6/19/2023

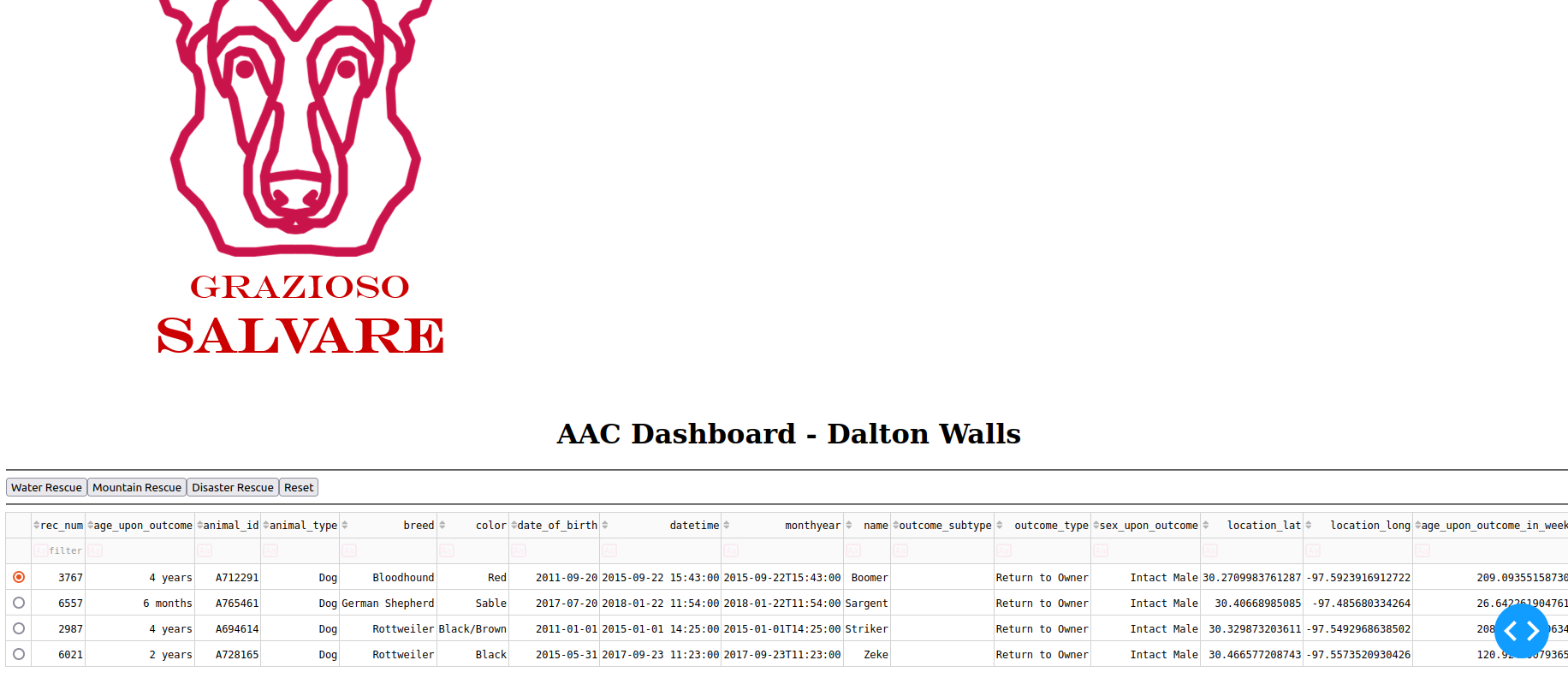
Dalton Walls

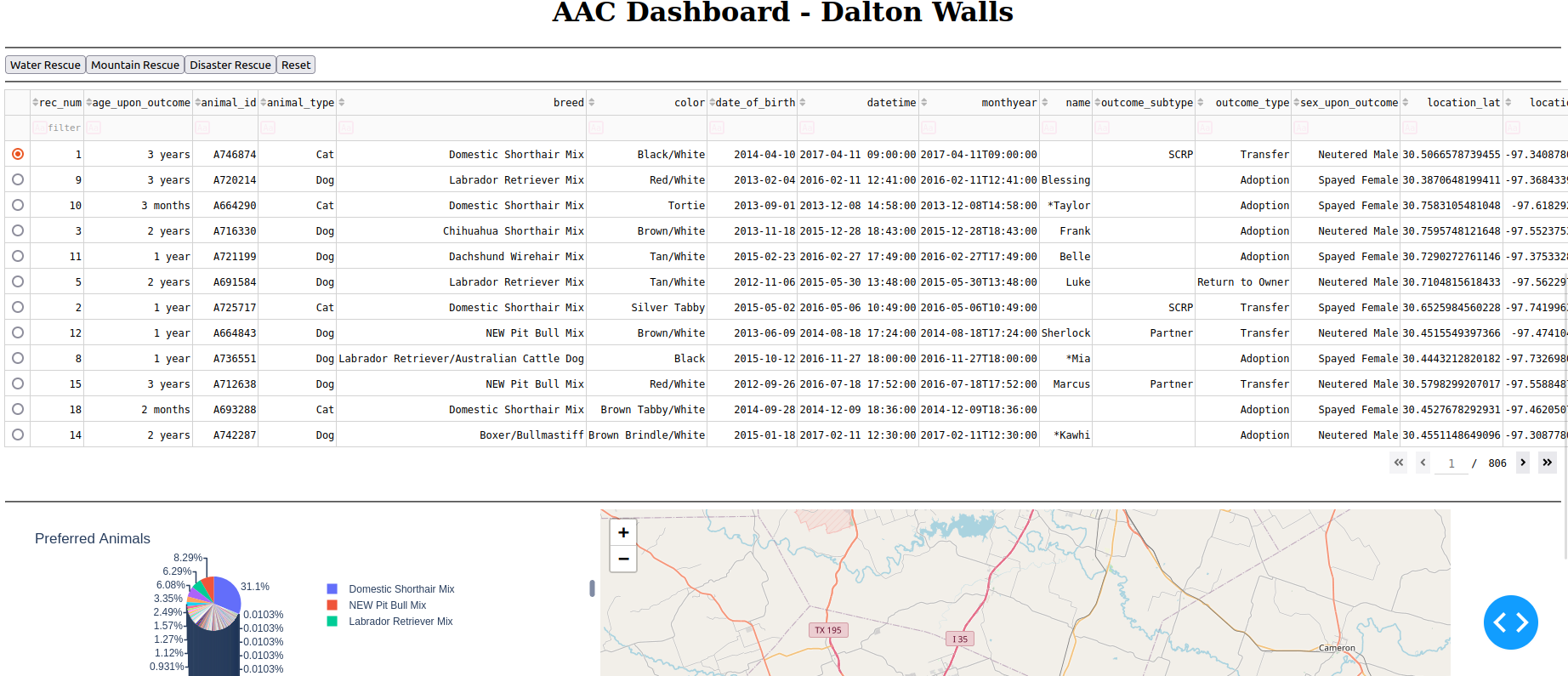
README for Interactive Dashboard View of Large Dataset

(Grazioso Salvare)

