# CS 340 README Template

## About the Project/Project Title

The purpose of this project is to create a means for the client, Grazioso Salvare, to effectively identify and categorize dogs from data provided by their partner Austin Animal Center. This will be accomplished using MongoDB and a Python module.

## Motivation

Grazioso Salvare is a rescue animal training company. Grazioso Salvare, in partnership with Austin Animal Center, wishes to identify and categorize potential rescue dogs from those that reside in one of five animal shelters around the Austin, Texas area. There are certain characteristics that Grazioso Salvare uses for identifying and categorizing potential candidates. These include, but are not limited to, age and breed.

## Getting Started

To get a local copy up and running follow the following steps:

1. Verify MongoDB is installed on your system or network. If not, follow the instructions found at [MongoDB's website.](https://www.mongodb.com/docs/manual/installation/) Take note of the database’s server address.
2. After verifying MongoDB is installed and set up, open the command terminal to connect to the MongoDB server.
3. From the directory storing the “aac\_animal\_outcomes.csv” file, import the CSV to MongoDB, using the database name “AAC” and collection name “animals”.

A screenshot of a computer screen

Description automatically generated with medium confidence

1. Create a user account, for example we use “aacuser”, with read and write access to the AAC database. A screenshot of a computer program

   Description automatically generated with medium confidence
2. In the “AAC” database “animals” collection create any indexes desired for search efficiency.
3. In the AnimalShelter.py file, update the USER, PASS, HOST, and PORT values to reflect the created account’s username and password, as well as the server address.

A screen shot of a computer code

Description automatically generated

1. In Jupyter Notebook, or similar program, in a new Python 3 environment import the AnimalShelter object from the AnimalShelter Package.
2. Establish an object that is AnimalShelter(). – This initializes the Python MongoDB Connection. In our example we used the object “shelter\_test”
3. From here you can create a new database item using the .create() function, for example shelter\_test.create({EXAMPLE DICTIONARY}), or find a database item using the .read() function, such as shelter\_test.read({“breed”: “Labrador Retriever”}).

Using the Dashboard:

1. From Jupyter Notebook, or similar, open the notebook file “ProjectTwoDashboard(1).iypnb” and verify that the values for username and password match the account created when setting up the database.

A close-up of a computer screen

Description automatically generated

1. Run the notebook file.
2. A local Dash App running notification will be at the bottom of the notebook. Click on the local web address provided to access the dashboard.

A computer screen shot of a computer code

Description automatically generated

1. The Grazioso Salvare Database Dashboard should display the company logo, that acts as a link to the Grazioso Salvare website, <http://www.snhu.edu>, selectable search filters through radial buttons, a list of animals, a graph of breed breakdown, and a map showing the current location of the selected animal.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Installation

The following will be required to utilize this software:

Python 3 or later

Remote or local MongoDB server

Jupyter Notebook or similar.

## Usage

### Code Example

*def create(self, data):*

*if data is not None:*

*result = self.database.animals.insert\_one(data)*

*if result != 0:*

*return True*

*else:*

*return False*

*else:*

*raise Exception("Nothing to save, because data parameter is empty")*

### Tests

*As described in Getting Started, tests can be run using Jupyter Notebook or something similar. Using Python 3, import AnimalShelter from AnimalShelter then initialize the AnimalShelter() object. In the screenshots below “shelter\_test” was used.*

