**CS-340 Dashboard Final Project**

Bobby Eigenfeld

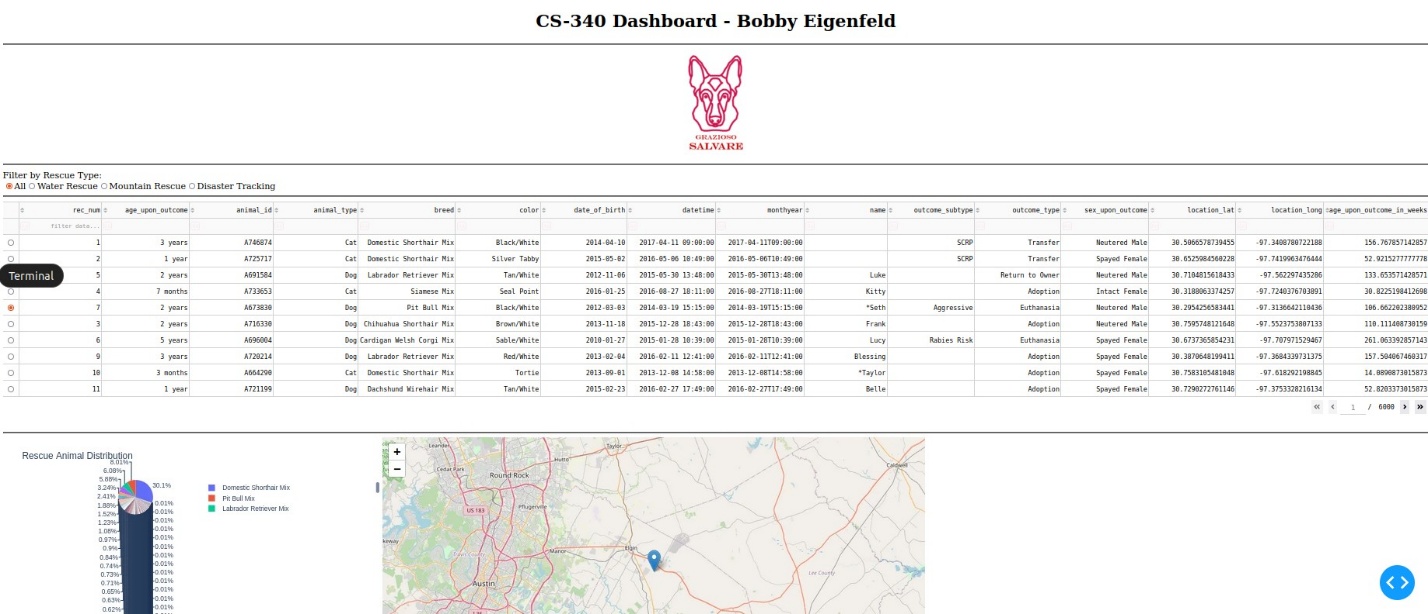
**Project Overview**

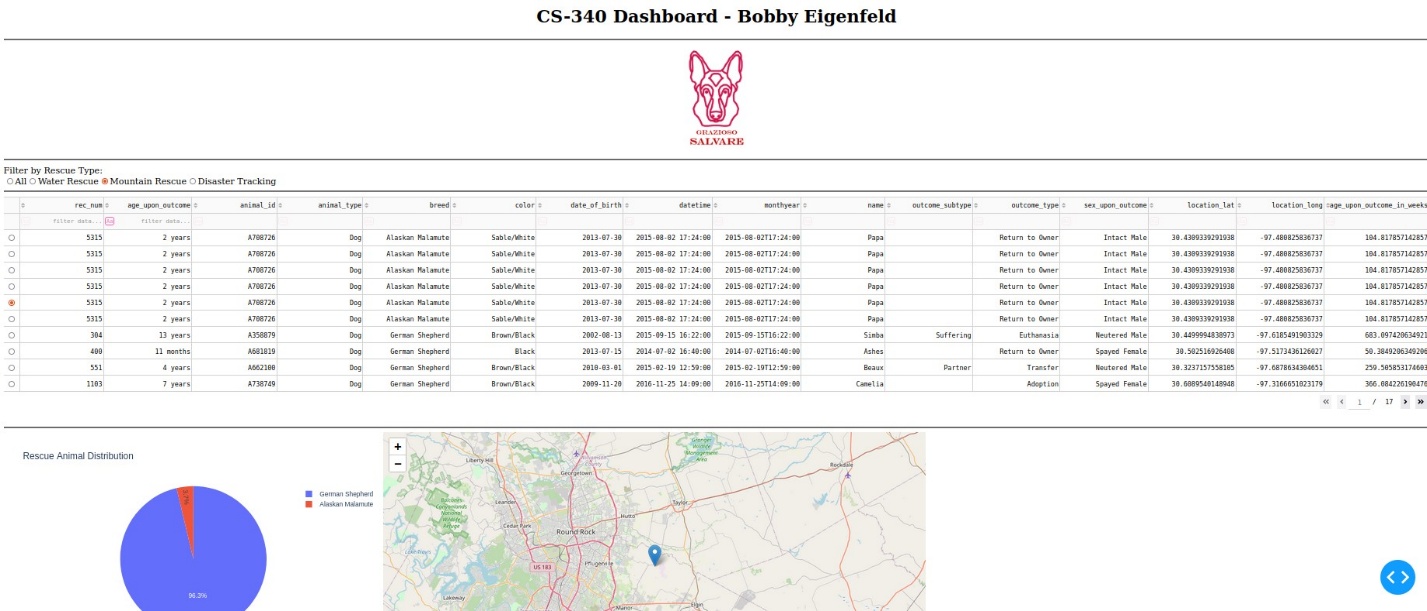
This project is a web-based interactive dashboard that allows users to visualize and filter data from an animal shelter database. The dashboard is built using **Dash**, **Plotly**, and **Dash Leaflet**, and it connects to a **MongoDB** database through a custom Python CRUD module (ProjectOne.py). Users can filter rescue animals by category, view data in an interactive table, generate graphs, and see location-based animal rescue data on a map.

**Features**

* Connects to a **MongoDB** database for animal shelter data
* Interactive filtering options for rescue categories
* Displays data using an interactive **Dash DataTable**
* Pie chart visualization of animal breeds
* Map integration using **Dash Leaflet** to display animal locations
* Interactive styling when selecting table columns

**Screenshots**

****

****

****

****

**Tools & Technologies Used**

**1. Python & Dash Framework**

* **Dash**: Used for building interactive web applications with Python. Provides structure for the **view** and **controller** of the dashboard.
* **Plotly**: Enables interactive graphing features.
* **Dash Leaflet**: Used to integrate maps and display location-based data.

**2. MongoDB (Model Component)**

MongoDB was chosen for this project because:

* It is a **NoSQL database**, which allows storing flexible and hierarchical data.
* It provides **fast querying** and **indexing**, making it efficient for retrieving animal shelter data.
* It is easily integrated with Python through the pymongo library.

**3. ProjectOne (CRUD Module)**

* ProjectOne.py is a custom CRUD module that handles database operations (Create, Read, Update, Delete).
