# CS 340 README

## About the Project/Project Title

The project is an application that allows users to access files from a local mongo database and to create different users to manipulate these files. The project allows a user to create, read, update, and delete documents from the AAC database, which these documents are animals’ information that are being held at a shelter.

## Motivation

This project exists because it is an assignment for my CS 340 at SNHU as Project 1. These assignments are there to help me develop my skills and knowledge in working with a database. Working with this database, I have learned how to create different users with access to the database. I have also learned the steps of the CRUD by being able to create, read, update, and delete documents within this database.

## Getting Started

1. The first step is to enter Mongo and then import the aac\_shelter\_outcome.csv file into the AAC database and animals collection within your local port by using mongoimport. After import, verify the documents were imported and check if both the database and collection were created.
2. The second step is to create a simple and compound index for these documents within the AAC database.
3. The third step is to create an admin and user account to allow the user to readWrite the documents within AAC.animals.
4. The fourth step is to check if your newly created login information works in order to manipulate the documents within the AAC database.
5. The fifth step is to pip install pymongo to make sure its installed and up to date.
6. The six step is to create both ipynb and py files to use the CRUD code and main, but also be able to test these codes.

## Installation

The tools needed to run this program are Python 3 and MongoDB installed.

Python 3 is needed because the files used for this project are created as .py and .ipynb files.

MongoDB needs to be installed because it is used to access the entire project.

## Usage

Please see below to see code examples, test information, and screenshots for this project.

### Code Example

Enter mongo by using a line of code like this example: /usr/local/bin/mongod\_ctl start-noauth. Next, the aac\_shelter\_output csv has to be imported into the database of AAC and the collection of animals. Using a line of code like this example code, mongoimport - -port 44663.sock Dir=/home/1696594\_snhu/mongodb

While using admin database, create admin account and give the admin the ability to readWrite using a code like this example: db.createUser( {user: “J.Silva”, pwd: “password”, roles: [ { role: “userAdminAnyDatabase”, db: “admin”}, “readWriteAnyDatabase” ]}). Once this admin is created, an output of a successful creation should be displayed to acknowledge the admin was successful with the information pertaining to the admin. The same type of code can be used to create a user account for the AAC database, but the code will have to be modified in certain areas.

After each account is created, it can be tested by exiting from mongo and re-logging into mongo by using a line of code like this example: mongo - -authenticationDatabase “admin” -u “J.Silva” -p . After this code is written, a request for the admin user’s password will be displayed. Type in your password (side note the password will not be displayed as you type it out). After entering the password, you will re-enter into mongo using the user account. An example code to enter mongo as the user “aacuser” is the following: mongo -u “aacuser” - -authenticationDatabase “AAC” -p

### Tests

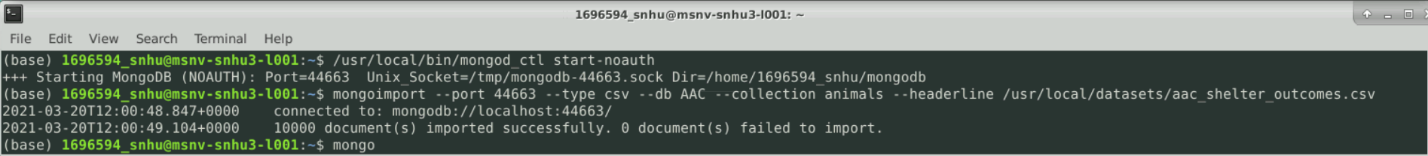
Looking at the main.ipynb file, I imported the animal\_shelter class AnimalShelter into this test. The test was able to show a display of when a document was successful created and when a document failed to be created. The same was done when a document was successful displayed to be read and when a document failed to display.