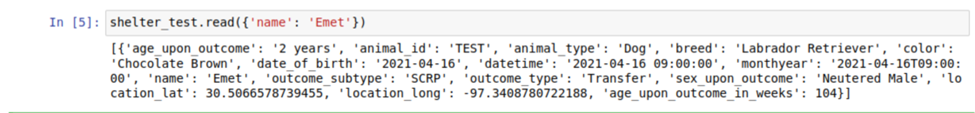
### Screenshots

*Provide screenshots that demonstrate your work.*

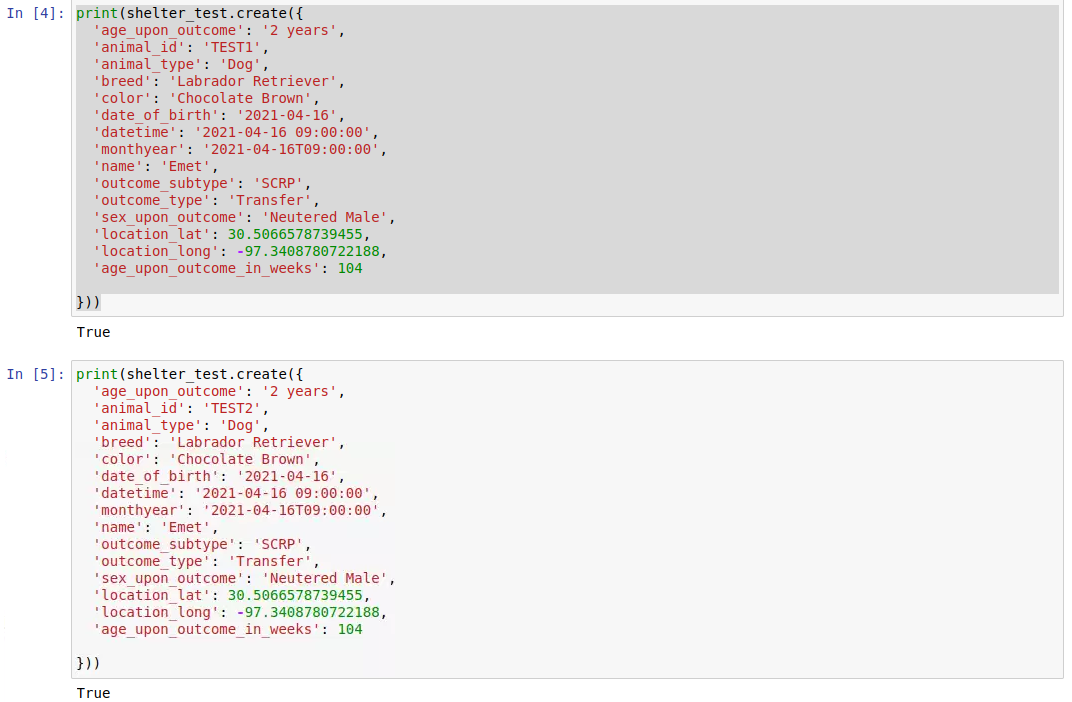
A screenshot of a computer

Description automatically generated with medium confidence

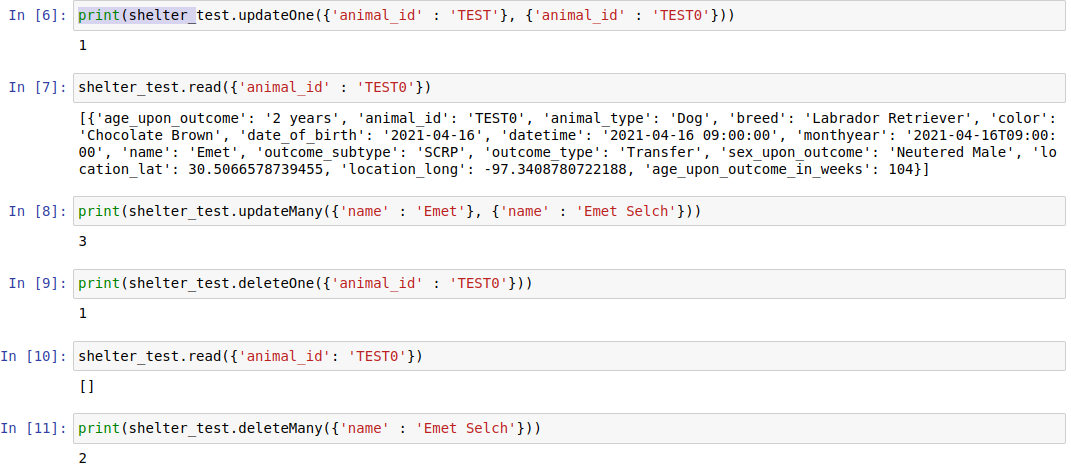
- The picture above shows testing the Python CRUD module by using Jupyter Notebook. Using Python 3 the AnimalShelter object is imported and initiated. Afterwards a new dog, 'Emet' is added to the database. True is shown to verify the item was added.

**

2 - The picture above shows using the read() function to find the newly added 'Emet' item in the database. The output is in list form



- The picture above shows adding two additional test dogs to demonstrate the update and delete functionality.



- The picture above shows using the updateOne(), updateMany(), deleteOne() and deleteMany() functions.

**Challenges Encountered:**

During creation of the dashboard filter errors were encountered due to changing the selected filter when a row with an index value greater than the number of results in the new filter was selected. In order to prevent this from occurring an @app.callback was implemented that reset the current selected row to the first row on the list.

## Contact

Lukas Pentowski