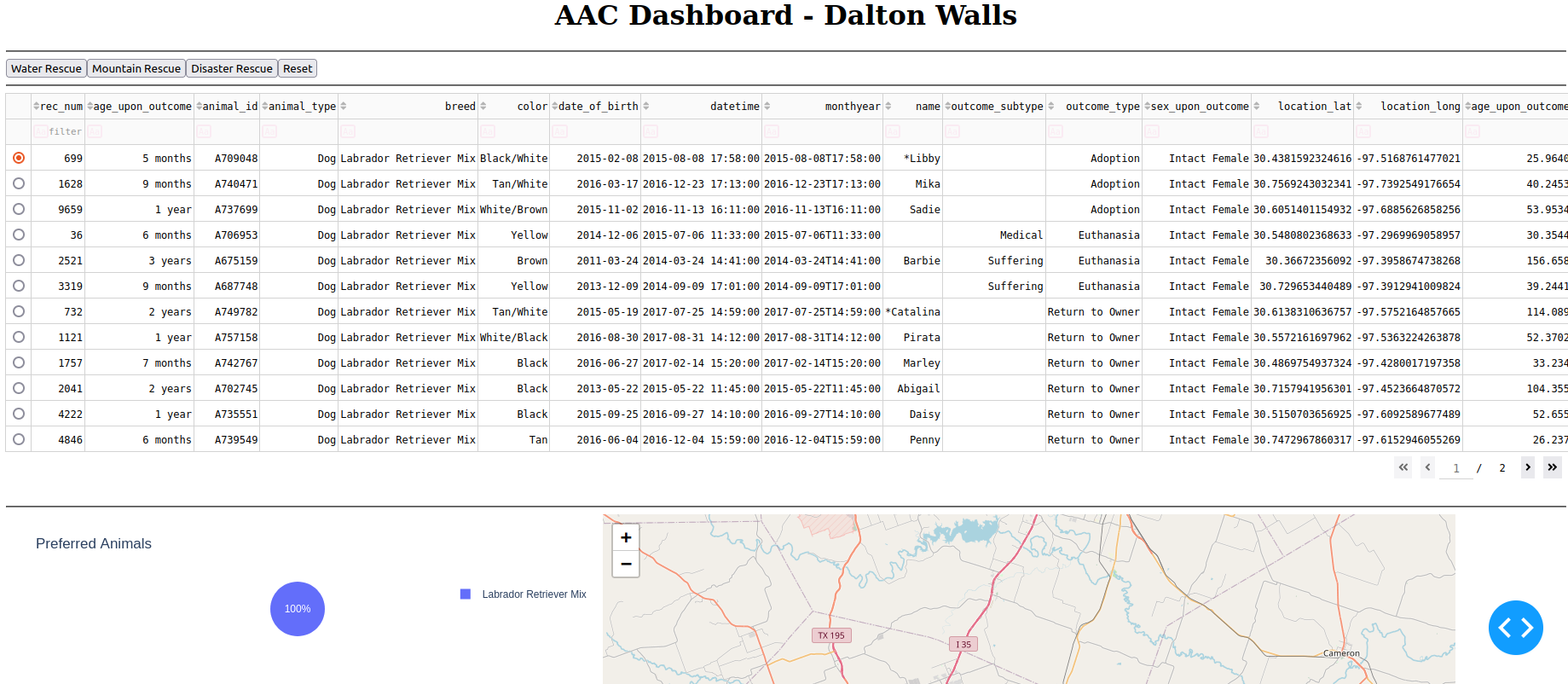
This project was to take a large dataset of animals for Grazioso Salvare, use them to create a database in MongoDB, create a Python middle layer with CRUD functionality, and