Ethan Klukkert

Professor Paruchuri

CS-340 Project Two

19 February 2023

Project Two README

Functionality

Grazioso Salvare is a company that specializes in training rescue dogs for a variety of disaster types such as water rescue, mountain and wilderness, and individual tracking. Grazioso Salvare requested a full stack application that works with existing animal shelter data, which allows for filtering options to display data in a meaningful way. Specifically, the client requested that the data be displayed in an interactive data table, map, and second chart of my choosing. I chose to display the data in a pie chart. The data should be filtered according to rescue type, and the map should display individual animal locations if selected. Below, we can see the data displayed, a map, and a pie chart in several screenshots:

SNHU CS-340 Dashboard

Ethan Klukkert, Dash and Python/MongoDB

Rescue Type:

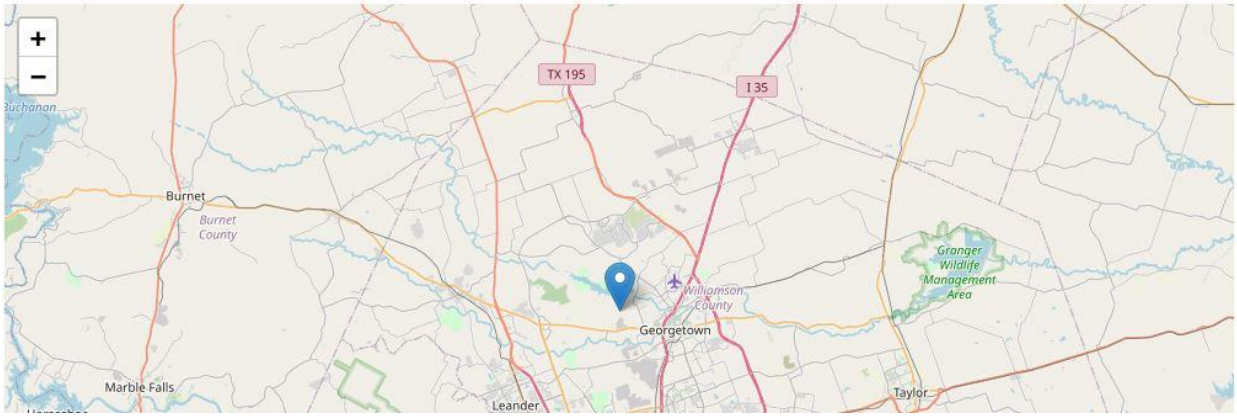
Reset

Rescue Type:

Reset

	1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthy
○	2	1 year	A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49
○	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41
○	10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01	2013-12-08 14:58:00	2013-12-08T14:58
○	11	1 year	A721199	Dog	Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49
○	12	1 year	A664843	Dog	Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24
○	8	1 year	A736551	Dog	Labrador Retriever/Australian Cattle Dog	Black	2015-10-12	2016-11-27 18:00:00	2016-11-27T18:00
○	13	1 year	A700408	Cat	Domestic Shorthair Mix	Brown Tabby/White	2014-04-13	2015-04-15 13:34:00	2015-04-15T13:34
○	14	2 years	A742287	Dog	Boxer/Bullmastiff	Brown Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30
○	15	3 years	A712638	Dog	Pit Bull Mix	Red/White	2012-09-26	2016-07-18 17:52:00	2016-07-18T17:52
○	16	5 years	A723742	Dog	Miniature Schnauzer Mix	Black/White	2011-04-05	2016-04-10 17:27:00	2016-04-10T17:27
○	17	6 months	A668960	Dog	Pit Bull Mix	Blue/White	2013-06-12	2013-12-27 16:56:00	2013-12-27T16:56
○	5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48
○	6	5 years	A696004	Dog	Cardigan Welsh Corgi Mix	Sable/White	2010-01-27	2015-01-28 10:39:00	2015-01-28T10:39
○	7	2 years	A673830	Dog	Pit Bull Mix	Black/White	2012-03-03	2014-03-19 15:15:00	2014-03-19T15:15
○	4	7 months	A733653	Cat	Siamese Mix	Seal Point	2016-01-25	2016-08-27 18:11:00	2016-08-27T18:11

« ‹ 1 / 668 › »

