

Name: Steven Valdivieso Lemus

Class: CS-499-13167-M01

Date: April 4th, 2025

5-2 Milestone Four: Databases

Artifact:

The artifact is AnimalShelter.py, a backend application that manages animal shelter data using MongoDB. It was created as part of CS 340 Project 2: Dashboard. Initially, the system allowed for basic CRUD operations to manage animal records but lacked efficiency in handling larger datasets. The enhancement involves adding advanced MongoDB features such as data indexing, aggregation pipelines, and user authentication with role-based access control. These improvements will optimize the system's performance and security, making it more efficient and scalable.

Justification for Inclusion:

I selected this artifact because it demonstrates my ability to optimize database-driven applications and enhance their functionality. The improvements I made, such as adding indexing and using aggregation pipelines, directly address performance bottlenecks. The addition of user authentication and role-based access control showcases my understanding of how to secure applications, which is a crucial skill in real-world software development. These changes enhance the system's scalability, security, and efficiency.

Data Structure Optimization: By adding indexing and using aggregation pipelines, I optimized data retrieval and complex query performance.

Algorithmic Efficiency: The aggregation pipeline enhances query performance by grouping and filtering data efficiently, improving the system's overall algorithmic performance.

Security and Access Control: Implementing user authentication and role-based access ensures that only authorized personnel can access sensitive data, highlighting my skills in securing software applications.

Specific Components Showcasing Skills:

Data Structure Optimization: Implementing indexing improves query performance by reducing lookup time, making operations faster and more efficient as the data grows.

Algorithmic Efficiency: The aggregation pipelines allow complex queries to run efficiently, enabling reports and detailed searches to be executed quickly.

Security and Access Control: Adding role-based access control ensures that only authorized users can perform sensitive operations such as updating or deleting records.

Course Outcome Coverage:

I successfully met the course outcomes related to software design, particularly:

Course Outcome 3: I designed and evaluated a computing solution by optimizing the AnimalShelter.py system with advanced database features like indexing and aggregation pipelines, applying algorithmic principles to improve performance.

Course Outcome 4: I implemented secure coding practices by incorporating user authentication and role-based access control, aligning with industry standards for building secure applications.

Reflection on the Enhancement Process:

The enhancement process provided valuable learning opportunities:

Optimization Of Performance: I gained insight into how indexing, and aggregation pipelines can drastically improve the performance of MongoDB queries, especially for large datasets. These enhancements make the system more efficient and scalable.

Improving Security: Implementing role-based access control was a valuable learning experience, as it ensured that only users with the correct roles could perform sensitive actions, such as updating or deleting animal records.

Challenges: One of the biggest challenges was ensuring that the new features worked harmoniously with the existing CRUD operations. Integrating the new MongoDB features required careful testing to avoid issues with the legacy code, particularly around user authentication and permissions. Another challenge was ensuring the system remained easy to use while adding complex database functionalities like aggregation and indexing.