# CS 340 README Template

## About the Project/Project Title

**Project Title: Grazioso Salvare Interactive MongoDB Dashboard**

This project is a full-stack dashboard application developed for Grazioso Salver, a rescue-animal training organization. The goal of the project is to assist the organization in identifying dogs that are suitable for specialized rescue training programs – such as water rescue, mountain/wilderness rescue, disaster recovery, and individual tracking.

The dashboard integrates a MongoDB database containing animal outcome data from shelters in the Austin, Texas area. It includes filtering tools, interactive data visualizations, and geolocation mapping to allow users to quickly identify and analyze dogs that match Grazioso Salvare’s training criteria.

This project builds upon a previously developed Python CRUD module and demonstrates the use of the Dash framework to create a responsive, client-facing web application tailored for data-driven decision-making.

## Motivation

The Grazioso Salvare Interactive Dashboard was developed to address a specific need: helping a rescue-animal training organization quickly identify dogs suitable for various types of specialized rescue work. Manually sifting through shelter records to find animals that meet criteria like breed and age is time-consuming and inefficient.

This project was created to streamline that process by leveraging the power of MongoDB and interactive data visualization tools. The dashboard simplifies complex queries into user-friendly filter options, giving non-technical users the ability to access, visualize, and act on critical data without requiring deep database knowledge.

Additionally, this project serves as a demonstration of how client-server applications can apply CRUD principles and full-stack development in a real-world, mission-driven context.

## Getting Started

To get a local copy of this dashboard up and running, follow these simple steps:

**Prerequisites:**

Make sure you have the following tools installed on your system:

* Python 3.x - [Download Python | Python.org](https://www.python.org/downloads/)
* pip – Comes bundled with Python, used for installing dependencies
* MongoDB – Local instance or access to a remote MongoDB server
* Jupyter Notebook – For running and testing the dashboard interactively

You can install Jupyter with:

pip install notebook

**Installation Steps:**

1. Clone the repository
   1. git clone https://github.com/--username--/grazioso-salvare-dashboard.git
   2. cd grazioso-salvare-dashbaord
2. Install required Python packages
   1. pip install pandas dash dash-leaflet jupyter-dash plotly pymongo
3. Configure MongoDB Access
   1. Ensure the MongoDB URI in script.py points to your MongoDB instance
   2. Default credentials use:
      1. Username: aacuser
      2. Password: password (or set your own password)
4. Launch the Dashboard
   1. Open ProjectTwoDashboard.ipynb in Jupyter Notebook
   2. Run all cells to launch the dashboard
   3. A web link will appear to open the dashboard in your browser.

## Installation

To use the Grazioso Salvare Dashboard locally, you will need the following tools and libraries installed:

**Required Tools and Libraries:**

1. Python 3.x
   1. Used to run all scripts and backend logic
   2. [Download Python | Python.org](https://www.python.org/downloads/)
2. pip (Python Package Installer)
   1. Used to install Python libraries
   2. Comes pre-installed with most Python installations
   3. Verify with:
      1. pip –version
3. MongoDB
   1. Required to store and manage the AAC Outcomes dataset
   2. You can use either:
      1. A local MongoDB server: [Download MongoDB Community Server | MongoDB](https://www.mongodb.com/try/download/community)
      2. Or a remote instance like MongoDB Atlas: <https://www.mongodb.com/products/platform/atlas-database>
4. Required Python Libraries
   1. Install all dependencies using pip
      1. pip install pandas dash dash-leaflet jupyter-dash plotly pymongo
5. Jupyter Notebook (for launching the dashboard)
   1. Optional but recommended for development and testing
   2. Install with:
      1. pip install notebook

**Summary of Tools:**

1. Python
   1. Language used to build and run the app
2. pip
   1. Python package installer
3. MongoDB
   1. NoSQL database for storing animal data
4. pymongo
   1. Python client to interact with MongoDB
5. pandas
   1. Data manipulation and transformation
6. dash
   1. Web framework for the dashboard
7. dash-leaflet
   1. Mapping/geolocation functionality
8. jupyter-dash
   1. Integration of Dash within Jupyter Notebooks
9. plotly
   1. Interactive charts and graphs

## Usage

This section shows how to use the Grazioso Salvare Dashboard, including examples of how the code works and what users can expect when interacting with the dashboard.

**Code Example:**

Here’s how to use the CRUD module (script.py) to interact with the MongoDB animal shelter database:

*from script import AnimalShelter*

*# Instantiate the AnimalShelter object with credentials*

*shelter = AnimalShelter(username=”aacuser”, password=”password”)*

*# Create a new animal record*

*new\_animal = {*

*“animal\_id”: “A123456”,*

*“name”: “Ranger”,*

*“animal\_type”: “Dog”,*

*“breed”: “German Shepherd”,*

*“color”: “Brown/Black”,*

*“sex\_upon\_outcome”: “Intact Male”,*

*“age\_upon\_outcome\_in\_weeks”: 104*

*}*

*shelter.create(new\_animal)*

*# Read all dog records*

*results = shelter.read({“animal\_type”: “Dog”})*

*for r in results:*

*print(r)*

*# Update a specific record*

*shelter.update({“name”: “Ranger”}, {“color”: “Black/Tan”})*

*# Delete a test record*

*shelter.delete({“name”: “Ranger”})*

**Dashboard Testing Instructions:**

1. Run the ProjectTwoDashboard.ipynb notebook to start the dashboard server.
2. Once the link is displayed, open it in your browser.
3. Use the radio buttons on the left to apply filters.
4. Select a row from the data table to view its geolocation on the map.
5. Observe the pie chart update dynamically based on filter selection.

**Screenshots:**

Below are the screenshots showing full dashboard functionality and user interaction:

1. A screenshot of a computer

   AI-generated content may be incorrect.**Reset Filter (Default Dashboard View)**
2. A screenshot of a computer

   AI-generated content may be incorrect.**Water Rescue Filter**
3. A screenshot of a computer

   AI-generated content may be incorrect.**Mountain/Wilderness Rescue Filter**
4. A screenshot of a computer

   AI-generated content may be incorrect.**Disaster Rescue/Tracking Filter**

## Roadmap/Features (Optional)

**Current Features:**

* Full CRUD Functionality: The module supports Create, Read, Update, and Delete operations on the MongoDB animal shelter database.
* Interactive Dashboard: Includes filtering options, a responsive data table, a dynamic pie chart for breed distribution, and a geolocation map.
* Filtering for Rescue Profiles: Users can easily identify dogs suited for specific rescue operations (Water, Mountain/Wilderness, Disaster/Tracking).
* User-Friendly Interface: Built using Dash for intuitive controls, clear visuals, and responsive layout.

**Planned Features:**

* Export Filtered Data: Add an option to export filtered results as CSV or JSON.
* Advanced Filtering Options: Incorporate more filters such as color, age, and outcome type.
* Login and User Management: Secure the dashboard for internal staff with administrative features.
* Performance Optimization: Improve query speed and map responsiveness with pagination or caching.

**Known Issues:**

* Map Position Errors: Some entries in the dataset may have missing or incorrect geolocation coordinates.
* Large Dataset Load Times: Rendering the entire dataset can cause initial slowdowns on systems with limited resources.

**What Makes This Project Stand Out:**

* Combines powerful backend data handling with real-time, user-friendly visuals.
* Uses MongoDB’s flexible NoSQL structure to efficiently manage unstructured animal records.
* Built with open-source tools and modular design for easy maintenance and future expansion.
* Designed specifically for Grazioso Salvare but adaptable for other animal rescue organizations.

## Contact

Trinity Anderson

trinity.anderson@snhu.edu