# CS 340 README Project 2

## About the Animal Shelter Application

The Animal Rescue Application is a web app built to effectively read, sort, and manage data. The app connects to a backend database that contains information on all of the animals at Grazioso Salvare. Through the front end app, you can add, delete, and sort this data.

## Motivation

Creating a secure means of interfacing with delicate data was the motivation of this application.

Using a Python middleware is a great tool for securing data. This minimizes direct access to the data and allows the developer to set up methods and access based on permissions. Additionally, it provides an easy way to interface with the data without dealing directly with Mongo. This module exists to perform operations like create and read on the data.

## Getting Started

To get a local copy of this project running, you must first install and set up your Mongo database. To do this, open a terminal and CD into your mongo directory. From there, you must start your server by running the following command: **mongo\_ctl start-noauth**

Make note of the port number returned from this command.

After the server is running, you can import the data from the CSV file by running the next command: **mongoimport –port** (port number here) **–db AAC –collection animals –type CSV –headerline ./aac\_shelter\_outcomes.csv**

This will load the JSON from the CSV into our mongo database.

You can add users to this database and assign access accordingly.

* In order to run the Python Module, you will need an admin account set up for mongo.

In the Python Module, you will need to link the IP Address of the server and your port number to connect to the running database.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## Installation

MongoDB: <https://www.mongodb.com/docs/manual/installation/>

This link walks through the installation/set up for MongoDB.

PythonIDE: [pycharm](https://www.jetbrains.com/pycharm/promo/?source=google&medium=cpc&campaign=14127625430&term=pycharm&content=603858680130&gclid=EAIaIQobChMIwJTK0oTu_AIVpIFbCh319QNJEAAYASAAEgK7G_D_BwE)

You can use any Python IDE to run the module, but above is a link to PyCharm.

Usage

### Code Example

**API**

**Create**

*Graphical user interface, text, application

Description automatically generated*

The create() method takes a key, value pair as a parameter, checks to make sure that parameter isn’t NULL, then inserts it into the database. If the insert returns 1, the method returns true, that a value was inserted successfully. If it does not insert successfully or if invalid data is passed, an exception will raise or the method will return false.

**Read****Graphical user interface, text, application

Description automatically generated**

The read() method takes a key, value pair as a parameters, checks to make sure the parameter isn’t NULL, then returns the location of the occurrences of that key, value pair. If invalid data is passed to the method, an exception is raised.

**Update**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

The update() method takes in two key, value pairs as parameters, checks to make sure neither parameter is empty, then returns the mongodb reference object. An exception will read in the case of an error or invalid parameters. NOTE: the second parameter must be passed as follows: {“$set”: {“key”: “val”, “key”: “val”}}

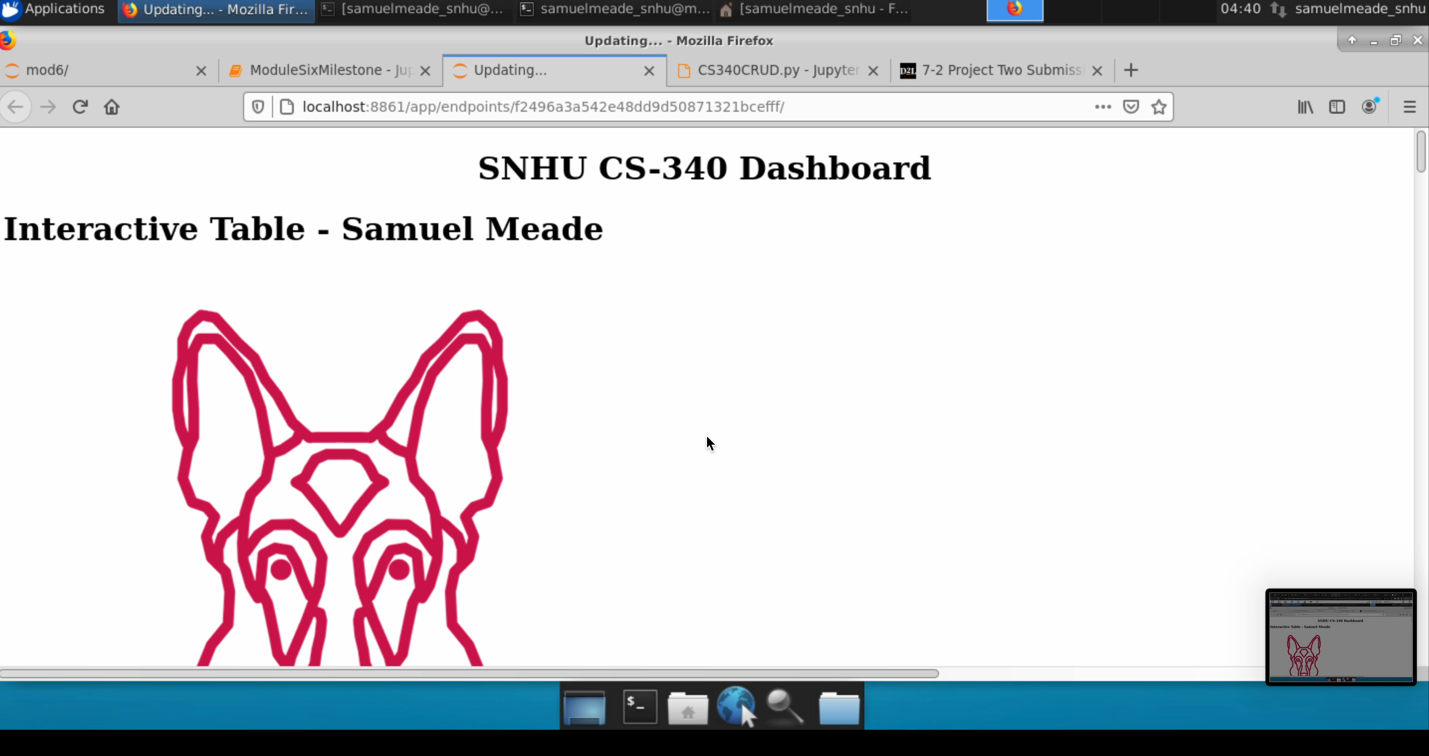
**Delete**

Graphical user interface, text, application

Description automatically generated

The delete() method takes in a key, value pair as a parameter, checks to make sure the parameter isn’t empty, then returns the mongodb reference object after deleting from the data. An exception will read in the case of an error or invalid parameters

Web Application

**** SCREEN RECORDING:

In the recording above, you can see the user navigate the Animal Shelter Dashboard. Here they can scroll through the data, toggle a search filter by animal type, and view the geo-location of the selected animal.

Table

Description automatically generated

This is a work in progress of the table component to be implemented. Currently the component is displaying but not linking to any data. Link this database to show the amounts of each animal-breed.

**Tests**

Graphical user interface, text, application

Description automatically generated

The tests import the module, then initializes the database using the create credentials for the database. The create test attempts to insert an arbitrary object into the data, and stores that result into the create variable. To ensure this was successful, we assert create is true, then attempt to read that addition in the next test. The read test takes in the same object as the create test to verify that it was created properly. We assert that the read count is more than zero to signify a successful search. To test the update functionality, we update the data we just created, modifying the numberOfAnimals to 1. Since the update method returns the new data set when successful, we assert that the data stored in update matches our input data. The delete method takes in one object as a parameter, and we use the same data we just updated. To confirm the data was removed, we perform a read and assert the read count is zero, indicating the data no longer exists.

## Contact

Samuel Meade