Juan Cervantes Ortiz

CS-499

February 4, 2025

Milestone 4

The project I have chosen for the final category of Databases is my CS-340 Animal Database project. This project consists of various components. First there is an animal database that holds the animal information such as name, breed, and age. Then there is a user dashboard where the data is shown on a data table, along with a pie chart broken down by breed, and a map that shows the animals' location. A python module is also created, it is responsible for connecting to the database as well as creating, reading, update, and deleting items from the database. Through the development of this process a testing class was used to ensure proper functionality of the python module. The final portion of this project was a README file that is created for users to see the system functionality and steps for installing the system on their local Windows machine.

This project allows me to show various skills so there are many reasons why I chose to include this project into my ePortfolio. The project shows my understanding of the Python programming language and its integration of MongoDB through my Python Driver Class. It also demonstrates my ability to use VS code in conjunction with Python and Jupiter Notebook to create, run, test, and debug my system. Lastly through the README file I can show my ability to create proper system documentation for technical and non-technical users and stakeholders.

The enhancements for this project all pertain to its migration to a different system. In the case of this project the current database and information is ported over to work on Windows. To begin the project I first created the AAC database. This was done using MongoDB compass. The animal.csv file was uploaded to MongoDB using the MongoDB compass app. This made it very easy to create the database since it didn't need any sort of command line arguments. Moving to the python driver class, the original Linux class was rather simple to add using VS code. Some things that needed to change were the database connection settings since it has a different port number. The dashboard works as expected but requires the proper libraries to be installed to first even function. The dashboard notebook had to be updated to ensure that its connections to the driver class were correct.

Part of ensuring that the system functions properly, I created a testing file. This file tests the various features of the python class and ensures proper connection to the Mongo database. The first few tests look at the diver's class ability to create data correctly. It tests it can add a 'Test' animal, try to add an animal with no animal data passed in. The first test returns true and the second throws an error. There are three tests that pertain to its ability to read data. One is to find our 'Test' animal, the second tries to search for something that is not in the database, and it returns a blank list, the last one tries to read the data when nothing is passed in. The system then has 3 important tests for the updating method. There is a test that updates the Test animals breed from test to Cat, there is then a read test that shows the Test animal to confirm the breed was updated properly. The next test updates a value that doesn't exist, and it returns updatedExisting as False since the item doesn't exist. The last update tests the system trying to update a document that has no information passed in. The last tests pertain to the delete method. There is a test that deletes our Tester animal, and a read test to ensure the item was deleted properly. A test to delete a document that doesn't exist return 'n' as 0 since it can delete something that isn't in the system.

Lastly a test to delete a document when no information is passed in performed, and it returns an exception.

The outcomes that pertain to the enhancement of this project are 1, 2, 3, and 4. It aligns with the first outcome because it is a system that would ideally be used by lots of people at once. This means that the system needs to allow a way for lots of people to interact with the database in a way that only runs proper methods. The second outcome is to create professional quality communications for specific audiences, and the README file addresses this outcome directly. Outcomes three and four refer to designing solutions withing the languages limitations and using the various tools to accomplish the industry-specific goal. In this case the program uses different systems to be created and ensuring that they all work as expected given their limitations is important for the system's functions.

While working on this enhancement there were things that I struggled with but other things that I excelled at. Something that I felt like I excelled in was creating a good README file that gives potential users clear instructions on how to properly use and install the system. The biggest struggle was being able to ensure that the proper dependencies were installed on my system. The dashboard used various imports, most of which are additional libraries, not a part of the standard Python library. This made it difficult to run the tests because it would produce an error if the wrong libraries were installed. Another thing that I struggled with was creating the database. The original project used the Linux command line to create the database. With Windows I used MongoDB Compass to create the database. It was easier to create the database with the use of the application, and then I had to add the authentication through the mongo shell directly available through MongoDB Compass. Something that I was unable to do was to get the authenticated user loaded into the admin database. I was able to create the authorized user in the admin database, but I couldn’t get the driver class to connect to the animal databases when the username and password were passed in. This is not that big of a security issue right now, since the dashboard can only read the data and not write to it.