Mike Miller

Purpose: The purpose of this project was to create a full stack web application which displays data about animals in Austin’s animal care centers which also allows the user to search for specific queries. Below are pictures of the app functioning, with the required radio buttons filtering results from the data table and chart.

A screenshot of a computer

Description automatically generatedA screenshot of a map

Description automatically generatedA screenshot of a map

Description automatically generatedA screenshot of a map

Description automatically generated

Tools: MongoDB was used to access the database containing information about the animals. A script was written to have access to this database with full CRUD functionality. It was used as its NoSQL architecture allows for the maintenance of a flexible and accessible database. It facilitates the back end of this project.

The Dash framework was used as the front end. Through it, an interactive webpage was made in Python. This handled the display of the webpage and sent search query requests to the back end. The front end code was written in a Jupyter notebook.

Links:

<https://www.mongodb.com>

<https://dash.plotly.com>

<https://jupyter.org>

Steps: The first step in completing this project was reading the necessary requirements as well as the accompanying Dashboard Specification Document. This detailed what needed to be done and the style in which it needed to be done. From there, I used the template project file to fulfill these requirements.

I added the needed visual features to the project first. This included the logo, the unique identifier, the map, customizing the data table, and adding radio buttons for search queries. After that, I gave functionality to the radio buttons so they would display the search results based on the input. After that, I added a pie chart to show a visual representation of the data.

The biggest challenge in completing the project was getting the data to show on the table after the query. I would get the data successfully, but then I would get an error saying it could not be displayed on the table. It took me a while to realize that it was because I was not dropping the \_id column being returned from MongoDB. When that is included, it crashes the table when it is loaded to the page. To fix this, I just dropped that column after retrieving it, but before sending it to Dash.