# CS 340 README Template

## About the Project/Project Title

Develop a database and a Python module enabling Crud functionality for MongoDB. Code a dashboard for the database interface meeting the specified requirements.

## Motivation

This project was created to provide backend support to be able to connect the user interface components to a database component. This code created will be able to interact with client-side code.

## Getting Started

To get a local copy up and running, follow these simple example steps.

First you need to import the csv file to the databse

mongoimport --port=port\_no --db=AAC --collection=animals --type=csv --headerline --file=aac\_shelter\_outcomes.csv

From here you are going to make sure that you have an admin and user account on the database you want to use. You can do this by starting mongo without authorization. Then you start mongo. Then write the following:

use admin

db.createUser(

{

user: "myUserAdmin",

pwd: passwordPrompt(), // or cleartext password

roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]

}

)

The you want to restart mongo without authorization and start mongo using the following to authenticate:

mongo --authenticationDatabase "admin" -u "admin" -p

You can then repeat these steps to create a user account.

Then you would create animal\_sheler.py file to be able to add CRUD functionality to the database. Next create a script to be able to test that the python file is working correctly and creating, reading, updating, and deleting a file in the database.

Now then you have all these components you can update the. ipynb file with your username and password so that you can properly run the program.

To run the application you would start up mongo without authentication and then run the .ipynb file.

## Installation

This project involved using the MongoDB as well as PyMongo Drivers to create CRUD functional access to the documents collection.

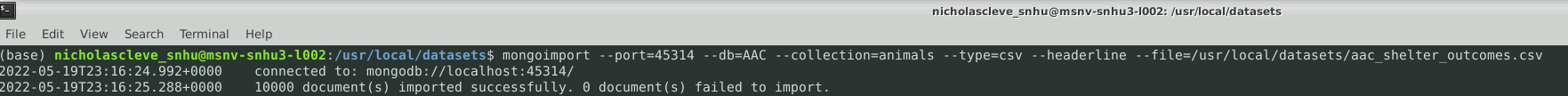
Jupyter Notebook was also used to create the python file, the program itself, and the script to test the code.

This program also takes advantage of the Dash framework that provides the view and controller structure for web applications.

## Usage

### Code Example

Importing a CSV file into database



Creating Admin and User Accounts

Text

Description automatically generated

Shape, rectangle

Description automatically generated

Animal Shelter Python Module

Table

Description automatically generated with low confidence

### Tests

Importing Animal Shelter Module

Text

Description automatically generated

Testing Create Method

Text

Description automatically generated

Testing Read Method

Text, letter

Description automatically generated

Testing Update Method

Text

Description automatically generated

Testing Delete Method

Graphical user interface, text, application

Description automatically generated

### Screenshots

Default View of Dashboard

Graphical user interface, table

Description automatically generated

Disaster Resscue View of Dropdown Selected

Graphical user interface

Description automatically generated with medium confidence

Mountain Rescue View of Dropdown Selected

Graphical user interface, application

Description automatically generated

Water Rescue View of Dropdown Selected

Graphical user interface, application, table

Description automatically generated

## Contact

Nicholas Cleveland