**CS 340 Project Two README**

**About Project Two**

Project Two exists so users can interact with a dataset of different animal breeds within a shelter. The project allows users to perform CRUD operations in the database: create, read, update, and delete.

**Motivation**

The motivation behind Project Two is so the client can interact with the database containing animal shelter rescues.

**Getting Started**

To get started, the user should navigate to the mongo shell through a terminal and use the AAC database by importing the csv file named “aac\_shelter\_outcomes.csv”. Next, the user will need to develop code in a python environment that follows the CRUD module. This file will need to include the user’s username, password, port number, and database name that they created earlier. Then, an ipnyb file is necessary so the user can run the dashboard on their desktop.

**Needed Installation**

Python environment

Jupyter Notebook

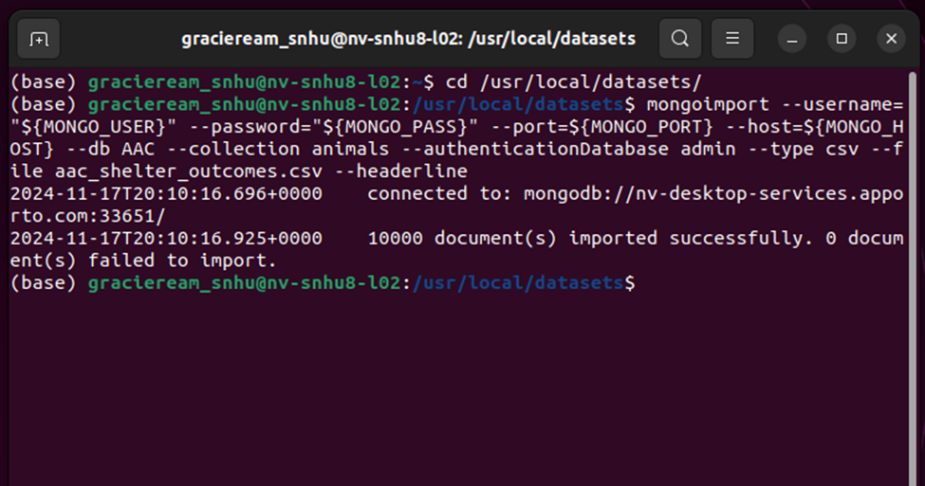
Terminal/MongoDB

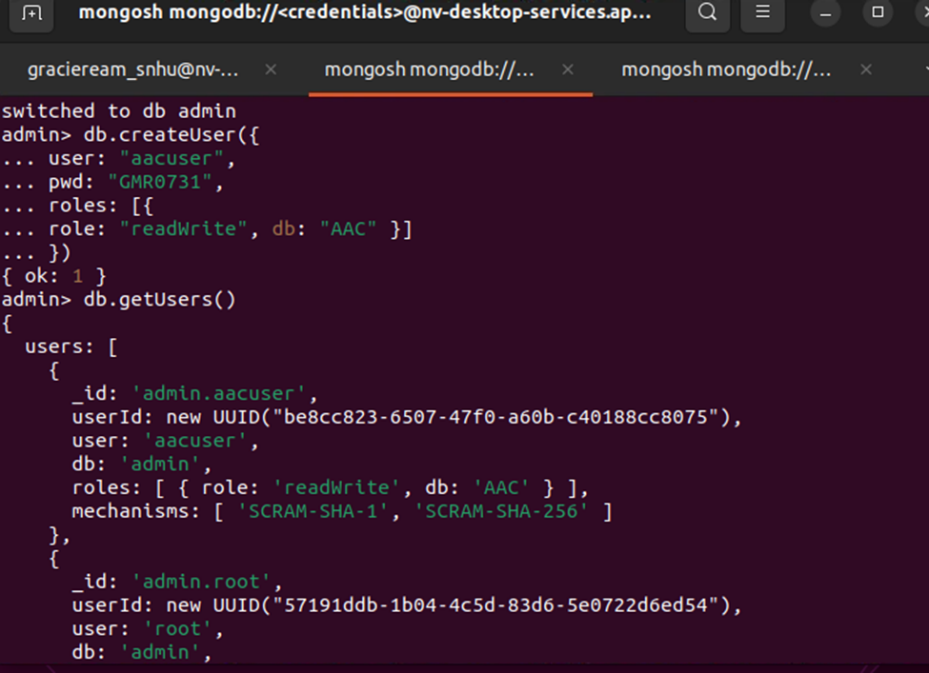
MongoDB was chosen because of its efficiency and seamless flow with the mongo driver, PyMongo. MongoDB provides for easy connectivity and easy usage so the user does not need to do any extra tasks.

