# CS 340 README Joel Hays

## Project Overview

The Grazioso Salvare Dashboard is a user-friendly web application designed to help the Grazioso Salvare organization identify and categorize dogs suitable for search-and-rescue training. The dashboard interfaces with a MongoDB database to dynamically display and filter data, providing valuable insights through various interactive widgets and charts.

## Required Functionality

The dashboard includes the following functionalities:

1. **Interactive Filters**: Options to filter data based on rescue types:
   * Water Rescue
   * Mountain or Wilderness Rescue
   * Disaster or Individual Tracking
   * Reset (to view all data)
2. **Dynamic Data Table**: Displays relevant information such as animal ID, breed, color, age, and location, and updates based on selected filters.
3. **Geolocation Chart**: Shows the exact location of the selected dog within the Austin, Texas region.
4. **Bar Chart**: Displays the ages of dogs by breed, updating dynamically based on selected filters.
5. **Branding**: Includes the Grazioso Salvare logo and a unique identifier with the developer's name.

https://www.loom.com/share/9f739d0130484ae08e4704fd96b9bbd0?sid=9b42e3c3-661c-4ad2-8650-12afb2f03623

*Purpose*

*The CRUD Python module provides a structured way to interact with the MongoDB database for the Grazioso Salvare project. This module allows users to perform basic database operations such as creating, reading, updating, and deleting documents in the animal collection.*

## Usage

We used the PyMongo driver for MongoDB in this module. PyMongo was chosen due to its robustness, ease of use, and extensive documentation, making it ideal for interacting with MongoDB from Python applications.

### Description

The primary components of this code include:

### 1. CRUD Python Module (crud.py):

### Create Method: Inserts new documents into the database.

### Read Method: Queries and retrieves documents from the database based on specified criteria.

### Update Method: Updates existing documents in the database.

### Delete Method: Removes documents from the database.

### 2. Testing Script (test\_crud.py):

### A script designed to test the functionality of the CRUD operations. This script demonstrates the module's capability to insert, query, update, and delete records in the database.

### Key Features

### Object-Oriented Design: The program uses object-oriented programming principles to ensure modularity and reusability of code.

### Error Handling: Each method includes exception handling to manage potential errors gracefully.

### Database Connection: Utilizes the PyMongo driver to interact with MongoDB, ensuring a robust and efficient connection.

### Authentication: Ensures secure access to the database by requiring valid user credentials.

### *A screenshot of a computer program Description automatically generated*Screenshots

User permission verification.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedImporting the appropriate database.

Example test code performing required functions of Create, Read, Update, and Delete.

## Contact

Your name: Joel Hays