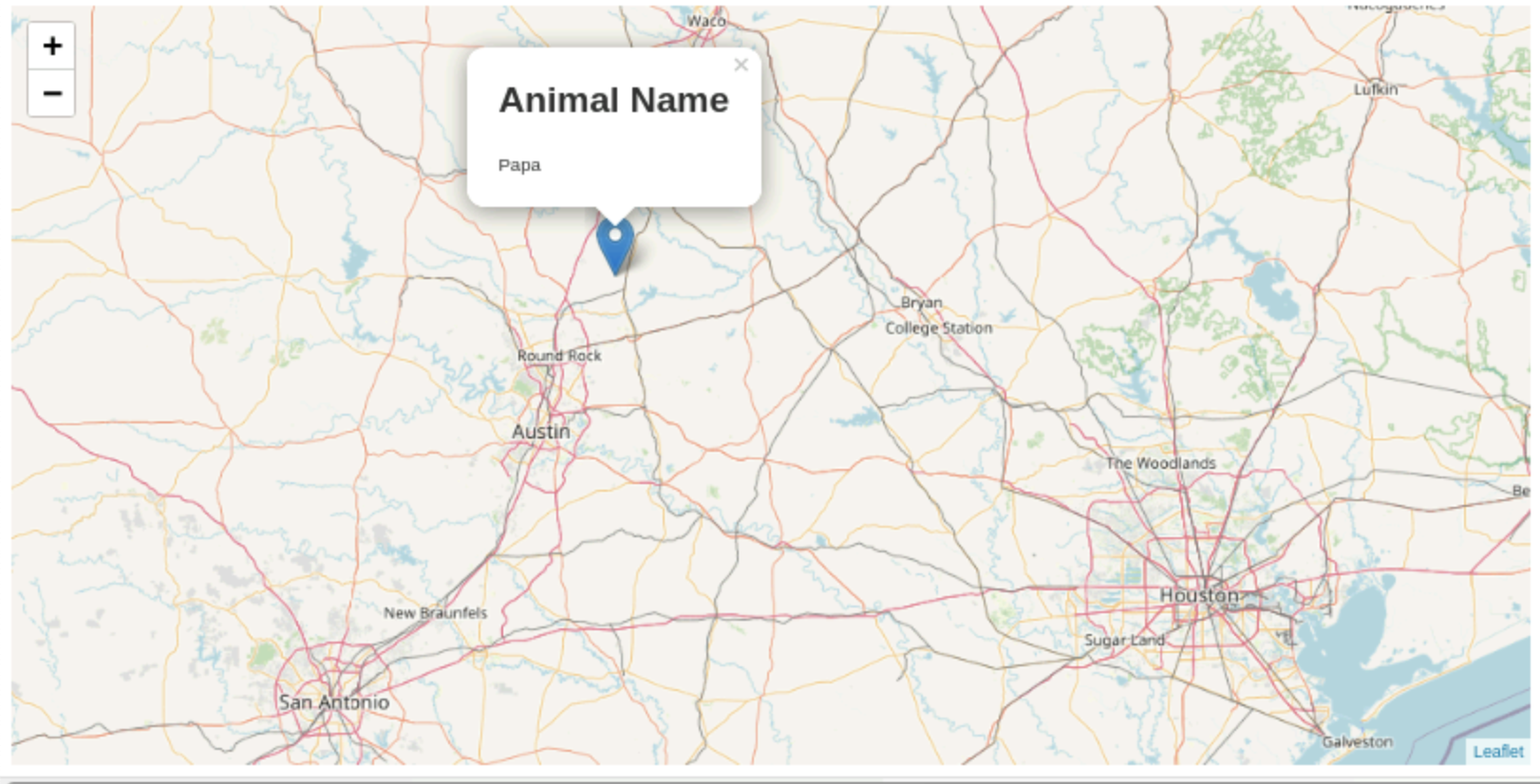
