# Grazioso Salvare README

## About the Project/Project Title

Global Rain was tasked to develop an open-source, full-stack, system for an international rescue-animal training company, Grazioso Salvare. Grazioso Salvare partners with local shelters in Austin, TX to find candidates to enroll in their search-and-rescue training programs, such as water rescue, mountain or wilderness rescue, and disaster rescue. By connecting to these shelters’ databases, this system aids in quickly identifying dogs by searching for specific qualities, such as breed, gender, and age. Using a CRUD python module, this system allows users to create, read, update, and delete documents. Also, a web dashboard is available to easily filter through the databases and find the animals’ locations.

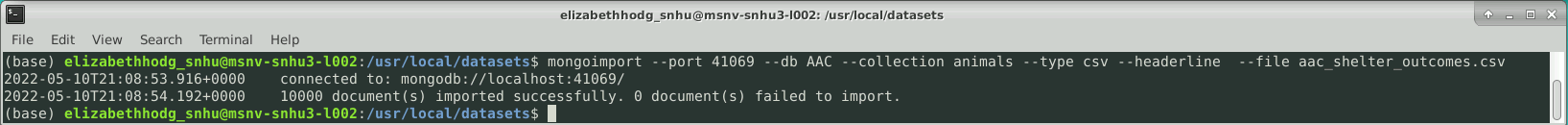
## Motivation

This system was developed to support an organization that saves dogs from being possibly euthanized and trains them to save peoples’ lives. The system will make it easier and faster for Grazioso Salvare to find trainee candidates, resulting in more lives being saved. Also, by being an open-source system, other organizations can edit this system to utilize for their different needs.

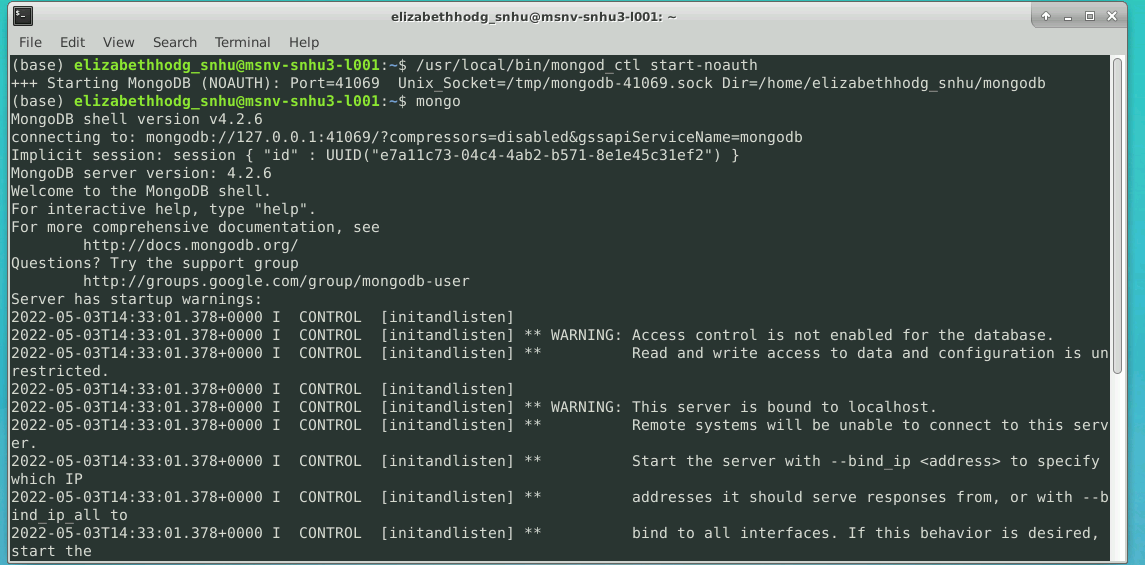
## Getting Started

Follow these steps to get a local copy of the project running:

