John Nguyen

October 19, 2025

CS-499

CS-499 Artifact Narratives

Introduction

For my ePortfolio, I chose to use the Travlr Getaways project, a full-stack travel booking web application that I developed using the MEAN stack (MongoDB, Express, Angular, and Node.js). This project demonstrates design, build, and enhance real-world software solutions. Over the course, I performed three major enhancements focused on software design, engineering, algorithms, data structures, and databases. Each enhancement increased the project’s scalability, security, and maintainability while showcasing skills I have developed throughout the Computer Science program.

Milestone Two: Software Design and Engineering

Artifact Description

The Travlr Getaways project was originally created in my CS-465: Full Stack Development course. It is a web application that allows users to browse and manage travel packages. For this enhancement, I focused on the backend API to improve modularity, maintainability, and error handling. The technologies used include Node.js, Express, and Angular for the front-end, and Winston for logging and CORS configuration for cross-origin requests.

Justification for Inclusion

I selected this artifact because it represents my growth in software engineering, specifically in applying modular architecture, security practices, and clean code design. The enhancement allowed me to refactor the application’s structure to align with full-stack development. I also implemented a logger utility using Winston to track server activity and errors. These improvements made the codebase easier to maintain and debug while supporting the real-world practices of software development.

Enhancements and Improvements

* Created a logger utility using Winston in /utils/logger.js to capture system events and error logs.
* Refactored API controllers (especially trips.js) to use structured error handling and improved responses.
* Enhanced security with JWT authentication and modularized middleware to protect administrative routes.
* Updated the app.js file to use environment variables (.env) for sensitive data such as JWT secrets and database credentials.
* Implemented consistent HTTP status codes and cleaner response structures for RESTful API standards.

Reflection

Through this enhancement, I expanded my understanding of software design patterns and the importance of clear, maintainable code. I learned how modularization improves scalability and how proper logging helps developers monitor application in production. One of the challenges was debugging API routes after refactoring the controllers. Implementing structured error handling made the code more predictable and easier to test. This enhancement connects to the program outcomes of designing quality software and using tools to improve existing systems.

Milestone Three: Algorithms and Data Structures

Artifact Description

For this milestone, I focused on improving the data flow within the Travlr Getaways backend. The original system was functional but lacked optimization in how it processed and validated travel data. Using Mongoose queries and custom validation, I established algorithmic improvements that made the backend efficient and reliable.

Justification for Inclusion

I chose this artifact because it demonstrates my ability to design and evaluate computing solutions using algorithmic principles. The improvement process allowed me to analyze how requests and data objects were handled throughout the system. I refactored the Trip controller logic to use asynchronous operations with async for non-blocking execution and improved overall API performance.

Enhancements and Improvements

* Rewrote trip queries to use Mongoose’s .lean() method, which optimizes read performance by returning plain JavaScript objects instead of full documents.
* Introduced input validation middleware (validators/tripValidator.js) to ensure data and reduce redundancy.
* Implemented secure password hashing and comparison using bcryptjs in the user.js model to enhance authentication.
* Added a custom sorting algorithm in the trips.js controller to display the newest trips first based on creation date.
* Optimized control flow using try/catch blocks for better error recovery and system resilience.

Reflection

This enhancement helped me strengthen my problem-solving and understand how algorithmic applies to backend development. I learned to identify performance bottlenecks and improve response times through efficient data retrieval methods. A challenge I faced was balancing query complexity with readability and too much optimization that made debugging difficult. Through repetition, I found the right balance between clarity and efficiency. This milestone demonstrates my competency in designing computing solutions that manage trade-offs.

Milestone Four: Databases

Artifact Description

The final improvement is improving the MongoDB database layer of Travlr Getaways. The project had a basic database connection file and limited data management capabilities. In this milestone, I expanded the system to include indexing, environment-based configurations, and a script for test data automation. These upgrades improved the project’s reliability, scalability, and real-world usability.

Justification for Inclusion

This enhancement highlights my ability to work with database design, data modeling, and secure connectivity. Managing data efficiently and safely is one of the important aspects of modern web development, and this milestone allowed me to demonstrate that. I implemented structured connection handling, added a .env file for secure MongoDB URIs, and created a seed script to populate sample data for testing.

Enhancements and Improvements

* Updated database.js to use Mongoose connection pooling with detailed error logging.
* Created a seed script (seed.js) to populate MongoDB collections with realistic test data.
* Implemented MongoDB indexes to improve query performance on frequently accessed fields (e.g., trip name, location).
* Added environment-based database configuration for development and production modes using .env.
* Developed a new analytics.js controller that uses MongoDB aggregation pipelines to generate trip statistics (e.g., most popular destinations).

Reflection

This enhancement taught me the importance of data management and security in full-stack systems. I learned how to structure a database connection that’s both efficient and secure. I also saw how MongoDB’s indexing and aggregation tools can improve query speed and add business intelligence capabilities to an app. The biggest challenge was configuring multiple environments for development and production. It gave me a better understanding of deployment workflows. This enhancement aligns with the course outcome of using well-founded techniques to implement computing solutions that deliver real-world value.

Conclusion

These three enhancements transformed Travlr Getaways into a more polished, secure, and scalable full-stack application. Each milestone highlighted a different area of computer science from software architecture and clean code to data efficiency and secure database management. These artifacts represent my growth as a developer and my willingness to contribute to software engineering.