**Category Two – Algorithms and Data Structures**

**Selected Artifact:**

For this assignment, I selected the artifact I completed in CS-340 Client/Server Development to exemplify my ability to implement enhancements in regard to algorithms and data structures. The original artifact consisted of a crud.py file that interfaced with a database of animals, and a Jupyter Notebook ProjectTwoDashboard that displayed the data to the user with accompanying analytics.

**Justification:**

This artifact was selected for this category due to its original, simplistic implementation. The crud.py only utilized basic CRUD functions and relied heavily on the client implementation to perform a lot of the computational work with the database results. The enhanced version of the application had improvements in three major ways. A simple data structure was implemented that contained frequently used indexes (including a compound index that was utilized by the application’s three rescue filters) to improve computational resources for lookups and sorting. Aggregation pipeline functionality was also implemented, offloading some algorithmic computation to the server, so a basic TTL Cache was implemented to save a list of results in a map, allowing for faster retrieval and reducing the need for frequent aggregations of the same type. The new version also implements a lazy loading algorithmic technique by fetching only one page's worth of data at a time from the database to improve efficiency in terms of time and memory.

**Outcome-Coverage Plan:**

With the completion of category two, I can say with certainty that I achieved more than the course outcomes attributed to this artifact.

1. Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science.

This artifact was modified and saved using git. Having an effective strategy for version control is necessary when constructing a large-scale application in a collaborative environment.

1. Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts.

The crud.py file is thoroughly commented on with explanations on the utilization of each method and how they are implemented.

1. Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices.

With the utilization of aggregation pipeline functionality, a simple TTL Cache, a data structure that can be referenced for frequently used indexes, and lazy loading algorithmic implementations, this artifact thoroughly displays an ability to design and evaluate computing solutions.

1. Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.

The original application was implemented in a virtual machine with many of the dependencies and aspects of the environment already configured. Migrating this project to my local machine required that I employ many strategies and tools to configure the application correctly. MongoDB Compass was used to load a new seed of test data points (the original collection was lost along with the virtual environment upon completion of the Server/Client Course), and Postman was necessary for thorough sanity checking.

1. Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

This artifact displays a moderate use of exception handling. This outcome was more thoroughly explored in the Category One portion of this assignment.

**Reflection:**

Before implementing this enhancement, I was not familiar with many of the concepts I used to enhance the artifacts. I had to conduct extensive research on aggregation pipeline functionality and pagination for implementing lazy loading functionality. The most difficult aspect of the application was configuring it to execute correctly on my local machine and setting up a new MongoDB collection that the crud.py could interface with. Below I included a screenshot of the application displaying each filtered screen.

**Reset:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Disaster or Individual Tracking:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Mountain or Wilderness Rescue:A screenshot of a computer

AI-generated content may be incorrect.**

**Water Rescue:**

**A screenshot of a computer

AI-generated content may be incorrect.**