README

Noah Duarte

Southern New Hampshire University

CS340: Client/Server Development

Professor Jeff Sanford

17 August 2023

Animal Shelter Dashboard

## About the Project/Project Title

This project aims to develop a web-based application that assists Grazioso Salvare in identifying and categorizing potential search- and rescue dogs from data provided by a non-profit agency operating an animal shelter in the Austin, Texas, region. The application includes a database and a client-facing web dashboard, allowing Grazioso Salvare to access and manage the data efficiently.

**Motivation**

Grazioso Salvare is crucial in search-and-rescue operations, utilizing specially trained dogs to save lives in challenging and life-threatening situations. Grazioso Salvare requires a tool to analyze animal shelter data and identify dogs with suitable profiles for search-and-rescue training to enhance its operations. This motivation stems from the necessity to ensure that only dogs with the most potential are selected and trained, optimizing the success of rescue missions.

The software application aims to streamline the identification process by considering factors such as age, breed, and rescue proficiency. By creating a user-friendly dashboard, Global Rain aims to provide Grazioso Salvare with a powerful tool that minimizes user errors, reduces training time, and enhances the overall efficiency of dog selection for training.

**Getting Started**

To set up the Animal Shelter Dashboard locally, follow these steps:

1. Clone the repository to your local machine.

2. Install the required Python packages by running: `jupyter\_dash, dash\_leaflet, pandas, matplotlib, and pymongo`

3. Ensure you have a MongoDB server running with the appropriate credentials.

4. Open the `ProjectTwoDashboard.ipynb` notebook using Jupyter or JupyterLab.

5. Modify the MongoDB credentials, host, and other parameters as needed in the notebook.

6. Run the notebook to launch the dashboard locally.

**Installation**

The following tools are required to run the project:

- Python 3.6+

- Jupyter or JupyterLab

- MongoDB server

**Usage**

The Animal Shelter Dashboard provides the following functionalities:

- Filter data by rescue type (Water Rescue, Mountain/Wilderness Rescue, Disaster Rescue/Individual Tracking).

- Sort and filter data in an interactive data table.

- Visualize breed distribution using a pie chart.

- Display animal locations on an interactive map.

**Code Example**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Tests**

To run tests for the AnimalShelter class from the `crud` module, use the following code:

# Create an instance of AnimalShelter with MongoDB credentials

db = AnimalShelter(username, password, host, port, db\_name, collection\_name)

# Test create method

data = {...}

db.create(data)

# Test read method

result = db.read({'breed': 'Labrador Retriever'})

print(result)

# Test update method

search = {'breed': 'Labrador Retriever'}

update\_data = {'age': 24}

modified\_count = db.update(search, update\_data)

print(f"Modified {modified\_count} records")

# Test delete method

delete\_count = db.delete(search)

print(f"Deleted {delete\_count} records")

**Execution Screenshots**

This screenshot shows the initial launch and query for all documents in the database.

A screenshot of a computer

Description automatically generated

This screenshot shows a water rescue filter.

A screenshot of a computer

Description automatically generated

This screenshot shows the Mountain/Wilderness Rescue:

A screenshot of a computer

Description automatically generated

This screenshot show the Disaster Rescue/Individual Tracking filter:

A screenshot of a computer

Description automatically generated

**Roadmap/Features**

Currently, the Animal Shelter Dashboard provides basic data filtering, visualization, and mapping functionalities. Future features could include:

- Additional interactive visualizations (e.g., time-based trends, adoption rates).

- Updates and notifications.

**Contact**

Noah Duarte