finally to create an interactive Dashboard to view the database.

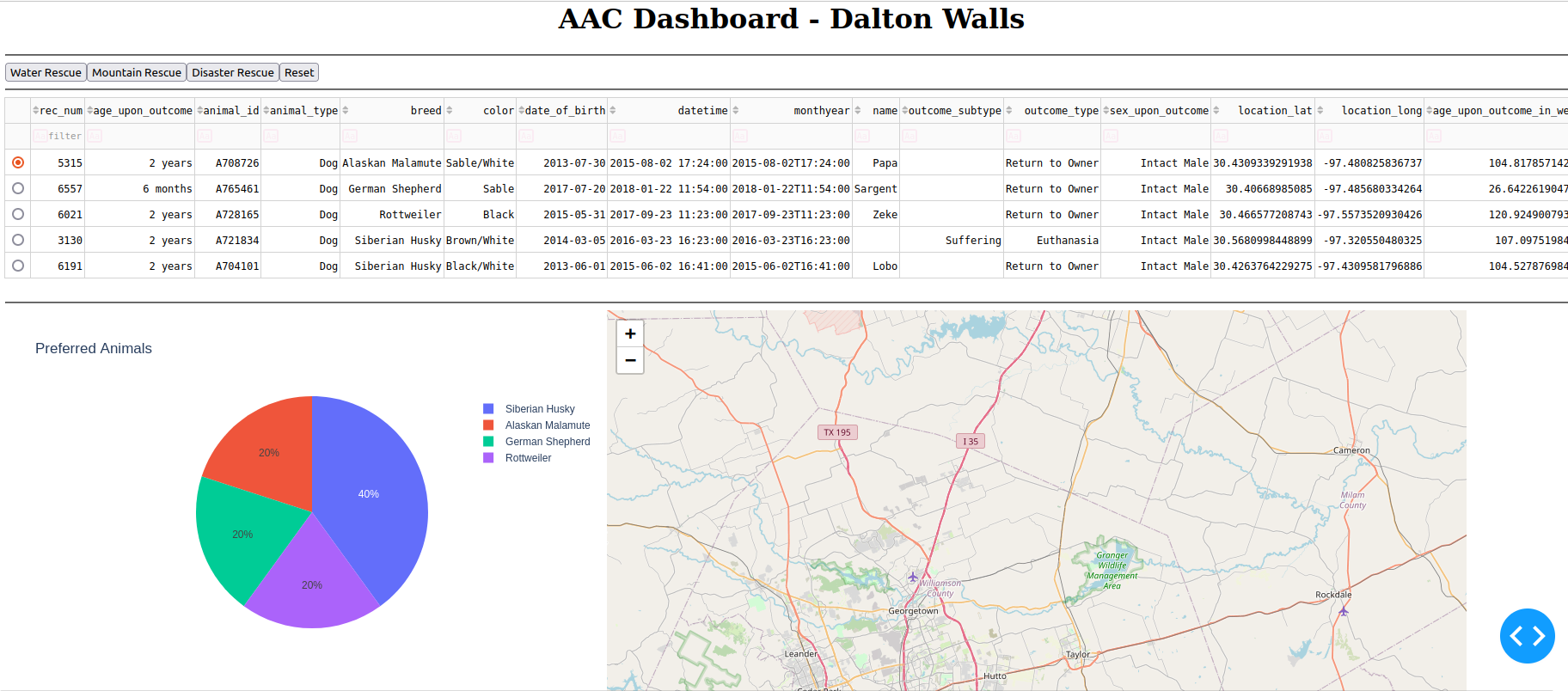
One requirement for this dashboard was to display the company’s logo as well as a unique identifier for myself:

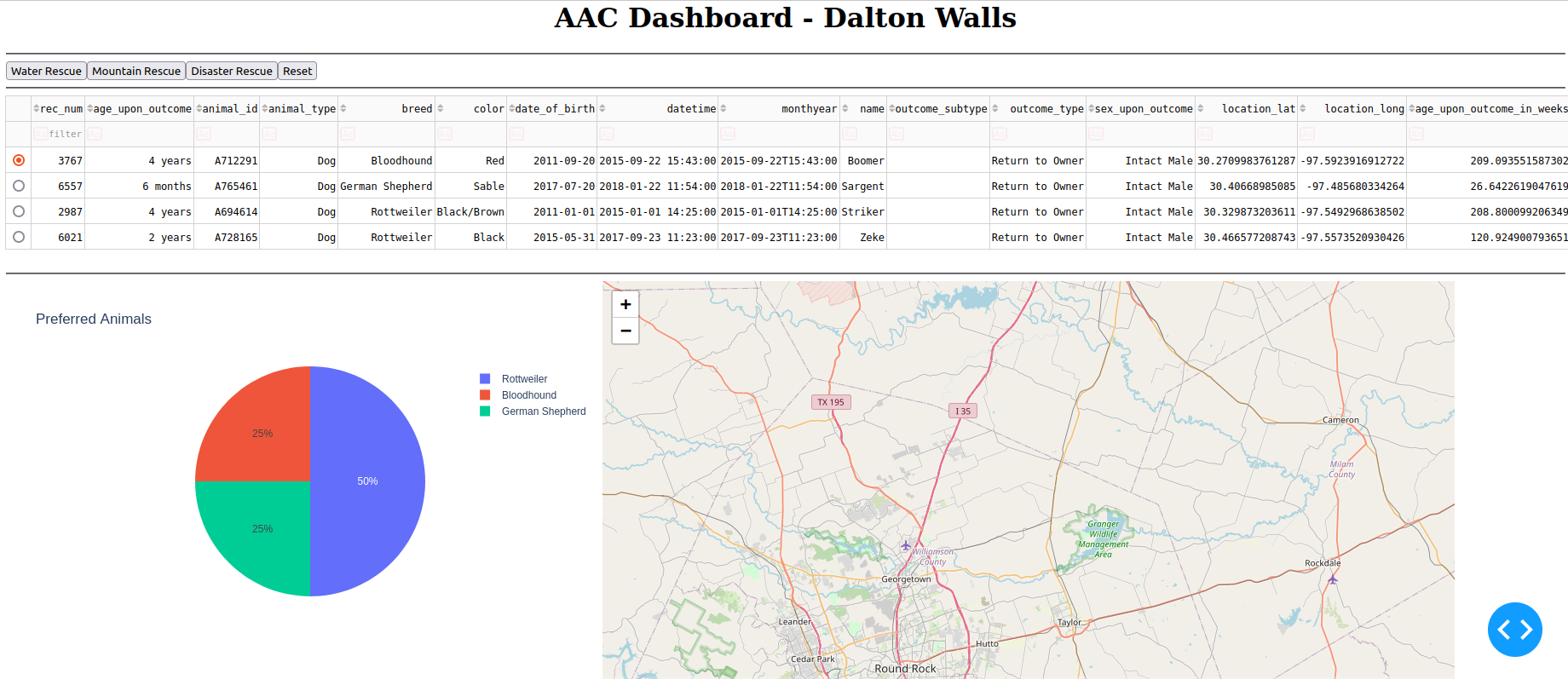
 Another requirement was to display a map corresponding with coordinate locations in the database, and a pie chart corresponding with the amount of breeds in the current filtered view of the database. Here is the unfiltered view of the entire database:



