outcome_type': 'Tra
nsfer', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'a
ge_upon_outcome_in_weeks': 52.92152777777778}
{'_id': ObjectId('63d7f2fb3a6c8ff61ee8fd66'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari', 'outcome_subtype': 'SCRP', 'outcome_type': 'Transfe
r', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_u
pon_outcome_in_weeks': 52.9215277777778}
{'_id': ObjectId('63df72160a2fbfb912085099'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari', 'outcome_subtype': 'SCRP', 'outcome_type': 'Transfe
r', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_u
pon_outcome_in_weeks': 52.9215277777778}
{'_id': ObjectId('63df72310a2fbfb91208509b'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari', 'outcome_subtype': 'SCRP', 'outcome_type': 'Transfe
r', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_u
```


In [12]:
```
delete_success = assignment.delete(search)
print(delete_success)
```

```
True
```

In [13]:
```
# call the read method
results = assignment.read(search)

# convert the cursor to a list
results_list = list(results)

# loop through the list to access individual documents
for result in results_list:
    print(result)
```


```
{'_id': ObjectId('63d7f2fb3a6c8ff61ee8fd66'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari', 'outcome_subtype': 'SCRP', 'outcome_type': 'Transfe
r', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_u
pon_outcome_in_weeks': 52.9215277777778}
{'_id': ObjectId('63df72160a2fbfb912085099'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari', 'outcome_subtype': 'SCRP', 'outcome_type': 'Transfe
r', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_u
pon_outcome_in_weeks': 52.9215277777778}
{'_id': ObjectId('63df72310a2fbfb91208509b'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari', 'outcome_subtype': 'SCRP', 'outcome_type': 'Transfe
r', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_u
pon_outcome_in_weeks': 52.9215277777778}
{'_id': ObjectId('63df781e0a2fbfb91208509d'), 'age_upon_outcome': '2 year', 'animal_id': 'A634647', 'animal_ty
pe': 'Dog', 'breed': 'Pitbull', 'color': 'Black', 'date_of_birth': '2019-09-04', 'datetime': '2019-09-04 10:4
9:00', 'monthyear': '2019-09-04T10:49:00', 'name': 'Sari', 'outcome_subtype': 'SCRP', 'outcome_type': 'Transfe
r', 'sex_upon_outcome': 'Female', 'location_lat': 30.6525984560228, 'location_long': -97.7419963476444, 'age_u
```