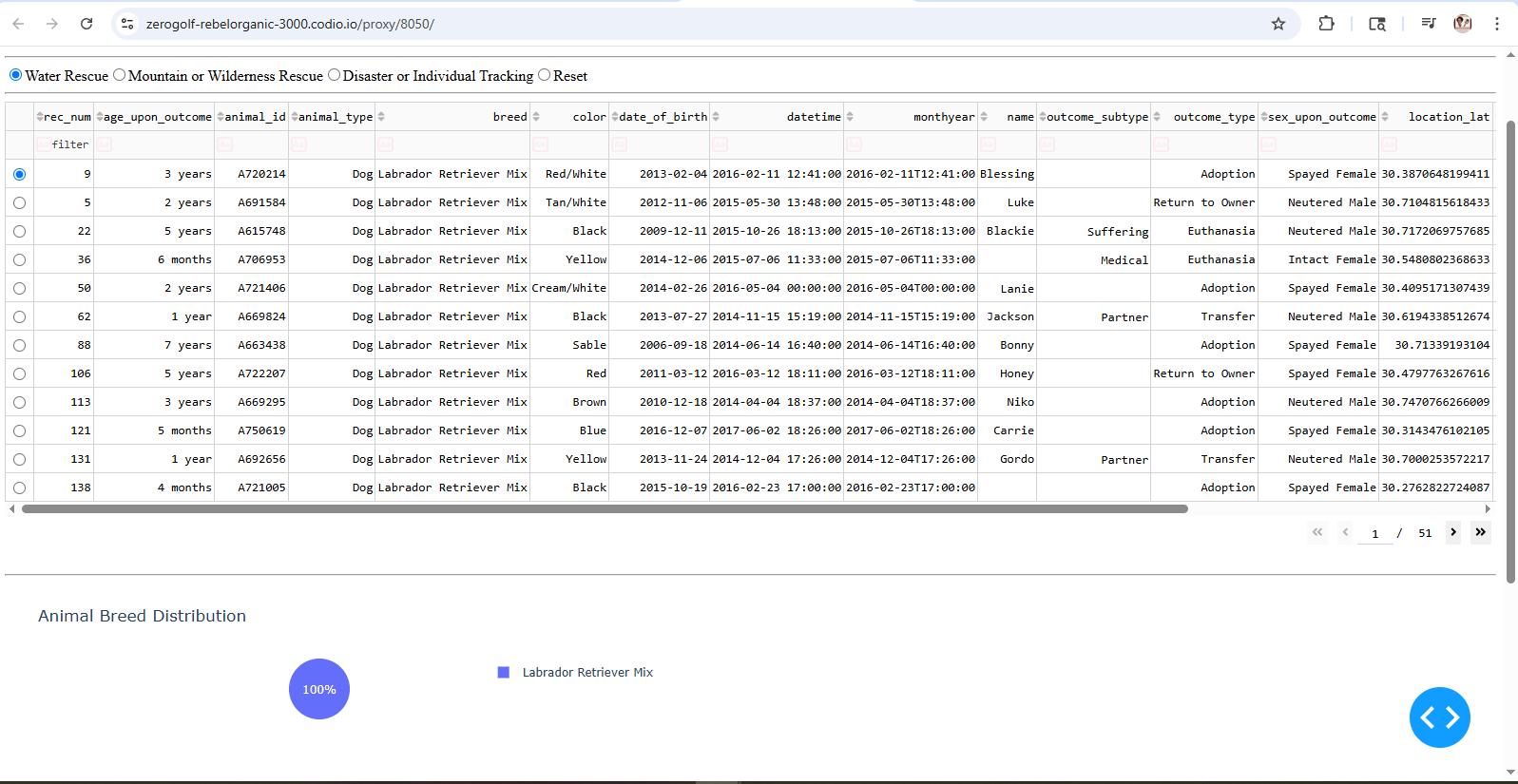
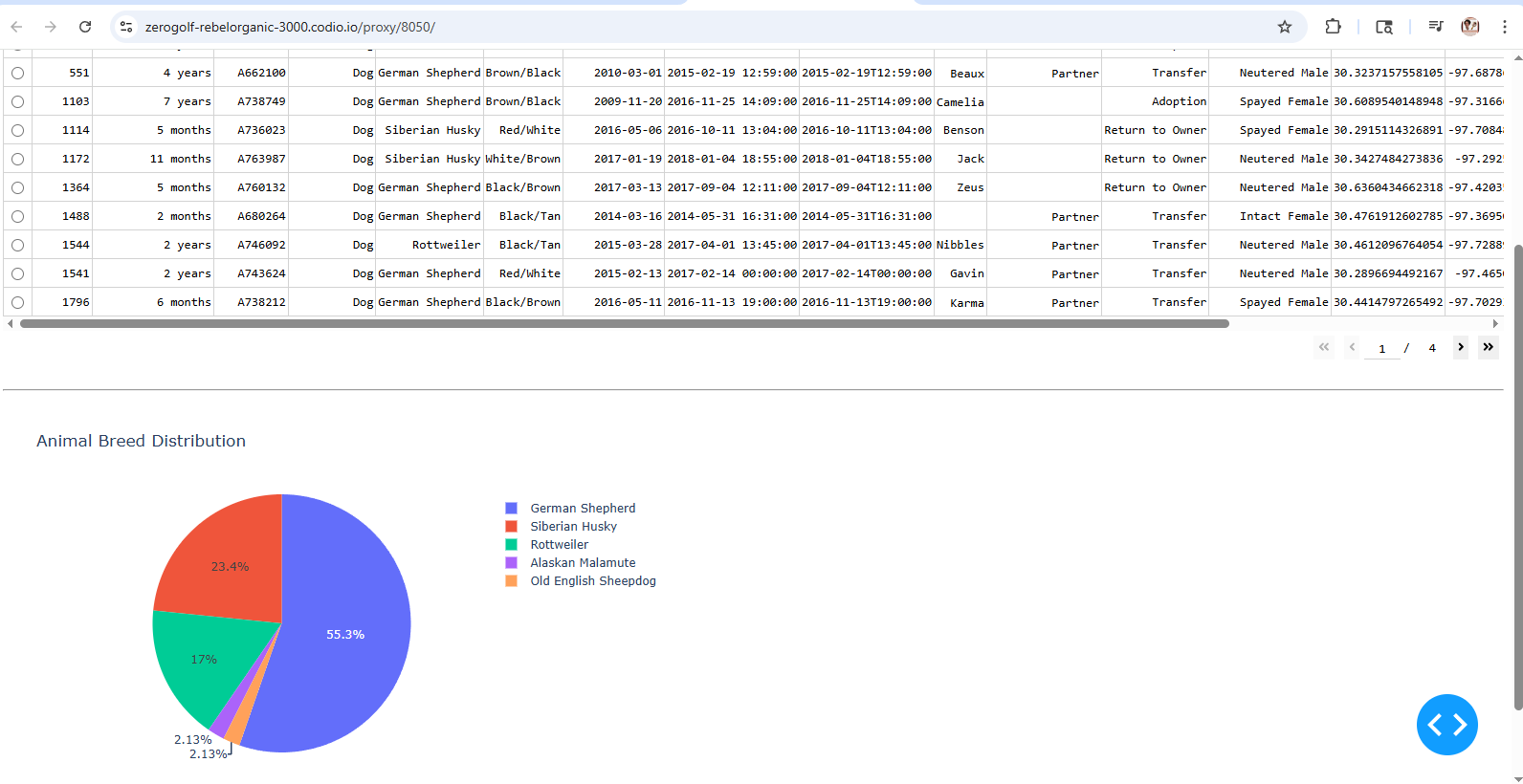
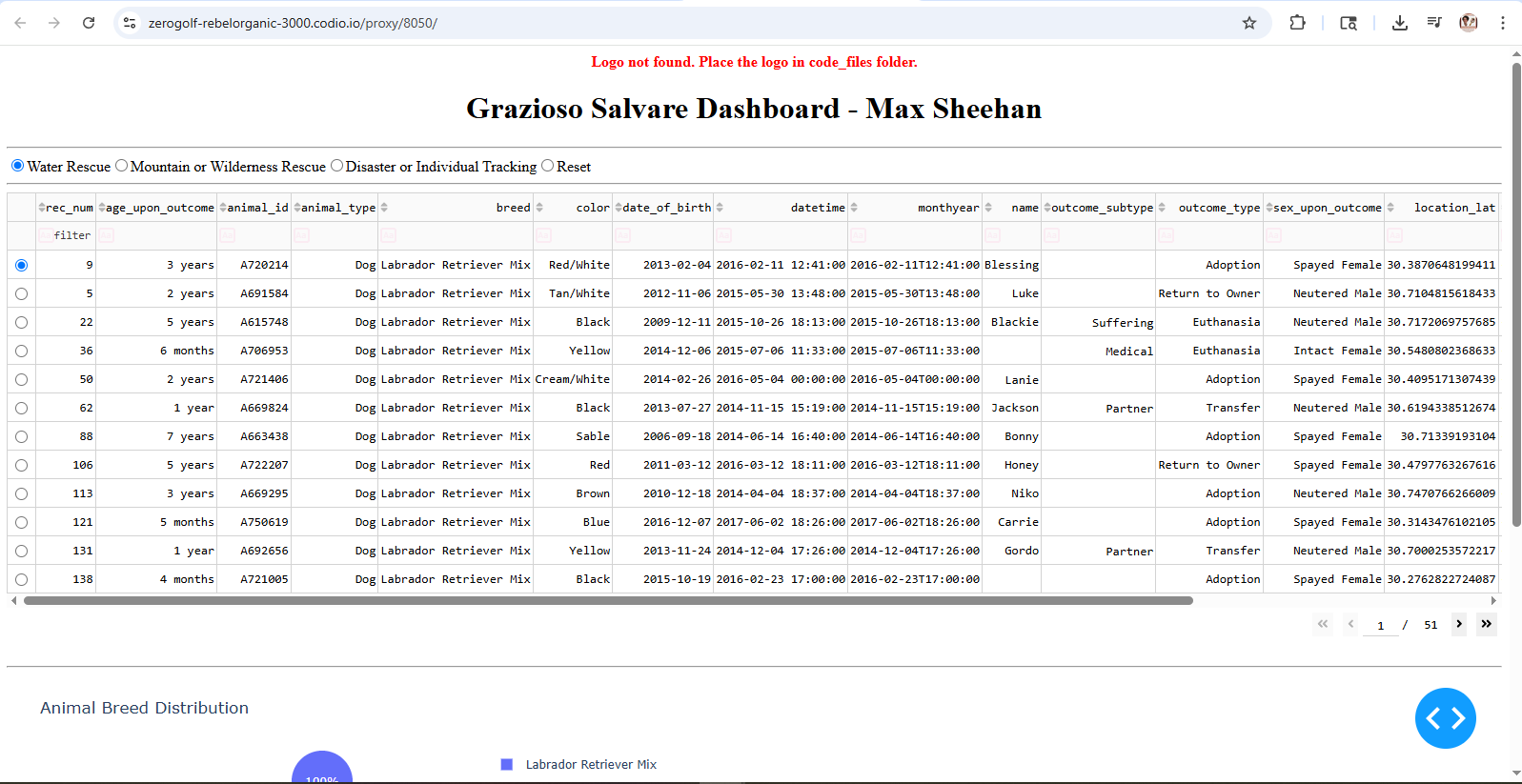
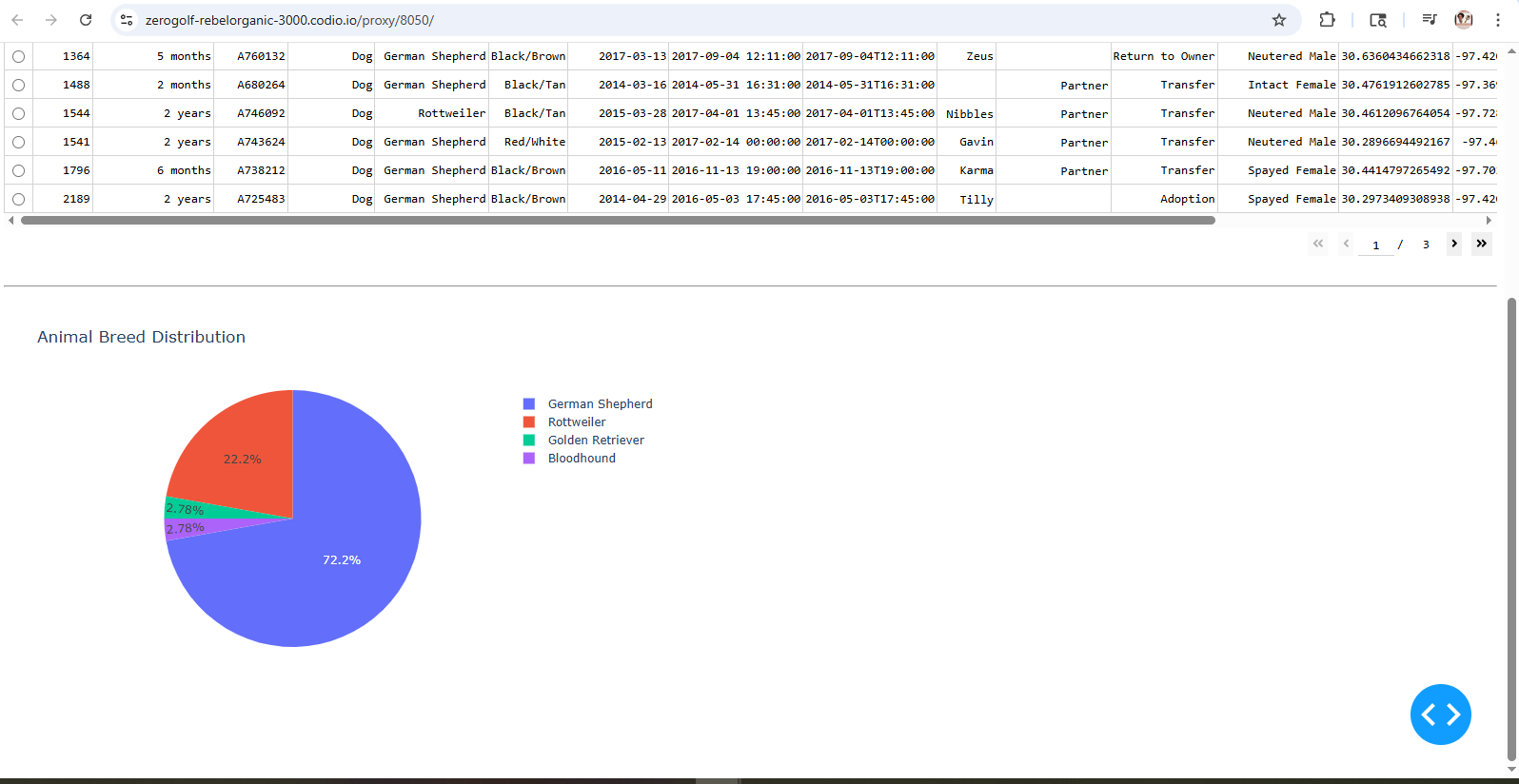
