Justin Phillips

10/20/2024

CS 340 Client/Server Development

Reuben Wilson

Southern New Hampshire University

GRAZIOSO-SALVARE DATABASE PROJECT READ-ME

About the Project

This project involves a MongoDB CRUD Python module and an interface built using PyjupiterDash, Matplotlib, Plotly Express, and Pandas. The objective is to allow users to navigate a database of pets characterized by qualities such as breed, sex, and age to determine the best candidates for emergency service operations. Our database is empowered by a non-profit agency partnered with us and affiliated with five animal shelters in the Austin, Texas area.

Motivation

Grazioso-Salvare is motivated by finding the best candidates for search-and-rescue training. These dogs, with the right training, could help to find and save the lives of humans and other animals.

Getting Started

First, install Python 3.x from python.org.

Next, go to Jupyter.org and install their notebook editor. This will allow you to take advantage of JupyterDash and its HTML hooks, which are perfect for building a web-app interface for your application.

Open your command prompt and run the following commands:

*```pip install urllib3```* - This HTTP library has a string parser that allows password handling with special characters. It also can provide networking functionality with powerful web API interactions. This is a hard dependency for our CRUDoperations module.

*```pip install pymongo```* - This is essential for working with MongoDB databases through Python and a hard dependency on our CRUDoperations module.

*```pip install jupyter\_dash```* - This integrates Dash into the Jupyter Notebook environment, which is a powerful web-based application that allows you to troubleshoot your webapps easily.

*```pip install plotly```* - This provides you with all sorts of charts and graphs you can use to visualize your data.

*```pip install numpy```* - Pretty essential for any data analysis. Numpy handles computations and statistics well and can be used for our data visualization.

*```pip install pandas```* - Pandas is also pretty essential for data science. It allows data manipulation and analysis, as well as providing the dataframe structure we will be using.

*```pip install matplotlib```* - Another plotting library.

You’ll next want to follow the instructions for CRUDfunctionality.

To get started, simply download the latest major version .py file and drop it into your working directory.

Alter the values in the \_\_init\_\_(self) function for the following variables (labeled under connection variables):

```

USER=urllib.parse.quote\_plus( “YOUR MONGODB USERNAME” )

PASS=urllib.parse.quote\_plus( “YOUR MONGODB PASSWORD” )

PORT= YOUR MONGODB PORT

HOST=urllib.parse.quote\_plus( “YOUR MONGODB HOST” )

DB = “YOUR MONGODB DATABASE”

COL = “YOUR DATABASE’S COLLECTION”

```

This is all the configuration needed. Ensure whichever account you use has the readWrite role for the database you intend to interface with.

Call for it to be imported with the following command:

```import CRUDfunctionality as CRUD```

When you want to instantiate your MongoDB interface, enter the following command where yourInstance is any variable for your instance you choose:

```yourInstance = CRUD.CRUDoperations()```

yourInstance can now have the following commands called: create(your list of parameters for a new entry) or read(your list of parameters for a query).

Create returns a true value if the creation of the new entry completes properly. If not, it will return false, and an exception will be thrown.

Read returns the entire results of your query, and if the query fails it will also cast an exception.

Be aware of case sensitivity as you interact with MongoDB. I had used capitals in place of lowercase for my database at first, which led to issues with readWrite role permissions not being assigned to “aac” – because “aac” did not exist, but “AAC” did.

You will need MongoDB, Python 3.x, a Python editor (I recommend PyCharm), and the following libraries:

Urllib to take advantage of the quote\_plus parser; This allows Python to share your complex passwords with pymongo

Pymongo to allow interaction with MongoDB through Python.

As you install MongoDB, follow its documentation to configure it.

Now, your environment is set up. Create a new notebook and add the following lines:

*```From jupyter\_dash import JupyterDash*

*import dash\_leaflet as dl*

*from dash import dcc*

*from dash import html*

*import plotly.express as px*

*from dash import dash\_table*

*from dash.dependencies import input, output, state*

*import base64*

*import os*

*import numpy as np*

*import pandas as pd*

*import matplotlib.pyplot as plt*

*from CRUDfunctionality import CRUDoperations```*

You will be using CRUDoperations to interface with your MongoDB database.

```db = CRUDoperations(username, password)``` is a good command to allow easy access to the CRUDoperations functionality. ‘db’ will act as your database cursor, enabling you to create new items, read existing items, update items, and delete items exactly as you need to.

```df = pd.DataFrame.from\_records(db.read({}))``` will give you a Pandas dataframe containing all your table’s information, which will allow you to interface with visualization software in Plotly Express.

```df.drop(columns=[‘\_id’], inplace=True)``` will drop the MongoDB v5+ “\_id” column, which could cause issues with the dataframe format. ‘inplace’ returns a new dataframe free of \_id.

Usage

A screenshot of a computer

Description automatically generatedYou’re welcome to browse the source code. If you’re using the application, be aware that it defaults to a view of all prospect canine candidates. You can interact with the drop down menu to see our three pre-defined filters, water rescue dogs, mountain and wilderness rescue dogs, and disaster/individual tracking rescue dogs.

Water rescue dogs will display Newfoundlands, lab mixes, and other dogs well suited for the water. Mountain dogs will show German Shephard and lab mixes, huskies, and other dogs well suited for those environments. Disaster dogs will show bloodhounds, rottweilers, and other clever dogs with sharp noses.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Here is a demonstration of each view:

All Dogs (Default view)

A screenshot of a computer screen

Description automatically generated

Disaster Rescue Dogs

A screenshot of a computer screen

Description automatically generated

Mountain Rescue Dogs

A screenshot of a computer screen

Description automatically generated

Water Rescue Dogs

A screenshot of a computer screen

Description automatically generated