# CS 340 README

## About the Project/Project Title

This project is about creating an easier way to develop a web application that connects a client-side user interface to a database. In this case, it is for Grazioso Salvare to increase and grow the company’s ability to find dogs that are available for adoption and that can be trained in search and rescue.

## Motivation

This project exists so that Grazioso Salvare can streamline CRUD functionality when interacting with their database from a user interface. This will allow for easier maintenance and scalability of their database.

## Getting Started

These steps will allow you to set up a project locally:

1. Have Linux operating system, Jupyter Notebook, and MongoDB installed.
2. Make sure to have your file you wish to import in csv format.
3. Use Linux Shell to upload the csv file into MongoDB
4. If needed, set up a user account for security reasons
5. Develop a python module using object-oriented programming to enable CRUD functions.
6. Create a testing script in Jupyter Notebook
7. Run the script to test

## Installation

The tools used to create this project are Python 3+/Jupyter Notebook, Linux operating system, PyMongo driver, and MongoDB. Users can install MongoDB directly from the MongoDB website. The same can be done for Jupyter Notebook and Linux as well. Websites are as listed: <https://www.linux.org/pages/download/> <https://jupyter.org/install> <https://www.mongodb.com/try/download/community>

**CRUD Python module**

The reason for a CRUD python module is so that there can be some sort of fundamental functions used to manage data in a software application. CRUD stands for Create, Read, Update, and Delete. These functions allow the maintainers of a database to easily modify a database. For Grazioso Salvare, it should help in finding dogs that fit their specific criteria to be trained in search and rescue.

**CRUD Python module Usage**

The Python driver that was used in this application was PyMongo. PyMongo was used in this application because it is the official MongoDB driver for synchronous Python applications. This is very helpful because all the tools needed can be downloaded for free. This makes it easier for this project to be open source and accessible on GitHub. The functionality of the CRUD operations is as follows:

* Create - adds a dog into the MongoDB database called AAC, true will be printed if added successfully.
* Read - allows a list of the dog to be found in the AAC database to be printed.
* Update - changes a specific attribute for a certain dog, example name, species, or age etc... This will return the number of objects modified.
* Delete - removes documents from AAC database, and returns how many documents were deleted as a result.

