# CS 340 README Template

## About the Project/Project Title

The project is called CRUD: In Python. Implementing the fundamental operations of creating, reading, updating and deleting documents in Python.

## Motivation

The Python middleware serves as ‘glue’ between the Python web application and the MongoDB. Creating a one-to-one relationship between a database and a web application by crafting reusable/adaptable code that will hold the development together while also laying a foundation for future events. The flexibility of reusable code allows for potential interface with multiple web applications and allows for a smoother transition to a different base or sever level in the event of a database change.

## Getting Started

Start by importing a csv database file to MongoDB. Create a new user account for the AAC database in mongo shell. Develop a Python module in a PY file with the capability of importing Python code as a module to support reusability. Include methods that insert and query documents as well as updating and deleting documents into a specified MongoDB and collection. Finally create a Python testing script in Jupiter Notebook that imports the CRUD Python module to call and test the ‘create’, ‘read’, ‘update’, and ‘delete’ instances for functionality.

## Installation

MongoDB: use the mongoimport tool upload a csv file and mongo shell (mongosh) to create a new user for the database.

Jupyter Notebook: import libraries MongoClient from pymongo and ObjectId from bson.objectid to create a PY File. A separate IPYNB file should import and instantiate an object from the CRUD library to affect changes in MongoDB.

## Usage

### Code Example

The code includes an Animal Shelter Class that initializes the Mongo Client and connects the specific variables from the ACC database in MongoDB. This uses the aac database, the animal collection and the aac user.

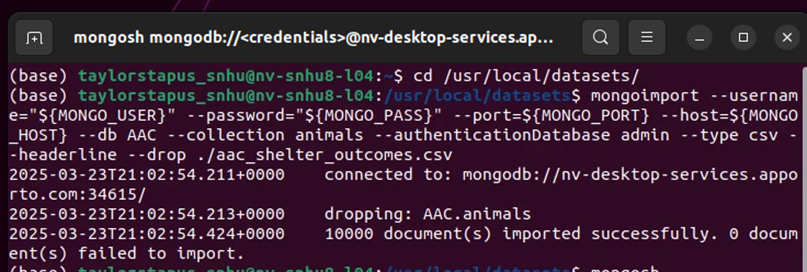
Methods are added to be able to create, read, update and delete new animal information that is inputted into the list.

### Tests

A IPYNB is created separately that imports the AnimalShelter Class from the PY file. The Class is then assigned to a variable. The test animal variable “Cat” is implemented with the respected fields. The create, read, update and delete functions are then called to verify that the animal was added correctly.

### Screenshots

**Import CSV:**



**Create aacuser:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**



**AnimalShelter Class:**

A screenshot of a computer program

AI-generated content may be incorrect.

**CRUD Method:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Test Functions:**

A screen shot of a computer code

AI-generated content may be incorrect.

A white background with text

AI-generated content may be incorrect.

A white rectangular object with red and black text

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

*.*

## Contact

Your name: Taylor Stapus