# Project Two README

## About the Project/Project Title

This project is a simple program that runs CRUD operations using Python and the Mongo Database. The company, Grazioso Salvare, has asked us to make a program that can help them easily search through a database for potential rescue dogs. Its main use is to search a database of animals in nearby animal shelters. On the dashboard, you can select any dog and see its location, and there is also a pie chart that shows all the dog breeds.

## Motivation

This project was created to simplify the process of creating, reading, updating, and deleting objects in a database. With an easily accessible dashboard, potential rescue dogs can be found much quicker. Hopefully many lives can be saved by rescue animals by using this software.

## Getting Started

To run the project, you will need to download the Project Two.zip file (this is in the same location as the readme file) and unpack it. You also need to download the animal shelter database file. Once that is done upload the .py file and the .ipynb file to Jupyter Notebook. Once it is uploaded, open the .ipynb file and hit run. This will open the dashboard in a new window for you to interact with.

## Installation

MongoShell:

MongoShell can be downloaded from mongodb.com, just follow directions on the website.

PyMongo:

Pymongo can be installed through the mongo shell. Just enter the command “$ python3 -m pip install pymongo” into the console.

Jupyter Notebook can be downloaded through their website.

**Usage/Functionality**

The first thing to do is upload the database into MongoDB, as seen below.

**MongoDB Import:**

A screen shot of a computer program

Description automatically generated

**User Authentication Execution:**

A screenshot of a computer program

Description automatically generated

Next, we must upload the crud file and the project file to Jupyter Notebook. Once this is done you can run the program and it should look like this: On the dashboard you can filter by rescue type and a map will appear when an animal is selected.

**Reset Selection:**

**A screenshot of a computer screen

Description automatically generated**

**Water Rescue Selection:**

**A screenshot of a computer screen

Description automatically generated**

**Mountain Rescue Selection:**

**A screenshot of a map

Description automatically generated**

**Disaster Rescue Selection:**

A screenshot of a computer program

Description automatically generated**A screenshot of a map

Description automatically generatedTesting Screenshots:**

**Read and Create Code:**

## A close-up of a computer screen Description automatically generated

These Screenshots test the read and create code functionality.

A screen shot of a computer code

Description automatically generated**Update and Delete Code:**

A close up of a text

Description automatically generated

These Screenshots test the delete and update functionality.

**Tools:**

MongoDB works well with Python. This is mostly because they store data in very similar ways that can interface well with one another. This program uses the Dash Framework to display data. Using this anyone can create a well-designed web application. One upside of Dash is that you only have to use one language, Python.

**Challenges:**

I had a hard time developing the queries I needed for the radio button functionality. After much tinkering I discovered that the queries on this modules quiz were almost exactly what I needed in my code. Once I discovered that everything else went by smoothly.

## Contact

Name - Jonathan Borntreger

Email - jonathan.borntreger@snhu.edu