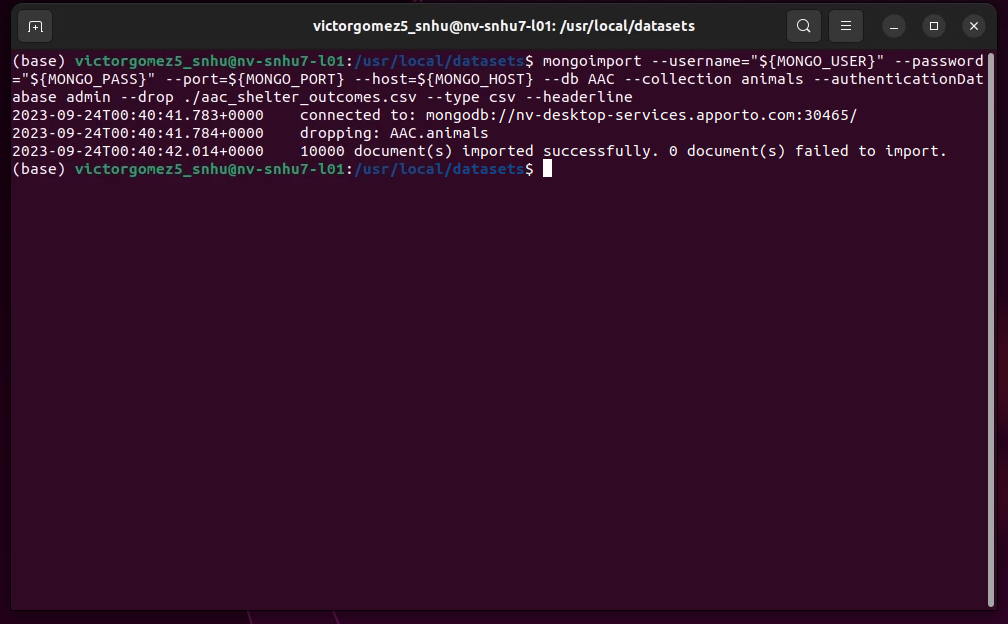
**Dash Framework Usage**

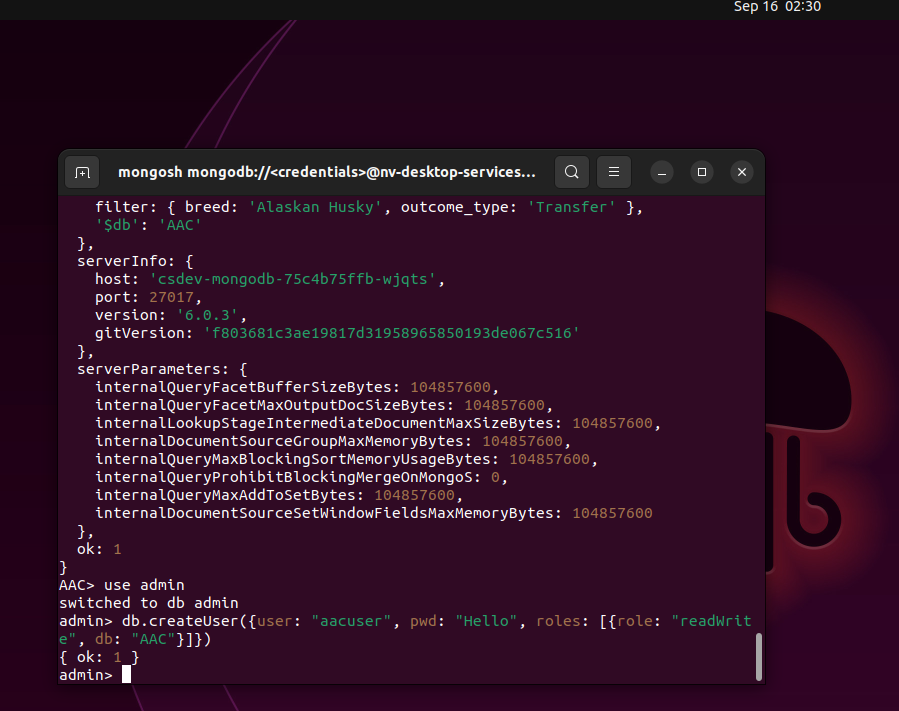
The Dash framework is an open-source framework that is usually used for building data visualization interfaces. Dash is also a very great option for this project because it is a python framework. The dash framework in this project helped with the view and controller aspect. The view displays the data to the user and allows for user modification, and it is easy to communicate with the controller from here. The controller is the requests handler in the framework which allows for the user to click on something in the view and get the correct view back from the controller. Resources for Dash framework: <https://dash.plotly.com/>

