CS340M4Assignment2

## About the Project/Project Title

CRUD summarizes the 4 most basic and important functions of a web application, such as a database. Using CRUD, we can create, read, update, and delete data in a database. This repository offers a python module that allows users to easily manage their NoSQL databases in MongoD*B.*

## Motivation

There are many different ways to interact a database in MongoDB: Atlas (cloud-based), mongosh (command-line/shell), MongoDB Compass (graphical user interface), as well as drivers for numerous programming languages, including Python. However, each of these methods require some knowledge of syntax, platform, or the programming language used, which can be overwhelming at first. In this project, the CRUD module offers a simple way to communicate with a MongoDB database; users can perform a CRUD operation with only a few lines of code!

## Getting Started

* To get started, first ensure all the requirements from Installation are met.
* Download the *CRUD* module (*CRUD.py)* from this repository and place it in your python project.
* Start and run the MongoDB service by running the following command in your terminal window:

sudo systemctl start mongod  
Copy

Note: There should be no response to this command in the terminal.

You are now ready to use the CRUD module in your project.

## Installation (Ubuntu 22.04)

Before using the CRUD python module, you must first install the following software on your system.

##### Requirements

* [*MongoDB* and *Mongosh*](https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu) - NoSQL DBMS (database management system) along with its shell
* [*Python* (version 3)](https://www.python.org/downloads/source) - Python programming language
* [*Pymongo*](https://pymongo.readthedocs.io/en/stable/installation.html) *-* MongoDB for Python

##### Recommended

The following is recommended (but optional) to help you visualize your database.

* [*MongoDB Compass*](https://www.mongodb.com/try/download/compass) *–* Graphical user interface software that can manage your mongo databases

## Usage

To begin usage,

* First import the AnimalShelter data class from the CRUD

from CRUD import AnimalShelter  
Copy

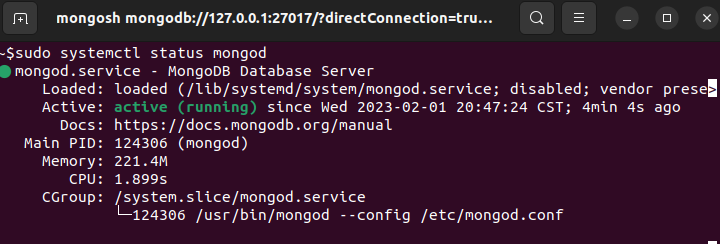
Next, we’re going to instantiate the AnimalShelter class call CRUD functions on the resulting object.

To instantiate AnimalShelter, we need to provide the following arguments in order:

* MongoDB username
* MongoDB password
* database name
* collection name
* MongoDB server port (This is optional and can be omitted to use the default value)

Example:

animal\_shelter = AnimalShelter("aacuser", "password1", "AAC", "animals")  
Copy

**Important:** If you receive an error message while instantiating, make sure an instance of MongoDB is running.  


##### Create

To insert a document (as python dictionary) into the database, use the create function.

Example:

animal\_shelter.create(  
 {  
 "age\_upon\_outcome": "200 years",  
 "animal\_id": "WHALE!",  
 "breed": "Sperm Whale",  
 "color": "Grey"  
 }  
)  
Copy

##### Read

To read from the database, first use the read function with a query argument (as python dictionary), then iterate over the resulting cursor object.

Example:

query\_results = animal\_shelter.read({"breed": "Sperm Whale"})  
  
for document in query\_results:  
 print(document)   
Copy

## Contact

Questions, comments, or feedback? Contact the author, Afshin Ahvazi, at [afshin.ebadehahvazi@snhu.edu](mailto:afshin.ebadehahvazi@snhu.edu)