John Nguyen

October 5, 2025  
CS-499

Databases Enhancement

The artifact I selected for my database enhancement is the Travlr Getaways project, a full-stack travel booking application developed during my coursework in Full Stack Web Development. The project includes a Node.js and Express backend connected to a MongoDB database. I created this project to show my understanding of data management, API integration, and CRUD functionality. For this enhancement, I focused on improving the database’s performance, structure, and security by optimizing queries, applying indexes, and implementing validation and error handling.

I chose this artifact because it represents complete and realistic software projects. The database serves as the backbone of the application, storing all travel package information, user accounts, and booking data. Enhancing the database, I was able to showcase skills such as indexing, query optimization, data validation, and secure CRUD operations. I added compound indexes and text indexes in MongoDB to support faster search queries and ranking results by relevance. I also improved error handling and ensured validation for a defined field, which makes the database reliable and maintainable.

This enhancement helped me meet the outcome related to demonstrating the ability to use innovative techniques, skills, and tools in computing practices for implementing database solutions that deliver value and accomplish industry-specific goals. By implementing optimized queries, compound indexes, and a schema structure. I supported the performance and reliability of the database system. These improvements connect with focusing on algorithmic and system design because indexes are data structures that improve query performance. My future goal is to extend this work by integrating data analytics features using aggregation pipelines or potentially connecting the application to a cloud-based service like MongoDB Atlas for scalability.

Working on this enhancement helped me better understand how database performance affects the responsiveness and scalability of an application. One major challenge was balancing flexibility with structures like MongoDB’s schema design, but it can also lead to inconsistent data if validation is not handled properly. Implementing stricter validation and using Mongoose’s built-in schema rules helped address this. Through testing and analyzing query performance using MongoDB’s. explain() function, I learned how to measure the trade-offs between speed and storage.

This milestone also reinforced my ability to think about how front-end queries impact database performance. For example, when I implemented server-side pagination in a previous enhancement, I noticed that using appropriate indexes significantly reduced query response time. Overall, this enhancement reinforced my confidence in working with databases and prepared me to manage data-driven applications.