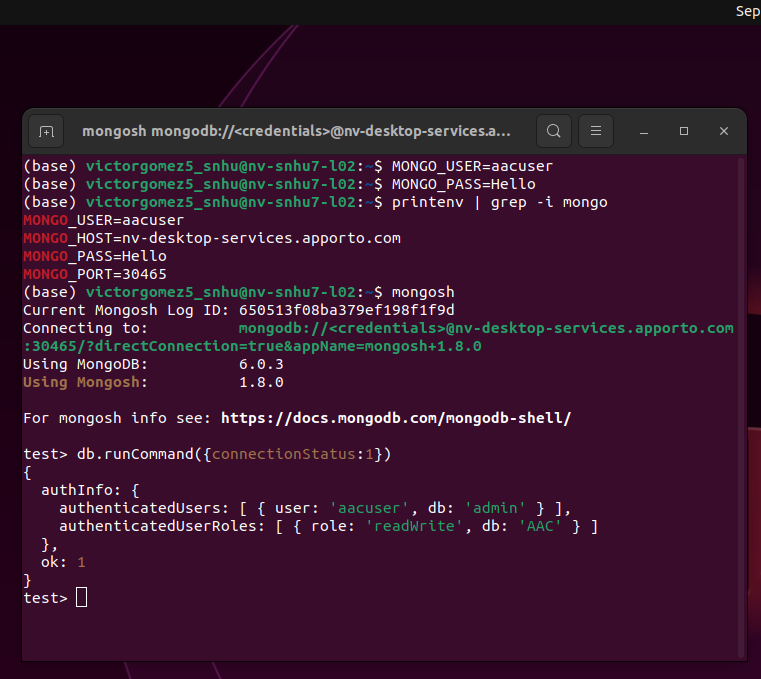
**Dash Framework Steps**

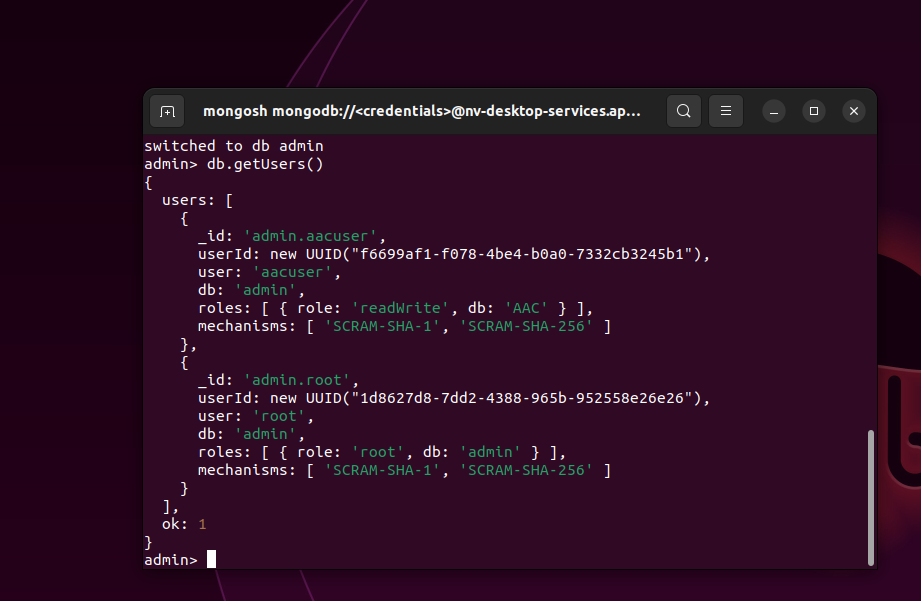
1. Import all the libraries needed
2. Import MongoDB into Pandas data frame
3. Make a simple dashboard layout
4. Edit the data table as wanted
5. Add callbacks for onClicks or RadioItems for smooth flow
6. Here you can have if/else statements that change the state of the data for interactive components
7. Make sure to always return cleaned table data so that no crashes happen
8. Add any type of data visualization wanted at this point such as charts, graphs, geolocation, etc….