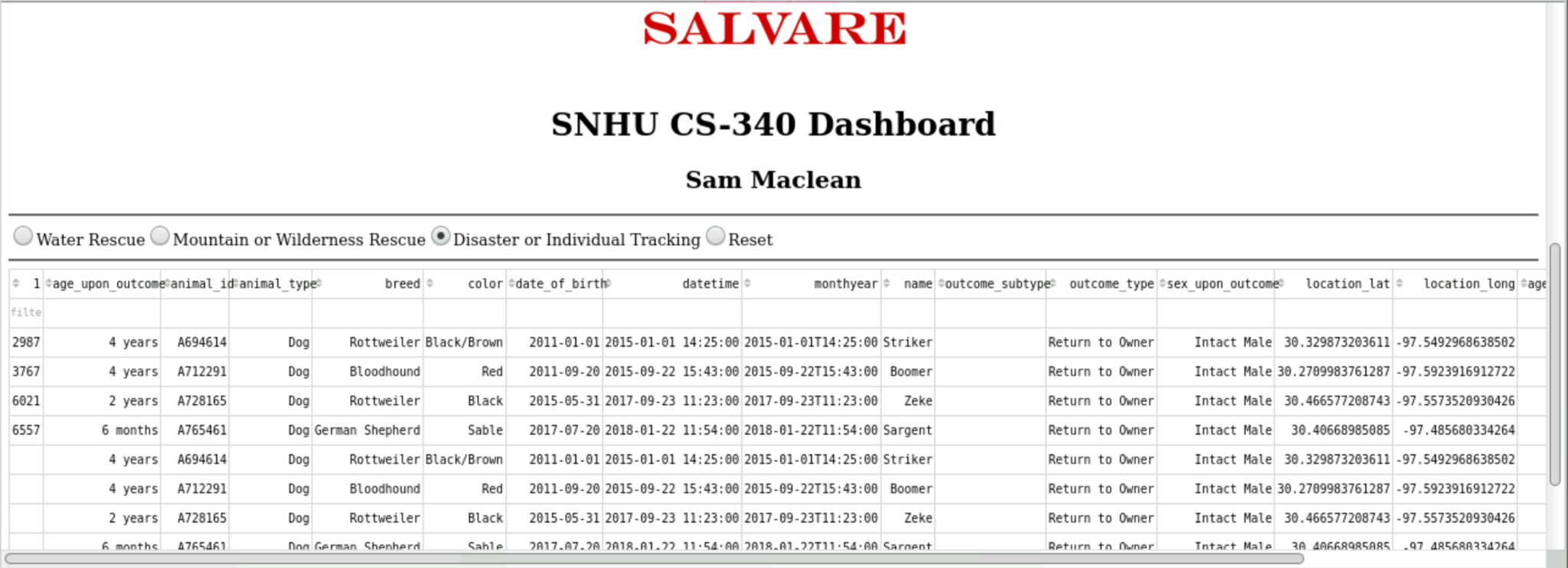
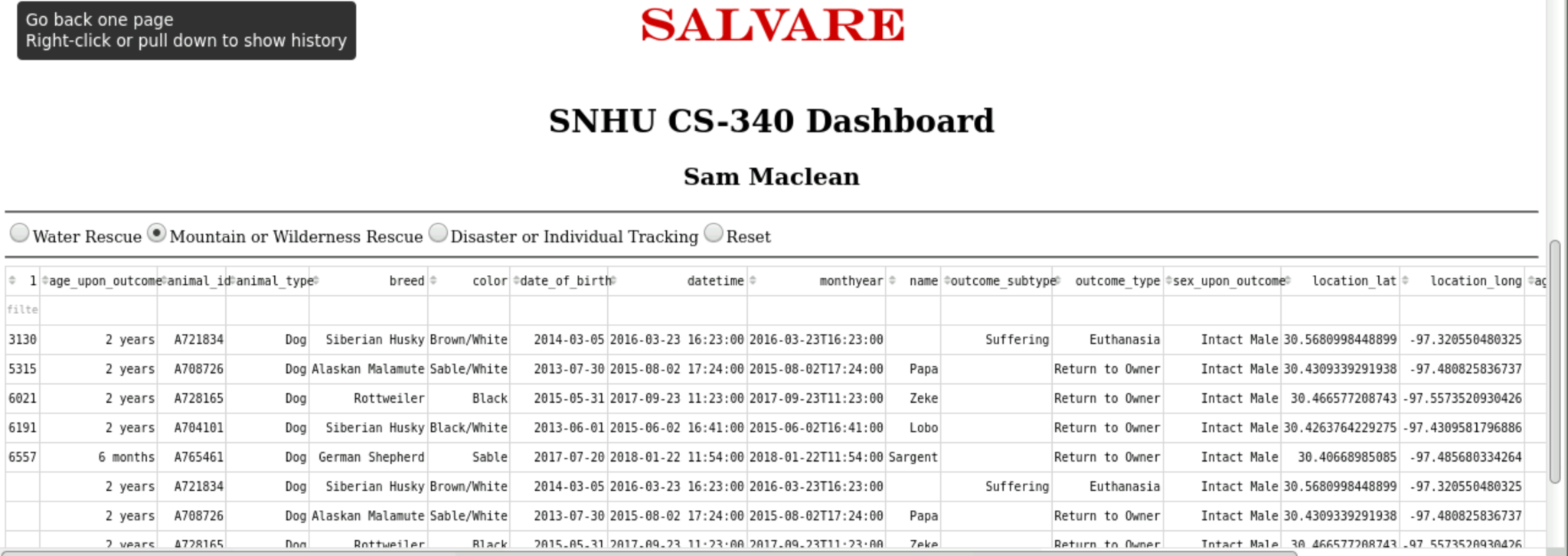
