Andrew Torrez

CS-499-10454

October 5th, 2025

Traveler App

Artifact 3: Databases

**Artifact Description**

The artifact I selected for my database enhancement is the **Travlr App**, a full-stack MEAN (MongoDB, Express, Angular, Node.js) web application that allows users to explore, book, and manage travel experiences. The original project was created during **CS 465: Full-Stack Development** and was later enhanced for my CS-499 Capstone to strengthen its **database functionality, authentication reliability, and admin management capabilities**.

The Travlr App includes a Node/Express backend, a MongoDB database managed through Mongoose, and a dynamic Angular-based interface. The database stores user, trip, and reservation information, while JSON Web Tokens (JWTs) handle authentication between the client and the server. Admin users can add, edit, and delete trips, while regular users can browse and reserve them. My enhancement focused on **fixing database communication issues and improving how the application handled and displayed data** between user interactions and MongoDB.

**Justification for Inclusion**

I chose this artifact because it represents one of the most challenging and rewarding projects of my academic journey. It demonstrates my ability to design and integrate a database, manage user authentication, and build secure backend systems that connect to an interactive user interface. The project also helped me practice realistic debugging and version control workflows, similar to what professional developers experience.

When I began the enhancement, the login screen loaded, but new user registrations were not being stored in the database. The admin page also looped endlessly to another login screen, and no reservation data appeared. Through careful debugging and restructuring, I fixed those routing and database issues and built a **functional authentication and reservation flow**. I also implemented **role-based access control** so that admin users could manage trip data while regular users had limited privileges.

This project demonstrates my ability to **diagnose, design, and execute real-world fixes** that transform a partially functioning prototype into a secure, reliable web application.

**Achievement of Course Outcomes**

This enhancement meets several **CS-499 program outcomes**, particularly in **Software Engineering, Database Design, and Security**.

* **Database Design:** Improved MongoDB schemas and Express routes for CRUD operations, ensuring data consistency and validation through Mongoose.
* **Security and Data Privacy:** Implemented JWT-based authentication, password hashing, and secure environment variable handling to protect sensitive information.
* **Collaboration and Professional Practice:** Used Git branching, commits, and rollbacks to track changes and recover from errors. These practices strengthened my understanding of professional development workflows and version control discipline.

These improvements collectively demonstrate my ability to build a reliable, secure, and maintainable full-stack application aligned with real-world industry practices.

**Reflection on the Process**

Revisiting this project after several months allowed me to understand full-stack systems more deeply. In earlier versions, I didn’t fully grasp how the backend, frontend, and database interacted. Through this enhancement, I learned to trace how user input travels from the interface to the database and back through the API. I discovered how a small issue, such as a missing route or configuration, could interrupt this flow.

One major breakthrough came when diagnosing why new user registrations weren’t saving. By creating a dedicated registration controller, properly defining API routes, and wiring up Mongoose models, I successfully restored full functionality. Implementing admin privileges and fixing the reservation logic deepened my understanding of **data flow and access control** in multi-user systems.

There were many frustrating moments — failed JWT verifications, broken layout files, and entire sections of the app crashing — but each setback taught me patience, persistence, and the value of testing and documentation. I also cleaned and reformatted my codebase, removing temporary console logs and rewriting all comments to explain intent and logic clearly. These efforts made my project easier to read, maintain, and scale, reflecting professional coding standards.

Ultimately, this enhancement helped me grow from simply following tutorials to truly understanding how modern web applications work across multiple layers.

**Learning and Growth**

This project solidified my confidence in several core areas:

* Debugging and interpreting server logs
* Designing and validating database schemas
* Structuring RESTful APIs and routes logically
* Applying secure authentication practices with JWTs
* Managing complex projects with Git branching and version control

It also taught me how to **balance scope with stability**, focusing on fixing broken features before adding new complexity. This mindset reflects real-world software engineering priorities and project management discipline.

**Summary and ePortfolio Connection**

Enhancing the Travlr App represents the culmination of my journey toward becoming a full-stack developer. It shows how I can approach complex systems, identify root causes of failure, and rebuild them into working, production-ready solutions. This artifact directly supports my **ePortfolio goals** by showcasing my ability to:

* Solve real-world technical challenges through structured debugging
* Integrate database and backend functionality with a secure user interface
* Produce maintainable, documented, and professional-quality code

This project not only demonstrates my technical skill but also my **growth as a problem solver and software engineer** capable of bridging theory and practical application.