**README**

**Project Overview**

The goal of this project is to create a web application that visually represents data using the Dash Framework while allowing users to interact with the database with a user-friendly interface.

**Required Functionality**

1. Connect to the MongoDB database and retrieve the data.
2. Display the data via an interactive dashboard.
3. Display the data in multiple formats such as pie charts, geological locations, and tables.
4. Give users the ability to filter data.
5. Deploy all of the above to a web server.

**Default Dashboard**

A red line drawing of a dog

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

**Water Rescue**

A screenshot of a computer

Description automatically generated

**Mountain or Wilderness Rescue**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A picture containing diagram, map, circle, text

Description automatically generated**

**Disaster or Individual Tracking**

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing map, diagram, text, screenshot

Description automatically generated

**Tools Used**

1. MongoDB: MongoDB makes it easy to store and manipulate structured and unstructured data and easily integrates with Python making it the obvious choice for querying our data.
2. Dash: Dash is an open-source Python framework that is primarily used to build analytical web applications. It excels in environments where data scientists are unfamiliar with web development because it is to work with.
3. HTML/CSS: HTML and CSS are the standard languages used for creating and styling web-based applications.
4. PyMongo: Pymongo provides useful tools to communicate with a MongoDB server as well as CRUD functionality.

**Steps Taken to Complete Project**

1. Planning: Read through the requirements to find the functionality and desired outcome of the project.
2. Data Import: Imported the Austin Animal Center data into the MongoDB database.
3. Dashboard Development: Developed a web-based dashboard using the Dash framework and HTML and CSS to create the front end.
4. Testing: Regularly and routinely tested the code during development to ensure that each piece of added functionality didn’t interfere with the working code.
5. README: Created a README file to document the steps taken to achieve a working database so otherwise can follow along to create their own databases using these files.

**Challenges**

I struggled with creating the buttons that give users the ability to filter data within the database. I overcame the issue by researching and reading through MongoDB documentation.

**Resources**

<https://www.mongodb.com/why-use-mongodb>

<https://www.tutorialspoint.com/python_web_development_libraries/python_web_development_libraries_dash_framework.htm>

<https://www.w3.org/standards/webdesign/htmlcss>