# 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 is about learning how to interact with MongoDB through Python. We can take this learning and implement it when developing web applications that needs to connect to the database to query the data. Throughout the project, we will learn Create, Read, Update, and Delete (CRUD) functionality, which is essential when working with databases.

## Motivation

This project highlights the importance of using MongoDB and CRUD functionality that can be easily use through PyMongo module. It shows the flexibility and use case of MongoDB in real life.

## Getting Started

The database for this project is very simple and it is about animal shelter, and what kind of species and their breed information we imported from csv file called “acc\_shelter\_outcomes.csv” in MongoDB. We created “aacuser” to run query against “AAC” database, where the animal shelter information is imported. We created separate functions for Create, Read, Update, and Delete functionality. So, it can be use in different projects. The difficult part was to connect with MongoDB through Python because without it, we can’t insert new objects or query the database. We were able to connect after making changes to the connection string in python file.

## Installation

For this project, make sure latest version of Python is installed on your local machine. In Python file, you will need to import MongoClient and ObjectId modules to communicate with MongoDB. Lastly, you will Jupyter Notebook to create test file to instantiate the object and use CRUD functionality.

## Usage

### Code Example

*Text

Description automatically generated*

Text

Description automatically generated

We are importing MongoClient and ObjectId modules. After that, we are creating a AnimalShelter class. So, it can be instantiated to connect to the database and run query against the database. When we instantiate the object of AnimalShelter, we need to pass username and password to the object and MongoClient will use it to connect to the database. Afterwards, we created ‘create’, ‘read’, ‘update’, and ‘delete’ functions to query the database.

### Tests

Text

Description automatically generated

We created the IPYNB file to test our code. We imported AnimalShelter module and instantiate AnimalShelter object by passing username and password. Then, we created a sample animal data and use create function and pass that data. Use read function to read all data exist in AAC database. We pass two values to update function, the first value will find the existing data in the database and replace the value with the second value. It will return the updated object. We only pass one value to the delete function, it will look for that value in the database and delete that object.

### Screenshots

Text

Description automatically generated

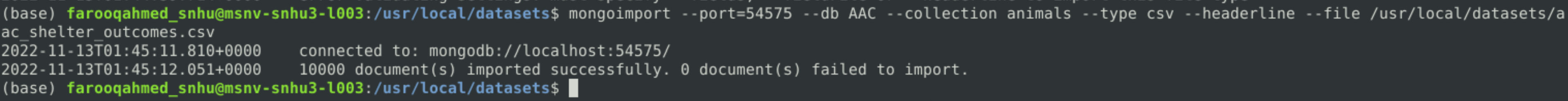
A screenshot of a computer

Description automatically generated with low confidence

Text

Description automatically generated

The above screenshot is the result after running the test script.

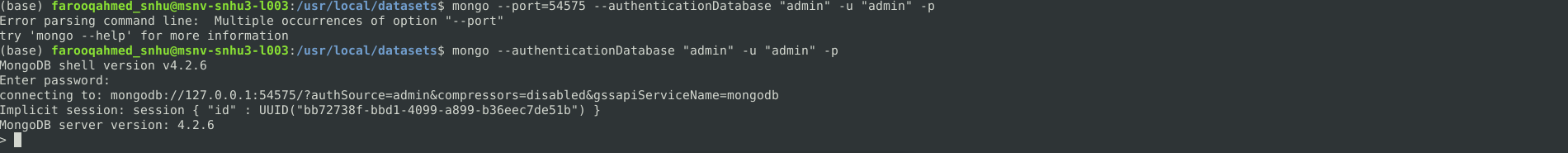


The above screenshot shows importing Austin Animal Center Outcomes dataset into MongoDB.

Shape

Description automatically generated with medium confidence

The above screenshot shows how to create admin account with admin privileges.

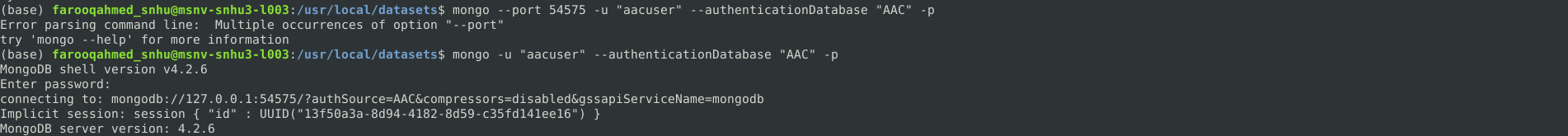


The above screenshot shows how you can connect to database using admin credentials.

Text

Description automatically generated

The above screenshot shows how to create user accounts.



Lastly, how you can login to database using user’s credentials.

Well, now let’s get into Main Dashboard.

A picture containing logo

Description automatically generated

It’s the main Dashboard with Grazioso Salvare logo on the main page.

Graphical user interface, application

Description automatically generated

As you scroll down, you will see the Unfiltered Animal Shelter Data with Pie Chart and location of Animal Shelter.



You have drop-down box with four filters, which are Water Rescue, Mountain Rescue, Disaster Rescue Dogs, and Reset (Unfiltered Data).

Graphical user interface, application

Description automatically generated

Graphical user interface, chart, application

Description automatically generated

Graphical user interface, chart, map

Description automatically generated

When we applied those filters, the Pie Chart changes along with the table. We wrote a query for each filter to pull the correct data from AAC database.

Dash Framework helped us achieve the beautiful User Interface with graphical data and it is interacting with Python and MongoDB very efficiently.

## Contact

Your name: Farooq Ahmed