Here is a look of the dashboard when the first filter button (water rescue) is selected:

  
  
  
Mountain Rescue:

  
  
  
and Disaster Rescue:

  
(**NOTE:** THE MAP IS UPDATING AS WELL, IT JUST DOES NOT LOOK LIKE IT IN THESE SCREENSHOTS, THE ANIMALS ARE VERY CLOSE TOGETHER IN THE SAME AREA)

**Tools Used**

**Database** – We used MongoDB for the database due to its open-source and flexible nature. We also chose it because its easy to interface with Mongo using Python code, thanks to the PyMongo API.

**Dashboard** – We used the DASH framework here due to the fact that its an easy and quick way to visualize data. We also imported Plotly modules for the graphs and maps.

**Code** – We used a middle layer of code written in Python to facilitate CRUD operations. We edited/tested this code in Jupyter Notebooks due to the fact that it can easily test a hosted server and related scripts. We also used some HTML for the page layout and interactive elements.

**Steps Taken**

1.) Create database in MongoDB

2.) Test

3.) Create Python CRUD module

4.) Test

5.) Create .ipnyb file in Jupyter Notebooks to test functionality

6.) Create dashboard using DASH framework

7.) Test in Jupyter Notebooks

**Challenges**

My main challenge here (as this was a school project) was understanding what exactly we were doing and using at any given time. A lot was thrown at us very fast with little explanation. Our school would give us template starter files to finish with us having very little understanding of the code. What troubled me the most was not knowing where all of the code came from, especially because I am used to using a more traditional IDE and not Jupyter Notebooks. We were using so many different new modules/frameworks/etc. (Plotly, DASH, MongoDB, Python, Jupyter, HTML) that I often found myself pretty confused on what code was what and how it all worked together. Additionally, many of the template files we were supplied were problematic from the start and required additional work to debug. A lot of the solutions, hints, and code stubs we were given were also not set up right. I would say that the biggest challenge here was basically working blind and being thrown into the deep end of an already started project. I can understand the value for a programmer to find solutions on their own, but I also think there’s a limit to that -- especially in an academic setting.

I overcame these challenges by Googling specific problems, and found that my biggest issues – the ones that took me hours upon hours to figure out – were simply due to errors in the directions and supplemental materials.

This course and project seemed to be all about connecting different modules and files, forming a “full stack”. I did successfully do that in the end, and I am glad for the experience, but I don’t think the course provided us with the knowledge that it promised. My main takeaway here is to do your own research, read official documentation, and use Google instead of trying to discern meaning from poorly revised instructions.

QUESTIONS:

1.) How do you write programs that are maintainable, readable, and adaptable? Especially consider your work on the CRUD Python module from Project One, which you used to connect the dashboard widgets to the database in Project Two. What were the advantages of working in this way? How else could you use this CRUD Python module in the future?

**Writing maintainable programs involves writing modular code that works together well without circular dependencies. Writing readable programs involves using clean code and proper comments. Writing adaptable code involves separation of responsibilities, and modularity again. The advantages of using a CRUD module instead of writing that behavior in one class is that it was easier to debug and focus on each module instead of all lines of code at once which would have been hard to read and think through. I could use this in the future if I ever needed to do CRUD operations on a database.**

**2.)**How do you approach a problem as a computer scientist? Consider how you approached the database or dashboard requirements that Grazioso Salvare requested. How did your approach to this project differ from previous assignments in other courses? What techniques or strategies would you use in the future to create databases to meet other client requests?

**I approach a problem in computer science by breaking it down into smaller steps. My approach differed here in the sense that I mainly had to Google how to connect different modules that were already written for me instead of writing my own code. In the future, I would start completely over in my learning about databases and learn it from scratch.**

**3.)** What do computer scientists do, and why does it matter? How would your work on this type of project help a company, like Grazioso Salvare, to do their work better?

**Computer Scientists are people whose job it is to perform research in the field of CS. That matters because our world is so dependent on computers that finding new and efficient ways of computing can make things in the world easier. Our work for this project could help a company like Grazioso Salvare understand their data better and be able to search through their data better.**