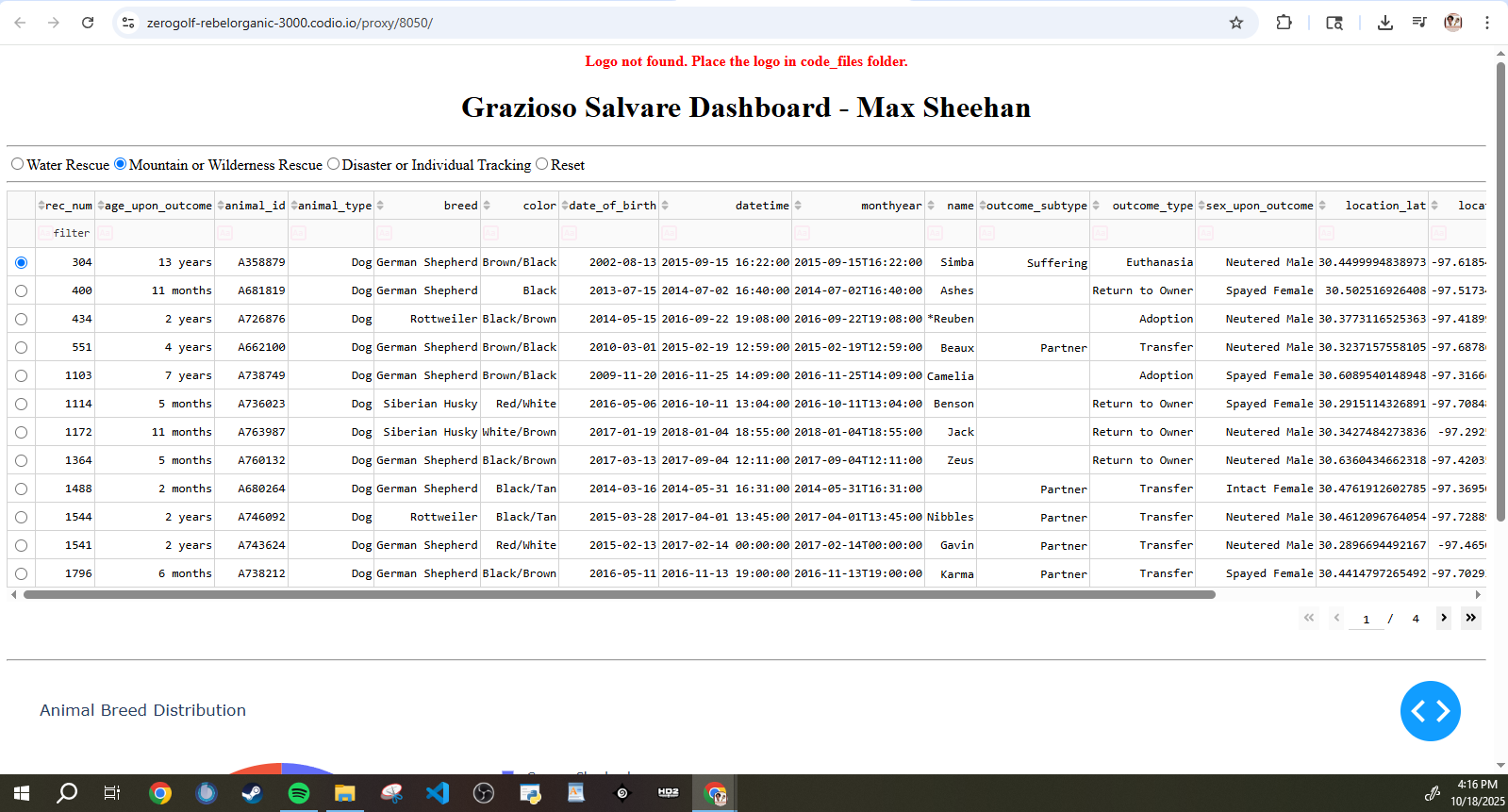
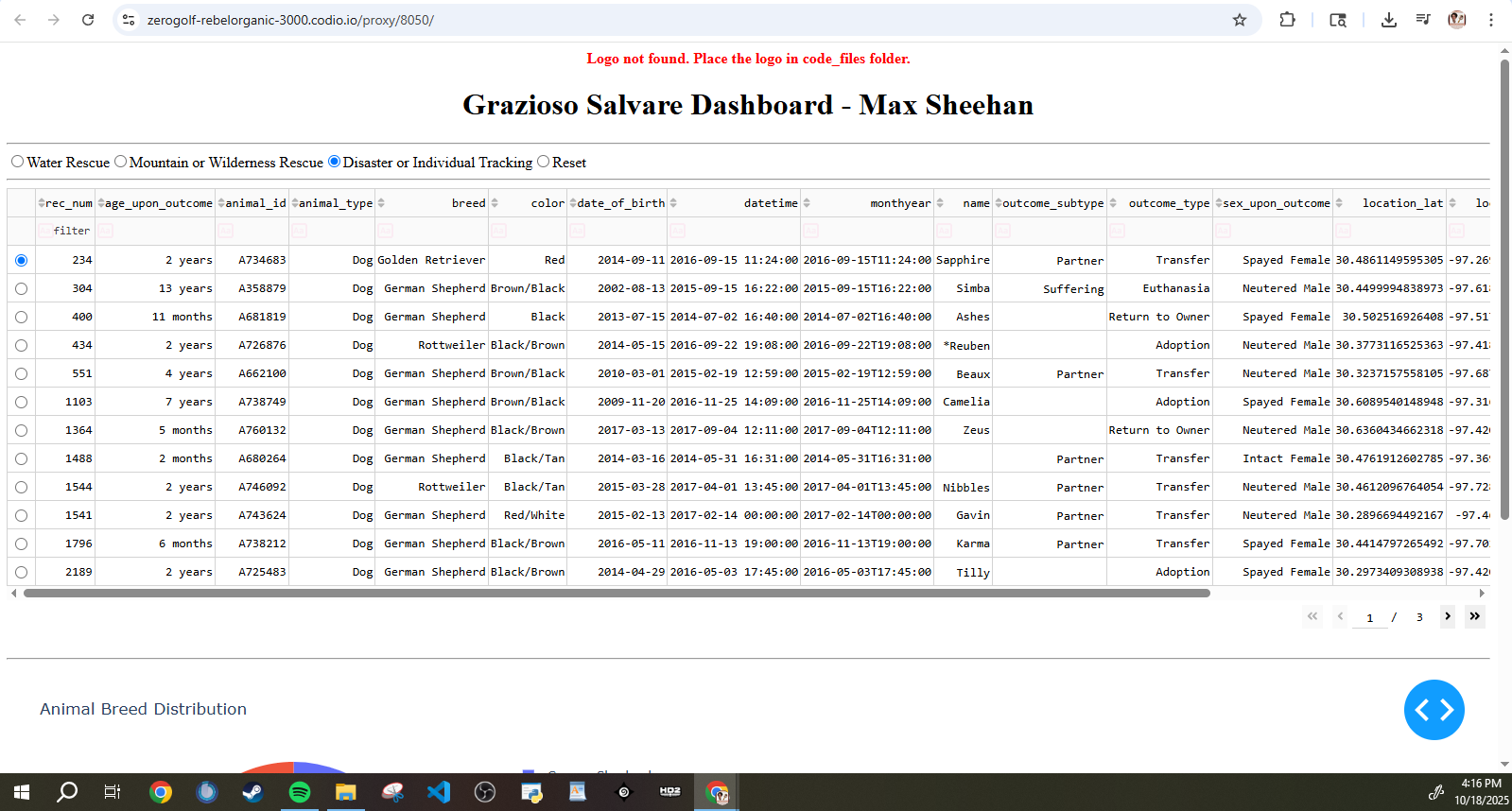
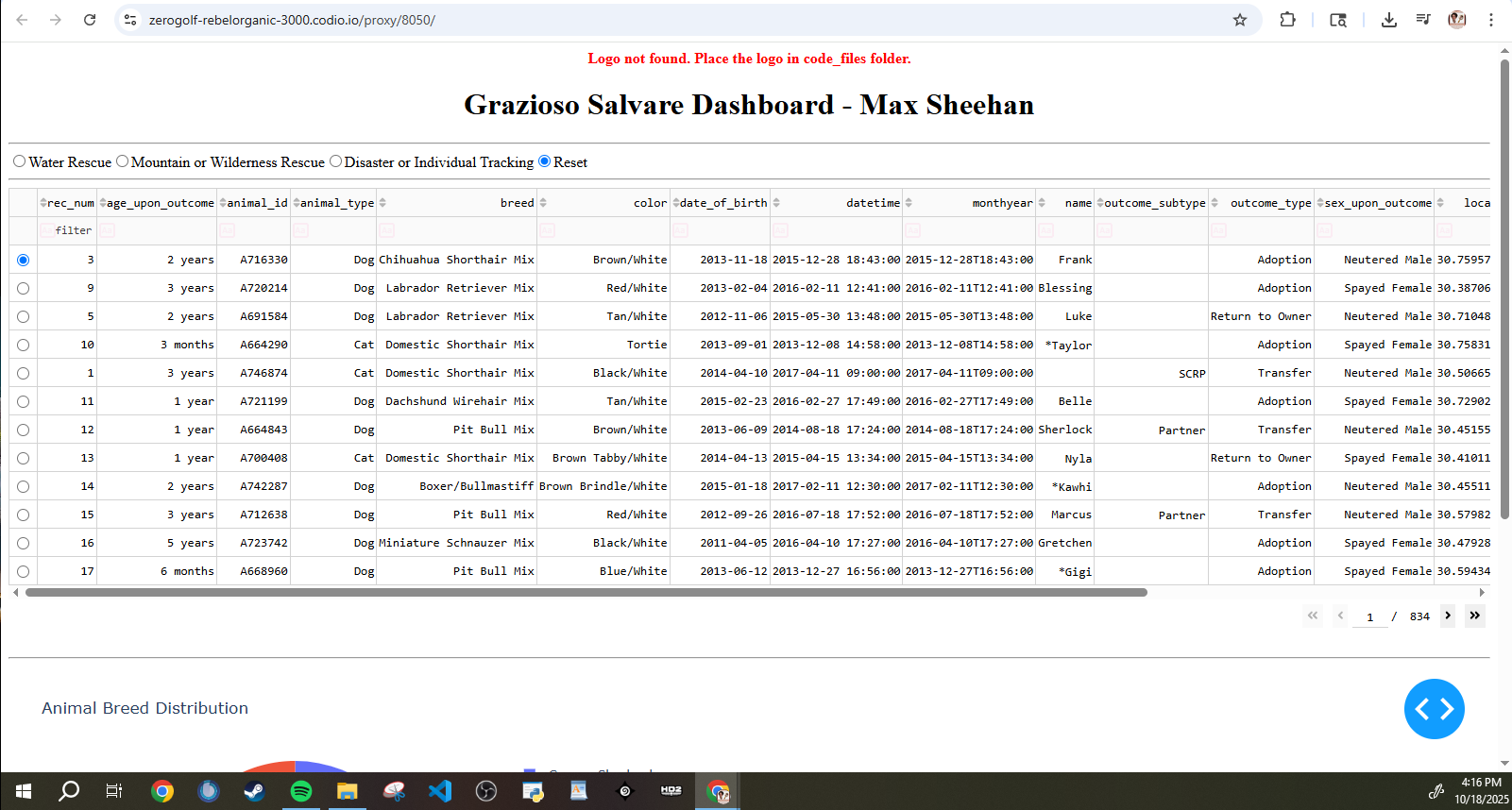
**Execution / Purpose**  
The Grazioso Salvare Dashboard is a web based dashboard that is tailor made to interact with users and allow organizations to visualize data simply and easily. Users can view all animals registered to a database with an interactive table that enables them to sort by key values. In this particular dashboard users can filter by how animals are reduced, we used water rescue, mountain / wilderness, and disaster rescues. Our dashboard allows users to sort each rescue by breed, and view their geolocation on a map. All of the breed data can also be observed through our interactive pie chart as shown below.   
  