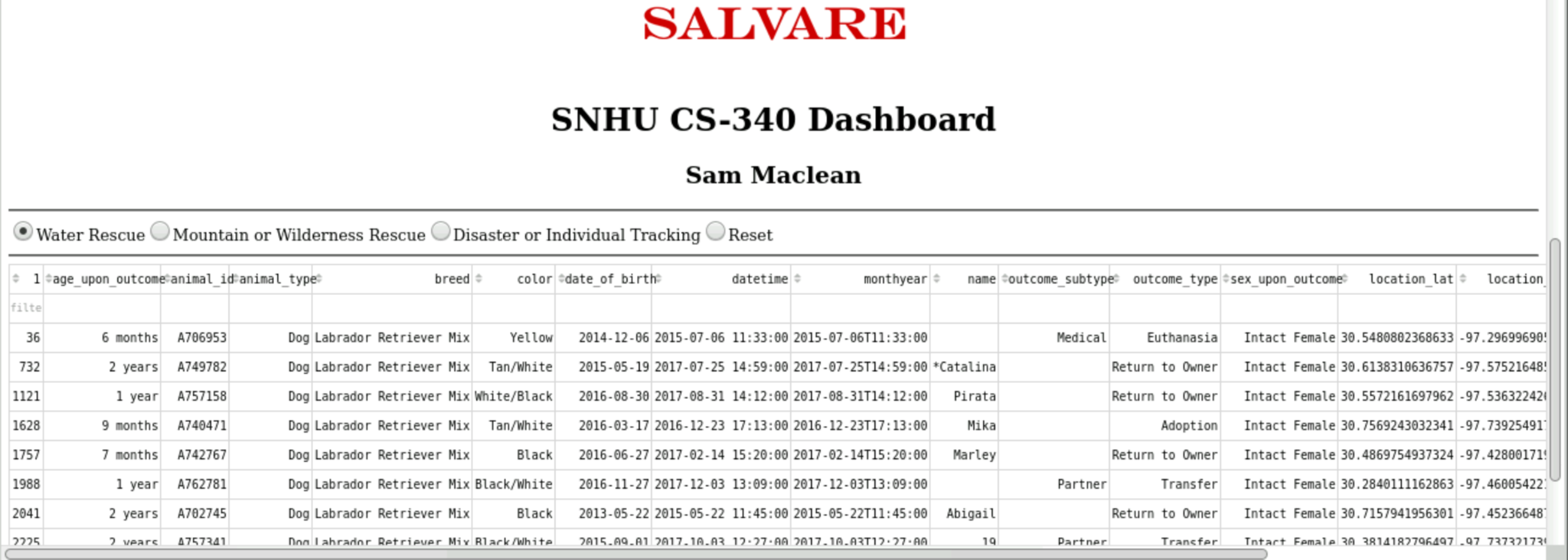
**Read Me**

