# CRUD Database Application for Grazioso Salvare

**Contents**

[**CRUD Database Application for Grazioso Salvare 1**](#_Toc127997763)

[**About the Project/Project Title 1**](#_Toc127997764)

[**Motivation 1**](#_Toc127997765)

[**Getting Started 2**](#_Toc127997766)

[**Installation 3**](#_Toc127997767)

[**Usage 5**](#_Toc127997768)

[**Code Example 5**](#_Toc127997769)

[**Tests 7**](#_Toc127997770)

[**Screenshots 7**](#_Toc127997771)

[**Roadmap/Features (Optional) 10**](#_Toc127997772)

[**Contact 11**](#_Toc127997773)

## About the Project/Project Title

This is a CRUD (Create, Read, Update, Delete) application that is currently configured to run for a database that is created in MongoDB for the company, Grazioso Salvare. The current configuration shows you all items that have been populated into a table as information. The information included is a Table, a Pie Chart, and a Map. These are completely interactive.

## Motivation

Grazioso Salvare, an international rescue-animal training company, is seeking a solution to help identify possible candidates for search-and-rescue training dogs. These dogs, when trained, will have the ability to find and help humans or other animals, which are often in life-threatening conditions. As part of a deal that Grazioso Salvare has made with a non-profit agency, I have been provided with a CSV file that contains a list of animals that are in five animal shelters in the region around Austin, Texas.

Particularly, Grazioso Salvare is looking for dogs with certain profiles to train. As an example, dogs that are less than two years old are easier to train for search-and-rescue. Certain breeds of dogs are proficient at different rescue types (water rescue, mountain/wilderness rescue, disaster rescue, and scent tracking for specific humans).

With this CRUD application, we can navigate any database. We are just specifically using this one for the company, Grazioso Salvare. We will be using this database to find the most suitable dogs for the search-and-rescue team.

## Getting Started

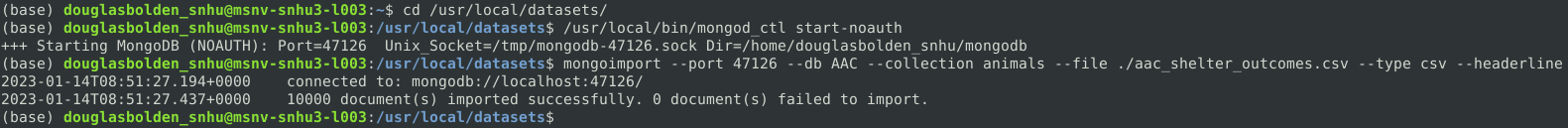
To get started, you will need to make a new database in MongoDB. After having done so, you will need to create user accounts. This will ultimately lead you to be ready to run the script that produces what is shown in the screenshot below.

Once you have the database setup, you will use this application to connect to your database and run the function you would like to run. Information on this below.

