# CS 340 README

## About Super-Pup Training Recruiter

*The purpose of the project is to provide a universal database consisting of available members of various animal shelters and also a client-facing web application that allows users to search for animals in this database using specific criteria. The information for the database is provided by the animal shelters and there is a need to compile this data into one entity then provide an easy way to search it. This is done through importing the provided information into the database and then coding a module that allows the web application to interact with the back-end database.*

## Motivation

*The Super-Pup Training Recruiter exists as part of a larger project by the team at Grazioso Salvare. The goal is to make it easier to identify animals that are available to be rescued and given purpose as search-and-rescue animals. Their research has shown that animals with specific criteria are better suited for these kinds of training roles and are looking for help with the database and software integration to identify these specific animals.*

## Getting Started

*Super-Pup Training Recruiter is a free, open source program that uses a MongoDB back-end database and a Python-coded front end script application. The user interacts with a web application created in Python using the Dash framework. In order to run a local version of it, you will need a terminal or GUI that can run a Mongo database and also a Python script. The Python code and script used here was created in Jupyter Notebook. The code will be freely available via GitHub for download and distribution. The Python script will produce the Dash web application that the end user will interact with to filter database results.*

*You will need to import the Austin Animal Center (AAC) Outcomes file using the mongoimport tool. The AAC database requires authentication so a universal user account was setup with a username of “aacuser” and password of “aapass”. You will need to use these credentials in order to access the database properly. Once the program authenticates into the running mongo database, you will have the ability to filter the database results using radio buttons that were created using pre-determined search criteria.*

*By default, the web application displays all of the dogs in the AAC database. It also displays the percentage of the dog breed types as a pie chart and a geolocation display and the name of the selected dog. Selecting one of the radio buttons ( Water Rescue, Mountain Rescue, Disaster Rescue, Reset) will query the AAC database and return results based on the filter options. This action will also update the pie chart and geolocation widgets accordingly and display values in real-time.*

*The project was created by first establishing the MongoDB structure and importing the AAC information. Next, admin and user accounts were created to allow proper permissions to the database and to give users a way to securely authenticate their sessions. Once the database was running, the Animal\_CRUD Python code was created to access the database and tested to ensure queries could be sent and the proper data was returned. This progressed further into creating a Python script for the Dash Framework web application. The app was also tested to verify the interface could properly call functions from the Animal\_CRUD.PY code, display the database results properly, and could be interacted with. Finally, customized features were added to the app such as radio buttons to filter for specific criteria and the real-time responses of the pie charts and geolocation widgets.*

*Please note that if using Jupyter Notebook for the Python application that you cannot simply create a new IPYNB file and rename it as a PY file type. You must create a PY file directly or create a Plain Text file which can be renamed as a PY file type. If these steps are not followed, the Python script will not be able to reference the Python code module and the program will fail with a Name cannot be null exception.*

## Installation

*You will need to be running the MongoDB to access the database. It can be downloaded it from the MongoDB Download Center* [*here*](https://docs.mongodb.com/manual/installation/) *.*

*OS-specific installation instructions are also available at the link above.*

*MongoDB is a NoSQL database that interfaces with Python to create fast and efficient queries while utilizing much less resources than other relational database designs.*

*To access the MongoDB, you will need PyMongo which is in the Python Package Index.*

*PyMongo is a native Python driver for MongoDB. This ensures that all functionality between Python and MongoDB have been certified, tested, and will receive official updates as long as the module is supported. It provides functionality to query, retrieve results, write, and delete data, and run database commands.*

*It is recommended to use* [*pip*](https://pypi.org/project/pip/) *to install PyMongo on all platforms:*