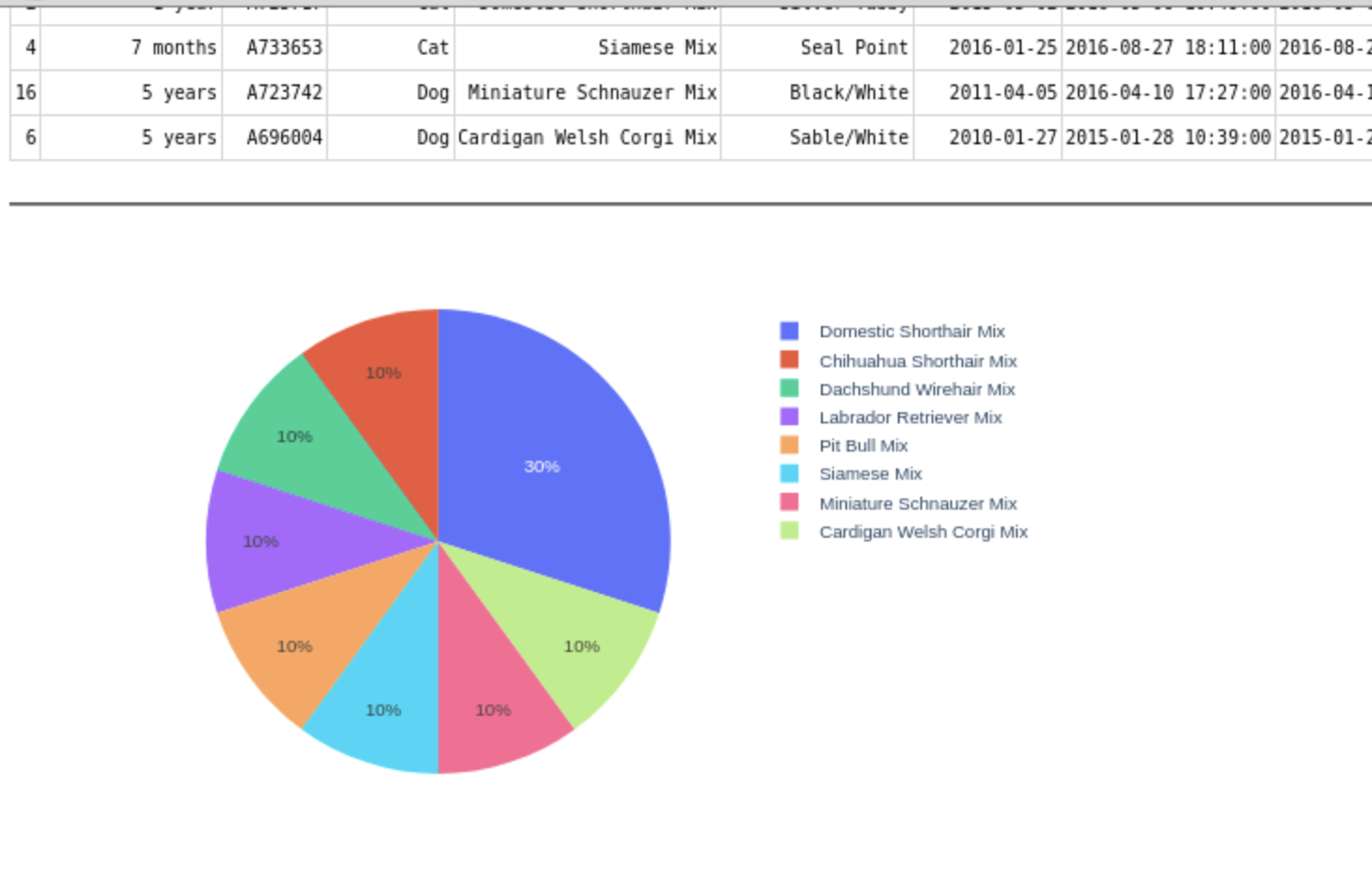
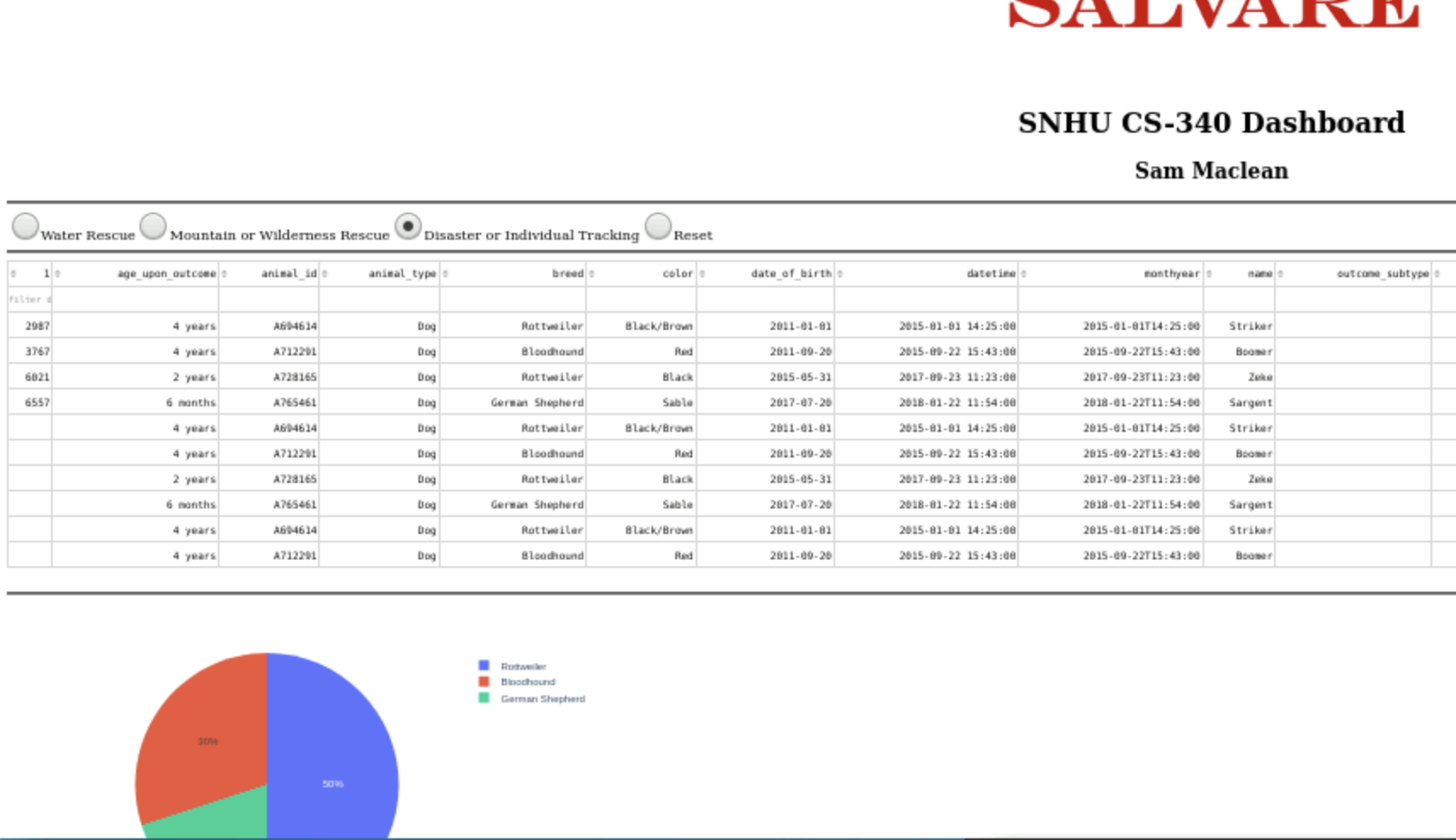
