# Project Two README

## About the Project/Project Title

The Grazioso Salvare Dashboard allows for a visual way to access and manipulate a database. It presents the user with a table, geographical chart, and pie chart. Above the table are buttons which allow the user to sort the table based on breeds best suited for the categories of: Water Rescue, Mountain/Wilderness Rescue and Disaster Rescue/Individual Tracking. The geographical chart will update with the location of the selected animal on a map. The Pie chart will also change to reflect the percentages of dog breeds within your selected category.

## Motivation

The motivation behind this project was to create a framework to easily interact with a data table. While the code is currently designed to work with the aac\_shelter\_outcomes data table, it is easily refactorable to work with other data tables of the users choosing.

## Getting Started

To get started, you will need Mongo DB and Jupyter notebook. You will also need the CRUD python module, the accompanying “.ipynb” code and the Grazioso Salvare logo.

## Installation

For this to work, you will need MongoDB and the CRUD python module. You will also require the accompanying “.ipynb” code and the Grazioso Salvare logo. The python module, “. ipynb” code, and logo must all be put in the same directory. Run the “. ipynb” code through an IDE, preferably Jupyter notebook. Lastly, you will require the aac\_shelter\_outcomes.csv database. If you followed these steps correctly, you could run the “. ipynb” code in your IDE and you should receive an output directing you to the Dash framework.

**Links:**

**MongoDB:** <https://www.mongodb.com/>

**Jupyter Notebook:** https://jupyter.org/

**Functionality:**

CRUD python module:A screen shot of a computer program

Description automatically generated

**Code Setup:**

The following code was written in Jupyter notebook. It interacts with the CRUD python module and the aac\_shelter\_outcomes.csv database. It also implements dash framework to display the interactive UI.

**Imports:**

A screenshot of a computer

Description automatically generated

**Data Manipulation:**

A computer screen shot of a message

Description automatically generated

**Dashboard Layout:**

A computer screen shot of a program code

Description automatically generated

**Buttons:**

A screen shot of a computer code

Description automatically generated

**Pie chart:**

A computer code with colorful text

Description automatically generated with medium confidence

**Map:**

A computer screen shot of a computer code

Description automatically generated

**Output:**

**Full Logo Image:**

**A screenshot of a computer

Description automatically generated**

**Default Table:**

**A screenshot of a map

Description automatically generated**

**Water Rescue:**

**A map with a red and white text

Description automatically generated with medium confidence**

**Mountain/Wilderness Rescue:**

**A map with a logo and a map with a map and a map with a map with a map and a map with a map with a map with a map and a map with a map with

Description automatically generated**

**Disaster/Individual Tracking:**

**A map with a red logo

Description automatically generated with medium confidence**

**Reset:**

**A screenshot of a map

Description automatically generated**

**Identify any challenges that were encountered and explain how those challenges were overcome:**

The process of completing this project proved not a significant challenge. The only thing that took time to complete were the buttons and displaying the logo. These two items created several errors due to improper syntax. These were fixed by simply reading the errors upon running the code and researching how to best achieve these tasks.

**Contact:**

Jeremy Depenhart