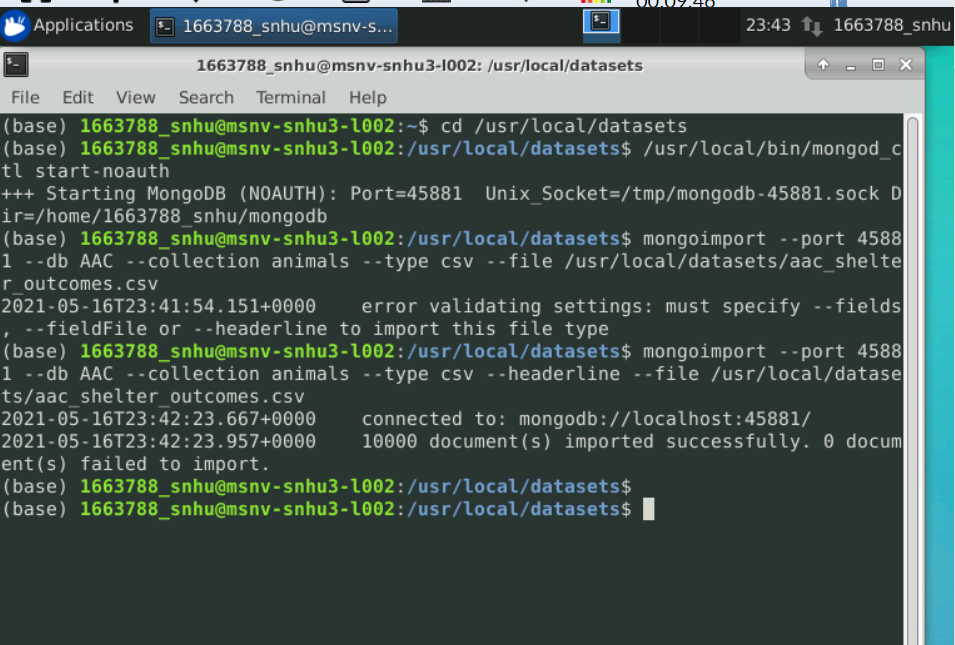


*The web interface is designed using the Python Dash Framework in a client-facing python script. This allows us to build a data visualization app with a highly customizable user interface within the Python code script.*

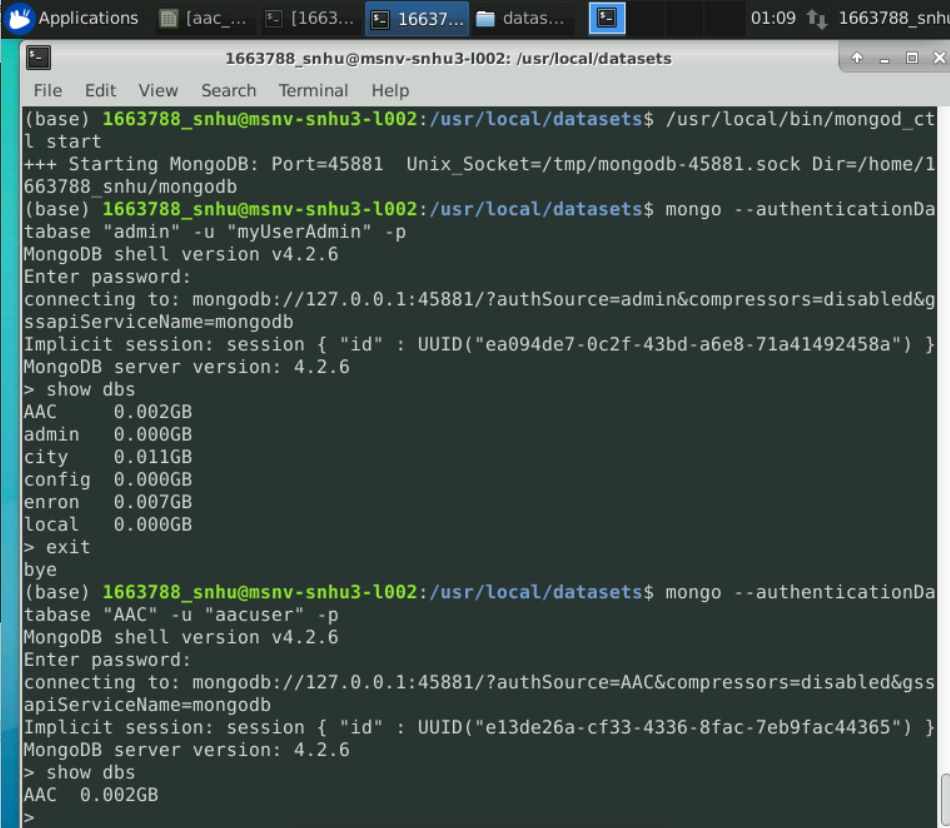
## Usage

### Initial Steps

*The screen below demonstrates the mongoimport tool execution and successful import of the CSV file into MongoDB*

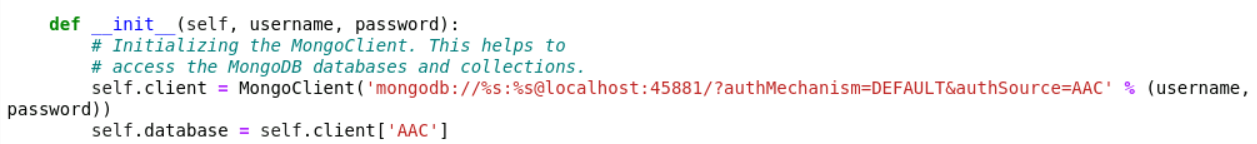


*Successful account creation and login to the database and collection using both the administrator and “aacuser” accounts to ensure authentication*

