Brylene Patrick

CS 340

Project Two

**Animal Shelter Database**

This project is a dashboard for an animal shelter that displays information about animals that are available for adoption. The dashboard includes a table that shows information about the animals, such as their name, breed, sex, and age, and allows the user to filter the table by rescue type. In addition to the table, the dashboard includes a chart that shows the distribution of animal colors and a map that displays the location of the shelter.

**Motivation**

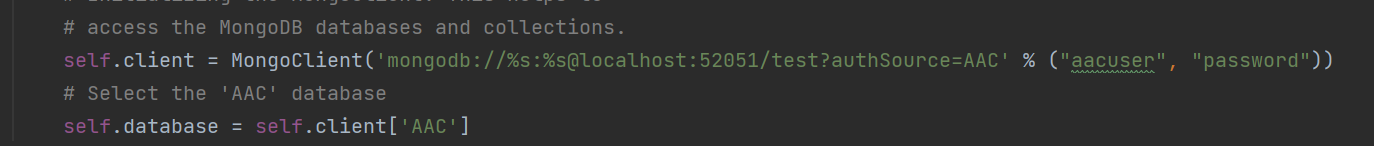
To create a dashboard that displays information about animals in an animal shelter. The goal is to help animal shelter staff and potential adopters easily find information about the animals. The dashboard allows users to filter the information by rescue type and view it on a table or on a map. This can help staff find the right homes for the animals and help potential adopters find the right pet for them.

**Getting Started**

To get started with this project, have a MongoDB database instance running and provide the necessary authentication details to connect to the instance.

**Database and user authentication**

In the AnimalShelter class, we are connecting to a MongoDB instance running on localhost with port 52051 and using the "AAC" database. We are also providing the username and password for the "aacuser" user to authenticate and access the "AAC" database.



**Retrieve data from the database**

Once the connection is established, the data is retrieved from the database using the read() method of the AnimalShelter class. The data is then converted to a pandas dataframe using the from\_records() method.

Text

Description automatically generated

Text

Description automatically generated

**Define the layout of the dashboard**

This includes adding HTML components such as headers, images, dropdowns, buttons, tables, and graphs to the layout. In this project, the layout is defined using the dash\_html\_components and dash\_core\_components libraries.

A picture containing chart

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Text

Description automatically generated

I encountered some challenges while working on this project, such as creating the dashboard layout using the right components and implementing the callbacks for interactivity between components. However, I was able to overcome these challenges by taking a step-by-step approach, reading up on how to use the tools, and experimenting until I found the best solution.

**Installation**

To get started with this project, you need to install the following packages using pip install in your terminal:

* Dash
  + A python web framework for building analytical web applications, was used to provide the view and controller structure for the web application. Dash makes it easy to create interactive and responsive dashboards by providing a set of high-level components for building HTML-based user interfaces.
  + <https://dash.plotly.com/>
* MongoDB
  + Used as the model component of the development because it is a document-based NoSQL database that is flexible, scalable, and easy to work with. Additionally, MongoDB provides an interface for Python that allows for efficient and convenient data manipulation and querying.
  + <https://www.mongodb.com/try/download/community>
* jupyter\_plotly\_dash
  + Allow users to create web-based dashboards within the Jupyter Notebook environment. This makes it easy to prototype, test, and deploy interactive dashboards without the need to leave the notebook environment.
* Pymongo
  + Provides an API that enables Python code to connect to a MongoDB server, insert and retrieve data, update and delete documents, and perform various other operations.
  + <https://pymongo.readthedocs.io/en/stable/>
* Python 3.8 or later
  + Python 3.8 was used as the main programming language to develop the module.
  + <https://www.python.org/downloads/>
* Pandas
  + A data manipulation library used to read data from MongoDB and create a pandas DataFrame from it.numpy
  + <https://pandas.pydata.org/>
* Plotly
  + A data visualization library that creates interactive charts and graphs. It is used to create the bar chart in the dashboard.
  + <https://plotly.com/python/>
* Base64
  + A module used to encode and decode binary data. It is used to encode the Grazioso Salvare logo image so that it can be displayed in the dashboard.
* NumPy
  + A numerical computing library used for scientific computing with Python. It is used to perform mathematical calculations on the data in the pandas DataFrame.
  + <https://numpy.org/>

**Usage**

Graphical user interface, text, application, Word

Description automatically generated

* This shows image as the header for the SNHU CS-340 Dashboard. The logo image is loaded from the 'Grazioso Salvare Logo.png' file using base64 encoding. The HTML layout is defined using the Div and Center components from the dash library. This image also shows a centered and bolded H1 header with the text "SNHU CS-340 Dashboard" and a centered and bolded H2 header with the text "Brylene Patrick".

Graphical user interface, application, table, Excel

Description automatically generated

* This image shows a filter and a data table using the Dash library. The filter is created using the Label and Dropdown components, with options for different types of rescues. A “Reset Filters” button is included. The data table is created using the DataTable component, with columns and data passed from a pandas DataFrame. The table has several features, including native filtering and sorting, multi-column sorting, multi-row selection, and pagination.
* The reset filter button is implemented using the reset\_filter function, which takes a list of widgets as input and returns all the widgets in the list.

Graphical user interface, application, table, Excel

Description automatically generated

* The filter is implemented using the water\_rescue\_filter function, which takes a list of widgets as input and returns only those widgets that are related to water rescue.

Graphical user interface, application

Description automatically generated

* The filter is implemented using the disaster\_or\_individual\_tracking\_filter function, which takes a list of widgets as input and returns only those widgets that are related to disaster or individual tracking.
* The filter is implemented using the mountain\_or\_wilderness\_rescue\_filter function, which takes a list of widgets as input and returns only those widgets that are related to mountain or wilderness rescue. Graphical user interface, application

  Description automatically generated

Map

Description automatically generated

* The resulting pie chart shows the distribution of outcomes in the animal shelter data, with each slice of the chart representing a different outcome type. The size of each slice is proportional to the number of animals that had that outcome. The pie chart allows for a quick and easy visual representation of the distribution of outcomes in the dataset.
* The resulting map shows the location of animals in the shelter data. Each marker on the map represents the location of an animal, with a tooltip displaying the breed of the animal and a popup displaying the animal ID. The map allows for a visual representation of the distribution of animals in the dataset and can be useful for identifying patterns or clusters of animals in certain locations.

**Contact**

Brylene Patrick