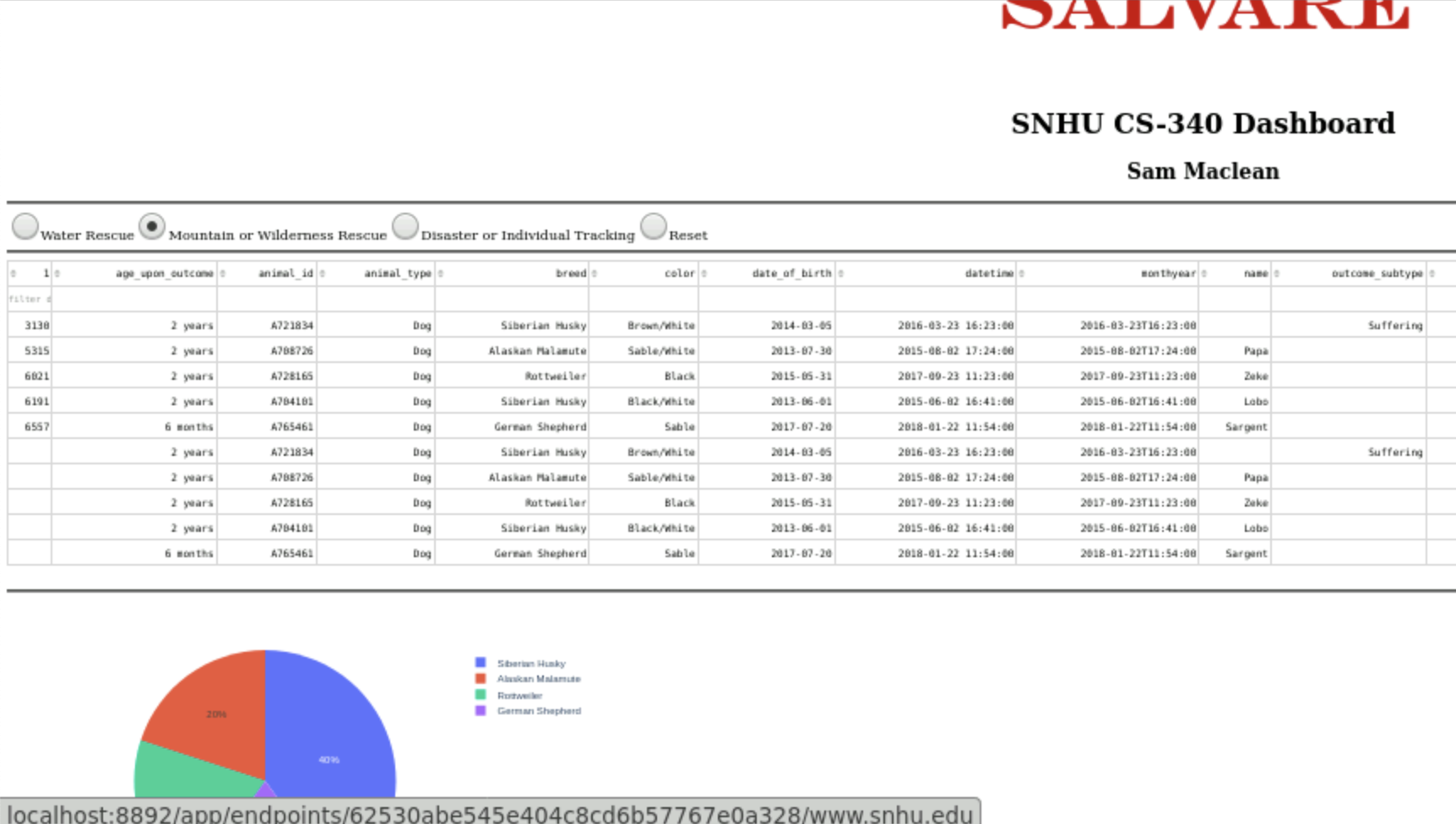
