Shawn Plaisted

CS-340

Southern New Hampshire University

01 June 2025

# Project Two

## About the Project:

This project builds a fully functional MongoDB dashboard for Grazioso Salvare, an animal rescue organization. The dashboard allows users to filter, visualize, and explore animal shelter data through interactive widgets using Dash and Python. It connects to a MongoDB database populated with Austin Animal Center outcome records and allows client users to query and interact with real-time data for selecting dogs best suited for search-and-rescue training.

## Motivation:

The goal of this project was to transition from backend CRUD operations to a front-end, user-accessible dashboard. By leveraging Dash’s MVC-style framework, the application empowers users with interactive controls to explore rescue-specific dog profiles. This enhances decision-making and improves efficiency in animal selection.

## Usage:

The dashboard includes the following components:  
- User authentication input  
- Interactive filter selection (dropdown)  
- Dynamic data table  
- Geolocation map and a secondary pie chart  
  
Example workflow:  
1. User enters MongoDB credentials.  
2. User selects a rescue type (e.g., Water Rescue).  
3. Table and charts update with matching animals.  
4. Reset button clears filters to show all records.

## Installation:

Tools used:  
- Python 3.9 (Jupyter Notebook in Apporto)  
- Dash & Plotly  
- Pandas  
- PyMongo  
- MongoDB (via Apporto)  
  
To launch the dashboard:  
1. Open ProjectTwoDashboard\_FINAL.ipynb in Jupyter Notebook.  
2. Run all cells.  
3. Access the dashboard using the local port link (e.g., http://127.0.0.1:16970).  
4. Enter username 'aacuser' and password 'Shawn1993' when prompted.

## Challenges:

The biggest challenge was managing local port conflicts and visual rendering errors within Apporto. Multiple restarts and reconfigurations were required to successfully load the dashboard interface in a new browser tab. Dash’s port usage conflicted with Jupyter Notebook’s socket binding, requiring cleanup of zombie processes before retrying.  
  
Despite technical setbacks, the dashboard met all functional requirements aside from live deployment. Screenshots were not included but can be easily generated once authentication and port issues are resolved.

