# Grazioso Salvare MongoDB Application

## About the Project/Project Title

This application is designed to allow controlled user access to a database to store, create, delete, or update data about animals to be selected for search and rescue training. It also features an interactive dashboard created using the Dash framework to filter and display results and locations of specific animals. Furthermore, it is designed to access from mongoDB via a created Python CRUD module. The results are displayed as a geolocation chart and a bar graph.

## Motivation

## This application will help manage a database of animals that are in several shelters and could be candidates for search and rescue training. Having the information organized and easily accessible will help the organization run more efficiently. MongoDB was selected as it is non-relational and allows for increased flexibility in this scenario vs SQL. We can use Python to create modules to interact specifically with MongoDB and manipulate the data. The Dash framework was chosen as it excels in data visualization. We can take/manipulate data from mongoDB and show geolocation data, data tables and other information in a user-friendly dashboard.

## Getting Started

1. Run MongoDB and import the aac\_shelter\_outcome.csv file, create the database “AAC” and the collection “Animals” following the import of the csv file.
2. Set up an admin account and establish a user for the AAC database (ex. “aacuser”) and ensure database authentication.
3. Create a simple index on the key “breed” and a compound index that will improve the performance of queries looking for breeds that have an “outcome\_type” of “Transfer”.
4. Import the AnimalShelter.py script into a Jupyter Notebooks script.
5. The AnimalShelter script imports pymongo and creates an object that allows initialization of the Mongo client. CRUD functions are then created to properly create and read documents from the collection. The use of pymongo allows for connectivity to mongoDB within other user created applications, easier.
6. Make sure to enter credentials correctly to be able to access and use the DB via the python AnimalShelter script.
7. Use the functionality of the imported python module to interact with the database.
8. With the help of the DASH framework, one can now filter and interact with the application. Radio buttons have been created to filter animals based on specific criteria. Simply check the button to see filtered results or select “reset” to see an unfiltered version of the database on the dashboard.
9. Each radio button was created to display results of “desired dog breeds” based on the recommended “rescue type”.

## Installation

This application is built using a combination of Python, Jupyter Notebooks, Dash, and MongoDB. Python is used in tandem with Jupyter Notebooks to create an importable module that will allow users to create, maintain and manipulate data within a non-relational database (MongoDB). We import the dash framework and work with plotly and other libraries to create a user dashboard and data visualization.

## Usage

### Code Example

The user will have the option to lookup and read an entry (animals.read({“name”:”Name”}) and this returns an iterated cursor object if there is a matching entry. Users will be able to create an entry by inserting dict data animals.create({“breed”:”Dachshund”, “animal\_type”:”dog”,…}) a user will know if it is properly inserted as the function returns a Boolean verifying the action.

### Tests

To search for an entry: print((animals.read({“name”:”Name”}))

To create animals.create({“breed”:”Dachshund”, “animal\_type”:”dog”,…}) will return false if nothing is input.

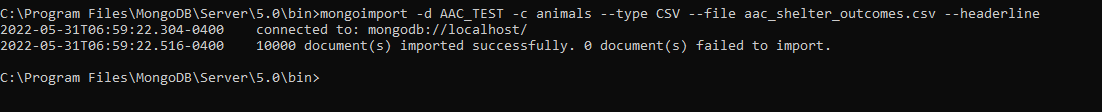
The last screenshot includes the use of the update and delete functions.

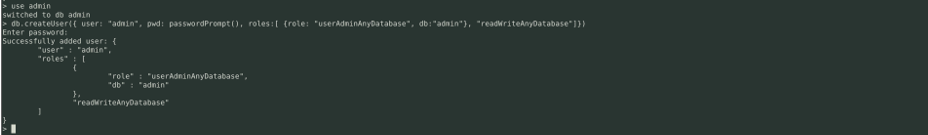
Update contains an extra parameter, the first to find the document to update and the second to input the updates animals.update({“breed”:”Dachshund”}, {“color””:”insert color here”}).