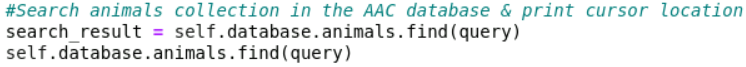


### Code Example

*First, the MongoClient gets initialized and authenticates in the AAC database:*

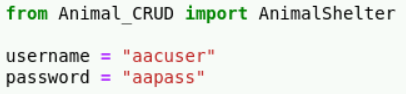


*This is used to query the database and return the documents:*



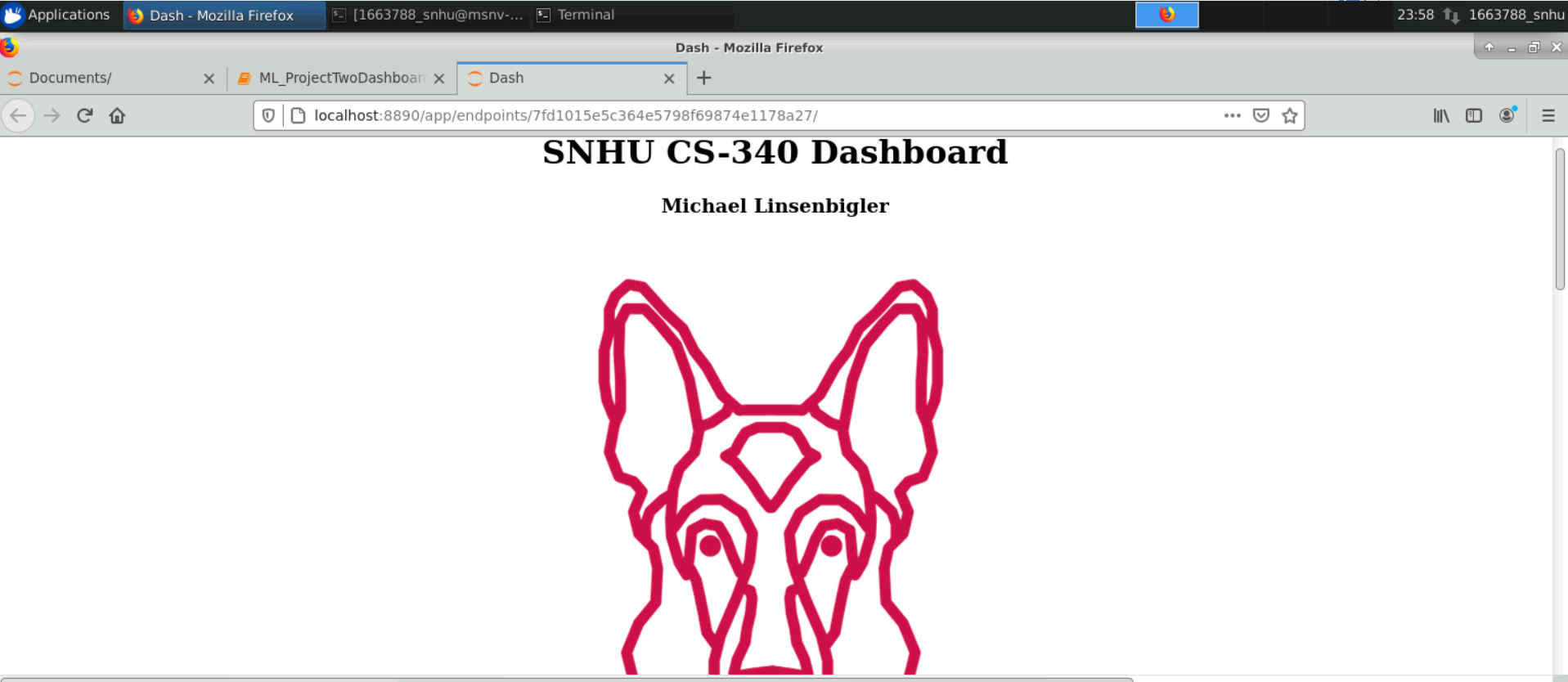
*This is how the script calls the AnimalShelter class from the PY code and sets the username and*

