Cs-340

Zeb Berry IV

Project 2 ReadME

**Purpose of the Dashboard Application**

This dashboard was developed for Grazioso Salvare, a company that specializes in identifying and training dogs for search-and-rescue missions. The dashboard represents the client-facing component of a full-stack solution built upon the AnimalShelter Python module created in Project One. Designed to provide a user-friendly and intuitive interface, the dashboard enables users to visually explore animal records pulled from a MongoDB database. The main objectives of the application are to display available dogs, apply rescue-specific filters, visualize breed distribution data, and pinpoint geolocation information for individual animals.

**Tools and Technologies Used**

This project utilizes a variety of tools and frameworks to ensure smooth interaction between the frontend dashboard and the backend database. MongoDB serves as the NoSQL database used to store and retrieve animal shelter records. The PyMongo library acts as the MongoDB driver for Python, allowing efficient execution of database operations within custom CRUD methods. Dash, an open-source Python framework developed by Plotly, is used to build the web-based dashboard interface. Dash Leaflet is used to render an interactive map displaying animal locations. pandas is used for transforming MongoDB documents into DataFrame objects, while Plotly Express is responsible for creating responsive charts such as the breed distribution pie chart. JupyterDash was chosen to facilitate development within a Jupyter Notebook environment.

**Dashboard Functionality**

The dashboard includes the following key components:

### 1. Interactive Filter Options

Users can filter animals using radio buttons:

* **Water Rescue**
* **Mountain/Wilderness Rescue**
* **Disaster/Tracking**
* **Reset** (to show all animals)

Each filter performs a database query through the AnimalShelter class and updates the table, chart, and map accordingly.

### 2. Interactive Data Table

A dynamic table displays animal records with the following features:

* Sortable columns
* Row selection
* Highlighted columns

### 3. Pie Chart – Breed Distribution

A responsive pie chart updates based on filtered data, helping visualize the most common breeds in each rescue category.

### 4. Interactive Map

Using Dash Leaflet, a map displays the selected animal's geolocation with:

* A tooltip showing the breed
* A popup showing the animal's name

**Filtering and Backend Query Details**

The dashboard utilizes custom filtering methods implemented within the AnimalShelter CRUD class. These methods return filtered datasets from the MongoDB collection based on criteria such as breed, age, and sex. For example, the filter\_water\_rescue() method returns Labrador Retriever Mixes, Golden Retriever Mixes, and Chesapeake Bay Retrievers that are intact males under two years old. Similar logic is applied to methods for mountain/wilderness and disaster/tracking filters, ensuring that only the most appropriate candidates are presented to the user.

Filtering is done using custom methods in the AnimalShelter CRUD module:

def filter\_water\_rescue(self):

return self.collection.find({

"breed": {"$in": ["Labrador Retriever Mix", "Golden Retriever Mix", "Chesapeake Bay Retriever"]},

"sex\_upon\_outcome": "Intact Male",

"age\_upon\_outcome\_in\_weeks": {"$lte": 104} })

**Testing and Dashboard Execution**

To launch the dashboard, the user simply runs the final cell in the Jupyter Notebook containing the app.run\_server(debug=True) command. Upon loading, the dashboard displays all available animals by default. When a filter option is selected, the data table and pie chart update accordingly. Users can then click on a specific row to update the map display, allowing for full visual interaction between the data components. This integrated functionality demonstrates real-time updates and ensures the dashboard meets the usability requirements set forth by Grazioso Salvare.

Proof of successful execution is demonstrated through a series of screenshots captured during the testing process. These include views of the unfiltered dashboard, each rescue filter applied individually, a reset view showing all records, and the map view after selecting a specific animal.

**Conclusion**

The Project Two dashboard completes the full-stack application for Grazioso Salvare. This client-facing tool integrates backend CRUD operations with an intuitive frontend interface. It allows users to explore animal records, apply mission-specific filters, visualize breed data, and geolocate animals on an interactive map. The dashboard demonstrates practical application of database integration, data visualization, and responsive UI design and represents a robust solution for supporting Grazioso Salvare’s mission of training dogs for life-saving rescue efforts.