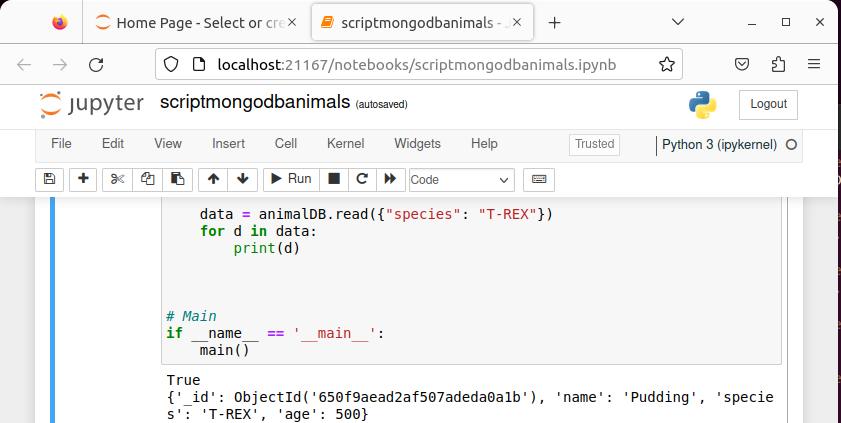
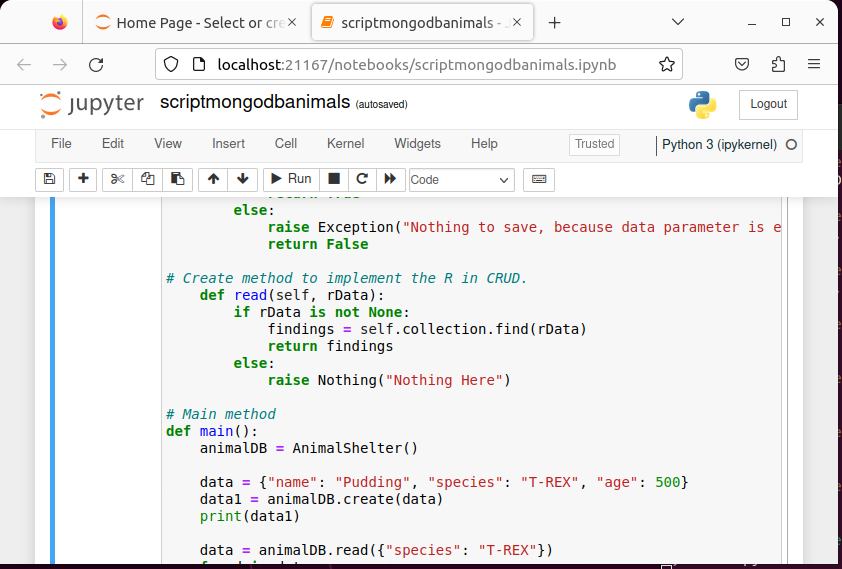
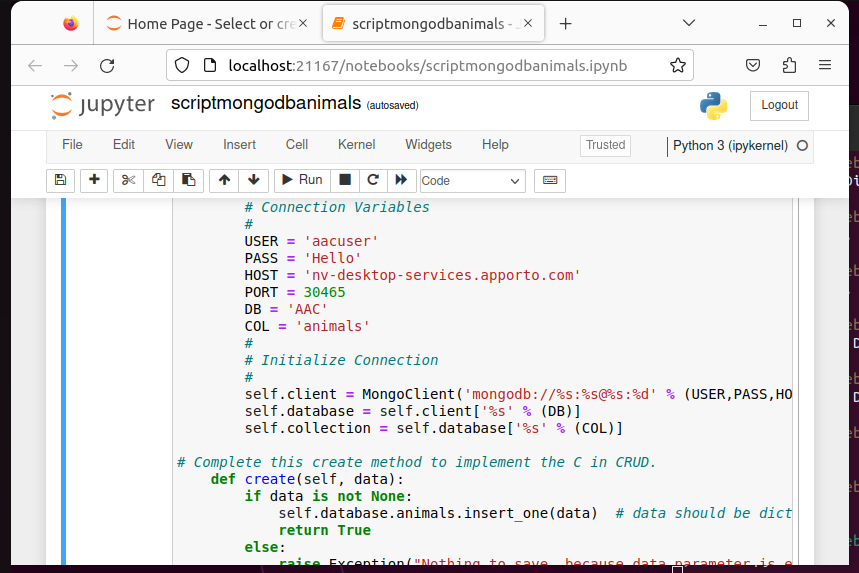
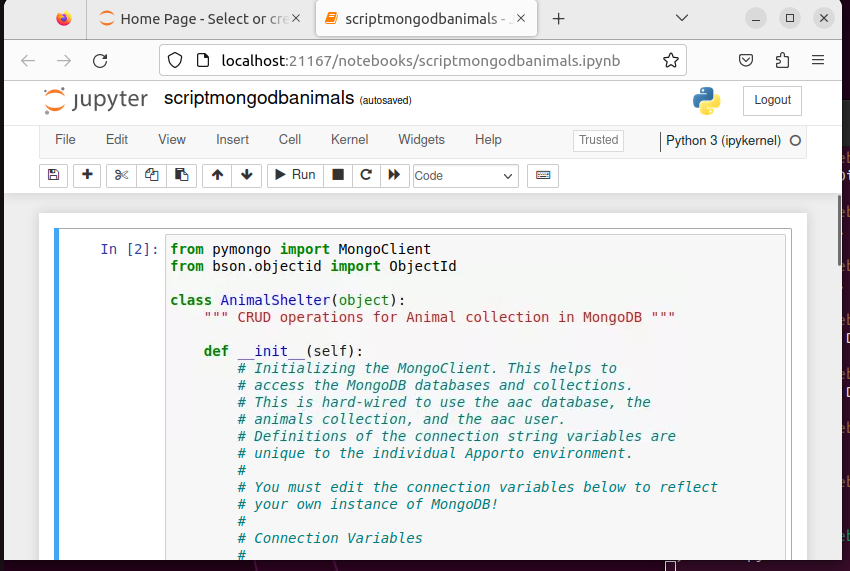
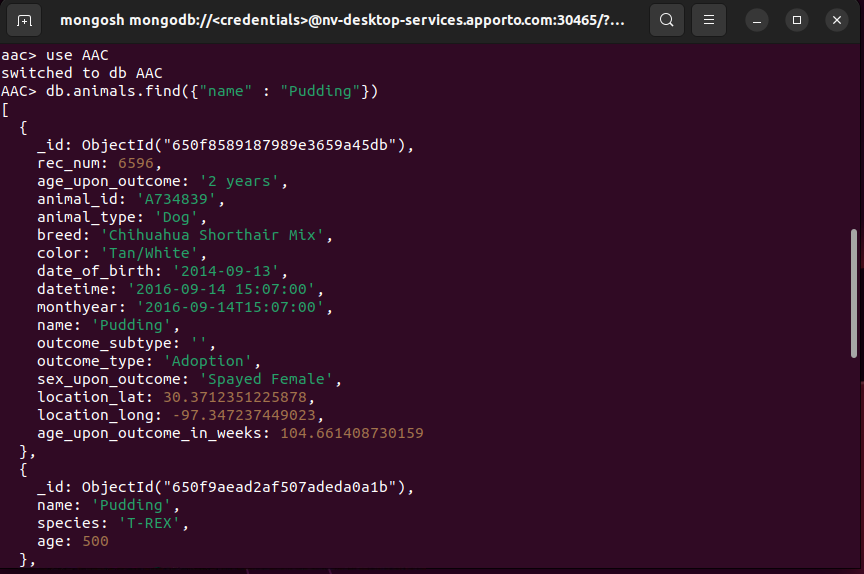
### Mongo DB Screenshots