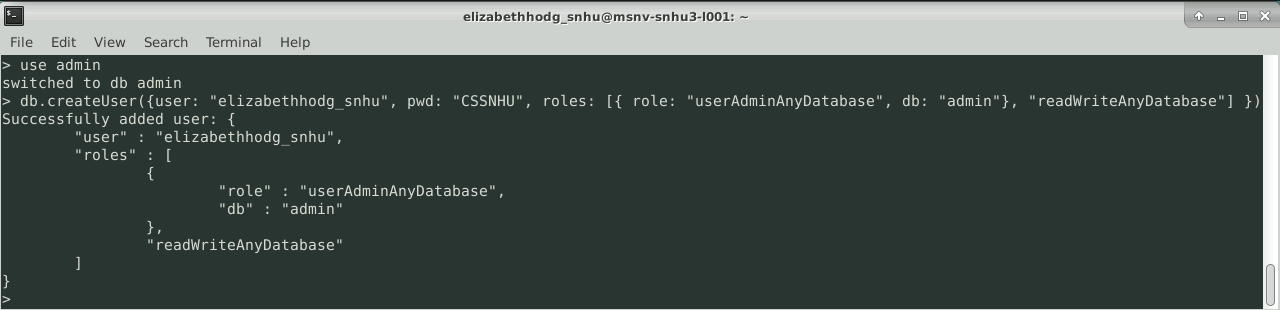
1. Download Shelter\_CRUD.py
2. Open terminal. Then, input and open the database directory (example: cd /usr/local/datasets)
3. Import desired database into MongoDB



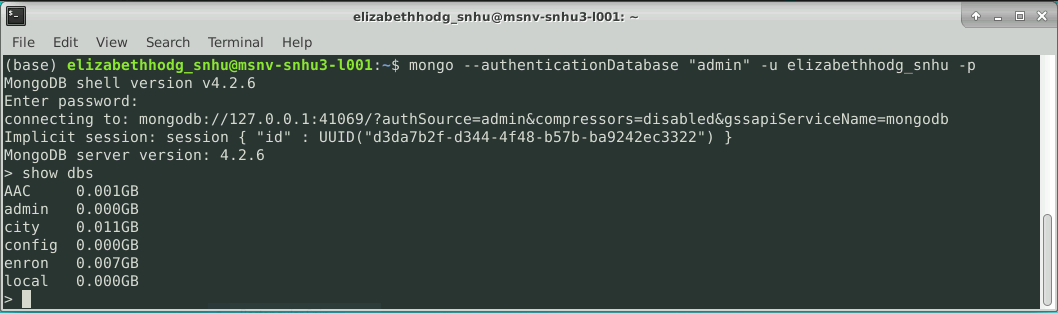
1. Start MongoDB with noauth



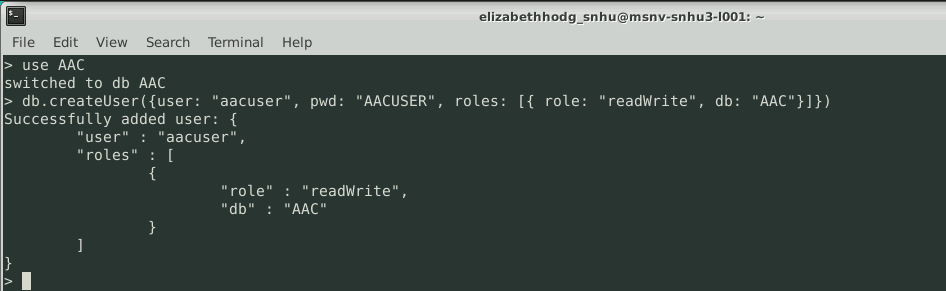
1. Use admin database and create admin account



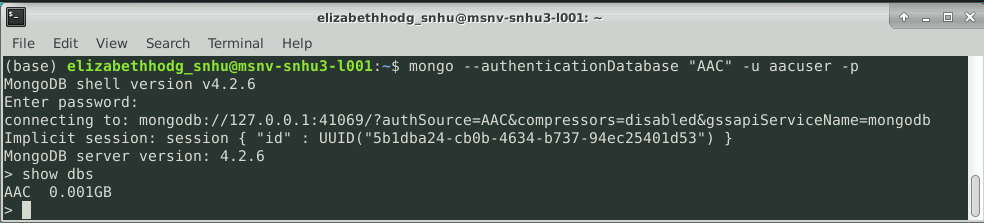
1. Exit Mongo, then restart Mongo using Admin account



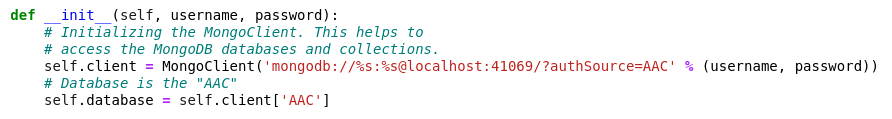
1. Use desired database and create database user



1. Exit Mongo, then restart Mongo by using database user account



1. Use desired database (This example shows “use AAC”)
2. Open Shelter\_CRUD.py in desired Python IDE
3. Edit Shelter\_CRUD.py to work with your database and Mongo port ( This example shows port #41069 and database “AAC”)



