# CS 340 README Template

*Use this template to complete your README file. When completing the template, keep the headings as they are so that your document has a clear organization. Remove the italicized prompt text after you have completed each section for a polished final document.*

## About the Project/Project Title

*This project implements a basic CRUD (Create, Read, Update, Delete) Python module, allowing users to interact with a database and perform data management operations. The module is designed to handle simple data records, enabling users to add new entries, view existing ones, and eventually update or delete them. It serves as the foundation for a more complex application in the future.*

## Motivation

*The motivation behind this project was to build a flexible and scalable Python module for managing user data in a database. The CRUD operations were implemented as part of a learning exercise to explore database interaction, user authentication, and building reusable code that can be expanded to include more features.*

## Getting Started

*To get this program started you would first,*

*1. Enter Mongo and import the csv file aac\_shelter\_outcomes.csv.*

*2. Create a simple and a complex index to parse the data stored within the document.*

*3. Create both an Admin account and an aacuser account to access the database.*

*4. Access or install python and run the program out of a notebook*

## Installation

*A current version of Python to run both the .py and the .ipynb files*

*A current version on Jupetyr Notebook*

*MongoDB - to access the database.*

## Usage

*This section demonstrates how to use the CRUD Python module.*

### Code Example

*from pymongo import MongoClient*

*class AnimalShelter:*

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

*def \_\_init\_\_(self, username, password, host='nv-desktop-services.apporto.com', port=31580):*

*""" Initialize the MongoClient with authentication """*

*try:*

*self.client = MongoClient(f"mongodb://{username}:{password}@{host}:{port}")*

*self.database = self.client['AAC'] # Use the AAC database*

*self.collection = self.database['animals'] # Use the animals collection*

*except Exception as e:*

*raise Exception(f"Failed to connect to database: {e}")*

*def create(self, data):*

*""" Create a new document in the collection """*

*if isinstance(data, dict):*

*try:*

*self.collection.insert\_one(data)*

*return True*

*except Exception as e:*

*print(f"An error occurred: {e}")*

*return False*

*else:*

*raise TypeError("Data should be a dictionary.")*

*def read(self, query):*

*""" Read documents based on a query """*

*if isinstance(query, dict):*

*try:*

*result = self.collection.find(query)*

*return list(result) # Return results as a list*

*except Exception as e:*

*print(f"An error occurred: {e}")*

*return []*

*else:*

*raise TypeError("Query should be a dictionary.")*

### Tests

*"from animal\_shelter import AnimalShelter\n",*

*"\n",*

*"username = \"aacuser\"\n",*

*"password = \"YourSecurePassword\"\n",*

*"shelter = AnimalShelter(username, password)\n",*

*"\n",*

*"new\_animal = {\n",*

*" \"name\": \"Buddy\",\n",*

*" \"age\": 3,\n",*

*" \"type\": \"Dog\",\n",*

*" \"breed\": \"Golden Retriever\"\n",*

*"}\n",*

*"create\_status = shelter.create(new\_animal)\n",*

*"print(\"Create Status:\", create\_status)\n",*

*"\n",*

*"query = {\"type\": \"Dog\"}\n",*

*"dogs = shelter.read(query)\n",*

*"print(\"Query Result:\", dogs)"*

*]*

*},*

### Screenshots

*A screenshot of a computer program

Description automatically generated*

## Roadmap/Features (Optional)

*Provide an open issues list of proposed features (and known issues). If you have ideas for releases in the future, it is a good idea to list them in the README. What makes your project stand out?  
  
Note: This section is optional for the purposes of this assignment. If you choose not to fill out this section, remove it from your final README file.*

## Contact

Your name: Kate Moran