* The read({}) function retrieves data from MongoDB and converts it into a Pandas DataFrame.

**4. JupyterDash**

* Used to run **Dash** applications inside Jupyter Notebook.

**5. Other Libraries**

* pandas – Used for data manipulation.
* numpy – Used for numerical operations.
* matplotlib – Additional plotting capabilities.
* base64 – Encodes the logo image for embedding in the dashboard.

**Steps to Reproduce the Project**

**1. Install Required Libraries**

Ensure you have Python installed and install the following dependencies:

pip install dash jupyter-dash plotly dash-leaflet pandas numpy matplotlib pymongo

**2. Set Up MongoDB Connection**

* Use a MongoDB database that contains the animal shelter data.
* Ensure ProjectOne.py contains the correct **username**, **password**, and **database details** for the connection.

**3. Run the Dashboard**

* Open Jupyter Notebook.
* Run the Python script that contains the **Dash application**.
* The dashboard should display the interactive table, charts, and map.

**Challenges & Solutions**

**1. Handling Data Formatting Issues**

**Challenge:** The \_id field from MongoDB caused datatype conflicts in the Dash DataTable.  
**Solution:** The \_id column was dropped before passing the data to Dash.

**2. Styling Issues in Dash DataTable**

**Challenge:** The NoneType error occurred when updating style\_data\_conditional.  
**Solution:** Ensured that selected\_columns was not None before iterating over it.

**3. Handling Empty Data in Callbacks**

**Challenge:** When no rows were selected in the DataTable, the graph and map callbacks would fail.  
**Solution:** Added conditions to check for empty viewData before processing.