Nathaniel Burt

CS 499

6/8/2025

Database Enhancement Summary

What is the artifact? When was it created?

The artifact is a full-stack rescue animal management application developed using the MERN stack — MongoDB, Express.js, React, and Node.js. It originated from a simple RescueAnimal.java class that was initially created in an earlier course back in 2023. In 2025, I transformed this model into a functional CRUD application with a backend database, RESTful APIs, and a styled, responsive frontend using React and React Bootstrap.

Why did you select this item for your ePortfolio?

I selected this artifact because it is a strong demonstration of my ability to apply full-stack development skills and integrate database systems into a modern web application. This project shows that I can build scalable, real-world applications that manage data persistently and provide user interaction through intuitive interfaces. It also reflects my ability to refactor legacy code and repurpose it using current technologies in the software development industry.

What specific components of the artifact showcase your skills and abilities in software development?

The artifact highlights several key skills, including backend API development using Express, MongoDB schema modeling with Mongoose, and frontend UI creation with React. It also showcases my ability to perform form validation, implement error handling, and manage

asynchronous data communication using Axios. The application includes modular components for editing, adding, and listing rescue animals.

How was the artifact improved?

The original artifact was a simple Java class used for storing and retrieving animal properties. It was improved by fully converting it into a modern CRUD application with persistent storage, form-driven interactions, and a dynamic frontend. I created a Mongoose model for database integration, built RESTful API routes for full record management, and implemented user-facing components with styling libraries. The project was also enhanced with inline comments, error-handling middleware, and responsiveness through a Vite-powered React interface.

Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

Yes, I met the outcomes planned for this enhancement. These include demonstrating CRUD operations using a NoSQL database, applying client-server communication principles, and structuring a project into maintainable, modular components. The project also meets objectives related to software integration, API development, and database design. I currently have no updates to my outcome-coverage plans as this enhancement aligns well with the intended goals.

What did you learn as you were creating and improving the artifact?

Through the enhancement process, I deepened my understanding of full-stack development workflows. I learned how to structure a RESTful backend that works smoothly with frontend requests and how to troubleshoot issues like port mismatches and form validation errors. I also learned how to use React state hooks (useState, useEffect) to handle data flow and form

interactivity. This project helped me better understand the relationship between frontend components and database logic in a production-ready architecture.

What challenges did you face?

One of the primary challenges was debugging communication between the frontend and backend, especially when dealing with Axios requests, server ports, and error messages related to missing schema fields. Another challenge was converting form inputs to match the required data types expected by the MongoDB schema. I also had to learn how to configure Vite correctly to support JSX, which taught me more about modern build tools and module resolution in React apps.