

Lauren Lindhurst

Southern New Hampshire University

CS-499

5-2: Milestone Four: Enhancement Three: Databases

February 4, 2023

Purpose

This paper is a narrative that accompanies the artifact for the Databases category and details why this artifact was included in my ePortfolio. It reflects on the approach used to create the artifact and focuses on the learning process that occurred thorough out the creation of it.

Prompt

The artifact selected for the databases category is the **Grazioso Salvare Animal Shelter Search Dashboard**. This web application was designed for my computer science course, CS-340: Client/Server Development. This web application is used with an existing database from animal shelters to categorize different animals (cats and dogs) by various and multiple factors. The application is developed in Python with Dash framework and with MongoDB, using PyMongo. This project best displays my knowledge with databases and how to utilize them. The project was originally developed on the Apporto platform, a virtual Linux system for student access. I enhanced this project by recreating it and using it on my own local machine running Windows Operating System.

The artifact was selected because it involves the development of a multi-tiered web application that utilizes the PyMongo driver and use of Python's dash framework to integrate with a MongoDB database containing the data for the animal shelter. The web application's functionality involves an imported ".csv." data file of existing animals in the shelter and that data file is manipulated through the use of python libraries, dash framework, python source code, and my created CRUD module. The application can be run in Jupyter Notebook or a CLI running alongside a web browser to display the application.

5-2: Milestone Four: Enhancement Three: Databases



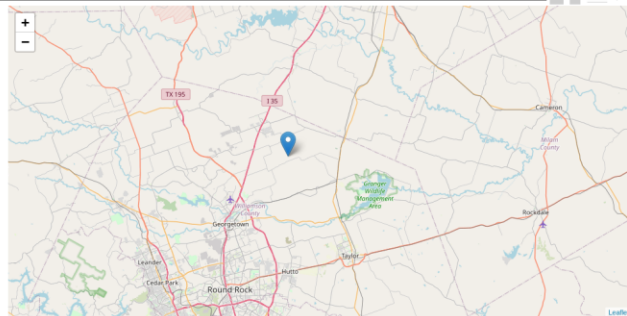
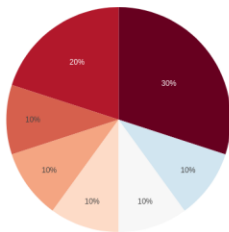
**GRAZIOSO
SALVARE**