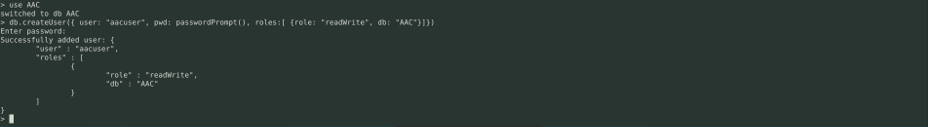
The delete function takes a specified document parameter and returns confirmation of the deleted object, animals.delete({“breed”:”Dachshund”}).

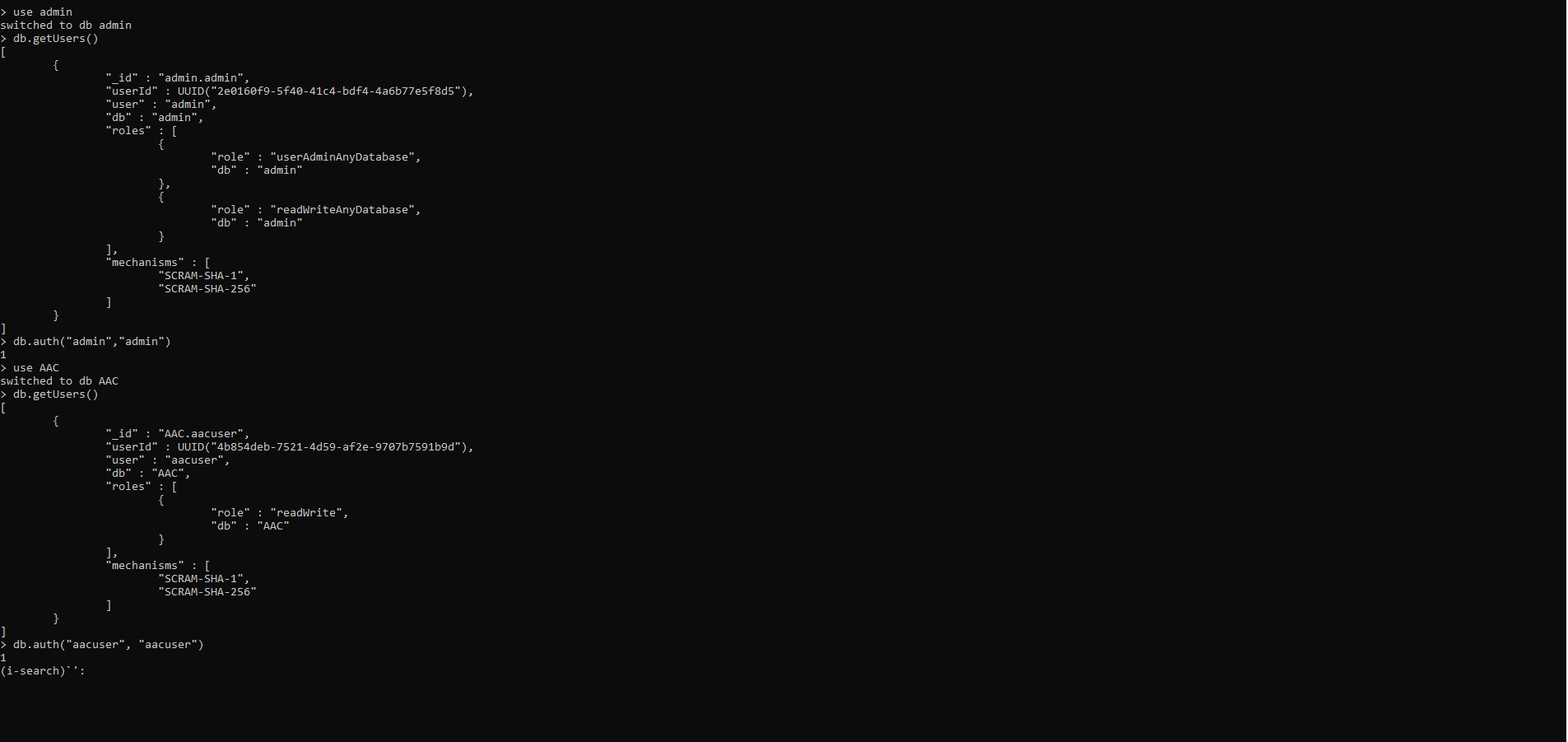
The last 4 screenshots show the created dashboard (from the dash framework) and the functions of the radio buttons to filter the documents in the database.

