The artifact I selected for my ePortfolio is Travlr Getaways. Its a full-stack web application designed to give users a perfect travel booking experience for customers and includes an administrative interface for managing bookings, customers, and trips. I created this project in the previous semester Summer 2025. It demonstrates the use of the MEAN stack (MongoDB, Express, Angular, and Node.js) which is good for developing scalable, dynamic and user-friendly applications. I chose the Travlr Getaways artifact for my ePortfolio because it shows my skills in full-stack web development. This includes designing innate user interfaces, implementation of secure data management, and making the connection of front-end and back-end systems. For this enhancement, Databases, I chose to make changes to the Trip Schema. This is located in travlr.js which is part of the back-end database design. This schema outlines the structure of all the trip data stored in MongoDB which have fields like code, name, resort, price, and description.

The changes I made to the Trip Schema show my skills in database design and protecting data integrity. I added new validation rules like requiring unique trip codes, making sure (perPerson) isn’t negative or zero, and only allowing future start dates. I also switched perPerson from a string to a number type so that prices would show in filters. On the Angular front end I updated the Trip model and the trip listing logic. This included removing the need to parse the price as a string and making sure the currency pipe worked right with numbers. These changes make the schema more professional and making it more reliable and easier to use.

I met the course outcomes I planned to cover in Module One. I was able to show that I can design and implement database solutions that apply data integrity and support the application. I also showed front-end and back-end consistency in this project since changing a schema type meant updating the Angular code as well. While enhancing this artifact I learned how small adjustments in a schema can have a big impact on the overall project. One of the problems I had was when I realized that changing perPerson to a number required updating more than the schema. Solving this issue was simple to me which gave me more confidence in my database and front-end integration skills.