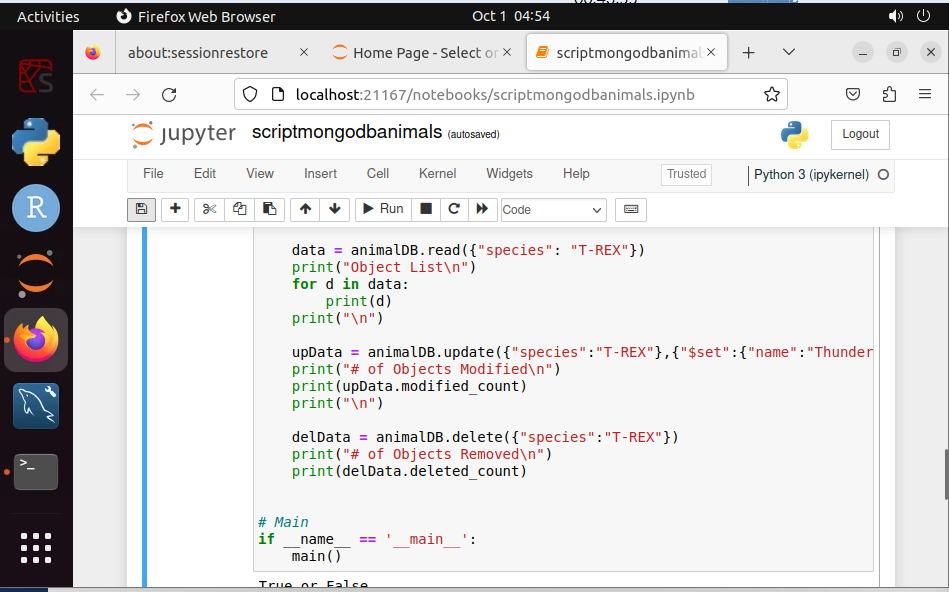


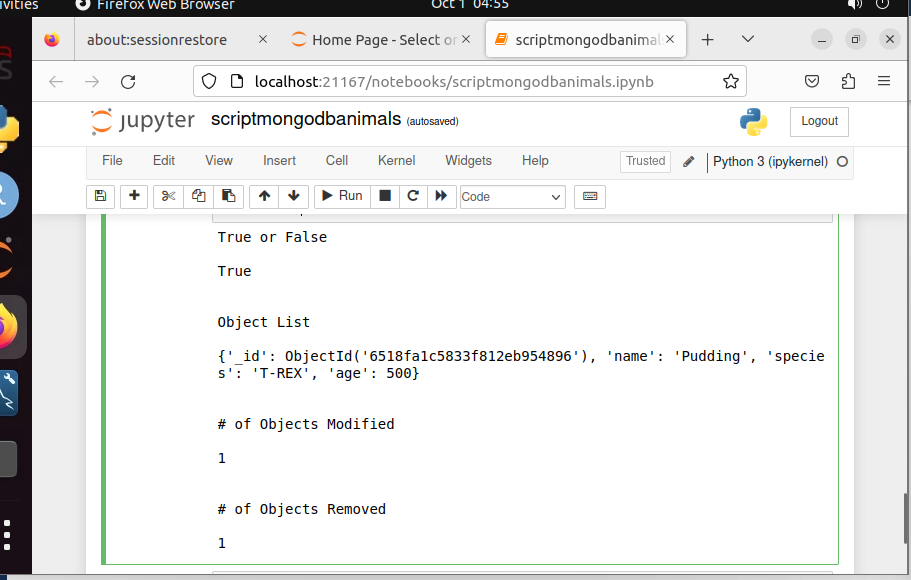






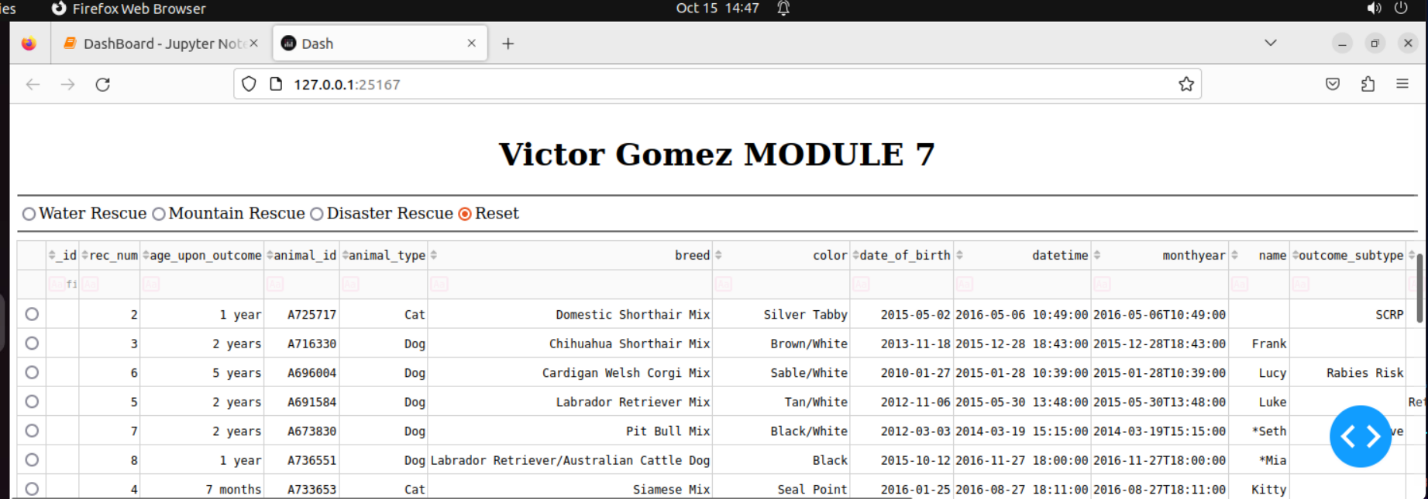