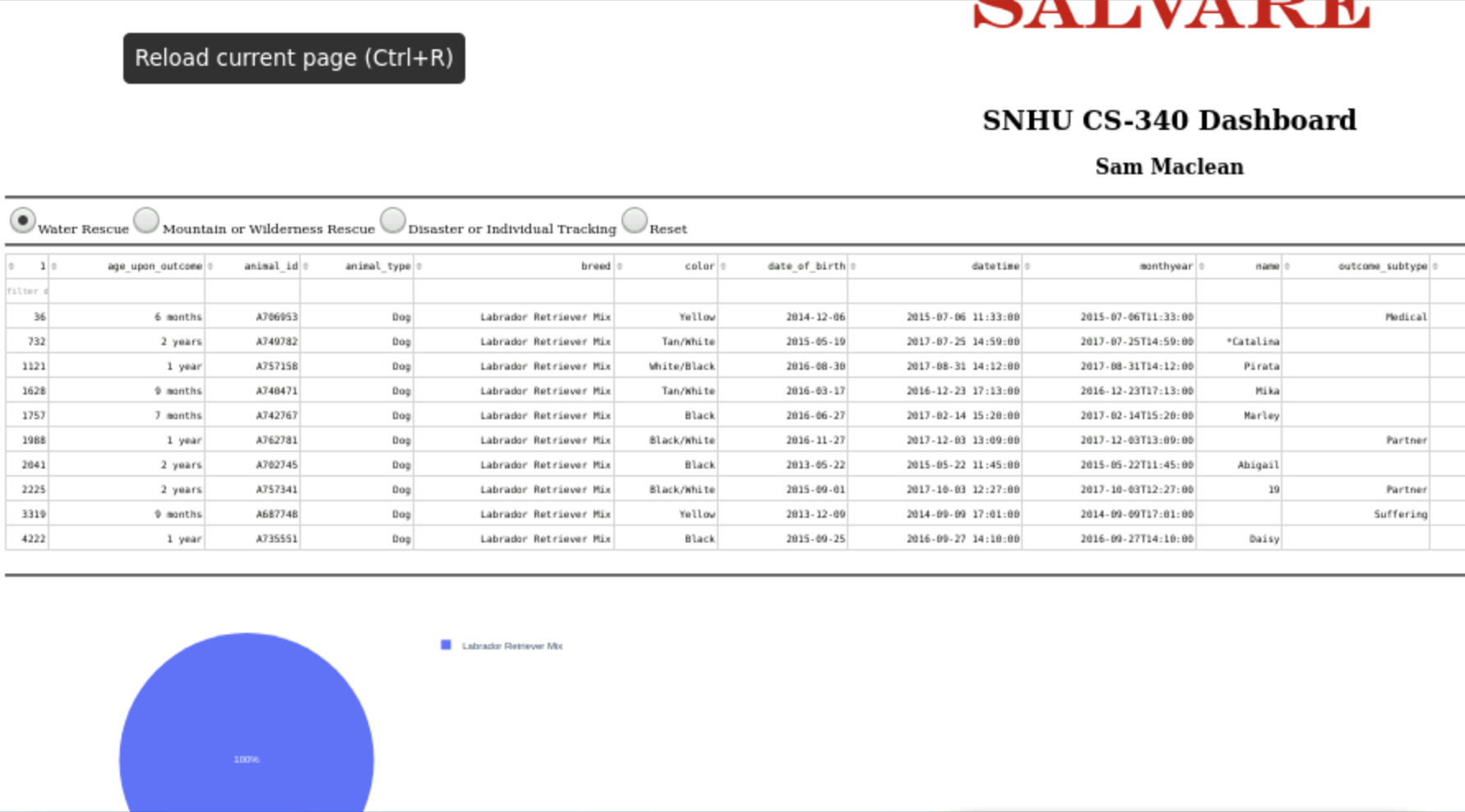
Developer: Sam Maclean