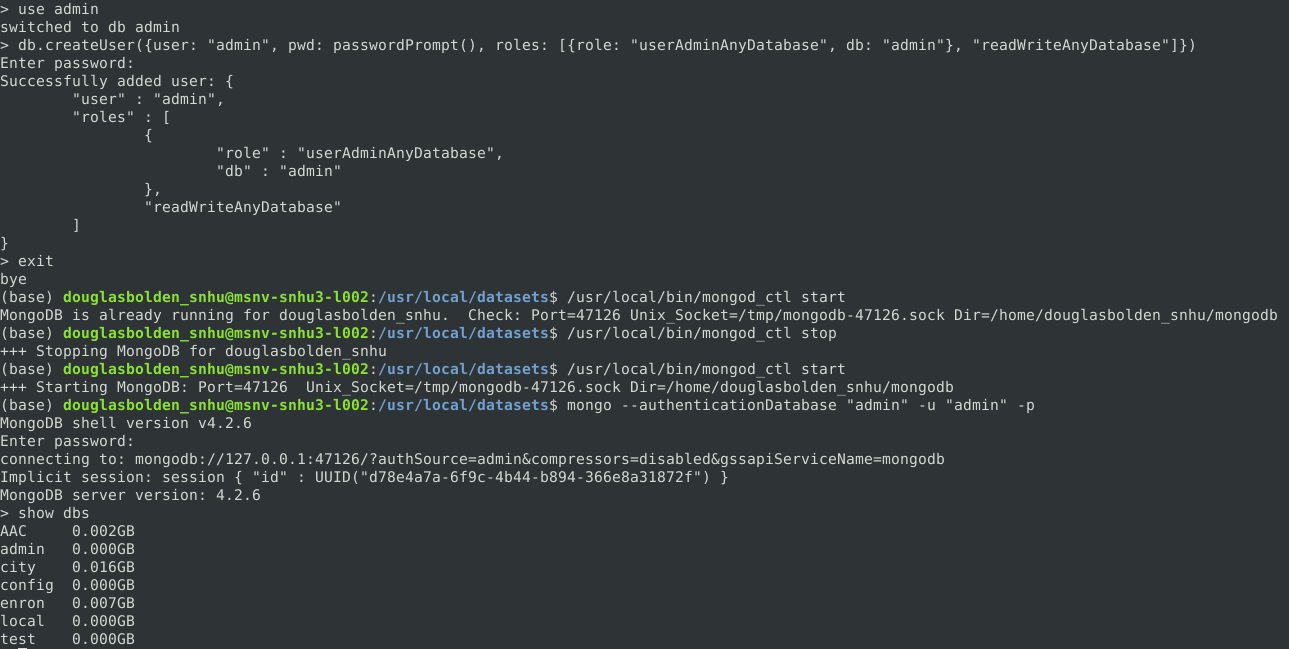
The purpose for using Python for this application is solely based on the languages exceptional number of libraries and ability to pull data from databases at a speedy rate.

BELOW IS SCREENSHOTS OF WHAT SHOULD BE PUT IN MONGODB to start you up.

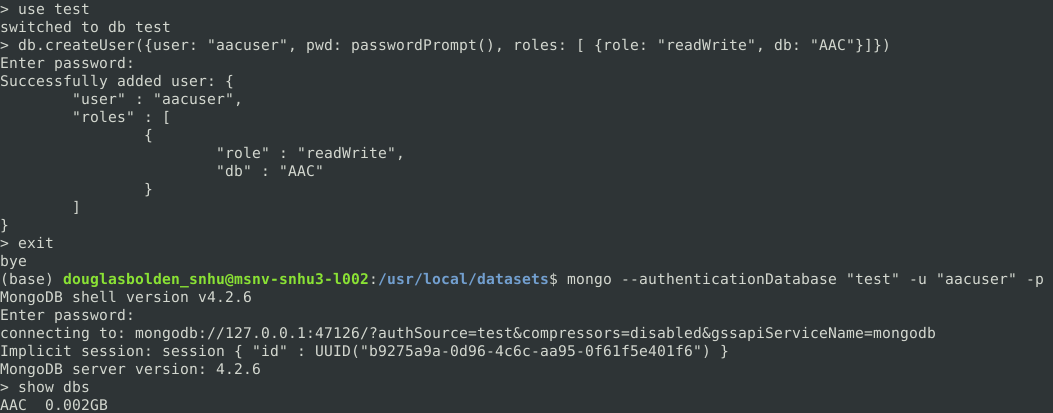
Importing the csv:



Creating Admin account:



Creating a user account:



## Installation

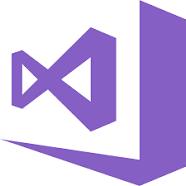
To use this application, you will need to download a few applications. These applications include:



1. <https://www.mongodb.com/>
   1. Learn about MongoDB [HERE](https://www.mongodb.com/docs/manual/tutorial/getting-started/). This will show you some beginner steps to learning some simple MongoDB shell commands.
2. You will need an IDE for Python. This could be:

 PyCharm

* 1. <https://www.jetbrains.com/pycharm/>

Visual Studio

* 1. <https://visualstudio.microsoft.com/>

IDLE

* 1. <https://docs.python.org/3/library/idle.html>

Jupyter Notebook

1. Optionally, [Jupyter Notebook](https://jupyter.org/).

## Usage

### Code Example

In the animal\_shelter.py file:

1. If you want to disable authentication change the \_\_init\_\_ function’s passed variables and uncomment the self.client with no variables attached:
   1. def \_\_init\_\_(self)
   2. self.client = MongoClient('mongodb://localhost:47126')
   3. “47126” Is the port number that mongoDB started on for my current instance. You will find this in the shell when you start your server. YOU WILL NEED TO CHANGE THIS NUMBER IF YOU WANT YOUR APPLICATION TO WORK.
2. Change the name of the database to the one that you want to use. It will be currently using a database labeled as AAC:
   1. self.database = self.client['AAC']

Using the user-defined script.py file:

1. First, you need to import the class from the library:
   1. from animal\_shelter import AnimalShelter
2. Second, you will have to instantiate a new AnimalShelter object and allow authentication for your user account accessing the database:
   1. shelter = AnimalShelter("aacuser", "password")
3. Functions available:
   1. create()
      1. Creates new entries in the database. Prints “Error Saving Entry.” If something is wrong.
      2. **Example:** shelter.create(data)
   2. read()
      1. Pulls an entry from the database and showcases all the information associated with it. Prints “Nothing Found.” If something is wrong.
      2. **Example:** shelter.read(data)
   3. update()
      1. Updates an entry (the first passed variable, *data*) with new information. (the second passed variable, *newData*) Prints “No Data Submitted.” If something is wrong.
      2. **Example:** shelter.update(data, newData)
   4. delete()
      1. Deletes one entry from the database. Prints “No Data Submitted.” If something is wrong.
      2. **Example:** shelter.delete(data)

Using the Grazioso Salvare.py file:

\*\*\*\*There are many imports at the top of this file. They include all the essential imports needed for the program to work correctly.

1. First, you need to import the class from the library:
   1. from animal\_shelter import AnimalShelter
2. Second, you will have to instantiate a new AnimalShelter object and allow authentication for your user account accessing the database:
   1. In this file, there are two variables, *username* and *password*. Change these values to the username and password combination that you want to use.
   2. shelter = AnimalShelter(username, password)
3. Populate the Table with data and load the image file for use in the file:
   1. df = pd.DataFrame.from\_records(shelter.read({}))

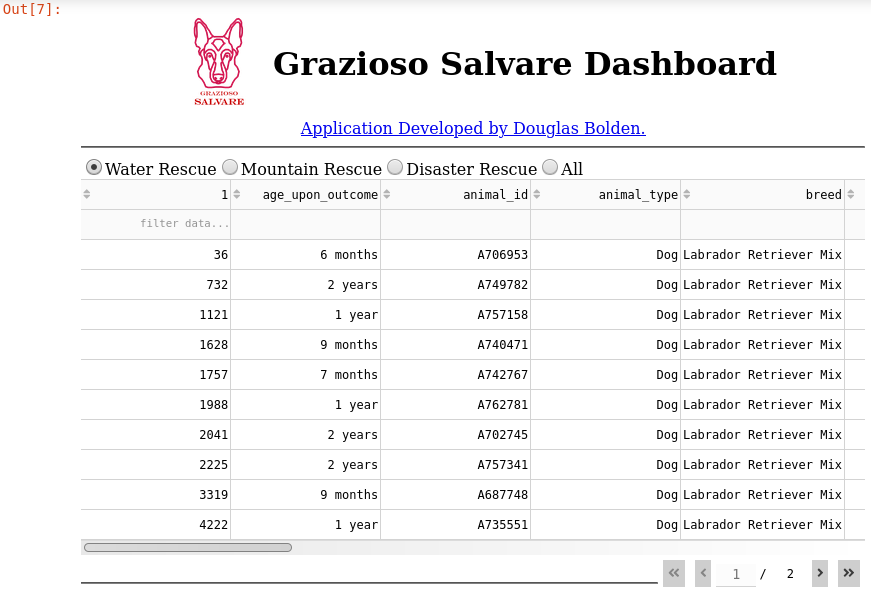
pil\_image = Image.open("assets/Grazioso Salvare Logo.png")