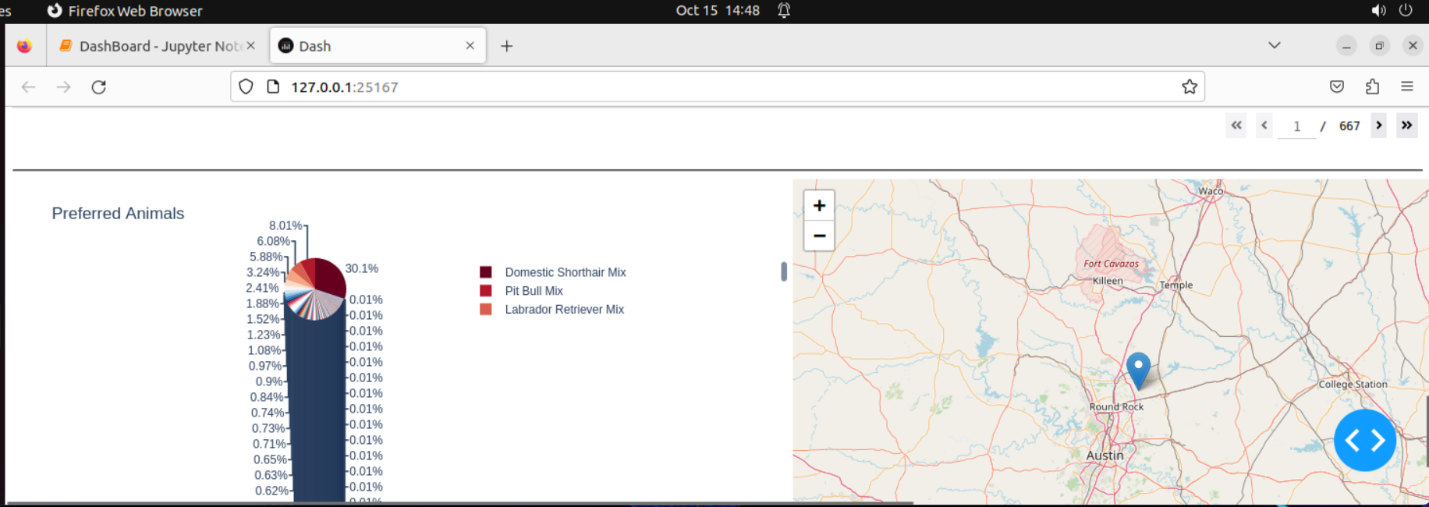


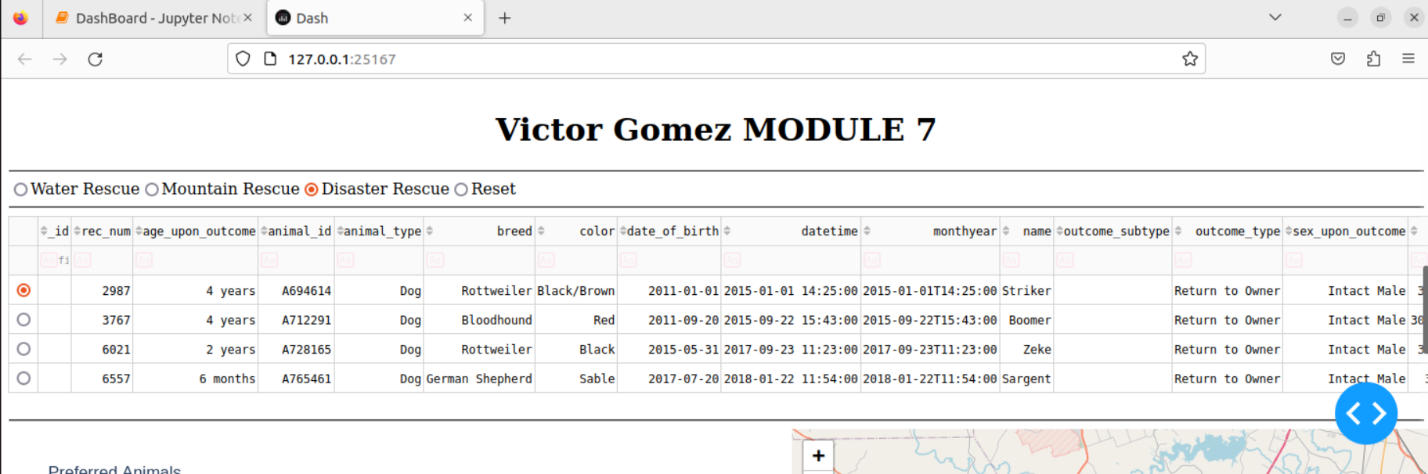