**Functionality**

This program functions as a visual representation of a Mongo database. The data is displayed in a chart that grabs information from the database and creates columns with various headings for each data type. The program also displays a pie chart that responds to the input depending on the type of filter applied. The filters for this project are Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and Reset.

**Screenshots**

****

****

**Tools**

MongoDB was used as a model component of the development because it provides instant easy access to databases and functionality with Python through the use of PyMongo. PyMongo is an efficient driver to work with both Python and Mongo when working with databases. Python has the ability to create, read, update, and delete entries in a Mongo database according to CRUD functionality to make for easy manipulation of a database.

The Dash framework provides an easy interface to create an interactive dashboard with basic functionality using built-in functions and commands. Using Dash, a Leaflet map can easily be created which can display the location of the particular entry in the database that is being focused on. Dash makes it simpler to create simple and advanced dashboards that are user-friendly.

The combination of Python, Mongo, and Dash combined to create a simple yet effective dashboard that can be used to easily access and adjust a database. Jupyter Notebook was used to test the Python program to ensure functionality. This program displays the output to the console and can be viewed in a separate window as if it were its own website.

**Steps**

To complete this project, I started by accessing a Mongo database and using Python along with the PyMongo driver to access this database and create create, read, update, and delete (CRUD) functionality. Then a dashboard was created to visually represent the data and use the Python script to easily access the data in the Mongo database rather than having to connect to Mongo directly from the Jupyter testing script. Dash was used to create a visual dashboard containing a data table with all keys and values in the database, a pie chart representing the currently selected filter, and a map that displays the location of the selected animal.

**Challenges**

One more challenge was figuring out how to properly index the database when dealing with the chart and the map. This challenge was overcome by researching the iloc documentation and Dash functionality. I also had issues figuring out how to get the visual components to display on the screen which was fixed by handling errors present in the program.