Lauren Lindhurst - SNHU CS-340

Grazioso Salvare Animal Shelter Web Application Dashboard

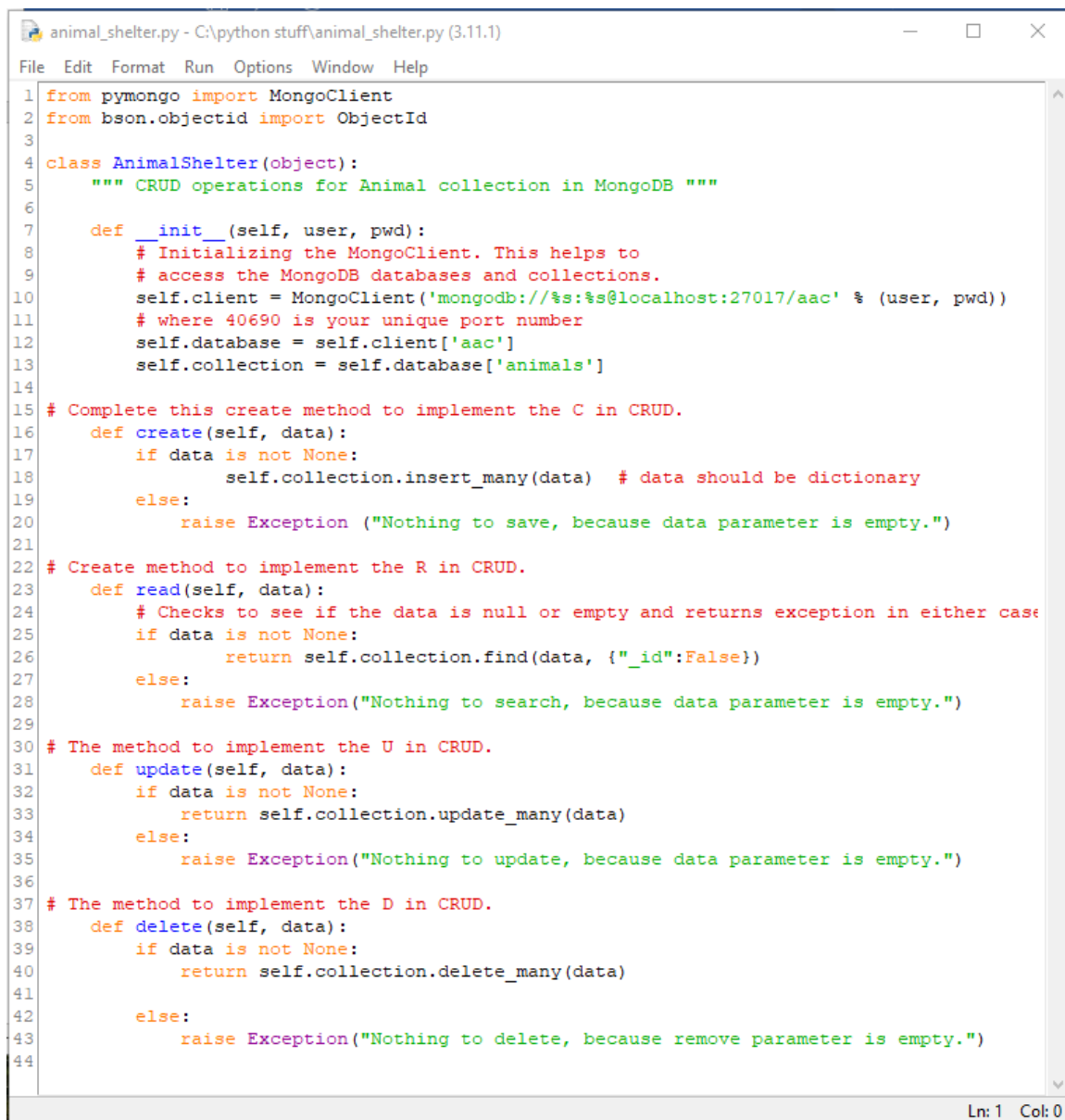
Water Rescue Mountain/Wilderness Rescue Disaster Rescue/Individual Tracking Reset- returns to unfiltered state

	1	age_upon_outcome	animal_id	animal_type		breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
11	3	2 years	A716338	Dog		Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7595748121648	-97.5523753887133	118.111408730159
6	5	5 years	A696004	Dog		Cardigan Welsh Corgi Mix	Sable/White	2010-01-27	2015-01-28 10:39:00	2015-01-28T10:39:00	Lucy	Rabies Risk	Euthanasia	Spayed Female	30.6737365854231	-97.7678715294657	261.063392857143
7	2	2 years	A673830	Dog		Pit Bull Mix	Black/White	2012-03-03	2014-03-19 15:15:00	2014-03-19T15:15:00	*Seth	Aggressive	Euthanasia	Neutered Male	30.2954256583441	-97.3136642110436	106.662282389952
5	2	2 years	A691584	Dog		Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7184815618433	-97.562297435286	133.653571428571
8	1	1 year	A736551	Dog		Labrador Retriever/Australian Cattle Dog	Black	2015-10-12	2016-11-27 18:08:00	2016-11-27T18:08:00	*Mia		Adoption	Spayed Female	30.4443212820182	-97.7326988338793	58.9642857142857
9	3	3 years	A720214	Dog		Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3870648199411	-97.3684339731375	157.504067468317
11	1	1 year	A721199	Dog		Dachshund Wirehair Mix	Tan/White	2015-02-23	2016-02-27 17:49:00	2016-02-27T17:49:00	Belle		Adoption	Spayed Female	30.7290272761146	-97.375328216134	52.8201373015873
12	1	1 year	A664843	Dog		Pit Bull Mix	Brown/White	2013-06-09	2014-08-18 17:24:00	2014-08-18T17:24:00	Sherlock	Partner	Transfer	Neutered Male	30.45155493997366	-97.474184510925	62.2464285714286
14	2	2 years	A742287	Dog		Boxer/Bulldog/Bulldog	Brown/Brindle/White	2015-01-18	2017-02-11 12:30:00	2017-02-11T12:30:00	*Kawhi		Adoption	Neutered Male	30.4551148649996	-97.3887788473978	107.931547619048
15	3	3 years	A712638	Dog		Pit Bull Mix	Red/White	2012-09-26	2016-07-18 17:52:00	2016-07-18T17:52:00	Marcus	Partner	Transfer	Neutered Male	30.5798299207817	-97.5588487936533	198.820634920635



The artifact involves validating input data and data structure when accessing database records. The source code is clearly and effectively documented, following coding best practices. It is easy to read and maintain, following the Python standards. The created CRUD module is imported by other Python scripts showing its reusability.