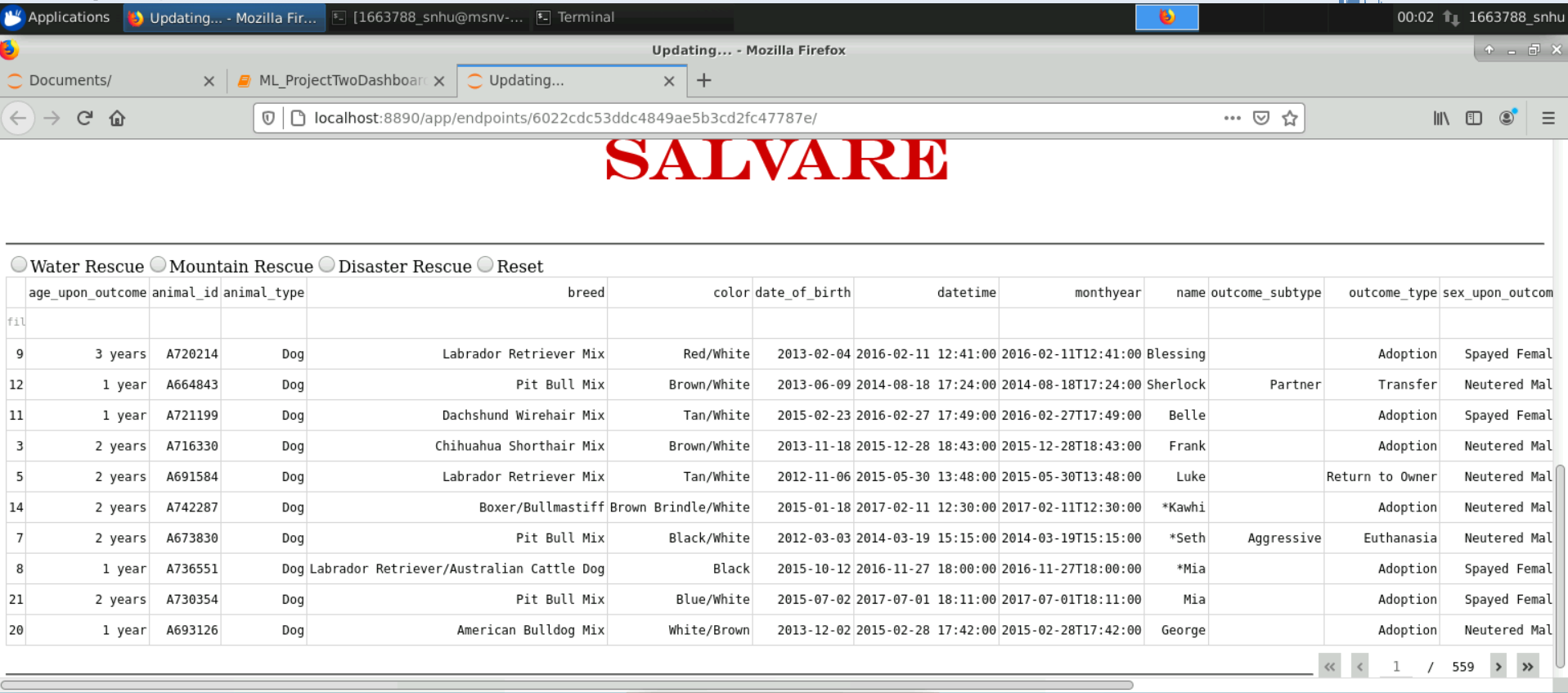
*password credentials:*



### Tests

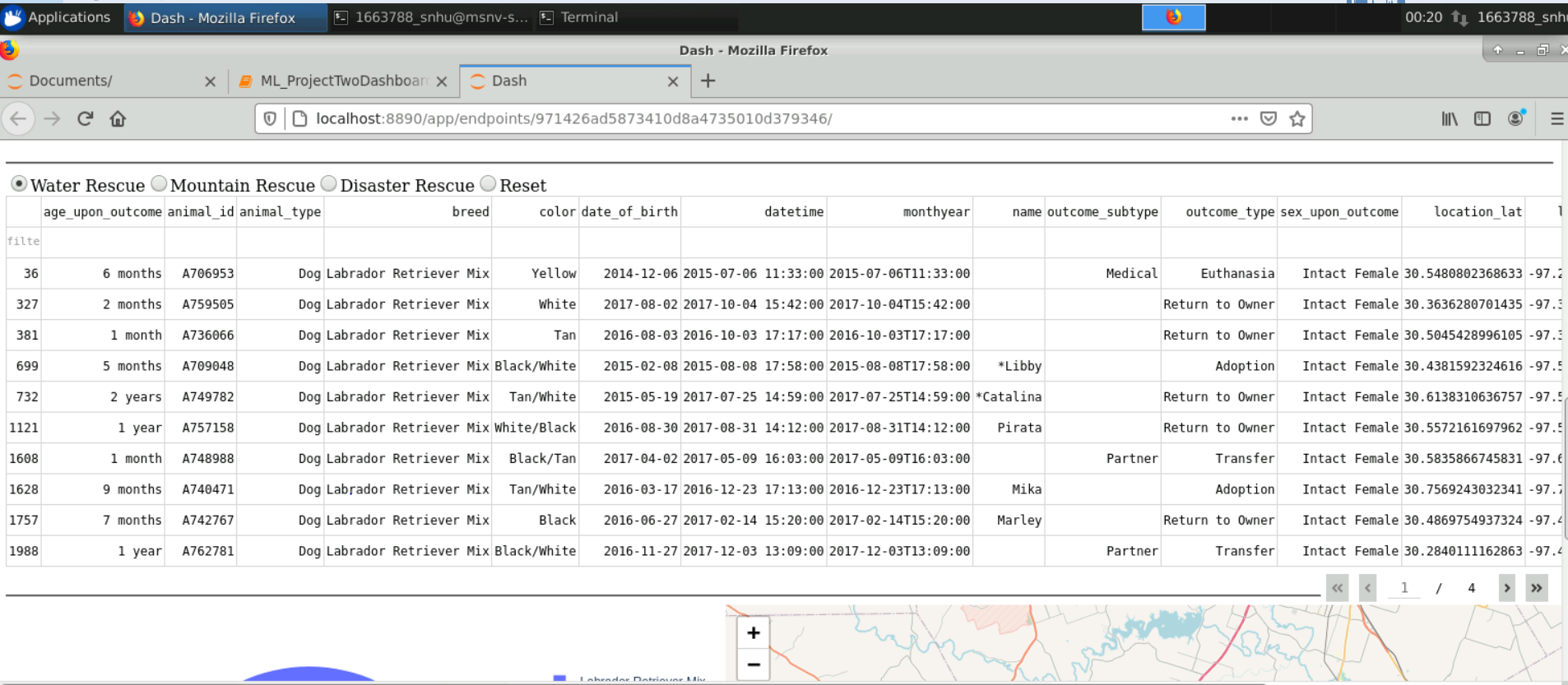
***Demonstrating the initial state of the Dashboard including Unique Identifier and company logo that links to homepage:***

