**Module 4: Milestone 3**

Zane Milo Deso

Southern New Hampshire University  
CS-499

**Milestone 3**

**Introduction**

This milestone gave me the chance to reflect on one of the most foundational skills I’ve picked up throughout the program. Algorithms and data structures were concepts I grasped early on but applying them meaningfully has taken longer to mature. In this narrative, I talk about the algorithm I built for matching recipes in my capstone project and how I’ve grown by thinking more critically about structure, logic, and performance.   
**Artifact Description**

The artifact I selected for this milestone is the matching system in my capstone project, a recipe recommendation web app. The enhancement I focused on was rewriting the ingredient matching logic to use binary vectors and a dot product function. The goal was to better match a user’s selected ingredients to stored recipes based on similarity. The algorithm checks for overlapping ingredients by representing both the user's input and each recipe as vectors and then scoring them by their dot product.

## **Justification for Inclusion in ePortfolio**

I chose this part of my project because it let me demonstrate algorithmic thinking in a way that’s directly tied to user experience. With the binary vector approach, the logic is clean, fast, and predictable. It shows my understanding of how to work with arrays, logical comparisons, and scoring systems. Even though this isn’t some advanced tree or graph traversal, it’s a good example of how I’ve learned to apply structured thinking to a real problem. It also gives me a solid starting point if I want to add weights or confidence scoring later.

## **Course Outcomes Addressed**

This enhancement supports the outcome related to designing and evaluating computing solutions using algorithmic principles and solid CS practices. I planned this enhancement back in Module One, and I stuck to that plan. I wanted to implement an algorithm that made sense, worked reliably, and didn’t overcomplicate the codebase. I feel like I’ve done that here. No major changes to the outcome plan, but doing the enhancement helped me realize how much more I could explore if I wanted to tune or expand this logic.

## **Reflection on the Enhancement Process**

Creating and implementing the dot product matching algorithm was straightforward in terms of logic but challenged me to think carefully about edge cases and performance. I had to ensure all recipes were vectorized with the same structure and account for missing ingredients or extra input. It also made me more aware of how even simple mathematical techniques can improve user experience when applied correctly. One challenge was deciding how to weigh ingredients or score matches more meaningfully, and this is something I might return to in the future. Overall, I learned how small algorithmic improvements can lead to better application behavior and clearer code.

## **Conclusion**

This milestone helped me step back and appreciate the value of thoughtful algorithm design. It’s easy to focus on the front-end or the visuals, but the underlying logic is what gives the application its backbone. Enhancing the matching system reminded me that real-world software can mainly be about complexity, although it’s jsut as much about creating functionality that works, scales, and can be built on later.