  
  
  
  


**Tools**

MongoDB: MongoDB serves as the database backbone of our application, it has fantastic utility with storing JSON like documents which are very well integrated into existing python libraries. MongoDB also allows us to have multiple variables to allow more tools to sort animal data, finally its quick integration with pymongo ties the webapp and database together smoothly.   
  
JupytrDash / Dash Framework: These tools allowed rapid deployment of the web server and dashboard. The Dash Framework specifically allowed a quick HTML deployment of our dashboard, combined with interactive data tools like filtering, and pie charts.

Dash works by having view tools, and controller tools. WIth the controller tools users can easily interact and update the dashboard they see on input, this is how we’re able to quickly filter and select different data types.

Plotyly: Plotly allowed us to easily generate the interactive pie chart of our animal breeds.

Pandas: another Python tool that easily integrated with MongoDB and Dash to allow quick manipulation and visualization of data.

**Steps to Success**

First you need to create a database or at least validate you have one that functions, we created ‘aac’ with our animal database. The database was already populated with our animal records but we used key values to allow quick querying like name , breed, age etc  
  
Next we created basic CRUD function of our database using the python class Animal Shelter, this allowed us to use pandas for efficient searching and CRUD Function.   
  
We then design a dashboard using Dash, this allows us HTML to format our info, and allows us to implement interactive tools like charts, data tables, and graphs.  
  
We then expended on our Dash and allowed filtering, we can filter by the type of rescue and our pie chart is pulled from the rescued animal data.  
  
All of our testing and verifications of functions worked and we documented them with our screenshots, but this type of QA is crucial for success.

**Challenges**

A variety of issues and challenges came with this project.Most of them came from how many different systems and parts came together to create our dashboard. I’ve never dealt with so many moving pieces and integrating them. We had a database, an html aspect, dash, pandas, just a variety of different tools intersecting to make our final product. I had many issues integrating the existing code into the templated code provided, most of my challenges were overcome from basic trail and error, and a lot of research. Luckily we’ve never existed in a better time for resources to trouble shoot errors and show examples of working products that exist with these tools already implemented together.