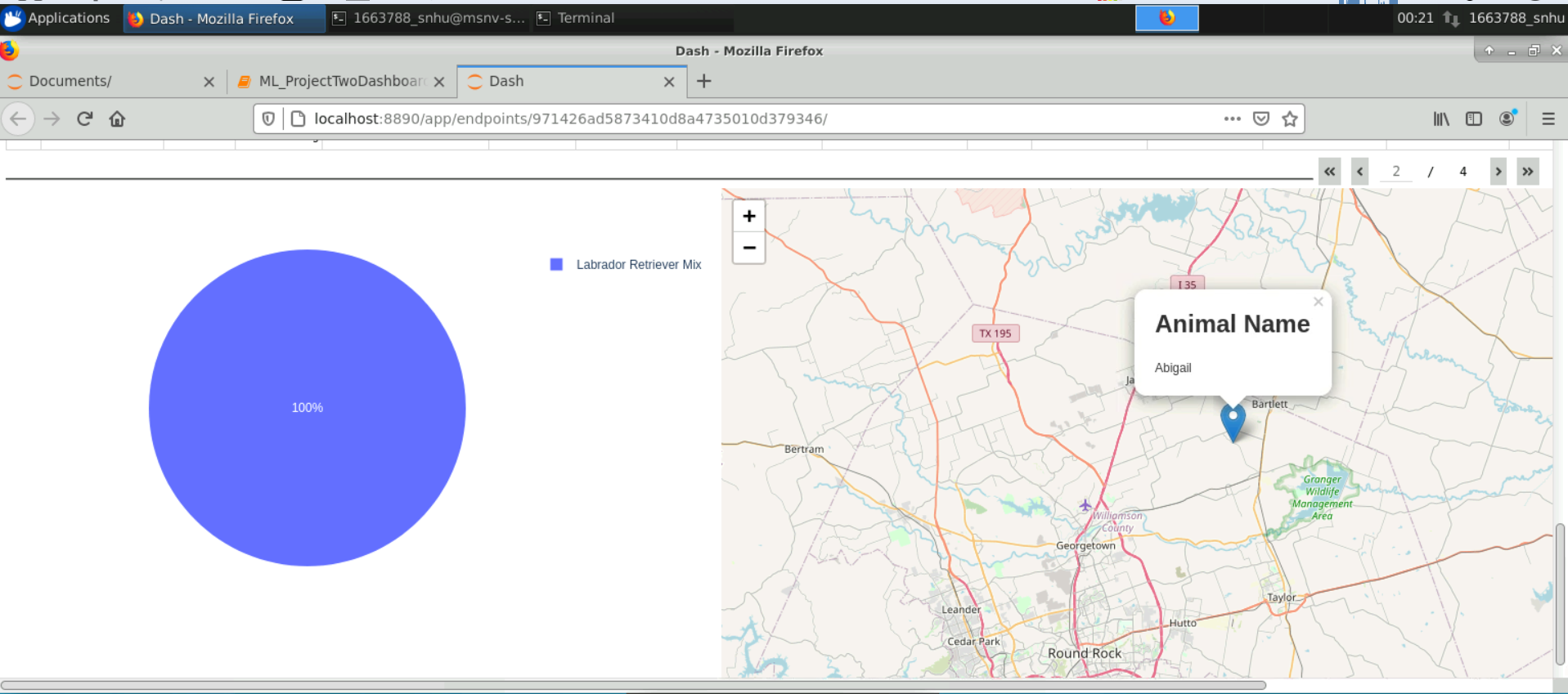




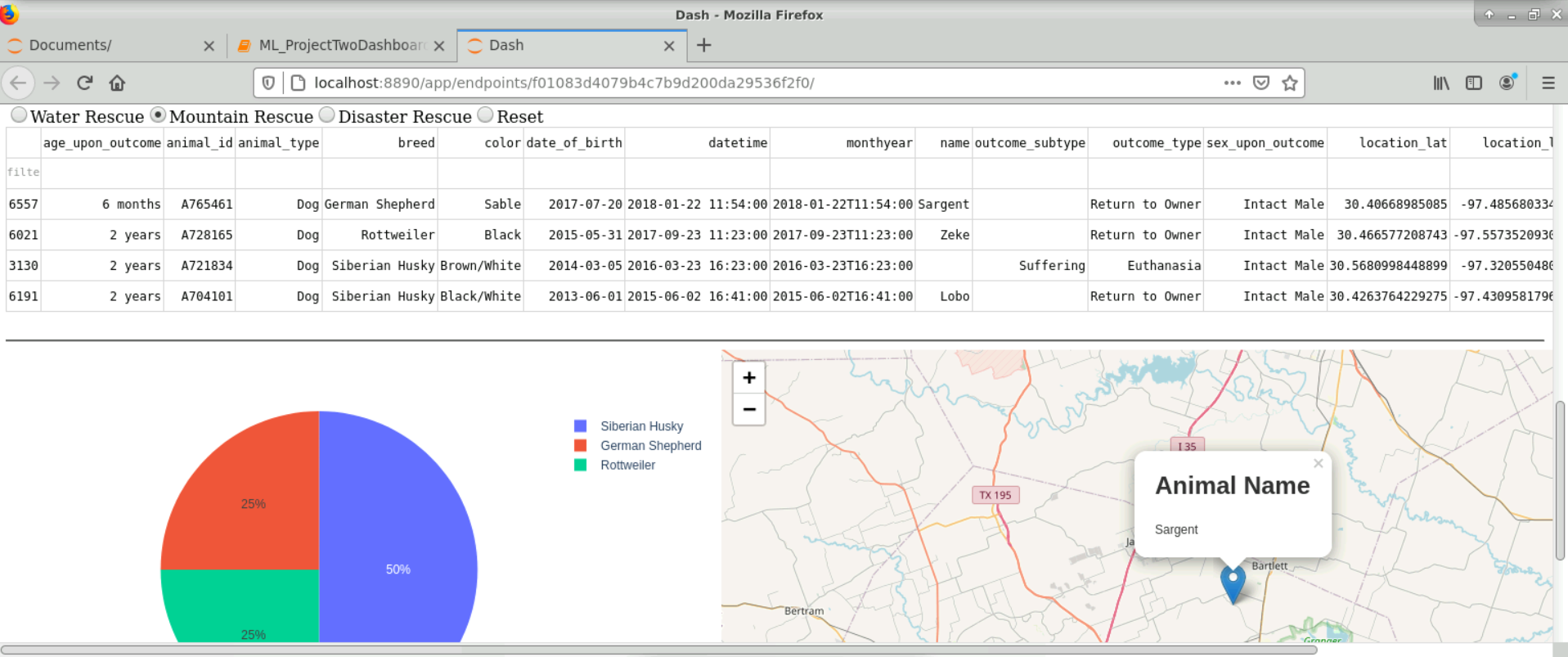


