**Narrative #1 – Software Design & Engineering**

**Artifact:** Travlr Getaways (Full-Stack Web Application)  
**Original Course:** CS 465 – Full Stack Development

**Description of the Original Artifact:**  
Travlr Getaways was a full-stack travel booking web application that I originally built using Node.js, Express, and MongoDB. The app exposed routes for trips, meals, and rooms, and used a static HTML front end to render pages for users. The original version had working CRUD routes but lacked strong organization and security practices.

**Enhancements Implemented:**  
For this enhancement, I refactored the backend structure to follow the MVC pattern using clear separation between controllers, models, and routes. I implemented environment variables through a .env file to secure sensitive information such as the MongoDB URI and JWT secret. I also improved error handling and organized static file serving so the front-end pages loaded dynamically from the /public folder.

**Skills Demonstrated:**  
These improvements highlight my understanding of scalable web architecture, modular design, and software engineering principles such as separation of concerns. This enhancement also demonstrates best practices for version control, deployment preparation, and secure server configuration.

**Program Outcomes Addressed:**

* Design and develop professional-quality software solutions.
* Apply engineering principles to produce efficient, secure, and maintainable applications.
* Employ industry standards for organization and modularity in codebases.

**Narrative #2 – Algorithms & Data Structures**

**Artifact:** Travlr Getaways Backend Logic  
**Original Course:** CS 310 – Software Development Lifecycle

**Description of the Original Artifact:**  
The backend originally handled trip listings and user data through basic Mongoose queries without efficient sorting or filtering. The logic was functional but not optimized for real-world scalability.

**Enhancements Implemented:**  
I enhanced the algorithmic logic by introducing query filtering and sorting within the trip controller (Trip.find(filter).sort({ startDate: 1 })). I also refactored loops into reusable functions, improved conditional checks, and streamlined response handling for better performance.

These changes optimized both the data retrieval and computation of filtered queries, reducing redundancy and making the code easier to maintain.

**Skills Demonstrated:**  
This enhancement demonstrates algorithmic thinking, efficient use of data structures in JavaScript, and the ability to implement optimized database queries using Mongoose.

**Program Outcomes Addressed:**

* Analyze complex computing problems and implement efficient solutions.
* Design modular algorithms and data manipulation routines.
* Optimize code for scalability and maintainability.

**Narrative #3 – Databases**

**Artifact:** Travlr Getaways Database Layer (Mongoose Schemas)  
**Original Course:** CS 340 – Client/Server Development

**Description of the Original Artifact:**  
The initial Travlr database used minimal schema validation. User and trip documents could be created with incomplete or incorrect data. There were no constraints or relationships defined between entities.

**Enhancements Implemented:**  
I introduced full validation and schema constraints for both user.js and trip.js. The User model now validates emails with regex and enforces required fields such as username and password. The Trip schema enforces required fields (destination, startDate, endDate, price) and validates numerical and date formats. I also improved seeding scripts to automatically insert consistent demo data and clear the database on reset.

**Skills Demonstrated:**  
These improvements show my ability to design secure, consistent, and validated data structures that prevent errors and maintain data integrity.

**Program Outcomes Addressed:**

* Design and manage data structures that support reliable and secure operations.
* Apply database normalization and validation techniques.
* Develop a security-focused mindset that protects data privacy.