# CS 340 README Project Two

Nii Amatey Tagoe

## Grazioso Salvare (International Rescue-Animal Training Company)

Grazioso Salvare is an innovative international rescue-animal training company that works with animals to teach them specific skills or correct problem behaviors, including search-and-rescue. Grazioso Salvare identifies dogs that are good candidates for search-and-rescue training and these dogs are trained to find and help rescue humans or other animals, often in life-threatening conditions. Grazioso Salvare has agreed with a non-profit agency that operates five animal shelters around Austin, Texas to aid them in identifying dogs for training. This project is based on a software application that can work with existing data from animal shelters to identify and categorize existing dogs. This project includes a database and a client-facing web application dashboard through which users at Grazioso Salvare will access the database.

## Motivation

Shelter dogs can be trained to become service animals, which can help people with mobility issues, visual impairments, mental health challenges, or emergency medical response dogs. Animal rescues save countless animals and human lives, providing medical care and rehabilitation so that animals can return to loving homes. Animal rescue organizations promote animal welfare policies and advocate for animal rights.

## Getting Started

* To get started, I first verify access to the environment by starting up MongoDB and the Mongo shell, In Apporto, I opened the terminal window to access the Linux shell and uploaded the Austin Animal Center (AAC) Outcomes data set into MongoDB by importinga CSVfile named **“aac\_shelter\_outcomes.csv” using the appropriate** MongoDBimpor**t** tool. I created a new user account called “aacuser” for the database AAC in the Mongo shell by modifying the commands, so the account is “accuser”. This was done to ensure **user authentication to the database and collection** was generated. I develop a Python module by using object-oriented programming methodology, to enable CRUD functionality. I created a Crud class when instantiated provides functionality for a Create method which inserts a document into the AAC database and Animals collection, a Read method that queries for documents from the AAC database and Animals collection, an Update method that queries for and changes documents from the AAC database and Animals, and finally a Delete method that queries for removes documents from the AAC database and Animal collection.
* The next step was to review the dashboard specifications document provided by the UI/UX developer at Global Rain. By using the Crud Python module to run a retrieve-all query and import the data from MongoDB I created an interactive data table on a dashboard that shows an unfiltered view of the Austin Animal Center Outcome data set.

A screenshot of a computer

Description automatically generated

## The next step was to develop database queries that match the required filter functionality for the rescue type and preferred dog breeds. I also created a create interactive options that allow for the selection of data based on the filtering function such as radio items in an IPYNB file. These interactive options are enable to control a bar chart which a dashboard widget on the website. The chart also responds to updates. Below are the screenshots.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Installation

Python: The CRUD Python module, which is crud.py was developed to provide a simple and efficient way to manage animal records in a MongoDB database for an animal shelter.

