**Project 2 Readme**

Zachary Mohler

Southern New Hampshire University

CS340

David Ostrowski

August 15, 2022

**Project Overview**

The purpose of this project was to create an interactive web application that takes an existing dataset and displays the data in an organized way. The client specifically requested data filters to be able to quickly identify data entries that met specific multi-tiered requirements.

The application uses MongoDB to host the database, a python script to translate between the database and client-side levels, and the Dash framework to draw all the information on an html page that the users can interact with.

**Database Level**

**About the Database Level**

The “animal\_shelter.py” script serves as the functional level to a database driven web application. Its functions serve to translate between python and mongodb query language. The purpose is to provide the middle-level grunt work for implementations of CRUD (Create, Read, Update, Delete) operations in mongodb. This simplifys the process of querying to a more front-end structure.

## Utilizing the Script to Perform Mongo Operations at the Database Level

To use the script, all you need to do is include the python file in your project directory. If the file is in a different location on your machine, an import will be necessary. As the script’s purpose is to implement CRUD operations using mongodb having mongo installed on your machine is necessary, as well as pymongo which is a common mongo to python driver. The script utilizes the mongoClient class from the pymongo module to establish a connection to the database you wish to manipulate. It also implements the ObjectId class which is a type used for storing mongodb id data.

To start, simply create an instance of the class, passing in the database’s log-in information: (**NOTE**: the user that is logged in must have the proper credentials for the operations to work)

databaseHandler = AnimalShelter(username, password)

Then you can call one of the handler’s functions, (read, create, update, and delete) passing in Python dictionaries as arguments.

### Tests/Screenshots

Here is an example of a test program and its output used on a live database:

A picture containing text

Description automatically generatedText

Description automatically generated

**Client Level**

The project’s implementation of the Python script for the client-side uses the dash framework to communicate the query results to an interactive html page to aid the user in accomplishing their goals.

**Layout**

**Chart

Description automatically generated with medium confidence**

The layout of the application is simple. A header is located at the top of the page, with logo (not visible in screenshot) and title information. Below that is the data-table which shows the information that is acquired from the existing dataset. This data-table can be manipulated by the user using any of the field headers to sort by value :

Graphical user interface

Description automatically generated with medium confidence

Or to search for specific elements in a field header :

A picture containing map

Description automatically generated

As you can see, the layout also contains a pie chart to show the distribution of results for each breed of animal in the resultant data-table. The layout also contains a geolocation chart which shows the location of the animal at the top of the list, or the selected animal using the checkboxes to the right of the page :

Graphical user interface, application

Description automatically generated with medium confidence

Another important feature is the filters above the data-table. The project purpose is specifically to find dogs that fit specific requirements, so the use of these “quick filters” is very important for the client’s purposes. Here are examples of the results when filters are enabled :

Chart

Description automatically generatedChart

Description automatically generatedGraphical user interface, chart

Description automatically generated

**Technical**

As mentioned, this is accomplished using the dash framework. The frame is ultimately a way to dynamically change html elements—so we can use our Python script to pass data into dash to then be displayed to the user. For example, the filter buttons are simply performing a specific query and passing that data on to dash :

Scatter chart

Description automatically generated with low confidence

## Contact

Zachary Mohler