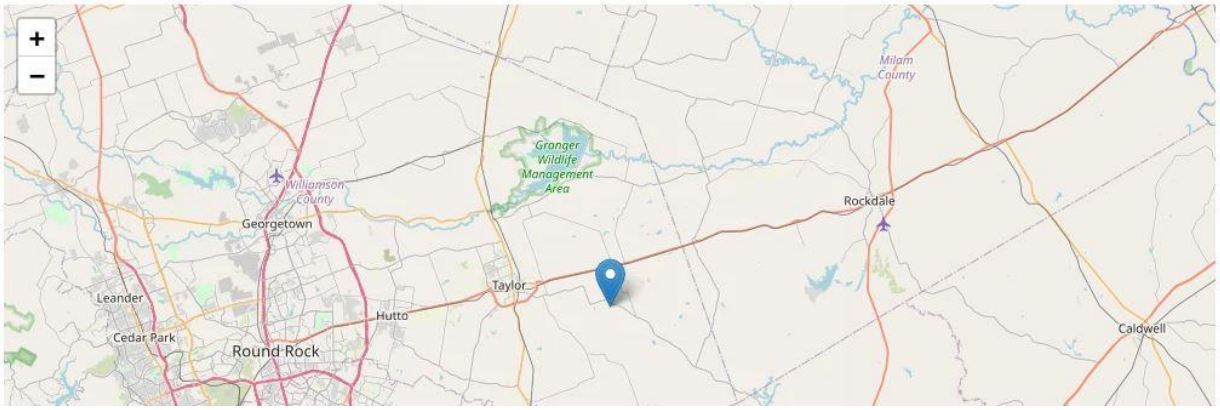


Rescue Type:

Water Rescue

	1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_sub
○	36	6 months	A706953	Dog	Labrador Retriever Mix	Yellow	2014-12-06	2015-07-06 11:33:00	2015-07-06T11:33:00		Medi
○	327	2 months	A759505	Dog	Labrador Retriever Mix	White	2017-08-02	2017-10-04 15:42:00	2017-10-04T15:42:00		
○	381	1 month	A736066	Dog	Labrador Retriever Mix	Tan	2016-08-03	2016-10-03 17:17:00	2016-10-03T17:17:00		
○	699	5 months	A709048	Dog	Labrador Retriever Mix	Black/White	2015-02-08	2015-08-08 17:58:00	2015-08-08T17:58:00	*Libby	
○	732	2 years	A749782	Dog	Labrador Retriever Mix	Tan/White	2015-05-19	2017-07-25 14:59:00	2017-07-25T14:59:00	*Catalina	
○	1121	1 year	A757158	Dog	Labrador Retriever Mix	White/Black	2016-08-30	2017-08-31 14:12:00	2017-08-31T14:12:00	Pirata	
○	1608	1 month	A748988	Dog	Labrador Retriever Mix	Black/Tan	2017-04-02	2017-05-09 16:03:00	2017-05-09T16:03:00		Part
○	1628	9 months	A740471	Dog	Labrador Retriever Mix	Tan/White	2016-03-17	2016-12-23 17:13:00	2016-12-23T17:13:00	Mika	
○	1757	7 months	A742767	Dog	Labrador Retriever Mix	Black	2016-06-27	2017-02-14 15:20:00	2017-02-14T15:20:00	Marley	
○	1988	1 year	A762781	Dog	Labrador Retriever Mix	Black/White	2016-11-27	2017-12-03 13:09:00	2017-12-03T13:09:00		Part
○	2041	2 years	A702745	Dog	Labrador Retriever Mix	Black	2013-05-22	2015-05-22 11:45:00	2015-05-22T11:45:00	Abigail	
○	2061	3 months	A677594	Dog	Labrador Retriever Mix	Black/White	2013-12-26	2014-04-26 18:00:00	2014-04-26T18:00:00		Part
○	2225	2 years	A757341	Dog	Labrador Retriever Mix	Black/White	2015-09-01	2017-10-03 12:27:00	2017-10-03T12:27:00	19	Part
○	2458	1 month	A756114	Dog	Labrador Retriever Mix	Black/White	2017-06-22	2017-08-17 17:09:00	2017-08-17T17:09:00		In Ker
○	2640	2 months	A723708	Dog	Labrador Retriever Mix	Black	2016-02-05	2016-04-10 11:29:00	2016-04-10T11:29:00		Part