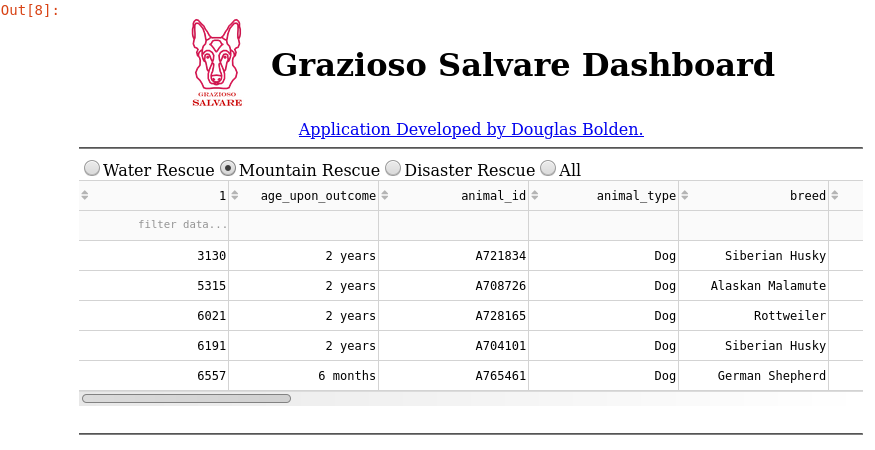
1. Just use my code, and this should be done for you. Very simple and easy to read. I made sure to make readability the number one priority.
2. Run the code.
3. You can interact with the table using the radio buttons above it, you can interact with the pie chart by using the dropdown, and you can interact with the map by clicking the map blip or by clicking and dragging the map.

### Tests

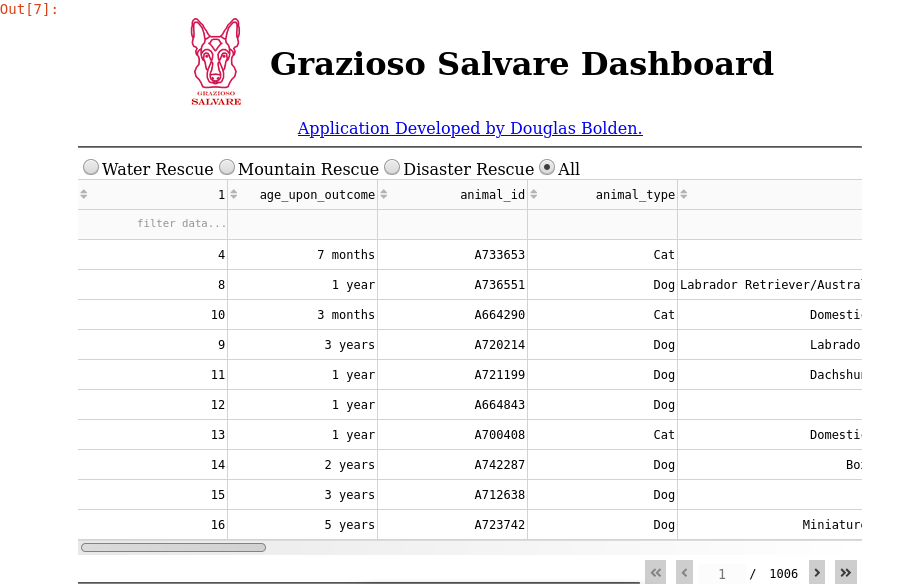
The tests that were ran involved making sure that all pieces that should function, were functioning correctly. IF the code is not workable, the IDE will throw exceptions. Due to errors with my VM, I was unable to confirm the map or pie chart. However; I did follow all methods and functional tests that should be applied to them to make sure that they work, so I am certain that they are good and intact.