### Screenshots



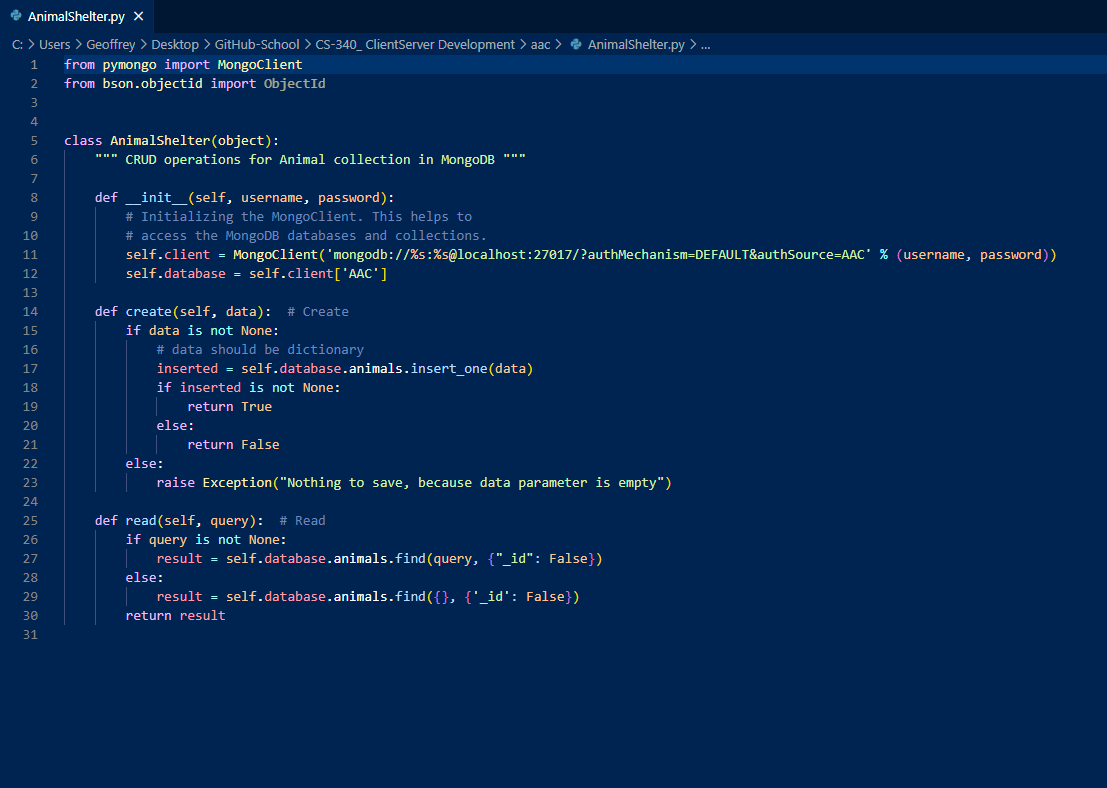






*Graphical user interface, application

Description automatically generated*

**

*Graphical user interface, website

Description automatically generated*

*A picture containing logo

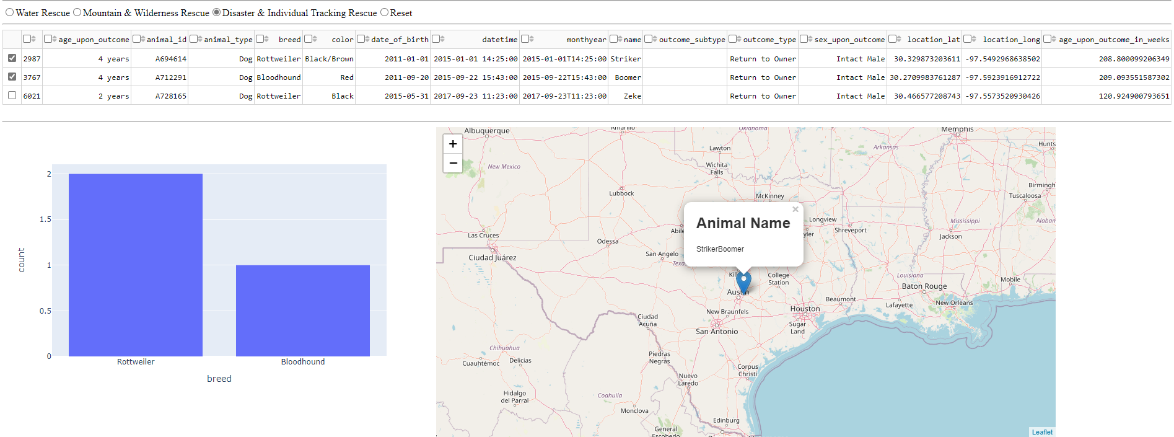