5-2: Milestone Four: Enhancement Three: Databases



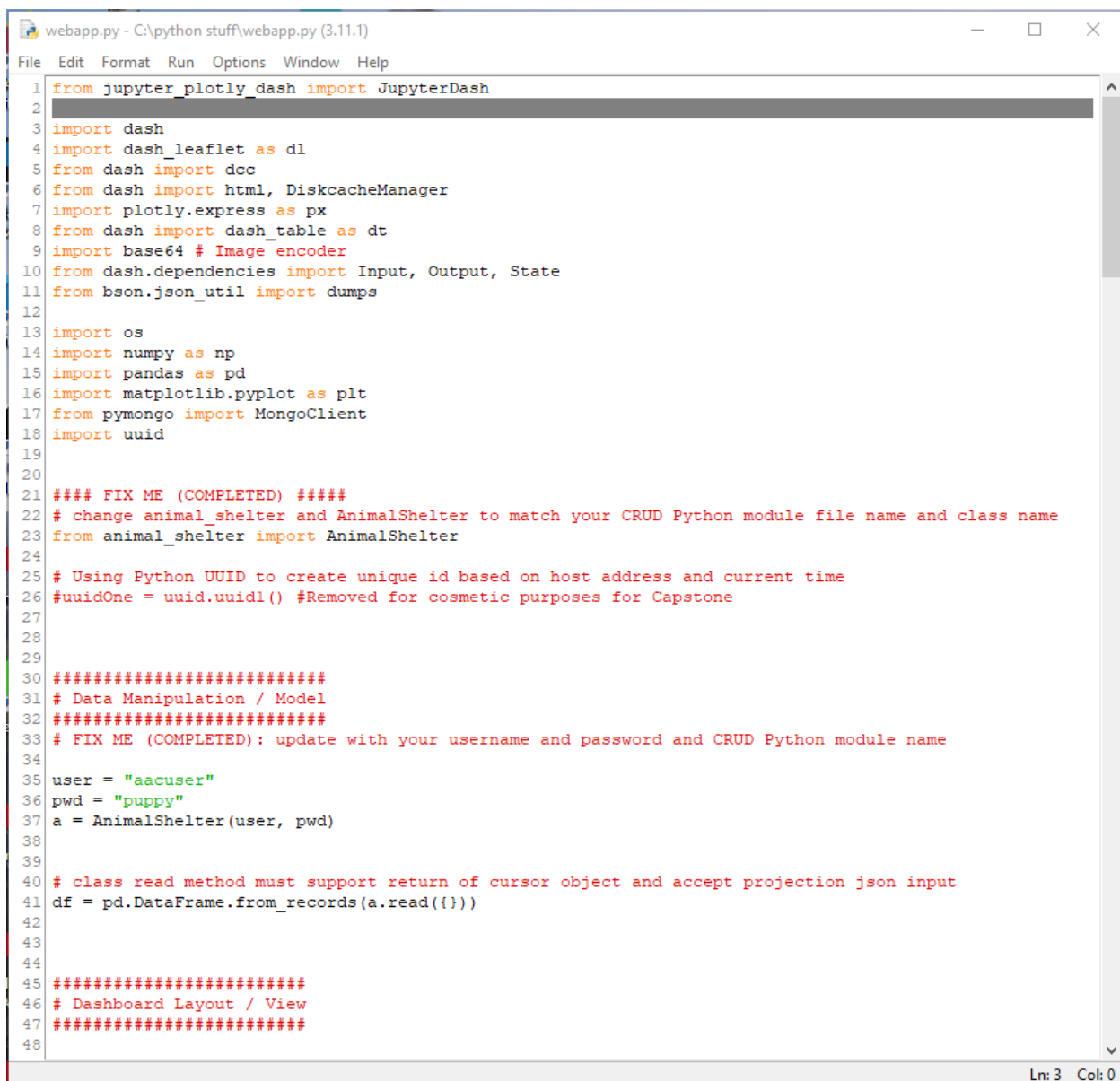
```
1 from pymongo import MongoClient
2 from bson.objectid import ObjectId
3
4 class AnimalShelter(object):
5     """ CRUD operations for Animal collection in MongoDB """
6
7     def __init__(self, user, pwd):
8         # Initializing the MongoClient. This helps to
9         # access the MongoDB databases and collections.
10        self.client = MongoClient('mongodb://%s:%s@localhost:27017/aac' % (user, pwd))
11        # where 40690 is your unique port number
12        self.database = self.client['aac']
13        self.collection = self.database['animals']
14
15    # Complete this create method to implement the C in CRUD.
16    def create(self, data):
17        if data is not None:
18            self.collection.insert_many(data) # data should be dictionary
19        else:
20            raise Exception("Nothing to save, because data parameter is empty.")
21
22    # Create method to implement the R in CRUD.
23    def read(self, data):
24        # Checks to see if the data is null or empty and returns exception in either case
25        if data is not None:
26            return self.collection.find(data, {"_id": False})
27        else:
28            raise Exception("Nothing to search, because data parameter is empty.")
29
30    # The method to implement the U in CRUD.
31    def update(self, data):
32        if data is not None:
33            return self.collection.update_many(data)
34        else:
35            raise Exception("Nothing to update, because data parameter is empty.")
36
37    # The method to implement the D in CRUD.
38    def delete(self, data):
39        if data is not None:
40            return self.collection.delete_many(data)
41        else:
42            raise Exception("Nothing to delete, because remove parameter is empty.")
43
44
```

