Milestone Four Narrative: Enhancement Three: Databases

The Artifact I selected for this milestone is the same Rescue Animal Management System that I enhanced for Categories 1 and 2. It began as a Java-based console program created ruing the IT-145 course. In the original version, all animal records were stored in memory using ArrayList, which meant data was lost every time the program closed. There was no real database in place. Just temporary data structures that existed only during runtime. While the program allowed basic intake and reservation of rescue dogs and monkeys, it was not suitable for real-world use, especially when persistence and data integrity were required.

To algin with the Databases category, I enhanced the program by implementing a proper data persistence layer using MongoDB. I used spring Data MongoDB to map Java objects to MongoDB documents and designed a unified Animal model to represent both dogs and monkeys in a single collection. The backend now communicates directly with a MongoDB instance, allowing all animal data to be stored and retrieved reliably, even across sessions and devices. Users no longer lose data when the program stops. Instead, all information is permanently stored in the database and can be updated or queried as needed.

I chose to include this artifact in my ePortfolio because it demonstrates my ability to design and integrate a real NoSQL database into a full-stack application. It shows that I understand how to create a data model, interact with MongoDB using repository interfaces and custom queries, and handle real-world concerns like input validation, document structure, and security. This work reflects my progress from writing console applications to building scalable, persistent systems that operate more like real-world software.

This enhancement specifically meets the course outcomes related to databases. It demonstrates that I can connect an application to a modern backend database, persist and retrieve documents, and use industry-standard tools to manage and interact with data. I followed the enhancement plan I submitted earlier in the course and completed all planned upgrades, so no changes were necessary. I also extended my work by refining how the MongoDB queries are constructed and how inputs are validated before being stored.

The process of integrating MongoDB taught me several things. I learned how to define model classes using Spring annotations like @Document, @Id, and @Field, and how to link those to MongoDB collections. I also learned how to use repository interfaces and query methods to perform CRUD operations, like finding animals by specific fields or updating documents based on user input. One challenge I encountered was ensuring that user input was validated properly before saving data. Because MongoDB is schema-less, it was important to use Spring’s validation features to protect the database from receiving bad or incomplete data. I added backend validation using annotations like @NotBlank and @Size, and handled exception for invalid inputs with proper error messages.

Another challenge I faced was learning how MongoDB handles ObjectId values and how to map those properly in Java using String types. Early in development, I experienced issues with retrieving records by ID due to type mismatches. I resolved this by standardizing ID field across the application and making sure the frontend and backend used the same identifier format. I also spend time organizing the database into a flexible structure that can be easily expanded. Instead of keeping dogs and monkeys in separate collections, I created a single, unified Animal model that stores all species and differentiates them with a type field. This simplified the code and made it easier to apply filters, queries, and updates consistently.

This enhancement supports Course Outcome 4, which emphasizes using well-founded and innovative tools and techniques in computing practice as well as Course Outcome 5, which focuses on secure data design and protection. Through this enhancement, I demonstrated the ability to design a secure and persistent data storage solution using modern tools. I also made progress on Course Outcome 3, by showing that I can evaluate the trade-offs between different storage models and implement a database that supports both performance and flexibility.

Overall, this enhancement transformed the application from a short-lived, memory-based program into a durable, data-driven system that can be accessed and managed across sessions and devices. It showcases my understanding of full-stack development with a focus on data persistence and structure. It also demonstrates that I can handle real-world data concerns like validation, performance, and expansion.