**READ ME**

**About the Project/Project Title**

This project is a CRUD (Create, Read, Update, Delete) module implemented in Python to interact with a MongoDB database.

**Motivation**

The goal of this project is to develop a Python-based CRUD system that efficiently manages data stored in a MongoDB database. This module is designed to provide an easy-to-use interface for inserting and retrieving documents, making it suitable for applications requiring database interactions.

**Getting Started**

To get a local copy up and running, follow these simple steps:

1. Ensure that you have Python 3.x installed.
2. Install MongoDB and set up authentication credentials.
3. Clone or download this repository.
4. Install required dependencies (see Installation section).
5. Import the animal\_shelter.csv (or any) into your AAC database/ animal collection (if being used)
6. Run the AnimalShelter module to insert and retrieve data.

**Installation**

To use this module, you will need the following tools:

* **Python 3.x**: The programming language used for the project.
* **MongoDB**: The NoSQL database used for storing and retrieving data.
* **pymongo**: A Python driver for MongoDB.

**Before Starting**

**Importing csv file**

Use the command Line to import your documents before starting (im using animal\_shelter.csv)

mongoimport --username="${MONGO\_USER}" --password="${MONGO\_PASS}" --host="${MONGO\_HOST}" --port=${MONGO\_PORT} --db AAC --collection animals --authenticationDatabase admin --type=csv --file='aac\_shelter\_outcomes.csv' –headerline

A computer screen with text

AI-generated content may be incorrect.

Once your database is ready to manipulate you are ready to use the CRUD functions.

## **Usage**

***Creating a Record***

The create function inserts a document into the MongoDB collection.

# Instantiate the AnimalShelter class

shelter = AnimalShelter()

# Create test data to insert into the database

data = {"animal\_type": "Dog", "breed": "Labrador", "age": 3}

# Call the create method to insert data into MongoDB

result = shelter.create(data)

# Check if the insertion was successful

print("Insert operation successful:", result)

### **Tests**

***Example Output:***

Insert operation successful: True

### **Screenshots**

A screenshot of a computer program

Description automatically generated

***Reading a Record***

The read function retrieves documents based on a query filter***.***

Store the query in a variable to make passing as a parameter easier.

# Define a query to search for documents with species "Dog"

query = {"animal\_type": "Dog"}

# Call the read method to retrieve documents

query\_result = shelter.read(query)

# Display the query result

print("Query result:", query\_result)

### **Tests**

***Example Output:***

Query result: [{'\_id': ObjectId('6796a754b27c88d1616b2275'), 'rec\_num': 8, 'age\_upon\_outcome': '1 year', 'animal\_id': 'A736551', 'animal\_type': 'Dog', 'breed': 'Labrador Retriever/Australian Cattle Dog', 'color': 'Black', 'date\_of\_birth': '2015-10-12', 'datetime': '2016-11-27 18:00:00', 'monthyear': '2016-11-27T18:00:00', 'name': '\*Mia', 'outcome\_subtype': '', 'outcome\_type': 'Adoption', 'sex\_upon\_outcome': 'Spayed Female', 'location\_lat': 30.4443212820182, 'location\_long': -97.7326980338793, 'age\_upon\_outcome\_in\_weeks': 58.9642857142857}

### **Screenshots**

A screenshot of a computer

Description automatically generated

***Updating a Record***

To update a record, you must already have to data necessary to insert into your document.

I created a variable that holds my new\_data as json to insert into the already existing document test\_data.

new\_data = {

"name": "BellaTron",

"animal\_type": "Dog",

"age\_upon\_outcome": "5 years"

}

# Calls the Update method on the first parameter with the second parameter

updated\_result = shelter.update(test\_data, new\_data)

# Display the updated result

print("Updated Result: ", updated\_result)

### **Tests**

***Example Output:***

Updated Result: 1

**Screenshots**



***Deleting a Record***

Deletes a record of your choosing based on the parameter you passed in.

# Calls the delete method on the parameter

deleted\_result = shelter.delete(new\_data)

# Display the delete result

print("Deleted Result: ", deleted\_result)

### **Tests**

***Example Output:***

Deleted Result: 1

**Screenshots**



## Roadmap/Features (Optional)

* Improve error handling and logging.
* Develop a user-friendly UI for managing database interactions.

## Contact

Your name: Darrell Walker

Email: darrell.walker@snhu.edu