**Usage**

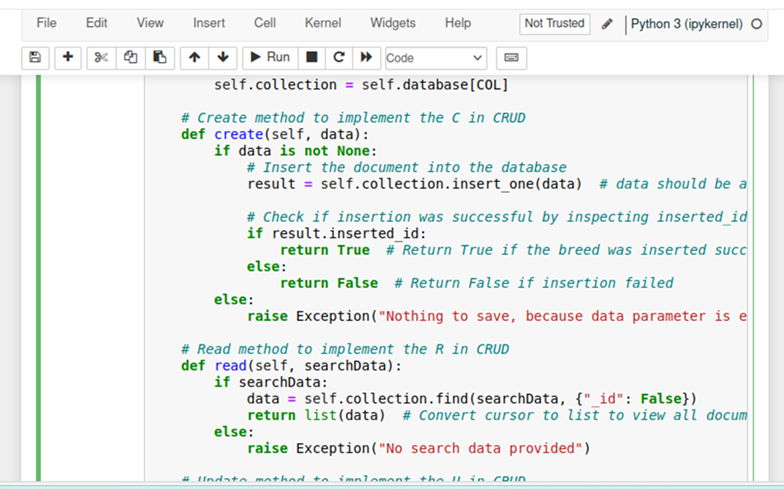
The following screenshots will show how the user can interact with the project and use it for themselves with explanations of the screenshots.

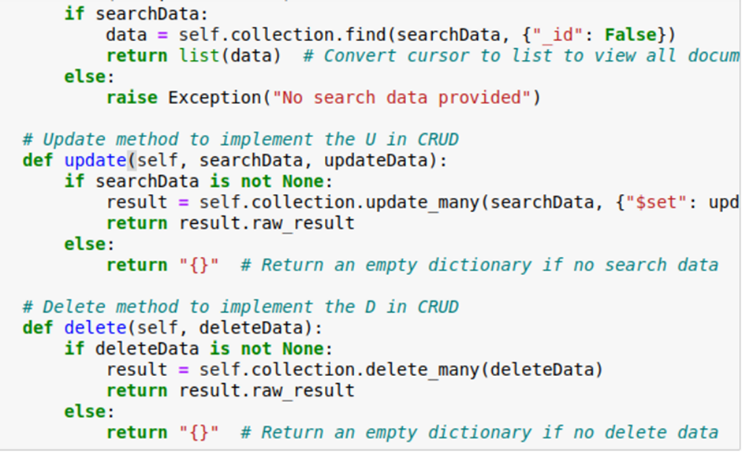
First the user should import the AAC shelter outcomes csv file while in the terminal. Then, create a personal user account and ensure connection to the database.



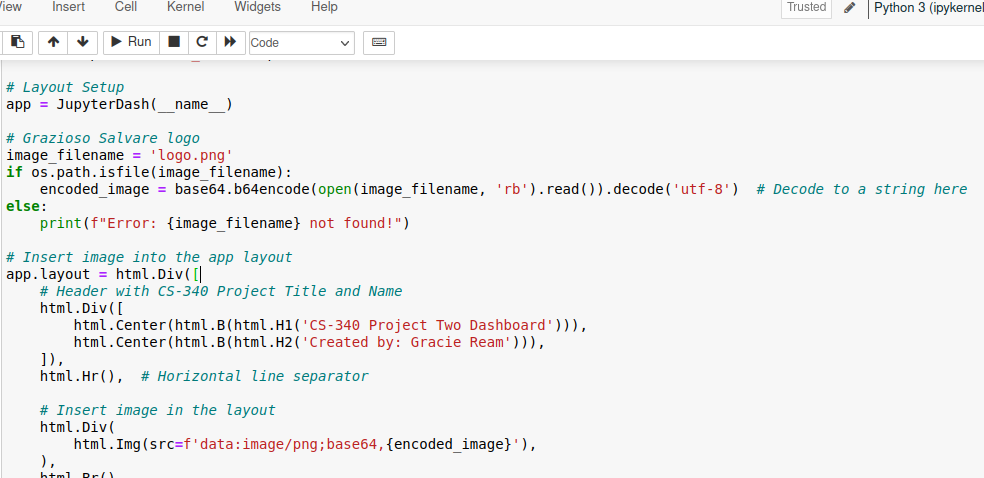
****

Following these steps, the user should create a CRUD module in a python environment that allows users to create, read, update and delete instances in the database.





The user should next navigate to the ipnyb file, where they can test instances of their database and ensure that the dashboard loads properly. In the case of project two, the user should have a geolocation chart, another example of a chart, a drop-down menu with different options, your name and header included, and an interactive data table.



Implementation of user’s name and header name/logo and dashboard to verify it appears.



A red line drawing of a dog

Description automatically generated

Once the user ensures their dashboard shows the client’s logo and the header name, they can create the filter drop down bar, where the client is able to click what breed they want to filter out and view.

A screenshot of a computer

Description automatically generated

Next, the user will ensure there is an interactive data table.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

As well as a pie chart or some other chart that updates when filtered.

A screenshot of a computer

Description automatically generated

Then a geolocation chart that goes along with the pie chart.

A screenshot of a computer

Description automatically generated

After implementing these, the dashboard should appear as this and update as the user clicks different breeds and filters.

A screenshot of a map

Description automatically generated

The user should have a drop-down menu as well as a reset button displaying on the dashboard.

A screenshot of a computer

Description automatically generated

**Challenges**

Overall, with project two, I had some challenges while implementing the pie chart and getting it to load correctly on the dashboard. Other than that, there were basic syntax errors that kept my program from running in the beginning.

**Creator**

Gracie Ream