***Demonstrating the Water Rescue Filter:***

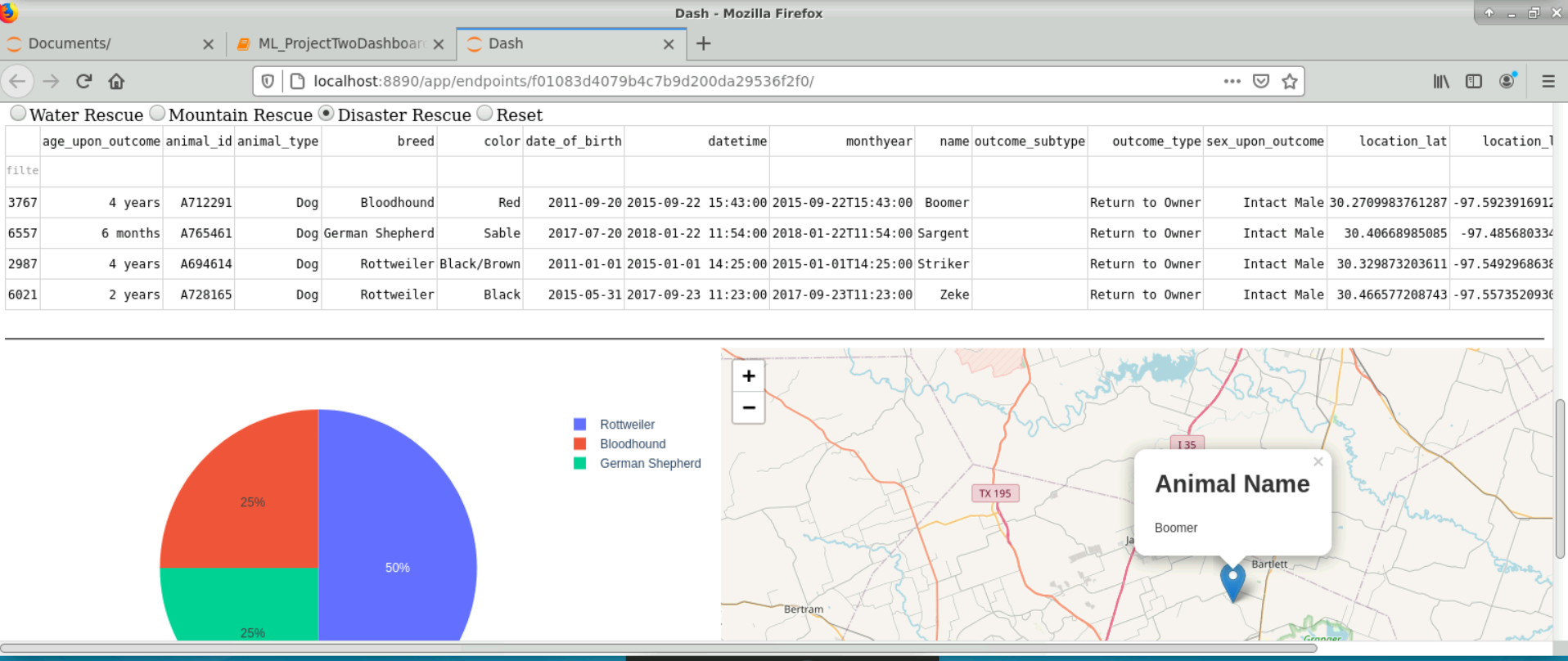