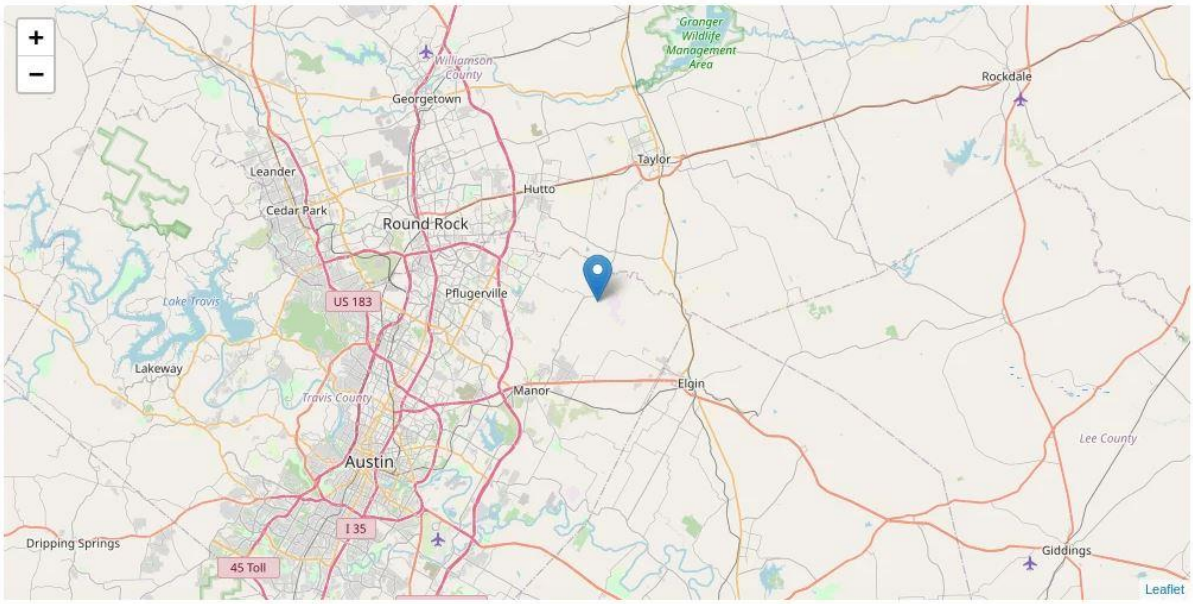
« ‹ 1 / 3 › »



Rescue Type:

Mountain Rescue

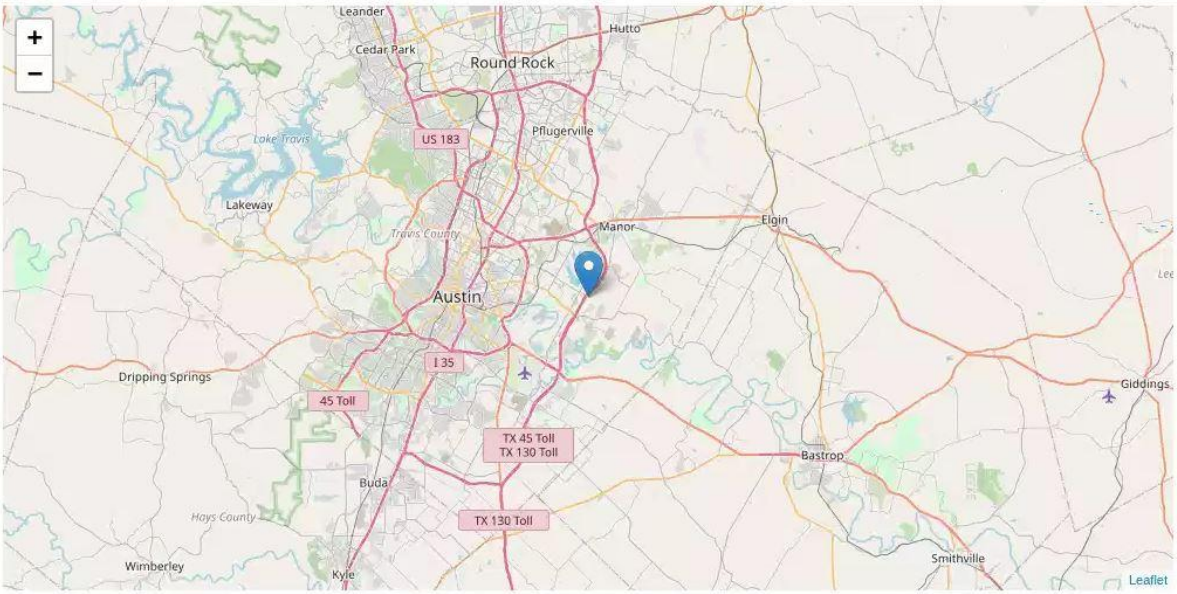
	1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	
○	5315	2 years	A708726	Dog	Alaskan Malamute	Sable/White	2013-07-30	2015-08-02 17:24:00	2015-08-02T17:24:00	Papa		Retu
○	6557	6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Retu
○	6021	2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Retu
○	3130	2 years	A721834	Dog	Siberian Husky	Brown/White	2014-03-05	2016-03-23 16:23:00	2016-03-23T16:23:00		Suffering	
○	6191	2 years	A704101	Dog	Siberian Husky	Black/White	2013-06-01	2015-06-02 16:41:00	2015-06-02T16:41:00	Lobo		Retu