1. Open Jupyter Notebook and upload both Shelter\_CRUD.py and ProjectTwoDashboard.ipynb
2. From this point, you will be able to utilize the CRUD functions with the database and open the Dashboard

## Installation

For this project, the user must have Python installed to run the Shelter\_CRUD.py and ProjectTwoDashboard.ipynb files. Also, MongoDB is used to access the databases to run with this program. We chose to utilize MongoDB for its database flexibility, scalability, and querying system. Next, MongoDB provides a specific Python driver, called PyMongo, making it easier to establish a connection. Furthermore, this project utilizes some PyMongo libraries, such as ObjectId and MongoClient, to access the database through MongoDB. Lastly, Jupyter Notebook is needed to run the files. Below are the links to get all the necessary tools installed.

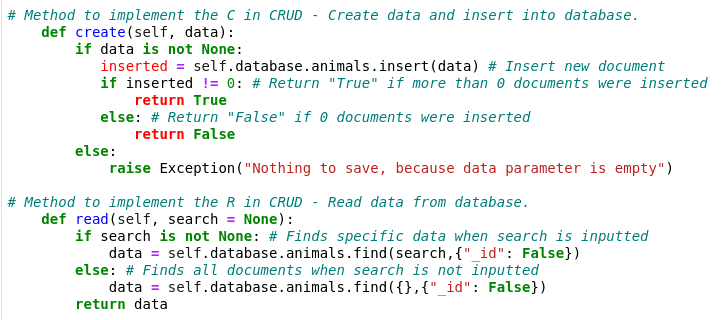
1. Python
   * Follow link if Python install is needed [Download Python | Python.org](https://www.python.org/downloads/)
2. MongoDB
   * Follow link if install is needed [Install MongoDB](https://www.mongodb.com/docs/guides/server/install/)
3. PyMongo Driver
   * Follow link if driver install is needed [pymongo · PyPI](https://pypi.org/project/pymongo/)
4. Pip Package (Needed for Jupyter Notebook)
   * Follow link if Pip Package install is needed [Installing Packages — Python Packaging User Guide](https://packaging.python.org/en/latest/tutorials/installing-packages/)
5. Jupyter Notebook
   * Follow Link if install is needed [Project Jupyter | Installing Jupyter](https://jupyter.org/install)

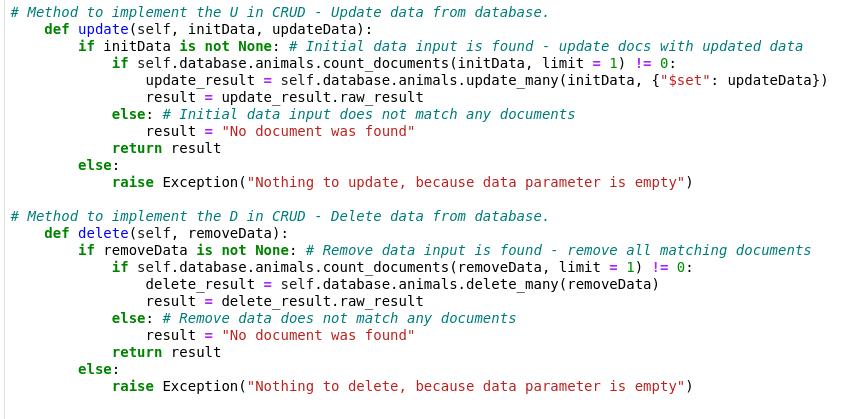
## CRUD Usage

By using the Shelter\_CRUD module, users can easily insert, search, edit, and delete documents from the databases. By going through the installation and “Getting Started” instructions, the CRUD module can get used and tested through Jupyter Notebook.

### Code Example

These are the CRUD functions, Create, Read, Update, and Delete, from the Shelter\_CRUD.py module



