# CS 340 README Template

## About the Project/Project Title

The project is a Python module for creating, reading, updating, and deleting (CRUD) operations for an Animal collection in MongoDB. The project is designed for the Animal Adoption Center (AAC) to store and manage information about the animals available for adoption. The goal of the project is to provide a simple and efficient solution for animal shelters to store and manage information about the animals they have for adoption, as well as to provide an easy-to-use interface for working with animal data stored in MongoDB.

## Motivation

The motivation behind the project is to help animal shelters keep track of the animals they have, their characteristics, and their availability. The module abstracts away the low-level details of connecting to MongoDB and provides a simple and intuitive API for working with animal data, making it easier for animal shelters to manage their animal information efficiently.

## Getting Started

1. First you need to make sure your mongoDB is running:

Graphical user interface, text

Description automatically generated

1. create an instance of the **AnimalShelter** class, passing in your MongoDB username and password as arguments:

Text

Description automatically generated

1. Once you have an instance of the **AnimalShelter** class, you can use the **create** method to insert new data into the animal collection:

Text

Description automatically generated

1. use the **read** method to retrieve data from the animal collection

Graphical user interface, text

Description automatically generated

## Usage

Here's how the Python CRUD module can be used/tested:

1. Instantiating the AnimalShelter class with a username and password.

**Example:**

Graphical user interface, text, application, website

Description automatically generated

1. Usage: Here's how the Python CRUD module can be used:

Instantiating the AnimalShelter class with a username and password.

**Example:**

**Graphical user interface, text

Description automatically generated**

1. Usage: Here's how the Python CRUD module can be used:

Instantiating the AnimalShelter class with a username and password.

**Example:**

Graphical user interface, text, application, website

Description automatically generated

**Installation**

For this project, MongoDB was used as the database to store animal information as it is a NoSQL document-oriented database that stores data in a JSON-like format. This makes it easy to work with Python data structures.

1. Install MongoDB: You can install MongoDB by following the official installation guide. <https://docs.mongodb.com/manual/installation/>

PyMongo, a Python library was used to interact with MongoDB. This library provides a convenient way to interact with MongoDB databases and collections.

BSON, a binary serialization format was used by MongoDB to store data in a binary format. PyMongo automatically converts Python data structures into BSON and vice versa.

Graphical user interface, text, application

Description automatically generated

Jupyter Notebook was used to write and test the code. Jupyter Notebook provides a user-friendly environment for writing, testing and debugging code.

Graphical user interface, text, application

Description automatically generated

The rationale for using these tools is that MongoDB provides a flexible and scalable solution to store large amounts of data. PyMongo provides a simple way to interact with MongoDB using Python, and BSON is used to handle the binary data that MongoDB stores.

### Tests

The above code code tests the functionality of the **AnimalShelter** class.

1. The test data is a dictionary representing an animal to be inserted into the collection.
2. The **AnimalShelter** class is instantiated by passing in the username and password for authentication.
3. The **create** method is then called, passing in the test data, and the result is stored in the **result** variable.
4. The **read** method is then called with a key-value lookup to find the animal that was just inserted, and the result is stored in the **result** variable.
5. Finally, the contents of the result are printed to the console to show the data read from the database.

## Roadmap/Features (Optional)

1. Implement the update method to modify the existing documents in the animals collection.
2. Implement the delete method to remove the existing documents in the animals collection.
3. Implement the search method to search and retrieve the specific documents in the animals collection.

Known Issues:

1. Necessary permissions for "aacuser" must be created for the correct data base:

# Connect to the MongoDB database as the admin user

# Switch to the AAC database

> use AAC

# Verify that the user has the necessary roles:

**> db.runCommand({usersInfo: "aacuser", showPrivileges: true})**

# If the user does not have the necessary privileges, you can use the grantRolesToUser method to assign the appropriate roles. For example:  
> db.createUser({user: "aacuser", pwd: "password", roles: [{ role: "readWrite", db: "AAC" }]})

This command grants the "readWrite" role to the "aacuser" user on the "AAC" database. You can also assign additional roles as needed.

## Contact

Your name: Aaron Ciminelli

Email: aaron.ciminelli@snhu.edu