InvenTitan and Travlr Getaways – Coding My Way to Success  
Southern New Hampshire University  
Cortland Burns  
27JUL2025

For milestone three, I will describe what I did for my two artifacts that can show my ability to create and add efficient solutions using algorithms and well-structured data systems. These projects, InvenTitan and Travlr Getaways both show a lot of growth in my journey as a Computer Science major, and each has given me plenty of challenges to dig into as I learn to understand how software handles real-world data.

The first artifact, InvenTitan, is a mobile inventory management application originally developed as a native Android application in Java. I created it around March 2025 and later migrated to Flutter with Dart in July 2025. The original application used an SQLite database with Java logic spread across some other close activities. In Flutter, I rebuilt the entire project with cleaner architecture using models, services, screens, and widgets. I used the sqflite package to maintain the full CRUD capabilities, keeping the original build, and even added a new field for optional descriptions. The best improvement came from redesigning the algorithm that handles the item\_list rendering and input validation by streamlining the database access and how the UI updates through the widgets and allowing it to be more efficient list-building logic. These changes made the app not only more scalable, but also make it easier to understand and reuse, especially for a cross-platform deployment.

The second artifact, Travlr Getaways, was built for my CS 465 course and has been improved throughout my time working on this capstone. It is a full-stack travel booking application using MEAN stack (MongoDB, Express, Angular, Node.js). Originally created between May and June 2025, this one has a much more intricate algorithmic foundation in both the frontend and the backend. I implemented token-based authentication using JWT and then updated a lot of the Angular components focused on the trip listing and search logic to support real-time filtering, sorting, and routing. I also remade the backend Express routes, so they work with the query parameters I set, giving the admin panel a more flexible and scalable build. These updates required me to understand both the data modeling and how it affects performance, especially when dealing with API route designs and client-side state management.

Together these projects meet the outcomes I planned on making in Module One by showing algorithmic thinking in both web and mobile environments. In InvenTitan, I took on the challenge of efficiency in local data handling using SQLite and Dart’s list structures. In Travlr Getaways, I focused on real-time interaction with a NoSQL database using optimized RESTful endpoints and reactive frontend filters. Both projects show my abilities with designing solutions that are not just functional, but they are clean, secure, and adaptable for changes needed.

The process of improving these artifacts pushed me to take on some newer technologies I am unfamiliar with and allowed me to think more critically about structure and troubleshoot the edge cases that were not always obvious at first. In Flutter, I had to learn new ways to manage form states and error handling without depending on the Android Activity lifecycle I was used to. In the MEAN stack, I had to rewrite older form logic to use Angular reactive forms and refactor the backend controllers to support cleaner routing and validation. These were not just surface-level changes, they were in depth refinements that I needed to make, and it really helped me become a better problem solver.

I selected these artifacts for my ePortfolio because they have and can show real, iterative progress on my skillsets. They reflect what happens when you keep working through problems, test different approaches, and optimize not just for code that works, but for code that can stay strong. From modular design to maintaining my data, both InvenTitan and Travlr they are showing how far I have come in algorithms, data structures, and software engineering. I am taking this class as more than just an assignment, but projects that I have built, broken, and rebuilt, and something that I can be proud to showcase and make a career out of.