**CS 340 Project 2: Grazioso Salvare Rescue Dashboard**

**Author**: Kain Mason  
**Course**: CS 340 - Client/Server Development  
**Project**: Two – Full Stack Dashboard Application

**Project Overview**

This project demonstrates a functional, full-stack dashboard application built for Grazioso Salvare, a company that identifies and trains rescue dogs. The dashboard pulls data from a MongoDB database using a custom Python CRUD module (animal\_Shelter.py) and displays it interactively using Dash by Plotly.

Users can filter dog profiles by rescue type and visualize outcomes both through a data table and geolocation/chart widgets.

**Technologies Used**

* **Python 3.x**
* **Dash & JupyterDash** – for interactive dashboards
* **Dash Leaflet** – for map visualization
* **Pandas & Plotly Express** – for data wrangling and charting
* **MongoDB + PyMongo** – for storing and retrieving structured animal records

**Authentication Setup**

This dashboard uses user-provided credentials to connect to the MongoDB instance. For this project, the aacuser account created in Module 3 is used.

Connection is established in the CRUD module with:

self.client = MongoClient(f"mongodb://{username}:{password}@{host}:{port}/?authSource=admin")

**How the CRUD Module Works**

The animal\_Shelter.py file defines a reusable class AnimalShelter that handles:

* create() – Adds new animal records to the MongoDB collection.
* read(query) – Retrieves data based on filters (supports empty query for full dataset).
* update(query, new\_values) – Updates one or more documents.
* delete(query) – Removes one or more documents.

This modular design makes it easy to connect the backend to any front-end dashboard or app.

**Dashboard Functionality**

The dashboard supports:

**Filterable Radio Buttons**:

* Water Rescue
* Mountain/Wilderness Rescue
* Disaster/Individual Tracking
* Reset (shows all animals)

**Data Table**:

* Displays dynamic dog records
* Supports filtering, sorting, and row selection
* Highlights a custom "rescue\_type" column based on selected filter

**Map Visualization**:

* Shows the selected dog's location using Dash Leaflet
* Centered on the latitude and longitude in the dataset

**Chart**:

* A bar chart (histogram) showing outcome types (e.g., Adopted, Transferred)

**Logo & Branding**:

* Custom Grazioso Salvare logo is displayed at the top of the dashboard
* Your name (Kain Mason) is included for identification

**Screenshots or Demo Proof**

Be sure to include these in your final documentation:

1. Dashboard start state (with logo, radio items, table, map, and chart visible)

A screenshot of a computer

AI-generated content may be incorrect.

1. Water Rescue filter activated

A screenshot of a computer

AI-generated content may be incorrect.

1. Mountain/Wilderness Rescue filter activated

A screenshot of a computer

AI-generated content may be incorrect.

1. Disaster/Tracking filter activated

A screenshot of a computer

AI-generated content may be incorrect.

1. Reset state showing all data

A screenshot of a computer

AI-generated content may be incorrect.

**File Structure**

├── Project2.ipynb # Dashboard UI + callbacks

├── animal\_Shelter.py # CRUD Python module

├── photo.png # Grazioso Salvare logo (shown at top)

└── README.md # This file

**How to Run**

1. Ensure MongoDB is running and contains the AAC dataset
2. Open the Project2.ipynb in Jupyter Notebook or Apporto
3. Run all cells to launch the dashboard