The implemented data structure solves the given problem using algorithmic principles and computer science practices and standards appropriate to its solution while management trade-offs involved in design choices.

The artifacts enhancements were successful. It aimed to recreate the web application in Windows Operating System environment. The process was challenging because the application was initially developed and ran on a Linux Environment. This meant I had to start from scratch,

5-2: Milestone Four: Enhancement Three: Databases

installing and setting up each driver and tool. I did struggle with getting the pie chart and map to display due to a callback issue, but was able to solve my issue and get the application running smoothly. The setup process and the code updates because of updated drivers and software show my ability to use well- founded and innovative techniques, skills, and tools in computing practices to implement computer solutions that deliver value and accomplish industry-specific goals.



```
webapp.py - C:\python stuff\webapp.py (3.11.1)
File Edit Format Run Options Window Help
1 from jupyter_plotly_dash import JupyterDash
2
3 import dash
4 import dash_leaflet as dl
5 from dash import dcc
6 from dash import html, DiskcacheManager
7 import plotly.express as px
8 from dash import dash_table as dt
9 import base64 # Image encoder
10 from dash.dependencies import Input, Output, State
11 from bson.json_util import dumps
12
13 import os
14 import numpy as np
15 import pandas as pd
16 import matplotlib.pyplot as plt
17 from pymongo import MongoClient
18 import uuid
19
20
21 #### FIX ME (COMPLETED) ####
22 # change animal_shelter and AnimalShelter to match your CRUD Python module file name and class name
23 from animal_shelter import AnimalShelter
24
25 # Using Python UUID to create unique id based on host address and current time
26 #uuidOne = uuid.uuid1() #Removed for cosmetic purposes for Capstone
27
28
29
30 #####
31 # Data Manipulation / Model
32 #####
33 # FIX ME (COMPLETED): update with your username and password and CRUD Python module name
34
35 user = "aacuser"
36 pwd = "puppy"
37 a = AnimalShelter(user, pwd)
38
39
40 # class read method must support return of cursor object and accept projection json input
41 df = pd.DataFrame.from_records(a.read({}))
42
43
44
45 #####
46 # Dashboard Layout / View
47 #####
48
```

Ln: 3 Col: 0

5-2: Milestone Four: Enhancement Three: Databases

References

Southern New Hampshire University. (2023). CS 499 Milestone Four Guidelines and Rubric Enhancement Three: Databases. SNHU.