Paige Heiney

CS-499

09 June 2024

Milestone Four Narrative

Description of the Artifact

The artifact in question is a user.js file, part of the backend code for the Travlr Getaways application. Initially created during CS-370 Full Stack Development 1, this file defines the user schema for the MongoDB database, handles password hashing and validation, generates JSON Web Tokens (JWTs) for authentication, and includes functions for user management such as listing, adding, and updating users.

Justification for Inclusion

I selected this artifact for my ePortfolio because it showcases a range of my skills in software development, particularly in the areas of database management, authentication, and security. The enhancements made to this file demonstrate my ability to implement robust data structures, improve security measures, and refine code for better performance and maintainability. Specific components, such as the implementation of password hashing with crypto, JWT generation for user authentication, and the addition of query parameter filtering in the usersList function, highlight my proficiency in developing secure and efficient backend solutions.

Meeting Course Objectives

Through this enhancement, I met several key course objectives. I employed strategies for building collaborative environments by writing clear, maintainable code and comprehensive documentation, facilitating teamwork and enabling diverse audiences to support organizational decision-making. The enhanced user.js file is now more readable and maintainable, aiding future developers in understanding and extending its functionality. Additionally, I demonstrated my ability to design, develop, and deliver professional-quality code that is technically sound and appropriately adapted to user authentication and security. By incorporating best practices in password hashing, JWT generation, and user data validation, the code is robust, secure, and ready for real-world deployment.

Furthermore, I designed and evaluated computing solutions using algorithmic principles and computer science standards, managing trade-offs such as balancing security with performance. The improved password handling and authentication mechanisms showcase this ability. I utilized well-founded and innovative techniques, skills, and tools to implement solutions that deliver value and meet industry-specific goals, particularly in software engineering and database management. Lastly, I developed a security mindset by anticipating potential adversarial exploits and implementing measures to mitigate design flaws, ensuring the privacy and enhanced security of user data through robust hashing algorithms, secure token generation, and comprehensive validation techniques.

Reflection on the Enhancement Process

The process of enhancing and modifying the artifact was both challenging and educational. One key lesson learned was the importance of implementing robust security measures, such as using salts and hashing algorithms for passwords and generating secure tokens for authentication. This experience deepened my understanding of how vital it is to protect user

data from potential breaches and attacks. I also learned about the complexities of managing user authentication and authorization, which are critical for any application dealing with sensitive user information.

Additionally, I improved my skills in writing cleaner, more maintainable code by refactoring functions, adding error handling, and ensuring proper validation. This included using express-validator to ensure that user input met required criteria before processing, which is crucial for preventing invalid data from entering the system. The importance of thorough error handling became clear as I worked to make the system more resilient to unexpected inputs and conditions.

The challenges I faced included ensuring compatibility with existing code and integrating new features without introducing bugs. This required a deep understanding of the existing codebase and meticulous testing to verify that enhancements did not negatively impact the application's functionality. Debugging issues related to asynchronous operations and ensuring consistent behavior across different parts of the application were particularly demanding but ultimately rewarding.