# Grazioso Salvare README

## About the Project/Project Title

This project is a full stack application that displays the information from a MongoDB database onto a user-friendly dashboard.

## Motivation

This project was created for Grazioso Salvatore, a rescue animal training company. They are working with multiple shelters in the Austin, Texas area. The shelters will give the information of available pets. By completing this project, Grazioso Salvatore can more easily narrow their search on pets who could be successful candidates for different times of search and rescue training. This will allow them to filter quickly and efficiently for available pets in the area that meet the qualifications for specific types of rescue training.

## Getting Started

1. The information was imported into a database named “aac” onto MongoDB.
2. A different user authentication was created, which only allows modification for the aac database. This helps prevent any editing or accessing of other databases.
3. A program was created utilizing PyMongo.
4. Used Jupyter Notebook to ensure CRUD functionality.
5. Utilized Plotly Dash to create the web-based dashboard that integrated CRUD functionalities. Added filtering options along with visual tools.

## Installation

There were many tools used to create this program. First, access to a Linux terminal. MongoDB was used to first create the database and load the given data from the shelters. PyMongo was used in order to access our database and create our CRUD functionality utilizing Python. Jupyter Notebook was used to test our Python program. Plotly Dash allowed us to create the dashboard and create the visual graph and map.

**Resource Links**

* [MongoDB](https://www.mongodb.com/)
* [Python](https://www.python.org/)
* [Jupyter Notebook](https://jupyter.org/)
* [Plotly Dash](https://dash.plotly.com/)

### Code/Screenshots

First, the aac database was imported onto MongoDB.

A computer screen with white text

Description automatically generated

After, a custom user authentication was created, which only allows modification for the aac database. This helps prevent any editing or accessing of other databases.

**A computer screen shot of a program

Description automatically generated**

A program was created utilizing Python to access the database using the authentication created.

A screenshot of a computer code

Description automatically generated

The “create” and “read” functionalities were created.

A computer screen shot of text

Description automatically generated

The “update” and “delete” methods were created.

A screenshot of a computer code

Description automatically generated

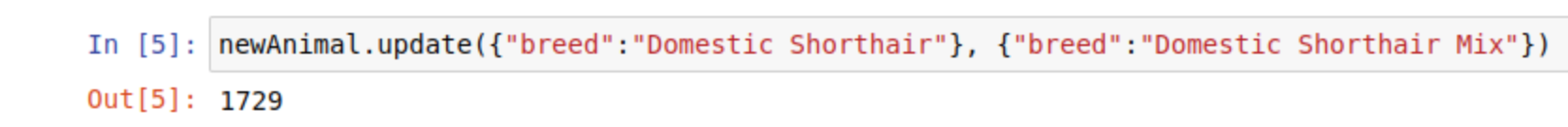
Testing in JupyterNotebook to ensure CRUD functionality was responding as expected.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated



A computer screen shot of a computer screen

Description automatically generated

Plotly dash was utilized to create the dashboard. The examples below show the different filtering options that have been instantiated and the map/graph update accordingly.

Water Rescue:

A screenshot of a computer

Description automatically generated

Mountain/Wilderness Rescue:

A screenshot of a computer

Description automatically generated

Disaster Rescue/Individual Tracking:

A screenshot of a computer

Description automatically generated

Reset (No filters applied):

A screenshot of a computer

Description automatically generated

## Contact

Your name: Tiffany Montero