# Project Two README

## About the Project

This software is intended to be used by the rescue-animal training company Grazioso Salvare. The software will allow the company to identify potential candidate dogs for animal rescue training. A database of information about animals such as their breed, age, location, etc… can be interfaced with by the user via a web application interface on the Dash framework. These two ends are glued together via a python CRUD (Create, Read, Update, Delete) module that allows for interactions between the user and database.

## Motivation

The main purpose for this application is to better improve the efficiency of determining appropriate dogs to be candidates for rescue training by the Grazioso Salvare staff.

## Getting Started

To install and use this program locally, creation/importation of a database into MongoDB is required. After which the CRUD python module for facilitating users interfacing with the database can be used to verify any authentication that can optionally be set up within MongoDB. From there, and after starting MongoDB with the desired database imported, the web application can be used to display and interactively sort through the, in this case, the animal shelter’s data.

## Installation

A required list of software includes MongoDB, Pymongo, Python, and Jupyter Notebook. MongoDB is essential for importing/storing the database files and allowing for the python module to interface with something to perform its functionalities. Python is the language the application is written in since it closely mirrors MongoDB’s data types and syntax, a main reason they were chosen to integrate with each other for this project. Pymongo is an API that allows for python to interact with MongoDB. Finally, Jupyter Notebook helps in developing the Python modules for creating and reading documents in a Mongo database, and also helps in developing test scripts to verify the modules work correctly. The Dash framework helps in creating, python-driven, analytical web applications and provides the foundational libraries for the display/filtering functionalities of the dashboard.

## Usage

When fully integrated with the CRUD python module, and an associated Mongo database, deployment of the Grazioso Salvare program dashboard’s starting state for the data table should look something like:

Graphical user interface, text, application, email

Description automatically generated

And the starting state for the graph and geolocation map would look like:

Graphical user interface, application

Description automatically generated

Execution of the dashboard after filtering for Water Rescue dogs:

Graphical user interface, text, application

Description automatically generated

Execution of the dashboard after filtering for Mountain or Wilderness Rescue Dogs:

Graphical user interface, application

Description automatically generated

Execution of the dashboard after filtering for Disaster or Individual Tracking dogs:

Graphical user interface, application

Description automatically generated

Execution of the dashboard after it has been reset:

Graphical user interface, text, application

Description automatically generated

## Steps taken/Challenges Faced:

This project is the culmination of three separate development processes. The first was importing and understanding the creation of a database in Mongo. The second was implementing CRUD functionalities within a python module that interacted with that database. Finally, the third process was developing the actual web application that produced the dashboard output enabling the user to filter the data specifically to their liking. Out of all of these development phases, this final project, developing the web application that brings them together and enables the filtering functionalities, was the most challenging. However, the only challenge I faced was formatting the HTML syntax within the program correctly as I am not as familiar with that syntax.

## Contact

Noah Archibald