Nelson Lohse  
 CS-499  
 10/04/2025

Module 4 Narrative

The original artifact was a set of four Java files created as part of an earlier course project (IT-145). These files implemented basic functionality for managing rescue animal data, including adding, viewing, and updating records. While functional, the program relied on simple in-memory lists and loops for storing and retrieving data, which limited efficiency and made it difficult to handle larger or more complex datasets.

I selected this artifact for my ePortfolio because it highlights my growth in algorithmic thinking and data structure implementation. The enhanced version of the Rescue Animal project is now a full-stack application featuring a Java backend, an Angular frontend, and a MongoDB database. The backend connects to MongoDB using two collections, Dogs and Monkeys, each stored as structured JSON documents that contain species-specific attributes. This setup allows for filtering and sorting across both collections, resulting in significantly improved organization and performance compared to the static lists used in the original version. On the frontend, data from each collection is displayed in interactive Angular tables that support both numeric and alphabetical sorting in ascending and descending order. This sorting logic demonstrates algorithmic design on the client side by efficiently reordering data directly within the user interface without needing to send queries to the backend.

To maintain data integrity and prevent invalid entries, I implemented structured input forms in the Angular frontend, utilizing dropdown menus and type-specific fields that restrict the types of data users can enter. This validation process ensures that all animal data entered into the Dogs and Monkeys collections follow a consistent format and logical structure. Together, these enhancements showcase my ability to design and implement efficient data models, apply sorting and filtering algorithms, and use appropriate data structures across both backend and frontend components.

This enhancement aligns with the program outcome of designing and evaluating computing solutions that solve a given problem using algorithmic principles and computer science practices. By structuring data in MongoDB collections, implementing client-side sorting algorithms, and integrating validation logic through Angular, I demonstrated my ability to apply algorithmic and data structure principles. My outcome-coverage plan now emphasizes structured data modeling, efficient retrieval methods, and interactive client-side data manipulation as key strengths developed through this enhancement.

Through this enhancement, I learned how effective data structures and algorithmic logic directly improve software performance, maintainability, and user experience. One of the primary challenges was ensuring that sorting, filtering, and validation operated smoothly together while maintaining accurate synchronization with the backend. I also have had very little experience using CSS until now, and it was a challenge to learn how to use CSS stylesheets to modify the Angular front end. Completing this enhancement reinforced my ability to design efficient, scalable systems using algorithmic and data structure principles in a professional, full-stack application environment.

