

CS-340 Project 1

Shari Storlie



Step 1: Upload the Austin Animal Center Outcomes data set into MongoDB by **inserting a CSV file using the appropriate MongoDB import tool**.

A screenshot of a computer

Description automatically generated

Step 2: Create a user account in the mongo shell to **ensure user authentication to the database and collection** you created.

A screenshot of a computer program

Description automatically generated

Step 3: Develop a Python module in a PY file, using object-oriented programming methodology, to enable CRUD functionality for the database. Other Python scripts must be able to import your Python code, so it must support code reusability. Develop a CRD class that provides the Create, Read, Update, and delete functionality.

A screenshot of a computer code

Description automatically generated

*A screenshot of a computer program

Description automatically generated*

*A screenshot of a computer code

Description automatically generated*

Step 4:  Test your Python module to make sure it works. Expand your script to call and test the **Update** and **Delete** functionality.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

## **CS 340 README**

## About the Project/Project Title

This project implements a CRUD (Create, Read, Update, Delete) Python module to interact with a MongoDB database. The module was developed to facilitate seamless integration with a MongoDB database for data manipulation. It allows users to create new documents, read existing ones, update data, and delete records using Python scripts.

The module is designed with object-oriented programming principles to ensure reusability, flexibility, and maintainability.

## Motivation

The motivation for this project was to create a reusable and modular Python solution for managing a MongoDB collection. This CRUD module makes database operations easy for developers working with MongoDB, enabling them to focus on application logic without dealing with the complexities of database queries. The project also provides the foundational opportunity to expand CRUD operations to support additional features and performance optimizations.

## Getting Started

**Prerequisites**

1. **MongoDB Database**:
   * The database (AAC) and collection (animals) must be set up beforehand.
   * User authentication (aacuser with password Shari1234) must be enabled.
2. **Python Environment**:
   * Ensure Python 3.x and the necessary libraries are installed.
   * Python 3.x and the necessary libraries are installed.

**Setup Instructions**

1. Clone the repository or copy the AnimalShelter.py file to your project directory.
2. Install the required Python libraries:
3. pip install pymongo
4. Update the connection credentials in the AnimalShelter class:
5. USER = 'aacuser'
6. PASS = 'Shari1234'
7. HOST = 'nv-desktop-services.apporto.com'
8. PORT = 33421
9. DB = 'AAC'
10. COL = 'animals'

## Installation

**Tools Used**

1. **Python**:
   * **Version**: 3.x
   * **Rationale**: Python is a versatile programming language with excellent libraries for MongoDB integration.
2. **MongoDB**:
   * **Rationale**: A NoSQL database that supports flexible schema and efficient storage of JSON-like documents.
3. **Pymongo Library**:
   * **Rationale**: Official Python driver for MongoDB, used to connect and interact with the database.
   * **Installation Command**:
   * pip install pymongo
4. **Jupyter Notebook**:
   * **Rationale**: Used for testing and documenting the CRUD module in an interactive environment.

## Usage

### Code Example

from pymongo import MongoClient

from bson.objectid import ObjectId

class AnimalShelter(object):

""" CRUD operations for Animal collection in MongoDB """

def \_\_init\_\_(self):

# Initializing the MongoClient. This helps to

# access the MongoDB databases and collections.

# This is hard-wired to use the aac database, the

# animals collection, and the aac user.

# Definitions of the connection string variables are

# unique to the individual Apporto environment.

#

# You must edit the connection variables below to reflect

# your own instance of MongoDB!

#

# Connection Variables

#

USER = 'aacuser'

PASS = 'SNHU1234'

HOST = 'nv-desktop-services.apporto.com'

PORT = 31580

DB = 'ACC'

COL = 'animals'

#

# Initialize Connection

#

self.client = MongoClient('mongodb://%s:%s@%s:%d' % (USER,PASS,HOST,PORT))

self.database = self.client['%s' % (DB)]

self.collection = self.database['%s' % (COL)]

# Complete this create method to implement the C in CRUD.

def create(self, data):

if data is not None:

self.database.animals.insert\_one(data) # data should be dictionary

else:

raise Exception("Nothing to save, because data parameter is empty")

### Tests

***Create Operation***

*from AnimalShelter import AnimalShelter*

*shelter = AnimalShelter()*

*data = {"name": "Charlie", "type": "Dog", "age": 2, "breed": "Labrador"}*

*print("Document inserted successfully.") if shelter.create(data) else print("Failed to insert document.")*

***Read Operation***

*query = {"type": "dog"}*

*results = shelter.read(query)*

*print("Query results:", results) if results else print("No matching documents found.")*

### Screenshots

PY Screenshots:

*A screenshot of a computer code

Description automatically generated*

*A screenshot of a computer program

Description automatically generated*

*A screenshot of a computer code

Description automatically generated*

Test Screenshots

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

## Roadmap/Features

*Future features could include advanced querying, bulk operations, data validation, logging, and testing framework.*

*Known issues are case sensitivity for database names and error handling could be more descriptive for easier use.*

*A specific known issue is that the update method in the py file returned a bool instead of the expected UpdateResult object from MongoDB’s update\_one method. Further troubleshooting could include investigating external dependencies for possible issues.*

*The project stands out for it’s ease of use, reusability and expandability.*

## Contact

Your name: Shari Storlie