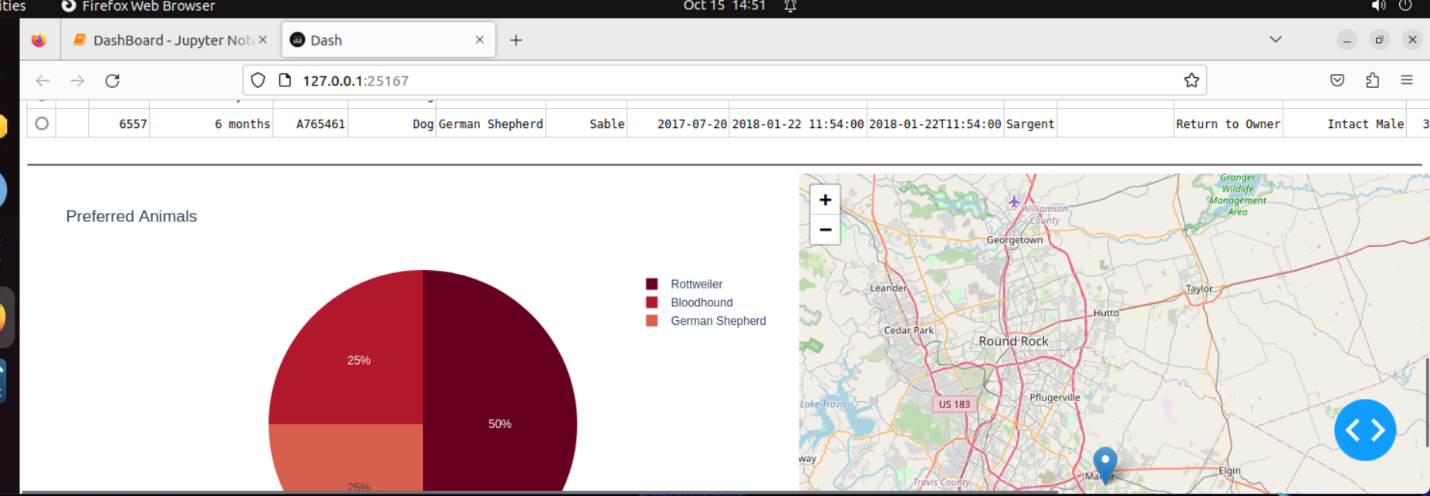
### Dash Frameworks Screenshots



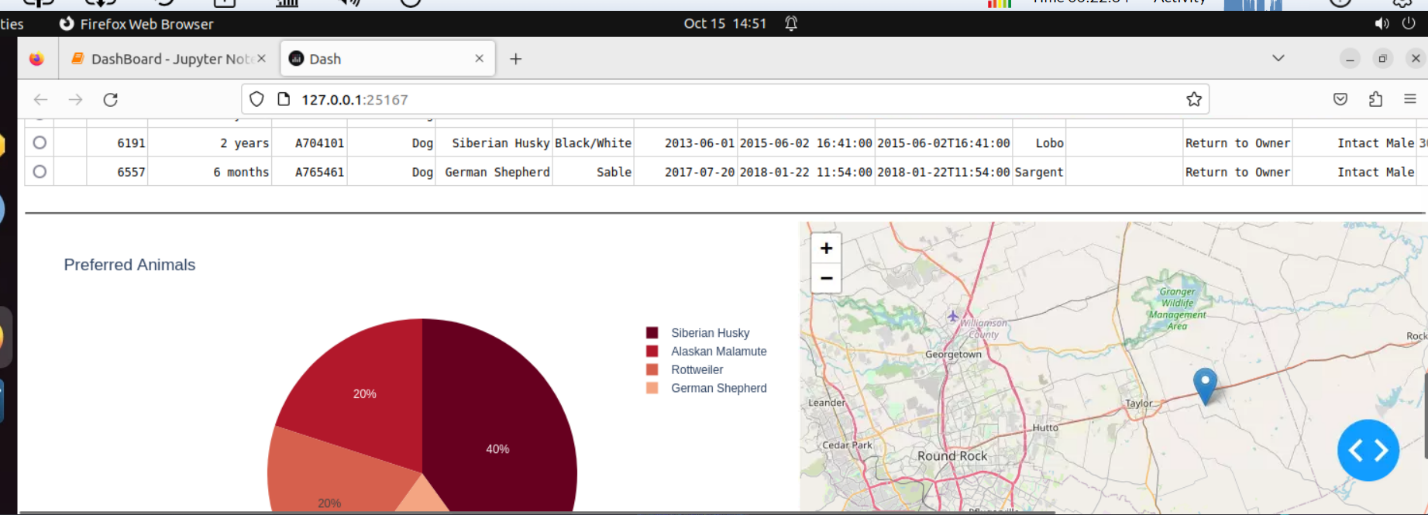


