# CS 340 CRUD README

## About the CRUD module

*The purpose of the CRUD module being developed is to provide the end user with the ability to create new documents, obtain or update existing documents, as well as delete documents altogether.*

## Motivation

*This project exists because the need to maintain a database is essential to productivity. If a user was not able to create, update, or delete records the database would remain mostly useless and would not provided accurate data. The decision to utilize MongoDB instead of a SQL database was due to the large amount of unstructured data that we’ll be working with. MongoDB and NoSQL databases are better suited to handling this form of data over a standard SQL database such as MySQL*

## Getting Started

*In order to utilize the CRUD module you would first need to confirm the database information for the database you would like to implement this this module on. The required fields are username, password, and database name for authentication. Both the create and read functions of the CRUD module take a standard python dictionary as input and output either a cursor for the read function or an object ID for the create function. The end user experience allows for viewing and filtering of results from the database. There are three grouped filter options located in the dropdown selection box as well as a reset function as requested. When updating the filter selection, the pie graph will display the ratio of animal color based on the selection (this option can be edited to show the ratio of any attribute such as breed, age, sex, etc.). The map marker can be updated by selecting the checkbox on the left-hand side, the marker would then show attributes relevant to the selected row.*

## Installation

*The main tools used for this implementation are Python and MongoDB. The libraries used were Pymongo and MongoClient as well as PPrint for some formatting assistance. The tools used for the implementation of the front end are Jupyter, Plotly, and Dash.*

## Usage

### Code Example

*An example of the code used to create a new database entry is provided below. In this example we take in the user input as a dictionary item, and then upon successful creation output the new object’s ID. If a new entry is not successfully created the function returns an error message.*

*A screenshot of a computer

Description automatically generated with medium confidence*

### 

### Import Example Text Description automatically generated

### User Authentication ExampleText Description automatically generated

### Test Example

Text

Description automatically generated

**End user****experience**

Initial site state

A picture containing graphical user interface

Description automatically generated

Water Rescue Filter

Graphical user interface

Description automatically generated with medium confidence

Mountain and Wilderness Rescue Filter

Table

Description automatically generated with low confidence

Disaster and Individual TrackingA picture containing graphical user interface

Description automatically generated

Reset Filter

Graphical user interface, application

Description automatically generated with medium confidence

**Challenges**

*I ran into several challenges when working on this project. Many of them were user error or my misunderstanding of how some of the APIs were used. I solved these by going over the proper documentation to make sure I understood how to properly implement certain functions. I would say that the most difficult challenge I encountered was using the virtual environment. This was difficult because the response was slow and when typing would sometimes miss keystrokes, making typos or omissions that would be difficult to notice when running the programs leading to frustrations. I solved this by installing everything needed to produce and use the project locally and developing in that local environment.*

## Contact

Erik Engvall

(209)417-7238

Erik.Engvall@SNHU.edu