Artifact Summary for Code Review Enhancement

# 1. Software Design and Engineering

Artifact Description:

The artifact selected is the Travlr Getaways MEAN Stack Web Application. This artifact is a full-stack travel booking app built using MongoDB, Express, Angular, and Node.js. It directly reflects software design principles, such as modular architecture, MVC structure, and RESTful API design.

Artifact Analysis:

This project uses custom routes, controllers, and services to manage trips and bookings. The controller logic and service abstraction demonstrate the application of object-oriented principles. However, the artifact initially lacked robust error handling and did not implement full validation or user feedback for unsuccessful operations.

Artifact Enhancement:

Improvements include adding detailed form validation, better error messages, and a refactored controller structure to handle exceptions cleanly. These enhancements address the deficiencies by increasing usability and maintainability, aligning with software design best practices.

# 2. Algorithms and Data Structures

Artifact Description:

The artifact chosen is the custom trip filtering logic in the Travlr application that sorts and displays available travel packages based on date, destination, and cost using sorting and filtering algorithms.

Artifact Analysis:

This section of the code uses arrays, filters, and custom sort functions in JavaScript to dynamically manage trip listings. Initially, the sorting algorithm lacked optimization and failed to handle edge cases like empty fields or date mismatches.

Artifact Enhancement:

Enhancements included implementing stable sort functions and conditional filters to handle missing or incorrect input. This improved performance and made the logic more robust and user-friendly. These changes directly address the identified algorithmic flaws.

# 3. Databases

Artifact Description:

The artifact is the MongoDB data model for storing trip information, users, and bookings. It’s a NoSQL schema design using Mongoose within the Node.js backend of the Travlr application.

Artifact Analysis:

Custom Mongoose schemas define relationships between users and bookings. However, the original version did not include proper indexing, and the schema lacked validation on several fields, increasing the chance of bad data entries.

Artifact Enhancement:

Enhancements included adding field validation, unique constraints, and indexing on frequently queried fields (like trip date and user ID). These changes improve database efficiency, integrity, and security — directly fixing the deficiencies found in the analysis.