Steven Blathras

Project Read me

4/20/2025

For my CS 340 final project, I created a dashboard application for a company called Grazioso Salvare. They specialize in training rescue dogs and needed a way to filter and view animals from different shelters in the Austin, Texas area. The goal of this project was to build a working dashboard that connects to a MongoDB database and helps the client quickly find dogs that are good candidates for specific types of rescue training based on things like age and breed.

The dashboard was built using Python and Dash, and I developed it in a Jupyter Notebook on Apporto. It pulls data directly from MongoDB using a CRUD module I made earlier in the course. That module, stored in a separate Python file called AnimalShelter.py, includes all the basic operations like create, read, update, and delete, and uses login credentials to connect to the database securely. In the dashboard itself, I used radio buttons to filter rescue types, a data table to show animal records, a map that pins each selected dog’s location, and a bar chart that shows breed counts based on the filter.

There are four main filters in the dashboard: Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, and a Reset option to return to the full unfiltered data. When a user clicks on one of the rescue types, the table updates to show dogs under two years old from specific breeds that match the category. The map updates to show where each selected dog is located, and the chart gives a visual of how many dogs from each breed are in the filtered set. I also included the Grazioso Salvare logo and added my name to the top of the dashboard for branding, like the assignment asked.

To build this, I started by importing the shelter data into MongoDB. I then finished writing and testing the CRUD module, which I reused from earlier assignments. After that, I worked on the layout of the dashboard and hooked up the widgets so they would respond to the filters. Once everything was working, I made a few changes to improve usability, like stretching the notebook layout so I could see more on the screen, zooming out the browser, and collapsing unnecessary code cells to make room for everything. This was especially helpful when taking screenshots of the full dashboard view.

One issue I ran into was needing to update my AnimalShelter class to accept a username and password as parameters. I had it hardcoded before, but that wouldn’t work for the final version of the project since it needed to handle login securely. I also had to adjust my filtering logic a bit because some breeds didn’t always match exactly with the ones in the client’s rescue type table. I fixed this by making sure the breed filters matched the data in the database correctly.

I’ve included screenshots in this document to show how the dashboard works. You’ll see the default view, and then separate screenshots for Water Rescue, Mountain or Wilderness Rescue, and Disaster or Individual Tracking. There’s also one showing the Reset filter in use again. All of the screenshots show that the table, map, and chart are responding correctly to the filters, and my name and the logo are visible on the dashboard.

For submission, I’ve included this README with the screenshots, my ProjectTwoDashboard.ipynb notebook file, and the AnimalShelter.py CRUD module. This wraps up my Project Two submission for CS 340.

SCREENSHOTS:A screenshot of a project

AI-generated content may be incorrect.A map with a location pin

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

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A map with a blue pin

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

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

AI-generated content may be incorrect.A map with a location pin

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

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

AI-generated content may be incorrect.A map with a location pin

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

AI-generated content may be incorrect.