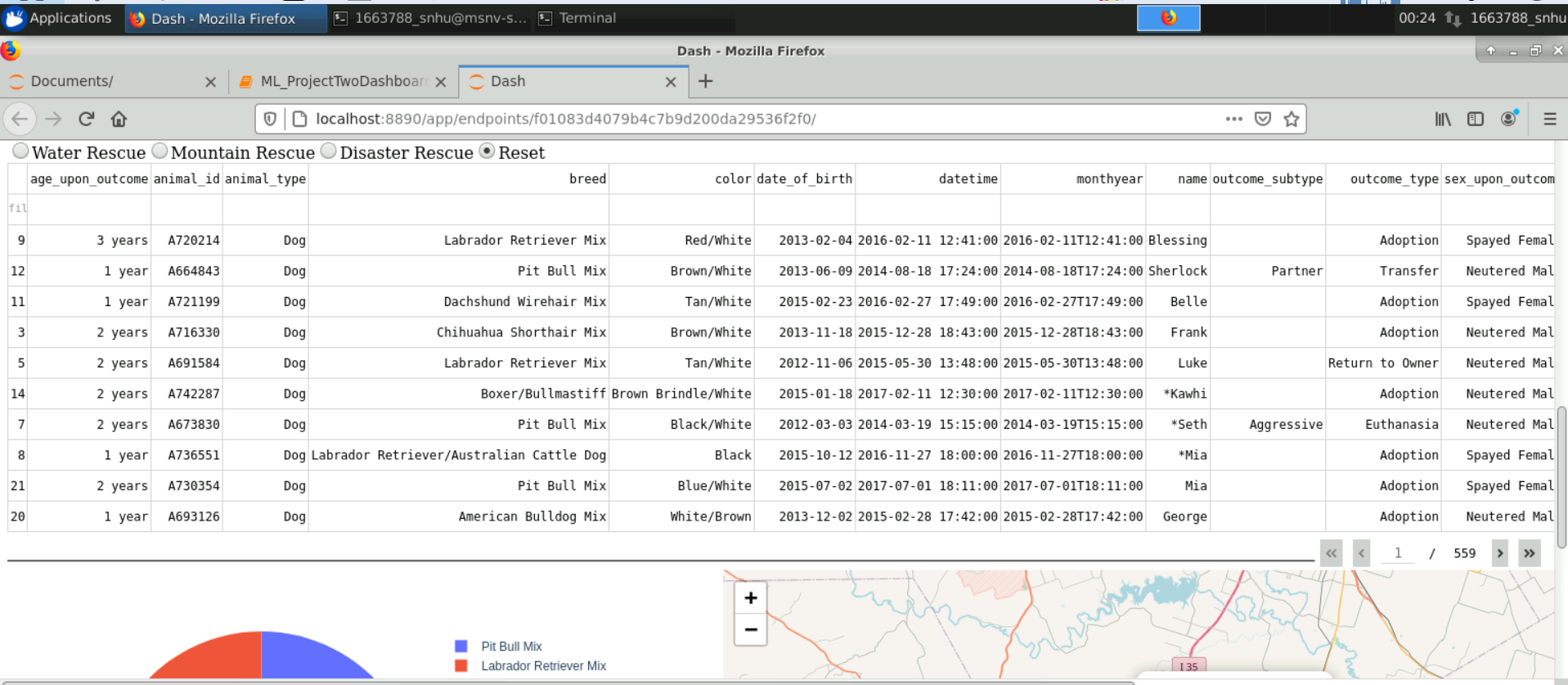
***Demonstrating the Mountain Rescue Filter:***

