CS 499: Milestone Four: Database Enhancement

Noah Khomer

Southern New Hampshire University

CS-499 (Capstone Project)

Joseph Martinez

10/5/2025

CS 499: Milestone Four: Database Enhancement

The artifact that I selected for the database category of my portfolio is the Travlr Getaways full-stack web application. This was something I originally developed in the Computer Science 465: Full Stack Development course at Southern New Hampshire University. This project was initially built as simple as it could be. However, one of the most complex aspects of it was that it used the MEAN stack.

That means this project was made using MongoDB, Express, Angular, and Node.js, so that it could simulate a travel planning platform where users can view, create, and manage trip listings with full creation, read, update, and delete (CRUD) functionality. The main purpose of this application during the CS 465 course was that it served as a strong demonstration of client-server architecture, authentication, and database integration.

However, for this course, I revisited the original codebase so I could enhance the backend logic and the database structure with a specific focus on improving performance, query design, and data integrity. The enhanced and updated version now aligns better with industry practices for data-driven web application development.

The reason I chose this artifact is because it allowed me to demonstrate real-world database integration in a dynamic, user-facing web application. A lot of the code was already provided through the CS 465 project, which was an incredibly challenging course, but it helped me build most of the project early on. It highlights my ability to work with and implement complex data operations using MongoDB, and to design schemas that support scalable functionality.

For this course, the main enhancements that were made (along with many others) included refactoring the Mongoose schemas so they better define the relationships between users, trips, and reviews. I implemented advanced queries and aggregation pipelines to support analytics and filtered search. I also established role-based access control, which is important for protecting sensitive operations like trip creation endpoints. Lastly, I worked on improving performance through index creation, which helped reduce redundant database calls.

All these changes highlight my database design and development skills while also aligning with modern backend development standards. I revisited, audited, and enhanced legacy code, then published the improvements to the main branch on GitHub.

One of the biggest challenges I had during this process was making sure that any changes I made did not break the Angular frontend. For example, the backend had to remain compatible with the existing API contracts, so I had to make sure any new indexes or aggregation queries did not negatively impact performance under testing conditions.

Overall, this enhancement taught me the importance of planning backend changes while understanding key principles such as performance, security, and structure when working on a database-driven application.

**References**

Baumann, P. (2024). Open standards for integrating Datacube backend, catalog, and visualization tools: [https://doi.org/10.1109/iceccme62383.2024.10797180](https://doi.org/10.1109/iceccme62383.2024.10797180%20)

Database schema integration. (n.d.). *SpringerReference*. [https://doi.org/10.1007/springerreference\_61915](https://doi.org/10.1007/springerreference_61915%20)

Google. (n.d.). *Aggregation queries  |  datastore  |  google cloud*. Google. [https://cloud.google.com/datastore/docs/aggregation-queries](https://cloud.google.com/datastore/docs/aggregation-queries%20)

Kapusta, J., Baran, W., & Ján, J. (2021). *Frameworks for Frontend Application Development (jQuery, Angular, Vue)*. [https://doi.org/10.17846/fpvai-2021-24](https://doi.org/10.17846/fpvai-2021-24%20)

*What is the mean stack? introduction & examples*. MongoDB. (n.d.). [https://www.mongodb.com/resources/languages/mean-stack](https://www.mongodb.com/resources/languages/mean-stack%20)