* An example of the line of code to display the failure to create a document is the following: print(animals.create( {0:0} )) and the output was an invalid document because must only have string keys and the key was 0. This is basically saying the entry for the document was invalid because strings were invalid.
* The same information was displayed for a failed reading of a document. The output for the invalid read document was the same as the invalid created document. An example of the line of code to display the failure to create a document is the following: print(list(animals.read( {0:0} ))).
* For an invalid document updated function, a large error was displayed to show the failure to delete a document. An example of code for a failed document was the following: print(animals.update({0:0})
* For an invalid document deleted function, a large error was displayed to show the failure to delete a document. An example of code for a failed document was the following: print(animals.delete({0:0})

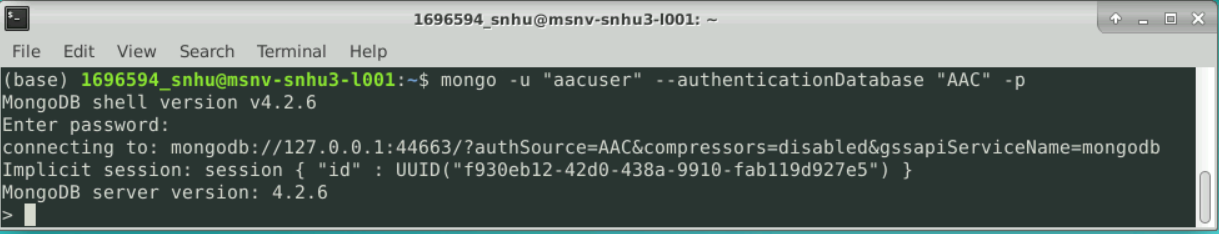
For the successful CRUD functions of a document, screenshot examples were placed in the heading below. Also, successful import and log in accounts were also shown in screenshots below.

### Screenshots

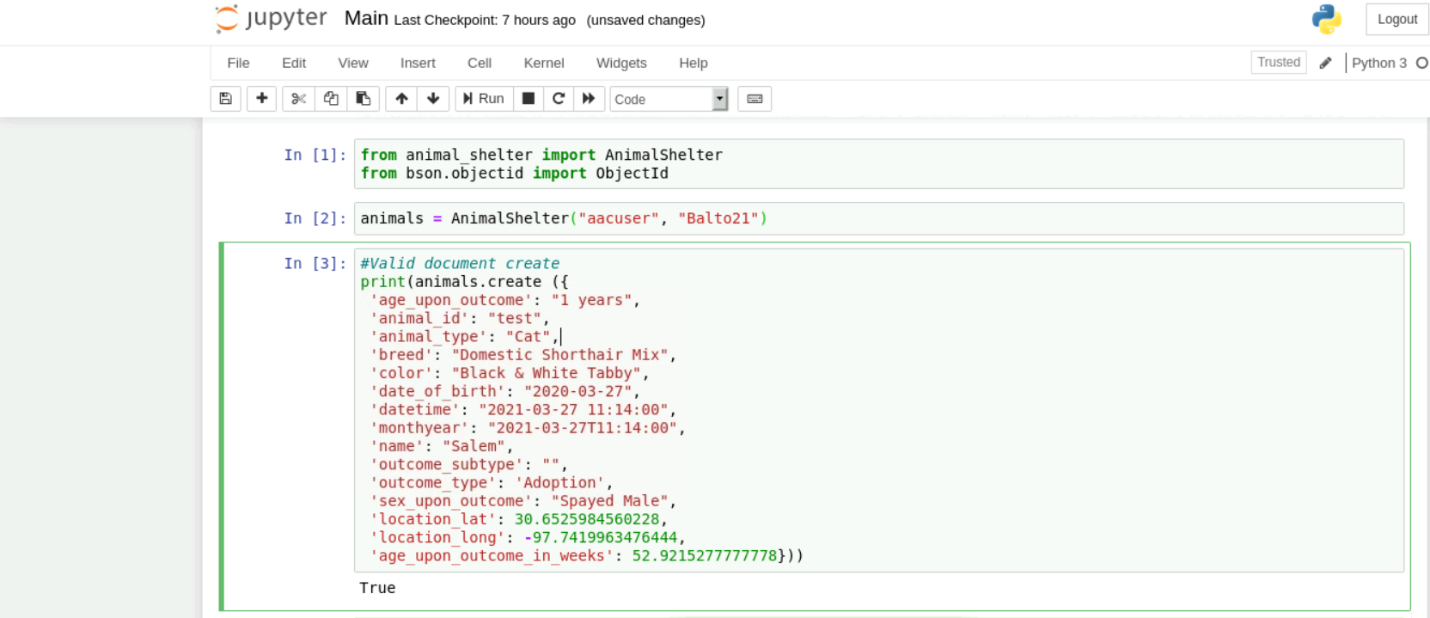
Example of a successful import of the csv file into the AAC database and animals collection.

