Shayna Mitchell

August 14, 2024

Project Two README

CS 340 Client Server Development

## About the Project/Project Title

This project works with manipulating data. It promotes interaction with databases and connects the user with their chosen database. It contains a portable Python module that allows for create, read, update, and delete functionality. The database used, in this case, is AAC or collection animals. The user can manipulate the data in this database, so for example: create a new animal, read about an animal, update an existing animal, and delete an animal. It incorporates a dashboard for the user to interact with and filter the database AAC. Specific data can be found and even visualized through a map and graph. This way, Grazioso Salvare are able to find specific dogs that are a certain breed, gender, and age. They will be able to find the correct animals they can train for specialized jobs like Rescue or Tracking. The user can see on a map where these animals are located so that they can reach out to the shelter to see if they would be good candidates for training.

## Motivation

This project allows the user to exercise their knowledge of databases and CRUD; or create, read, update, and delete functionality. It combines the database AAC with a user friendly dashboard so that the right animals can be found for rescue training.

## Getting Started

This project needs a couple steps to get it running. You will need to open up terminal and open up MongoDB. Here in the terminal, open up Jupyter Notebook to run commands. Here you can look for the data needed or manipulate in any way withing the CRUD functionality.

## Installation

The tools for this project can be downloaded and used easily. Search for these websites online and download the appropriate links according to user specific operating system. Examples given in this README file are from a Linux operating system running Apporto. MongoDB is a database tool that will be needed in this project, specifically. Another tool that would be needed is the Python development language. The Dash framework will be needed for this project to be able to use the dashboard feature. It integrates with plotly so that visualizations can be used on the dashboard. Plotly and Dash are easily accessible just like the other features of this project. Using the PyMongo driver allows for the easy use with MongoDB. Everything works very well together. Dash allows the developer to easily focus on building app logic using Python.

Necessary python packages:

‘pip install dash dash\_leaflet plotly pandas pymongo jupyter-dash’

You will need to customize your connection with your own username, password, host, port, database name, and collection name.

Your Jupyter Server will need to ne configured to run Dash applications.

<https://www.mongodb.com/docs/>

<https://github.com/plotly/jupyter-dash>

<https://dash.plotly.com>

<https://www.python.org/downloads/>

<https://www.mongodb.com/try/download/community>

## Usage

This module inserts a document into the user’s chosen database. If it works, the user will see “True”, and if it doesn’t, “False” is returned. This allows the user to know whether the module is responding properly. User functionality involves the use of queries and the method find(). Here you can look for data and manipulate data from your chosen database. Taking it further, a user can use the dashboard to find specific animals. This is achieved through the use of filtering data, a data table, a pie chart, and a geolocation map.

### Tests

Write tests to see if your data is being queried and manipulated properly. These can be coded in Jupyter Notebook to display results according to the information needing tested. When using this example database, the functionality of CR has been tested in the examples provided. The dashboard will need to be worked on and tested before it looks exactly as needed. Changing various things in your dashboard in small amounts will give you the right look or functionality for the user. For example, you can change filtering through the use of drop downs or radio buttons. Change the dashboard how you like until it looks correct.

### Screenshots/Examples:A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generatedA screenshot of a map Description automatically generated

## A screenshot of a graph Description automatically generatedA screenshot of a dashboard Description automatically generatedA screenshot of a dashboard Description automatically generatedA screenshot of a dashboard Description automatically generatedA screenshot of a dashboard Description automatically generated

MongoDB was chosen for this project’s database because it is NoSQL and it works so well with Python. Because it is NoSQL, the developer is able to make changes frequently for testing without experiencing downtime.

Make sure all code is reviewed thoroughly to prevent unnecessary time waste. Filtering and graphic visualizations were a cumbersome task because of typos thorough a lot of this project initially. The multiple callback functions were hard to synchronize in this project. Incremental development was key here. Otherwise, this project is quite doable as long as you follow all of the steps.

## Contact

Your name: Shayna Mitchell