# CS 340 README Template

## About the Project/Project Title

*This application is called Grazioso Salvare Data fetcher (innovative international rescue-animal training company) and is for demonstrating using standard python CRUD method calls to mongoDB API. The application can create, view, update, and delete documents in a mongoDB database. The application is currently setup for use as an ‘aauser’ which grants access to the AAC animal database. This project is for demonstrating capabilities with MongoDB and Python, along with learning CRUD functionality. The project has a dashboard which is used for interacting with the data in a table format, along with a pie chart, and interactable map. All this data can be filtered using the table.*

## Motivation

*The motivation behind this project is to demonstrate Python CRUD methods for a mongoDB API, as well as displaying interactable data in a fashionable and professional manner. The data has been provided via the international rescue-animal training company.*

## Getting Started

*To begin, firstly we’ll download the python module BlazeHaldermanProjectTwo and the Austin Animal Center data set, as well as the IPYNB file. Once we have that downloaded, we’ll simply need to run our IPYNB file through Meteor. The project module has pre-existing MongoDB credentials that have been setup for AAC database. The username and password are pre-setup with our database. This will display the interface of the application.*

## Installation

*Python 3.7.\**

* *Download from the Python site: https://www.python.org/*

*MongoDB 4.2.\**

* *Download: https://www.mongodb.com/docs/manual/installation/*

*PyMongo*

* *Download: https://pymongo.readthedocs.io/en/stable/installation.html*

## Usage

### Code Example

* This example illustrates using the module and making a “create/insert” call to the mongoDB API as well as a “read” call to show our data is being created in the database properly. (AAC Database provided via SNHU CS340)

Import BlazeHaldermanModuleFour as PythonCRUD

ShelterObject = PythonCRUD.AnimalShelter() -> creates a new connection instance for MongoDB connection

statusExample = shelter.create({key: value}) -> inserts a document into our collection with provided document, statusExample will be True or False depending on success or failure.

Data = shelter.read({key: value}) -> query documents in our collection and returns them in a list datatype

* This example explains using the module and making an “update” call to the mongoDB API as well as a “delete” call to show the data is being removed properly and updated properly. (AAC Database provided via SNHU CS340)

ShelterObject.update({key: value}) -> updates the specified query document(s) and returns the amount updated as well as an exception if not possible

ShelterObject.delete(key: value) -> deletes the specified query document(s) and returns the amount deleted as well as an exception if not possible

### Tests

* *We will first need to create an instance of our AnimalShelter class which has pre-configured credentials for “aauser” type in our AAC mongoDB database. When this is created using: “Shelter = CRUD.AnimalShelter()”, we are validating the credentials for the AAC database.*
* *The create method will return True if the value was successfully created in the database, and false if it failed. (status = shelter.create({key: value})) -> MongoDB document format*
* *The read method performs a MongoDB find call and returns the list Cursor with the data. (data = shelter.read({key: value}) -> MongoDB Query format)*
* *The update method performs a MongoDB insert call and returns an update mongo object with all information on the execution. If the execution of the update fails, it will throw an exception. (status = shelter.update(key: value))-> MongoDB Query Format)*
* *The delete method performs a MongoDB deletion call and returns the mongo object with all information from the execution. If the execution of the deletion fails, it will throw an exception. (status = shelter.delete(key: value)) -> MongoDB Query Format)*

### Screenshots

Example of Austin Animal Center data set import using MongoDB

Text

Description automatically generated

Authentication Example (operation auto-completed when initializing new object from class)

Text

Description automatically generated

Example Code for testing the module

Graphical user interface, text, application

Description automatically generated

Results example for code base

*Graphical user interface, text, application, email

Description automatically generated*

Example of using the interface of the application (OPENING IPYNB FILE)

1. DEFAULT OPENING OF THE APPLICATION

Logo

Description automatically generated

Graphical user interface, application

Description automatically generated

Map

Description automatically generated

1. FILTERING AN OPTION: WATER RESCUE

A picture containing table

Description automatically generated

Chart

Description automatically generated

1. FILTERING BY SELECTION OF ROW: ROW 5

Graphical user interface, application, table

Description automatically generated

Graphical user interface, application, map

Description automatically generated

1. FILTERING AN OPTION: MOUNTAIN RESCUE

Graphical user interface

Description automatically generated

Graphical user interface, map

Description automatically generated

1. FILTERING AN OPTION: DISASTER RESCUE

Graphical user interface, map

Description automatically generated

1. FILTERING AN OPTION: RESET

Graphical user interface, application

Description automatically generated

## Roadmap/Features (Optional)

*\*(Completed) CRUD functionality to the module.*

*\*(Competed) Interface with database functionality.*

*\*(Completed) Intuitive design and filtering options with a table, graph and map view for data.*

## Contact

Your name: Blaze Halderman