# CS 340 README Template

**About the Project/Project Title**

This project is the second project in the class. This project is part of a full-stack development initiative for Grazioso Salvare. The goal of this project is to build a user-friendly dashboard that enables the company to identify and categorize dogs from regional shelters that are suitable for search and rescue training. The dashboard is created using Python and MongoDB providing full CRUD functionality along with dynamic data visualization and filtering capabilities. It is designed to reduce user errors, minimize training time, and provide intuitive access to shelter data.

**Motivation**

The motivation behind this project is to help Grazioso Salvare filter and visualize shelter data to identify dogs that meet specific criteria for training.

**Getting Started**

Someone is able to setup and run my module by completing these simple steps:

* Install MongoDB and ensure it’s running locally
* Import and load database named aac and a collection named animals using  command: mongoimport --type=csv --headerline --db aac --collection outcomes --drop ./aac\_shelter\_outcomes.csv
* Create MongoDB user (I used aacuser and AAC\_ADMIN)
* Install required python packages (pip install pymongo dash pandas plotly)
* Run the dashboard from ProjectTwoDashboard.ipynb

**Installation**

I used a few tools to complete this project. Tools help with simplicity, flexibility, and real-world  relevance in backend development. These tools include:

* Python 3.10 (which is the primary programming language)
* MongoDB (noSQL database for storing animal records)
* Pymongo (Python library that interacts with MongoDB)
* Jupiter Lab (For testing and demonstrating the code)

**Usage**

This project delivers a fully interactive web-based dashboard built with python and dash, enabling users at Grazioso to visualize and filter animal shelter data stored in a MongoDB database. The dashboard is designed to support decision-making by identifying dogs suitable for various types of search and rescue training. This can be seen with the code below which shows filtering options to help users find animals that meet their requirements:   
app.layout = html.Div([

html.Center(html.B(html.H1("SNHU CS-340 Dashboard"))),

html.Center(html.B(html.H1("Project 2 Dashboard: Chris Clark"))),

html.Hr(),

logo\_img,

dcc.RadioItems(

id="filter-type",

options=[

{"label": "Water Rescue", "value": "water"},

{"label": "Mountain/Wilderness Rescue", "value": "mount"},

{"label": "Disaster Rescue and Individual Tracking", "value": "disaster"},

{"label": "Reset", "value": "reset"},

],

value="reset",

style={"textAlign": "center"}

**Code Example**

Below is my animalshelter.py code:   
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.  
this ties into my ProjectTwoDashboard.ipynb code that allows user in the company to filter, view visual tables, graphs, and maps within a personalized dashboard.

A screenshot of a computer

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 screenshot of a computer screen

AI-generated content may be incorrect.

My test script lauches a window within the codio platform that displays my personalized dashboard with a table, graph, and map with the company logo.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a map

AI-generated content may be incorrect.

**Contact**

Your name: Christopher Clark