x

**Module 7:**

**Read Me**

**Harley J. Reimels**

**CS 340 – Client/Server Development**

**Dr. J. Webb**

**August 18th, 2024**

**Module 7: Read Me**

**Project Overview**

**This project involves the development of a client/server application for Grazioso Salvare, an international rescue-animal training company. The application is designed to interface with a MongoDB database and provide a user-friendly web dashboard for interacting with the data. The dashboard will allow users to filter, view, and visualize data related to dogs identified as candidates for search-and-rescue training.**

**Functionality**

**This is an interactive dashboard where we see a data table displaying unfiltered animal data from the Austin Animal Center Outcomes dataset.**

**Original State**

**A screenshot of a computer

Description automatically generated**

**A map of the world

Description automatically generated**

**Water Rescue**

**Here we see filtered showing the dogs that meet the requirements for water rescue.**

**A screenshot of a computer

Description automatically generated**

**A map of the world

Description automatically generated**

**Wilderness / Mountain Rescue**

**Here we see filtered showing the dogs that meet the requirements for wilderness and mountain rescue.**

**A screenshot of a computer

Description automatically generated**

**A map of the world

Description automatically generated**

**Disaster Rescue**

**Here we see filtered showing the dogs that meet the requirements for disaster rescue.**

**A screenshot of a computer

Description automatically generated**

**A map of a road

Description automatically generated**

**Tools and Technologies**

**MongoDB**

**Chosen for its flexibility in handling unstructured data and its robust support for CRUD operations through Python. MongoDB's scalability and compatibility with Python made it an ideal choice for the model component of this application.**

**Python**

**The primary programming language used for developing the application, including the CRUD operations and dashboard components.**

**Dash**

**A Python framework used to build the web application dashboard. Dash was selected for its simplicity in creating interactive, web-based data visualizations and its seamless integration with Plotly for charting and data visualization.**

**Plotly**

**Used for creating dynamic, interactive charts that respond to the dashboard’s filters.**

**Project Setup and Execution**

**Ensure you have the following installed:**

* **Python 3.x**
* **MongoDB**
* **Required Python packages: Dash, Plotly, pymongo, pandas**

**Now import the crud module named AnimalShleterClass, and pass in the parameters; Username, Password, Host, port, Database, Collection**

**A screenshot of a computer

Description automatically generated**

**From here you can build your own dash layout with any widgets you choose. By using PyMongo you are able to create, read, update, and delete anything in the collection.**

**Module 7: Project Two**

References

Dash [Documentation](https://dash.plotly.com/)

[MongoDB](https://www.mongodb.com/docs/) Documentation

Plotly [Documentation](https://plotly.com/python/)