****

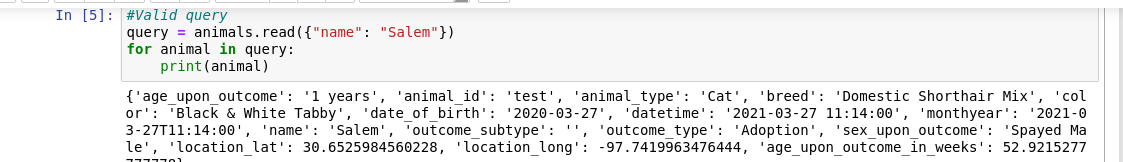
Example of a successful login using newly created account



Example of a successful created document



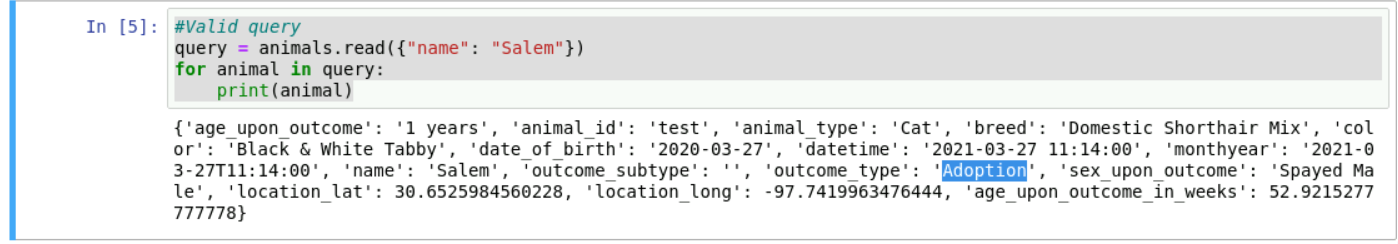
Example of a successful read document

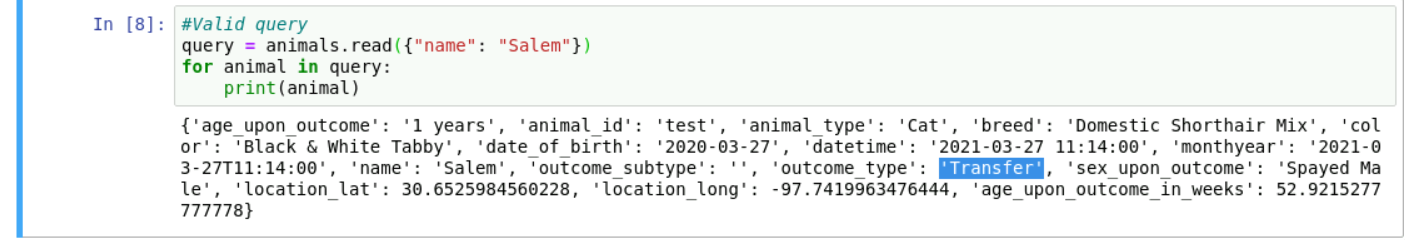


Example of a successful updated document



(Please see blue highlight in the next two screenshots to view the difference from the original query to the updated query)





Example of a successful deleted document



## Contact

Joseph Silva Jr.