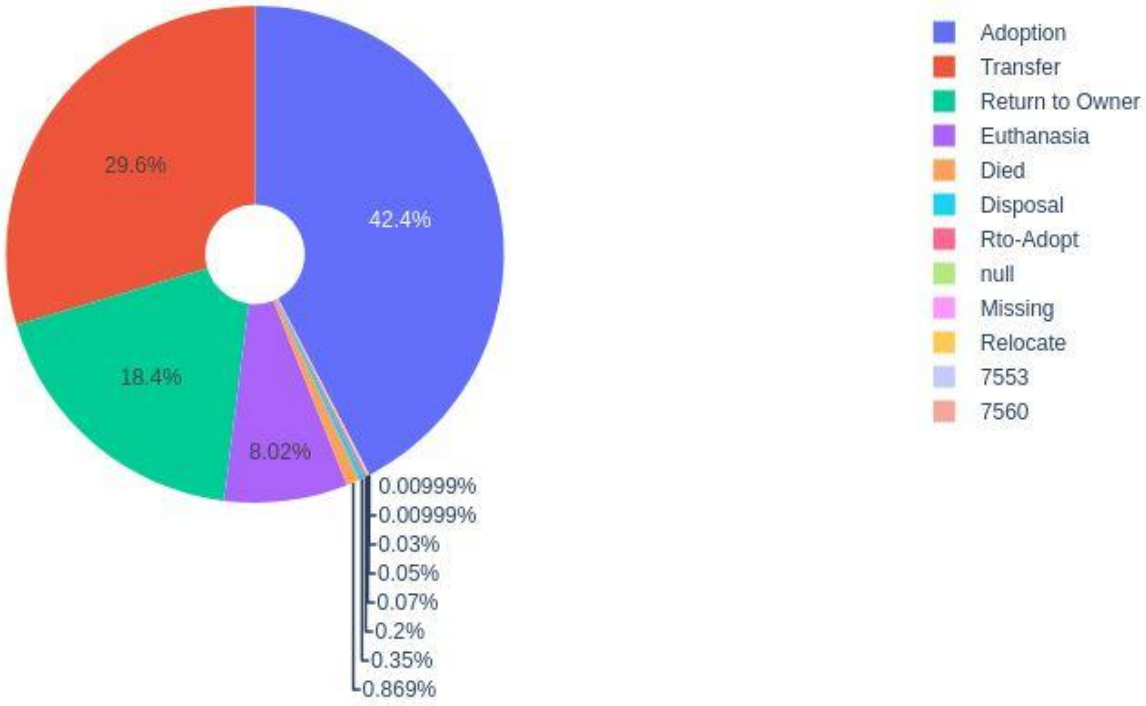


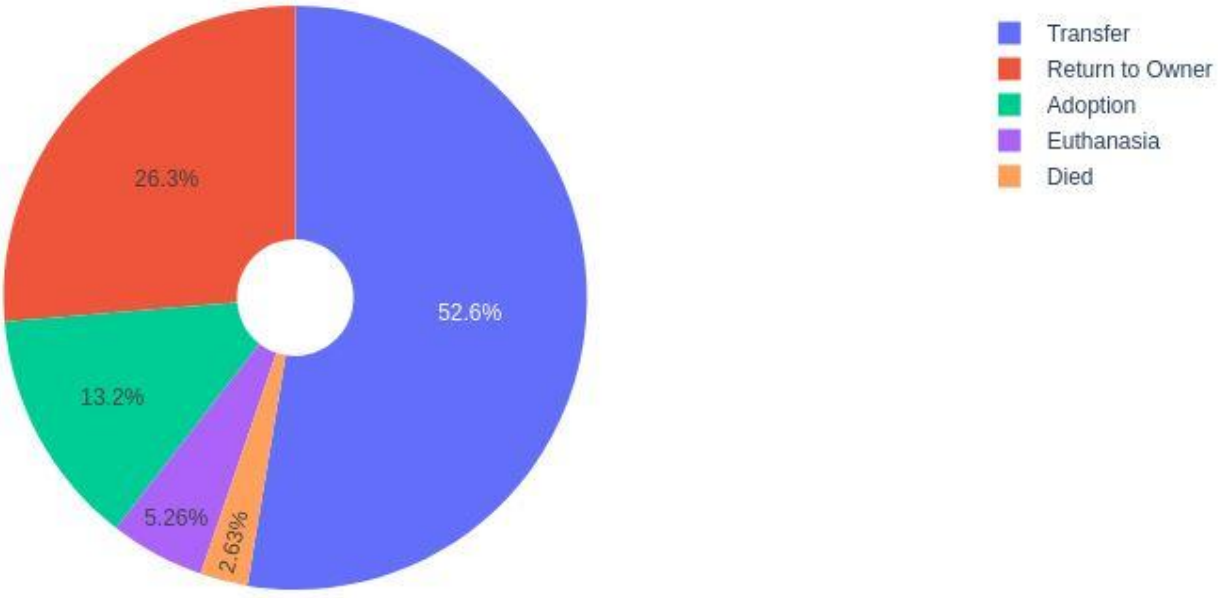
Rescue Type:

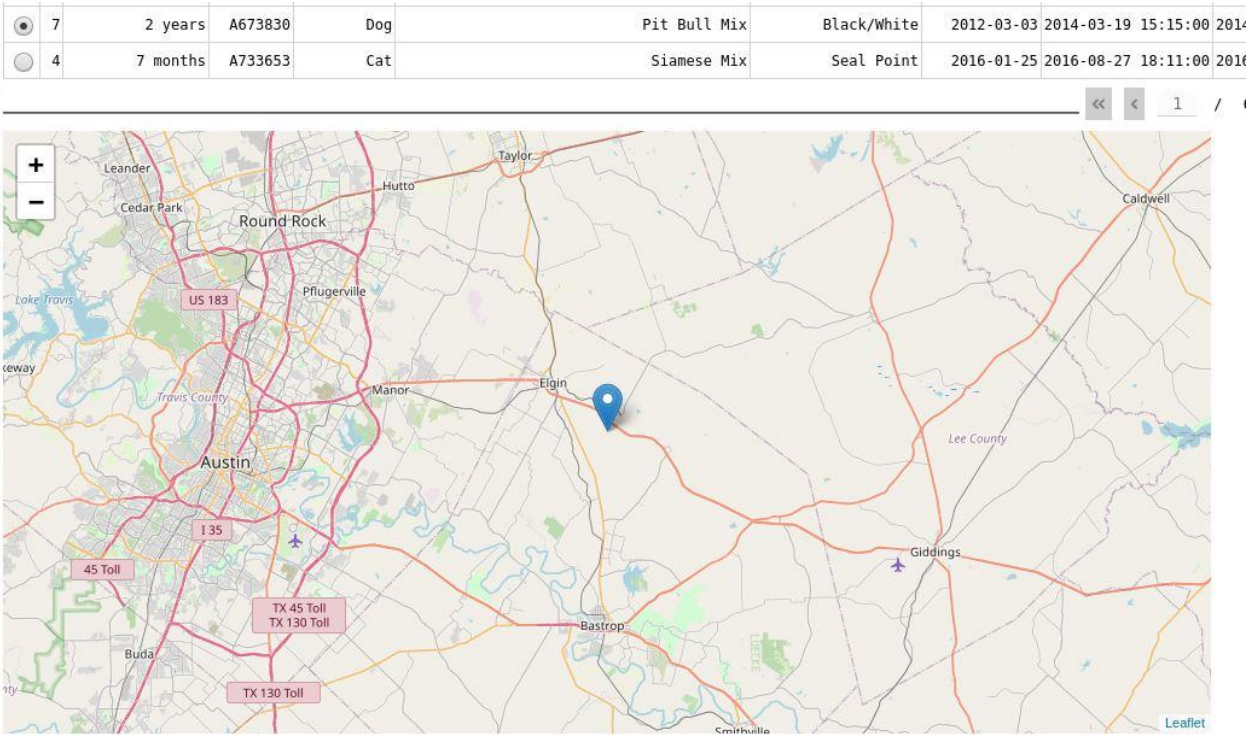
Disaster Rescue

	1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	o
○	3767	4 years	A712291	Dog	Bloodhound	Red	2011-09-20	2015-09-22 15:43:00	2015-09-22T15:43:00	Boomer		Retu
○	6557	6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Retu
○	2987	4 years	A694614	Dog	Rottweiler	Black/Brown	2011-01-01	2015-01-01 14:25:00	2015-01-01T14:25:00	Striker		Retu
○	6021	2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Retu




29.6%
42.4%
18.4%
8.02%
0.00999%
0.00999%
0.03%
0.05%
0.07%
0.2%
0.35%
0.869%
Adoption
Transfer
Return to Owner
Euthanasia
Died
Disposal
Rto-Adopt
null
Missing
Relocate
7553
7560


26.3%
52.6%
13.2%
5.26%
2.63%
Transfer
Return to Owner
Adoption
Euthanasia
Died



I also have a screencast demonstrating the functionality in action.

Tools Used and Why

MongoDB is a non-relational data platform that is used in this project. We used MongoDB because of how quickly queries can be structured, indexes can be established for quick access, and how Python works very well as the middle-ware. Python has a driver called Pymongo that interfaces with MongoDB to allow a frontend to be connected easily to a backend, in this case the database. The lack of tables in a NoSQL database allows a Python frontend to work smoothly with the data, as it can be easily converted to a dictionary type. The Python frontend used was the Dash framework. Dash allows the web application to be handled by the view and controller structures. The framework allowed us to set up HTML, CSS, and data state changes to manipulate the data displayed. We are able to use “callback” annotations to get input and output data within the frontend to properly manipulate and share changes on the screen, such

as a filter setting change or page change. The sources that particularly were helpful were the documentations of Dash and Plotly Express linked here:

Dash: https://dash.plotly.com/dash-core-components

Plotly Express: https://plotly.com/python-api-reference/generated/plotly.express.pie.html

Steps For Completion

First, I had to set up MongoDB with a CRUD module and authentication. The authentication is done within the MongoDB shell by creating an admin account, then initializing another account with the admin account. The collection can then be loaded into the database that the other account has permissions to. Then, the CRUD module is created using the Pymongo driver. This module handles the basic logic of CRUD by handling hard-coded authentication and queries. Next, I set up the Dash framework dashboard