John Nguyen

September 28, 2025

CS-499

Algorithms and Data Structures Enhancement

For this milestone, I chose my Travlr Getaways full-stack web application again, which I created in CS 465. This project is a travel booking app where customers can see available trips and administrators can manage trips through an Angular admin panel. The backend uses Express.js with MongoDB, and the front end uses Angular. When I first built the project, the data handling was simple and basically just basic CRUD operations to get and show all the trips.

I included this artifact in my ePortfolio because it gave me a chance to show off my skills in algorithms and data handling. The original version didn’t use advanced data structures or optimization, so I knew I could make improvements. For my enhancement, I added server-side pagination, so the app only loads a set number of trips at a time, instead of all of them. I also added MongoDB text search with indexing so users can find trips by name or description. On the frontend, I added a debounced search feature, so the app doesn’t send too many requests at once, and I used a Map structure to cache trips for quick lookups. These changes make the project run smoother and show that I can apply algorithmic thinking to improve performance.

I’ve made progress on the program outcome for designing and evaluating computing solutions using algorithmic principles. Adding pagination, search, and caching shows that I can think about and scalability. I also practiced making trade-offs, like deciding what should happen on the server versus the client. My original outcome plan hasn’t changed, and I’m on track to meet the algorithms and data structures outcome for the final project.

Working on these updates helped me see how important algorithms are in applications. At first, I didn’t think pagination or caching would matter much for a small project, but once I simulated data, I realized how slow the original version was. These changes have taught me how big an impact even small algorithm improvements can have.

One of the hardest parts was getting the debounced search in Angular to feel right. If the delay was too short, it still spammed the server with requests; if it was too long, the app felt laggy to use. I had to test it a lot to get it balanced. Another challenge was making sure pagination worked correctly together with filtering and searching. That took some debugging, but in the end, I figured out how to keep everything in sync.

Overall, these enhancements made the project much more professional and something I’d feel confident showing in my ePortfolio. They also gave me more practice thinking like a software engineer who must consider performance and scalability.