**Grazioso Salvare Dashboard for Austin Animal Center (AAC) Database**

**About the Project/Project Title**

**Project Title:**  
Grazioso Salvare Dashboard for Austin Animal Center (AAC) Database

**About the Project:**  
This project involves developing a fully functional web application dashboard that interacts with the Austin Animal Center's data stored in a MongoDB database. The dashboard provides Grazioso Salvare with the ability to filter, visualize, and manage animal shelter data efficiently. The dashboard includes CRUD (Create, Read, Update, Delete) operations via a Python module and features an intuitive interface that displays interactive filtering options, an interactive data table, a pie chart, and a geolocation map.

**Motivation**

The motivation behind this project is to create a user-friendly, intuitive dashboard that enables Grazioso Salvare to manage and visualize animal shelter data effectively. By integrating CRUD operations with interactive visualizations, this project aims to streamline data management processes, assist in identifying suitable rescue dogs, and enhance the overall user experience when interacting with the data.

**Getting Started**

**Prerequisites**

* MongoDB installed and running
* Python installed
* Python packages: pymongo, bson, dash, plotly, dash\_leaflet, pandas, matplotlib, numpy

**Setup**

**1. Database Setup:**

* The AAC database is hosted on a MongoDB instance. Ensure that the MongoDB service is running and accessible. The user account aacuser should be created with the readWrite role on the AAC database.
* Import the data using the mongoimport tool with the provided aac\_shelter\_outcomes.csv file.
* A screenshot of a computer

  Description automatically generated

**2. Installation:**

* Ensure a stable connection to Apporto services through the student portal lab access.

**3. MongoDB Setup:**

* Verify that MongoDB service is running and accessible using the following connection details:
  + **Host:** nv-desktop-services.apporto.com
  + **Port:** This is unique to each user and must be verified on your own machine.
  + **Database:** AAC
  + **Collection:** animals
  + **User:** aacuser
  + **Password:** snhu1234
  + A screenshot of a computer program

    Description automatically generated

**Installation:**

1. Clone the repository or download the project files.
2. Navigate to the project directory.
3. Install the required Python packages using pip:

pip install -r requirements.txt

1. Run the Jupyter Notebook or the Python script to start the dashboard.

**Usage**

**Running the Dashboard:**

1. Start your MongoDB server and ensure it is running on the specified host and port.
2. Run the Jupyter Notebook or Python script (ProjectTwoDashboard.ipynb) to start the dashboard.
3. Open the provided local server URL in your web browser.

**Dashboard Features:**

* **Interactive Filter Options:** Allows users to filter the AAC data by rescue type (Water Rescue, Mountain Rescue, Disaster Rescue, or Reset to view all).
* **Interactive Data Table:** Displays the filtered data dynamically based on the selected filter.
* **Pie Chart:** Visualizes the preferred breeds by rescue type.
* **Geolocation Map:** Displays the location of the selected animal on a map based on the filtered data.

**Code Example:**

To use the underlying CRUD operations, incorporate the AnimalShelter class and call the create, read, update, and delete methods as shown in the example below:

from animal\_shelter import AnimalShelter

# Connect to the database

shelter = AnimalShelter(username='aacuser', password='snhu1234')

# Create a new record

new\_animal = {

'animal\_id': 'A100001',

'name': 'Buddy',

'breed': 'Labrador Retriever Mix',

'age': '2 years',

'color': 'Black/White'

}

shelter.create(new\_animal)

# Read records

query = {'breed': 'Labrador Retriever Mix'}

animals = shelter.read(query)

# Update a record

update\_query = {'animal\_id': 'A100001'}

update\_values = {'$set': {'age': '3 years'}}

shelter.update(update\_query, update\_values)

# Delete a record

delete\_query = {'animal\_id': 'A100001'}

shelter.delete(delete\_query)

**Screenshots**

**A red line drawing of a dog

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**Contact**

**Jose Lara Hernandez**