by iterating on each component. First I set up the datatable and map. Then, I made a dropdown that changes the datatable which later causes a new query. The new queries retrieve data according to the specifications of Grazioso Salvare, which then are displayed in the table. Next, the table allows rows to be selected, where the latitude and longitude data points are sent to the map, allowing users to select animals and find their locations. Finally, the pie chart is implemented. I kept it simple by hard coding in the data that it uses, this being the outcome type of the animals.

Challenges Encountered

I came across several challenges throughout this project, mainly stemming from my unfamiliarity with the Dash framework. The first major issue I ran into was the passing of data from the filter dropdown to the datatable. I had to work with the datatable and which attributes to change in order to get the filter to work. Another issue I ran into was getting the row selection to work on the table. I had to be able to get the row selection to not only work, but be able to

retrieve the selection and output it to the map. Another issue I ran into was the pie chart. Again, most of my issues stemmed from how I did not know how the data could be passed from components while working with the Dash framework. The last issue I ran into was a but that occurred with the pagination and row selection. Row selections return the index of the data within the whole set, while the map could only get the data from the view model. Essentially, selecting row one on the first page returned the data index of “0”, but on the second page it returned “16”, while the map could only read 0-15 indices (depending on how many are allowed to display on each page, in this case it was 15). This was solved by using the modulo operator in Python by doing the index with a modulus of 15.

Besides the last issue, almost all of these issues were resolved by trial and error while examining the documentation of each component. I also looked up stackoverflow posts and YouTube videos to help me understand what kind of code the framework required to work properly. Overall, I learned that looking up problems works best as a process. First, I looked up the problem specifically to see if anyone had the same problem. Then, I searched for the whole problem in general, usually with an example to reference my issue with. Lastly, if all else fails (which it did quite often in this project), I would read the documentation in detail to verify that I am using the code properly.