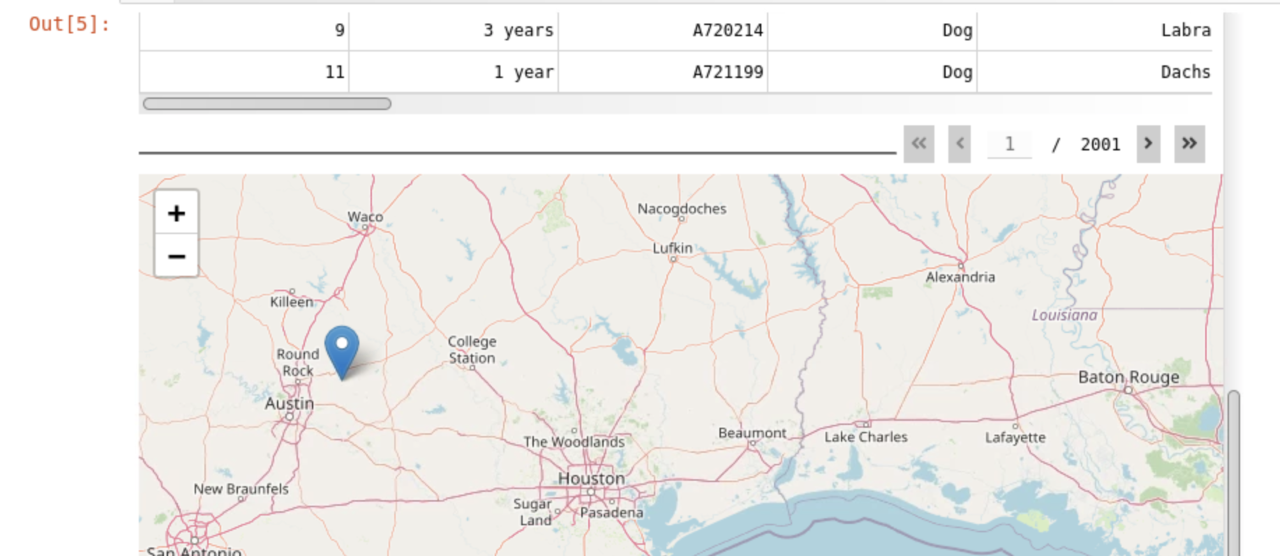
### Screenshots











This should display the first animal in the list. If there is no animal, it should show nothing. I could not see what I was doing, as the VM that I was using was not functioning correctly, and I could not get the technical support that I needed in time.

## Roadmap/Features (Optional)

Features:

* CURRENT:
  + Create Database entries based on passed parameters.
  + Read Database entries based on passed parameters.
  + Check whether an item exists in a database and stop duplicate entries.
  + Update Database entries based on passed parameters.
  + Delete Database entries based on passed parameters.
  + Filter a Table of populated data that was passed in main script.
  + Search a Table of populated data that was passed in main script.
  + Use Radio Buttons to filter populated data that was passed in main script.
  + Filter a pie chart of data that contains all the breeds in the table of populated data.
  + See a map with the current location of entry one from the table of populated data.
* BUGS:
  + Issue when using radio buttons that allows the radio buttons to be used only three times. I am unsure as to why this happens.
  + I can’t see any more information other than the table, as my VM is not working correctly.

## Contact



Douglas Bolden

Phone: +1 (423) 300-2190

Email: [douglas.bolden@ymail.com](mailto:douglas.bolden@ymail.com)

LinkedIn: <https://www.Linkedin.com/in/douglasbolden/>

GitHub: <https://www.github.com/douglasbolden>