# Project Two README

## About the Project/Project Title

This project is an interactive dashboard for the company Grazioso Salvare. This dashboard uses a python CRUD module to connect to a database containing records from multiple animal shelters around the Austin, Texas region. This dashboard allows users to view and categorize animals based on the company’s requests. The dashboard includes pages of data containing selectable animals. The animal selected will then be pinpointed on a map below, which will also contain the animals name if available. Additionally, a pie chart is included to showcase the breeds shown on the current page. The user can filter through the data by selecting the radio buttons in the dashboard, which will then change the shown data, chart, and map to reflect accordingly.

## Motivation

This project exists to provide a method for authorized users to have access to a database for CRUD functionalities, and to learn more about their implementations. It is also motivated by the need for the company, a rescue-animal training company, to identify dogs with key characteristics to train for different rescue jobs.

## Getting Started

To get a local copy up and running, you will need to follow the following steps:

1. Start an instance of Mongo in your terminal without authentication
2. Import the aac\_shelters\_outcomes.csv as AAC database and animals collection into mongo
3. Open Mongo Shell and create the following users using the CreateUser function:
   1. Using the admin database, create an admin user with permission to read and write all databases on file. Give a secure username and password.
   2. Using the AAC database, create an admin for the database who can read and write only to the AAC database. This user should also have a secure username and password.
4. Upload AAC\_Shelters.py and ProjectTwoDashboard.ipynb to Jyputer notebook to run the application

## Installation

*MongoDB:*

This tool allows access to the database system and provides methods to creating authenticated users, which is necessary for this project. This install will be specific to your system. Install manual should be followed from [Install MongoDB — MongoDB Manual](https://www.mongodb.com/docs/manual/installation/?_ga=2.106587843.1987463599.1675629606-329420269.1672779152).

*Python:*

Python is used to create the beginning CRUD functions, create and read, in this project. It is also used to create basic tests for those functions. If not already on your system, you will need to [Download Python](https://www.python.org/downloads/) .

*Pandas:*

This is a python library used to create the dataframe within the dashboard to be used. If not already installed on your machine, it can be installed with using the pip command to install after installing python.

*Dash:*

This is a required framework to run this application. It is used to display the datatable, create interactive buttons and elements, and provide layout areas for the map and pie chart If not already on your machine, it will need to be installed here. Manual should be followed from: [Part 1. Installation | Dash for Python Documentation | Plotly](https://dash.plotly.com/installation)

*Jupyter:*

This is used to run the application of this project.

*Required files for this project:*

AAC\_Shelter.py

ProjectTwoDashboard.ipynb

aac\_shelter\_outcomes.csv

## Usage

### Code Example

In this code, you’ll need to import the required database into MongoDB. The following in an example of a successful import:

Text

Description automatically generated

You will also need to create two authenticated users: an admin, and a user of AAC database. The following is an example of successful user authentication execution, which will need to be done for both users:

Text

Description automatically generated

Text

Description automatically generated

Next you will need to be sure that the ProjectTwoDashboard.ipynb contains your user’s username and password to create an instance of the CRUD module:

Graphical user interface, text

Description automatically generated

**How It Was Completed**

Interactive filtering was added using radio buttons and an app callback. The app callback is used when a radio button is selected. At this time it calls to the CRUD modules read function with a filter for the wanted animal type, breeds, sex, and age for the rescue method:

*A picture containing text

Description automatically generated*

Example of first filter:

*Text

Description automatically generated*

A pie chart was then added to the dashboard via an app callback, with its function using pandas to pass in the animals breeds as the charts data and provide a title for the user:

Graphical user interface, text, application, email

Description automatically generated

Lastly, a geo map was added. This was modified from the base code to have the tool tip show on the selected animal in the dashboard. This was accomplished by using the iloc function and passing in the selected row array:

Graphical user interface, text, application, email

Description automatically generated

### Challenges

The main challenge I faced in the development of this project was updating the geo map to reflect the selected animal. I easily got it to showcase the first animal shown on each of the filtering options, but couldn’t get it to change otherwise. I first tried using the loc function and passing in the latitude and longitude of the selected row. After being unsuccessful with this, I looked more in pandas documentation and decided to use iloc, as then I could use integers instead to indicate the positions.

**Results:**

The following screenshots are the various display pages. I couldn’t fit the logo, dashboard, map, and pie chart all in one photo so only the opening photo contains the company’s logo.

Opening dashboard:

A picture containing text

Description automatically generated

Graphical user interface, map

Description automatically generatedWater rescue filter:

Mountain/Wilderness filter:Graphical user interface, map

Description automatically generated with medium confidence

Map

Description automatically generatedDisaster filter:

## Graphical user interface Description automatically generated with medium confidenceReset filter:

## Contact

Katelyn Perry