Paige Heiney

CS-499

17 May 24

Code Review Journal

Part 1: Code Review

A code review is a systematic examination of source code by developers other than the author. This practice is crucial for computer science professionals as it ensures code quality, identifies bugs, promotes knowledge sharing, and enforces coding standards. Best practices for code reviews include ensuring thorough error handling, input validation, and proper documentation. Code reviews should occur before merging code into the main branch to catch issues early, preventing costly fixes later.

Part 2: Recording a Code Review

For recording the code review, I will use the screen recorder software for Windows 11. The recording will follow an outline based on the rubric and code review checklist, covering three main categories: Software Design and Engineering, Algorithms and Data Structures, and Databases.

Category One: Software Design and Engineering

I will review the trips.js file, which handles trip-related operations like listing, finding, adding, and updating trips. The functionality is well-implemented, but error handling can be more consistent, and input validation is missing. Enhancements include implementing input

validation, improving error handling, and adding a basic search algorithm to improve functionality and user experience.

Category Two: Algorithms and Data Structures

The index.js file contains route handlers for Travlr Getaways, including user authentication and CRUD operations on trips, secured by JWT middleware. While the code is clear and efficient, it lacks authorization logic and detailed error handling. Testing is also absent. Enhancements involve improving error handling, adding authorization, and introducing sorting functionality for trips, which demonstrates proficiency in algorithmic principles and data structures.

Category Three: Databases

The user.js file defines a Mongoose schema for users, handling password hashing and JWT token generation. While it uses a cryptographic hash, it lacks proper salting, error handling, and secure secret management. Enhancements include replacing SHA-512 with bcrypt for better security, implementing error handling, and securing the JWT secret in environment variables. These changes will improve security, error handling, and overall code reliability.

In conclusion, the existing code across all files demonstrates good structure and adherence to coding standards, but improvements in error handling, input validation, and security practices will enhance robustness and maintainability. The proposed enhancements will align the code with best practices and course outcomes, ensuring a more secure and user-friendly application.