# CS 340 README Template

## About the Project/Project Title

**Title***:* Python Database Interface

**About***:* This project focusses on creating, viewing, updating, and deleting database entries via the MongoDB shell. The purpose of this project is to provide a set of code files in which creating, reading, updating, and deleting database entries in the MongoDB shell is simplified via object-oriented Python code.

## Motivation

The motivation behind this project was that using a command-line was somewhat tedious for every day, database use. Thus, an interface between the MongoDB core shell and the end-user was created via Python 3 code in which object-oriented techniques were used to provide CRUD (Create, Read, Update, and Delete) functionality.

## Getting Started

To get started, follow these steps:

1. Install Python 3.x and the Mongo shell. (This is shown in the following section.)
2. Download the Python executable file and zip folder that contains all of the code needed for the database interface.
3. Run the executable Python file and begin creating, viewing, updating, and deleting data as needed.

## Installation

The tools needed to run this project are:

1. MongoDB – a database management tool in which the foundation of this project is based. It is used to create, read, update, and delete data entries, along with the Python 3 interface and Dash framework.
   1. Go to [www.mongodb.com/docs/manual/installation/](http://www.mongodb.com/docs/manual/installation/) to install the latest version of MongoDB Community Edition for your platform (Windows, Mac, Linux, etc.).
   2. Once MongoDB is downloaded, click on the file to install it and follow the installation instructions.
   3. When installed, MongoDB will provide you with a powerful NoSQL database management tool in which this project is based.
2. Python 3.x
   1. Go to [www.python.org](http://www.python.org) to install the latest Python version.
   2. Click the download link.
   3. Once downloaded, click on the Python installation and follow the installation instructions.
   4. Once Python is installed, you will be able to run the Python interface code for this project.
3. The Python Dash library
   1. This tool is included with the standard Python 3 download and is used to create dashboard interfaces.
4. Your operating system in which to install the above tools and the downloadable code and executable files
   1. You should have enough space to not only download the tools above but to also manage both small and large databases – depending on what the Python interface is being used.

## Usage

### Code Example

Example of the project’s primary code

Graphical user interface, text, application

Description automatically generated

As you can see in the above code, a class and functions are created to support code reusability. For each element of CRUD, a separate function is defined. Because of that, the code follows reusability guidelines in which other programmers will be able to rename variables and functions to represent their own ideologies. Following this, database entries will be gathered from the user and stored in variables via Python input functions. Then, the data that was provided by the user can be read (or viewed), updated, and deleted (if granted to do so by an administrator).

### Tests

The test case for the project

Text

Description automatically generated

The above test case illustrates an example of user data being entered in the main program and then stored in a “data” variable. Once the data is entered, the program will check to see in the data was indeed created and output an error if not. Note, further versions of the code will include the rest of the CRUD functionality – reading, updating, and deleting data entries.

### Screenshots

1. Import the AAC database.

A screenshot of a computer

Description automatically generated with medium confidence

1. Administrator authentication

Graphical user interface

Description automatically generated

1. CRUD functionality

Text

Description automatically generated

**Steps to Complete the Project**

1. Create the CRUD Python file in which database entries can be managed by the user.
2. Create a test script to tell if all CRUD functions run properly.
3. Finally, the Python Jupyter Notebook dashboard was created in which an interface was defined for the end-user.

**Project Issues / Fixes**

Because this was my first experience with MongoDB, there was a steep learning curve in its syntax and how it is used. I also had some issues with the Python dashboard in which some bugs still exist with the code in that file.

## Contact

Brett Ferguson

([brett.ferguson@snhu.edu](mailto:brett.ferguson@snhu.edu))