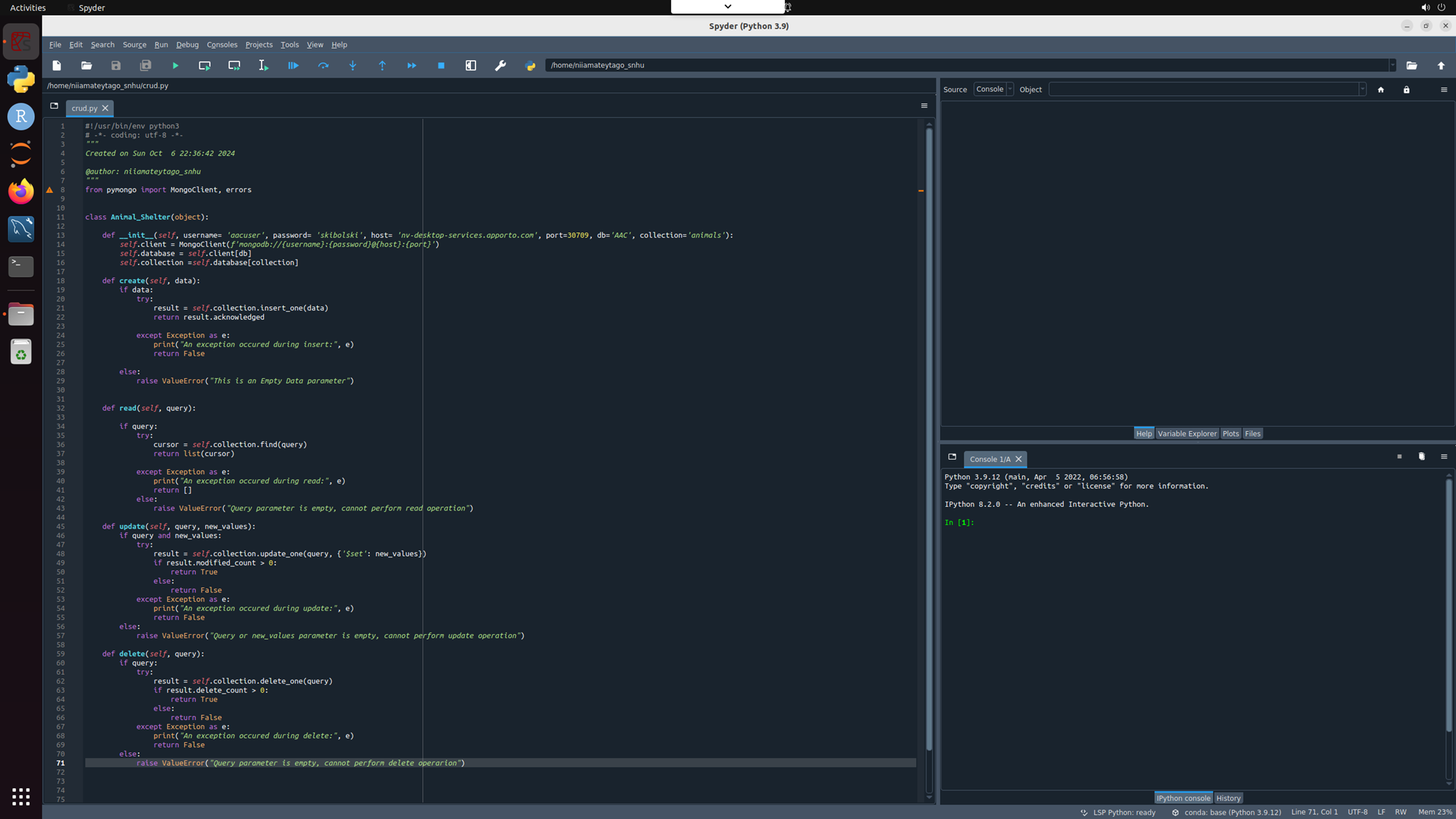
MongoDB was used as a database to store all the records of animals in the shelter.

Jupyter Notebook was used to create an IPYNB file in order to develop and test out the dashboard and its functionalities.

## Usage

### Code Example

*Below is a screenshot of crud.py and projectTwoDashboard.ipynd*

**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

This project involves developing an interactive web application for Grazioso Salvare animal shelter using Python. The application allows users to filter, visualize, and map animal data to aid in decision-making processes, particularly for rescue operations. The application is built using the Dash framework and interfaces with a MongoDB database to retrieve and manipulate data. MongoDB was chosen as the database for this project due to its flexibility in handling unstructured data. The animal shelter data includes various fields that may not be consistent across all records, and MongoDB's document-based storage allows for this variability. Dash is a productive Python framework for building web applications with interactive visualizations. It is ideal for data-driven applications and allows for seamless integration with Plotly for visualizations. Some challenges indentation error in the update\_dashboard callback function caused a syntax error.

Finally, this project successfully demonstrates the integration of MongoDB with a Python-based web application using the Dash framework. The interactive dashboard provides valuable functionalities for the animal shelter, allowing users to filter data, visualize breed distributions, and locate animals on a map.

## Contact

Your name: Nii Amatey Tagoe