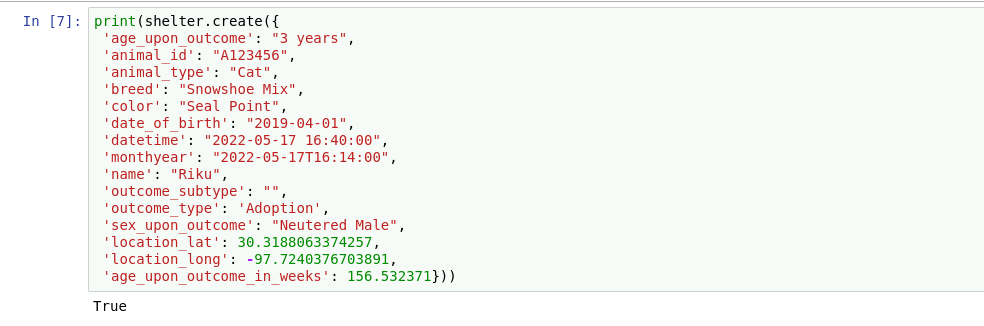
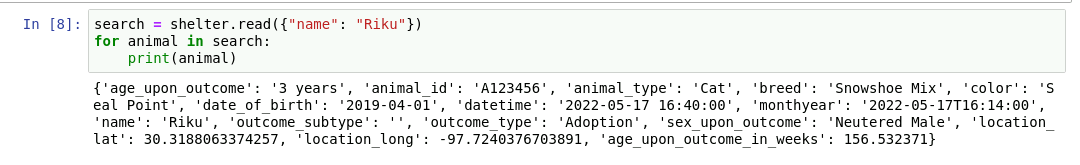
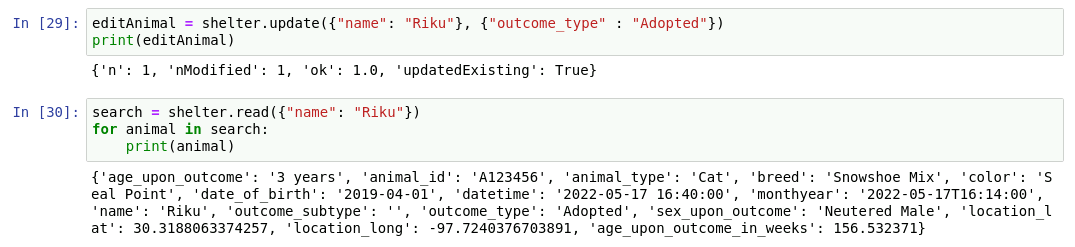


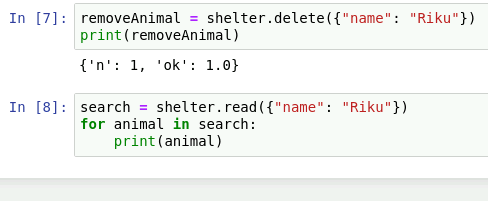
### CRUD Use and Tests

Users can use and test the CRUD functions by creating a new Python 3 test file in Jupyter Notebook. Follow these steps to utilize the CRUD functions.

1. Import Shelter CRUD and use authentication in test file



1. Create a new document 
2. Read a document 
3. Update a document 
4. Delete a document (Since the animal is no longer in the database, nothing is printed)



To Run the CRUD functions in Jupyter Notebook, click “Cell” and select “Run All” Graphical user interface, text, table

Description automatically generated

### Open Dashboard

Follow these instructions to run the dashboard using Jupyter Notebook.

1. Input the proper user authentication for the database (user that was created above)

Graphical user interface, text, application

Description automatically generated

Once desired user is connected, run the dashboard in Jupyter by clicking on the cell and selecting the “Run” button

Graphical user interface, text, application

Description automatically generated

Once the dashboard is opened, scroll down, and click on the “Open in new window” button to view it in a separate window.

Graphical user interface, text, application, email

Description automatically generated

In the new window, the user can utilize the dashboard easier. Also, the Logo is clickable to bring the user to the client’s home page, [www.snhu.edu](http://www.snhu.edu).

A picture containing Word

Description automatically generated

Above, the filter options can be seen. When clicked, the database becomes filtered, and the charts are updated. Next, to view a specific animal, the user can click on a row. This action shows the user where that animal is located on the map.

**All Filter**

**Map

Description automatically generated**

**Water Rescue Filter**

**Map

Description automatically generated**

**Mountain Rescue Filter**

**Graphical user interface, map

Description automatically generated**

**Disaster Rescue Filter**

**Map

Description automatically generated**

## Project Challenges:

The biggest challenge for me was getting the dashboard to run properly. I hit a lot of roadblocks during this project, such as getting the charts to display. Though, after some research and digging in the resources, I was able to figure out the issues. For example, one problem I was having was that my dashboard would crash when I had an animal selected and when I changed the filter. Though, I found out it had to do with the map callback. I was using derived\_viewport\_data and derived\_viewport\_selected\_rows as the inputs, instead of derived\_virtual. After changing the callback inputs, the dashboard was able to function properly without crashing.

## Contact

Elizabeth Hodgman

Elizabeth.Hodgman@snhu.edu