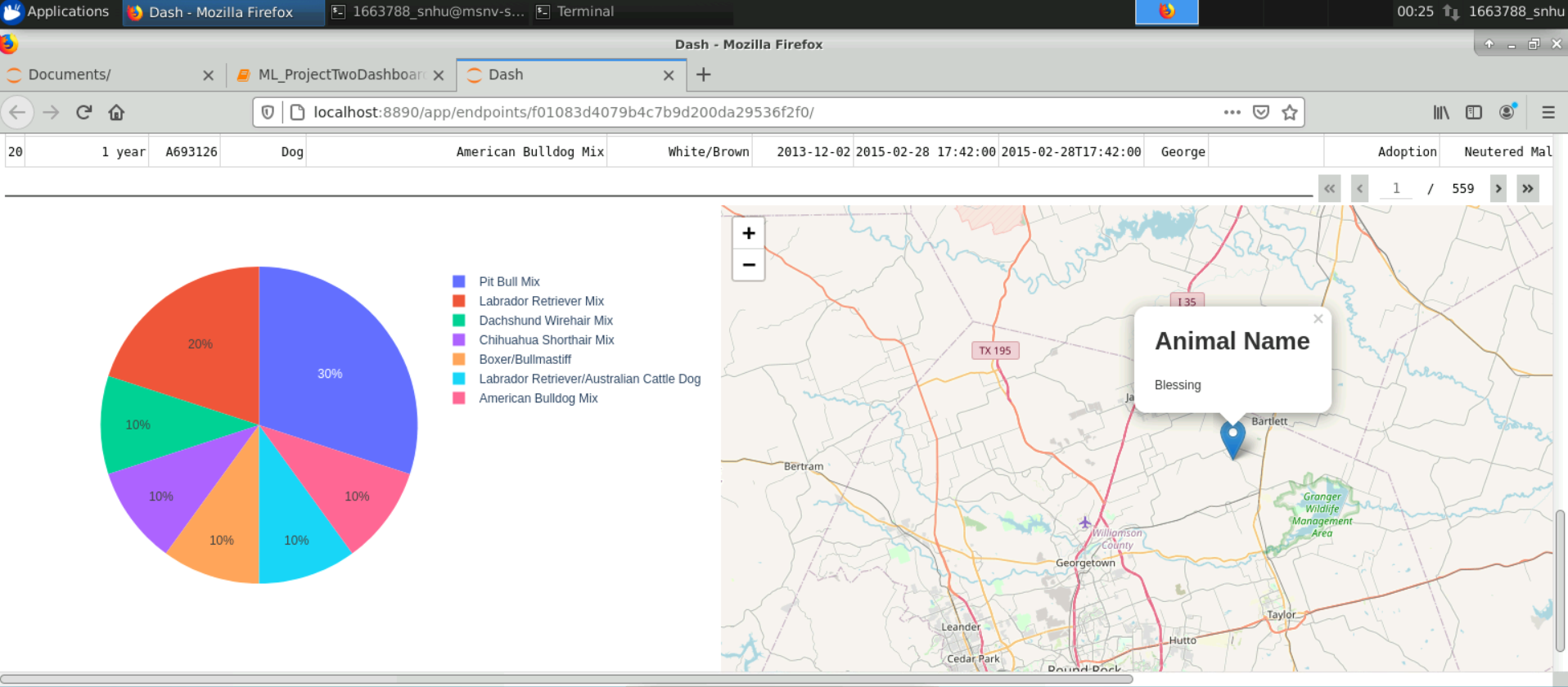


***Demonstrating the Disaster Rescue Filter:***



***Demonstrating the Reset Filter:***





## Contact

Your name: Michael Linsenbigler