# CS 499 Module 4-2 Narrative

# Algorithms and Data Structure

**Briefly describe the artifact. What is it? When was it created?**

The artifact is an Inventory Management System initially created as an Android mobile app during a course project for CS-360, focusing on Android development. This project, developed using Java and Android SDK, allowed users to manage inventory by adding, updating, and deleting items. Recently, I translated this mobile app into a web application using the MEAN stack, which includes MongoDB, Express.js, Angular, and Node.js.

**Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved?**

Here’s a refined version of your content, focusing solely on data structures and algorithm enhancements:

This artifact is included in my ePortfolio because it exemplifies my ability to adapt and optimize software across different platforms, showcasing my growth as a developer. The migration from a relational database in SQLite to a NoSQL document-oriented model with MongoDB required a deep understanding of data structures and algorithms. One significant enhancement was the implementation of indexing on critical fields such as email in the Users collection and name in the InventoryItems collection. This optimization improved data retrieval efficiency, particularly in queries that handle user authentication and inventory management.

I also designed and implemented flexible and well-structured Mongoose schemas to manage the application's data, specifically in the User.js, InventoryItem.js, and Notification.js files. These schemas allowed for more dynamic and scalable data handling, accommodating a variety of data types and structures while maintaining data integrity. By leveraging MongoDB’s document-based model, I was able to create a more adaptable and efficient data storage system, which significantly improved the overall performance of the application.

Additionally, the artifact demonstrates the use of async/await patterns for non-blocking operations, which are crucial in handling asynchronous database queries and enhancing the application’s performance. These algorithmic enhancements ensured that the application could handle larger datasets and more complex operations without compromising on speed or efficiency.

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

Yes, I successfully met the course objectives by showcasing advanced technical skills, particularly in data handling and full-stack development. The enhancement involved optimizing CRUD operations, implementing user authentication algorithms, and incorporating user notification systems—all of which demonstrated my ability to design and execute algorithms that are efficient and scalable. Moving forward, I plan to further develop my focus on security and data privacy, ensuring robust protection mechanisms in my projects.

**Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

Enhancing and modifying this artifact provided valuable insights into full-stack development, particularly the importance of efficient data structures and algorithms in a NoSQL environment like MongoDB. One significant learning outcome was understanding how to effectively use MongoDB’s indexing to improve query performance and manage large datasets. Challenges included integrating real-time data updates in the Angular frontend and managing asynchronous operations in Node.js to handle concurrent requests efficiently. Overcoming these challenges strengthened my ability to integrate various technologies and deliver a cohesive, high-performance software solution.