**Milestone 3 Enhancements**

**Artifact Description**

The artifact selected for this enhancement is from my animal shelter dashboard project originally created in CS 340, which focuses on data processing and retrieval. This project, initially created to build a dashboard for managing animal shelter data, involved connecting to a MongoDB database to retrieve and display information about shelter animals. The original artifacts were animal\_shelter.py and ProjectTwoDashboard.ipynb, but now consist of animal\_shelter.py and multiple files derived from the ProjectTwoDashboard.ipynb file in the first set of enhancements.

**Justification for Inclusion**

I selected this artifact for my ePortfolio because it showcases my understanding and skills in algorithms and data structures, specifically within the context of database interactions and optimization. By choosing this artifact, my goal was to demonstrate my ability in identifying performance bottlenecks and optimizing data retrieval using indexing and query optimization techniques. These optimizations reduced data load times and enhanced the application’s efficiency, showcasing my ability to apply algorithms and data structure principles in practical, impactful ways.

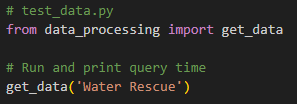
The specific enhancements include:

1. The addition of indexes on frequently queried fields in MongoDB to improve query efficiency.

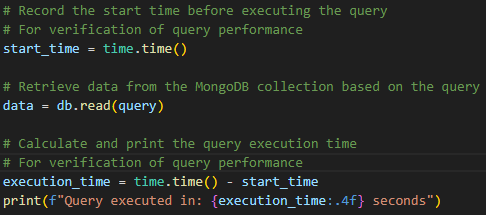


1. Testing and measuring query execution times before and after indexing using test.py, which I created to quantify the improvements.

First, I created a test script(test.py) to show query execution time before the indexes were created to get a baseline:



Then with a few lines of code created a print statement to display query execution time when the script ran:



And this was the result:



I then created the indexes and ran the script again multiple times. The query execution time was less than before the indexes were created:



The query execution time was reduced by almost half.

1. Adding additional records to the database to simulate real-world, high-volume data, allowing for a more realistic performance assessment.

These enhancements required an understanding of database indexing and the ability to analyze algorithm efficiency, making it an ideal artifact to represent my skills in this area.

**Meeting Course Outcomes**

Through this enhancement, I met several course outcomes related to algorithms and data structures. I planned to address Outcome 3, which involves designing and evaluating computing solutions using algorithmic principles, and Outcome 4, which focuses on using techniques and tools in computing practices to deliver valuable, relevant solutions. By optimizing database query processes through indexing and testing for efficiency, I demonstrated my ability to manage data effectively for optimal performance. These adjustments align with Outcome 4 as they demonstrate my ability to improve quality through existing computing practices.

I do not see any updates needed to my outcome-coverage plans, as this enhancement effectively fulfills the initial goals. However, I will continue to refine the artifact based on instructor feedback to ensure it fully represents my skills.

**Reflection on the Enhancement Process**

Enhancing this artifact taught me a great deal about the impact of database indexing on performance, especially when working with large data sets. Initially, I encountered challenges with measuring query execution times accurately. To address this, I created test.py to monitor performance before and after indexing. This helped me clearly quantify the optimization’s impact.

Another challenge was simulating real-world data volume, which I addressed by adding numerous records to the database. This addition allowed for a more realistic evaluation of the indexing enhancements and confirmed that the changes provided a noticeable improvement in query efficiency. In a real world scenario I might add or simulate the addition of thousands of records to ensure scalability.

This process underscored the importance of testing and benchmarking code during optimizations. Measuring performance both before and after the enhancements was important for evaluating the success of my changes. I also learned that even minor optimizations, like adding indexes, can significantly improve performance in applications, which is invaluable knowledge for my career in software engineering and my current job as a Business Analyst.