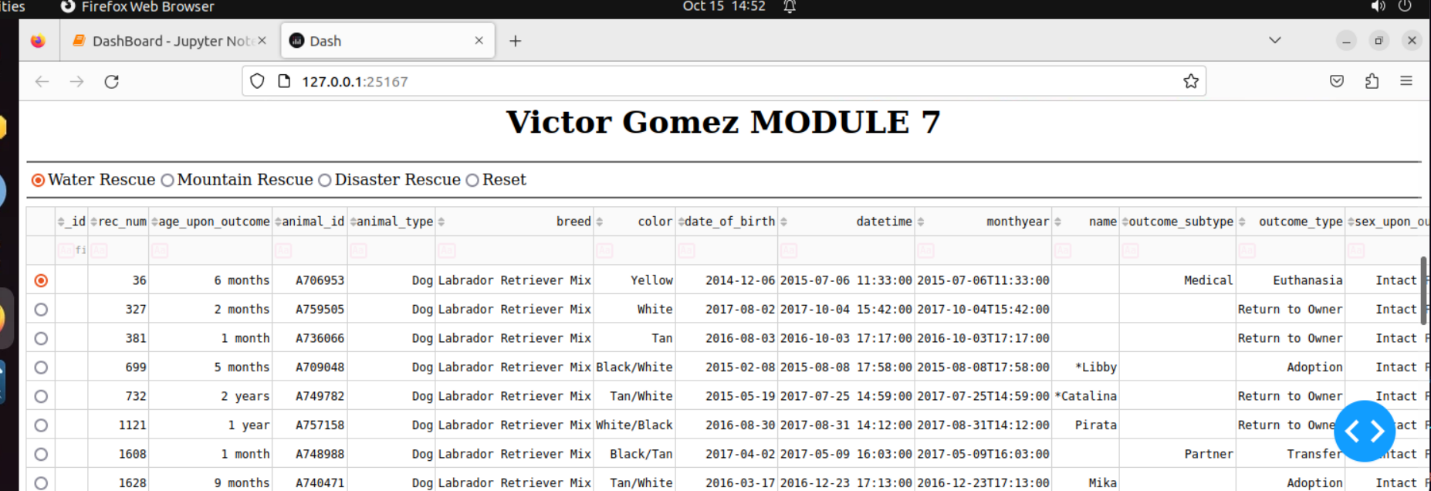


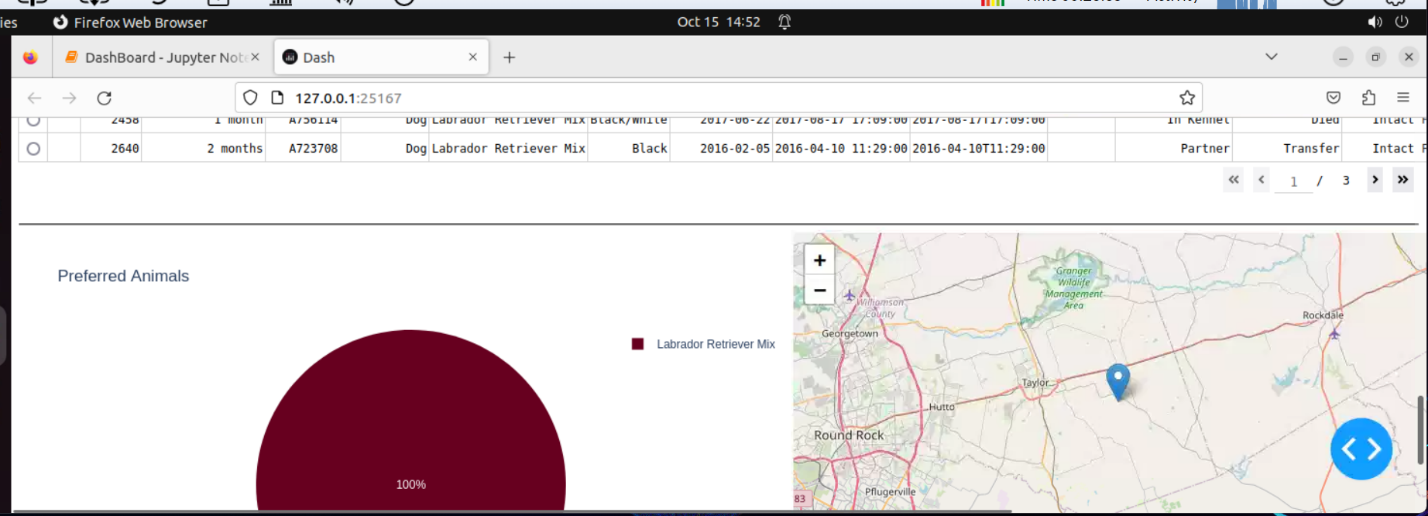












## Contact

Victor Gomez