Description automatically generated*

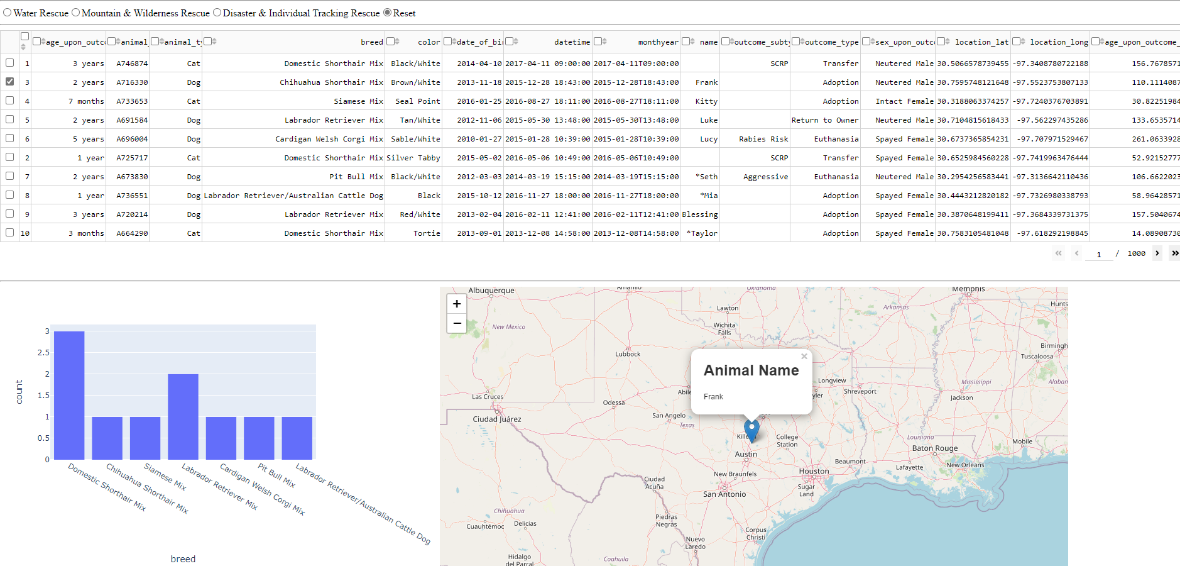
*Graphical user interface, application, map

Description automatically generated with medium confidence*

*Map

Description automatically generated*

**

**

## Contact

Your name: William Marshburn