# CS 340 Project Two - README

**Grazioso Salvare – Interactive Dashboard with MongoDB and Dash**

This project builds upon the CRUD module from Project One to deliver a fully interactive, web-based dashboard using Dash and Plotly. The dashboard connects directly to a MongoDB collection containing animal shelter outcome data and provides filtering, charting, and map-based interactivity. The goal is to help Grazioso Salvare identify and evaluate dogs for specialized search-and-rescue training more effectively.

## Motivation

After building a CRUD module to manage animal records, Grazioso Salvare needed a user-facing dashboard that would let them visualize key metrics, interact with the database, and filter animals by training criteria. The dashboard offers a clean interface for searching animals, analyzing outcome types, and viewing geo-location details. This aligns with their mission of identifying adoptable dogs suitable for water rescue, mountain rescue, and disaster tracking programs.

## Getting Started To run this project locally or in Apporto:

1. Clone/download the folder containing:
   * animal\_crud.py (CRUD class)
   * ProjectTwoDashboard.ipynb (dashboard notebook)
   * Grazioso Salvare Logo.png (branding)
2. Launch ProjectTwoDashboard.ipynb in Jupyter.
3. Ensure you’re authenticated using:  
   db = AnimalShelter("aacuser", "123456")

## Installation

The following tools and libraries are required:

* **Python 3.9+**: Primary programming language
* **Jupyter Notebook**: Used for testing the Python module interactively
* **PyMongo**: Python driver for MongoDB
* **MongoDB Atlas or local MongoDB instance**: Database service

To install dependencies, run:

pip install pymongo dash plotly dash-leaflet jupyter-dash

## Usage

This project includes two main files:

1. animal\_crud.py: A Python module with an AnimalShelter class
2. ProjectTwoDashboard.ipynb: A Dash-based interactive dashboard that displays real-time animal data, outcome types, and map locations.

**Features**

* Authenticated MongoDB access via PyMongo
* Interactive DataTable with filtering by animal\_type
* Pie chart of outcome types per selection
* Map showing animal location by latitude/longitude
* Real-time callbacks and updates based on user selections
* Branding with Grazioso Salvare logo

**Code Example: MongoDB Integration**

from animal\_crud import AnimalShelter

db = AnimalShelter("aacuser", "123456")

df = pd.DataFrame.from\_records(db.read({}))

if '\_id' in df.columns:

df.drop(columns=['\_id'], inplace=True)

### Tests

All testing was performed in a Jupyter Notebook. Functionality tests include:

* Connection to MongoDB via AnimalShelter class
* Filtered results by animal\_type (Dog, Cat)
* Pie chart dynamically updating by selection
* Map correctly updating location markers
* Logo successfully loaded in layout

All methods returned expected results and confirmed correct integration with MongoDB.

### Screenshots

MongoDB CSV Import Execution:

A screen shot of a computer

AI-generated content may be incorrect.

User Authentication Execution:

A screenshot of a computer screen

AI-generated content may be incorrect.

**Dashboard Filtering and Display**

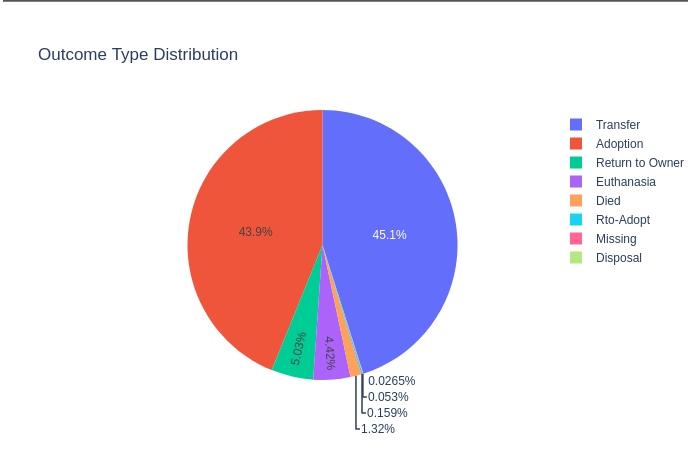
DataTable showing only Dogs:

## A screenshot of a computer AI-generated content may be incorrect.

DataTable showing only Cats:

A screenshot of a computer

AI-generated content may be incorrect.

Pie chart of outcome types:  


## Map correctly updating location markers:

Reset option:  
A screenshot of a computer

AI-generated content may be incorrect.

## Contact

Jose Medina

CS 340 – Southern New Hampshire University  
Project One – Grazioso Salvare CRUD Module