**Merrik Wright**   
**CS499**  
  
  
Algorithms & Data Structure: Status Page and Encryption Enhancement

Artifact Description

The artifact enhanced for this milestone is the encryption and configuration subsystem of the TRAVLR full-stack web application. TRAVLR is a Node.js/Express-based portfolio project developed during my Computer Science program. In this milestone, I expanded the backend and frontend functionality to include a dynamic status reporting page and improved encryption utilities. These updates were added in July 2025 in week 4 to meet Milestone Three’s requirement for demonstrating algorithmic thinking and structured programming enhancements. The date can be found in the file system on the Github.[Hearin1/CS499 at TRAVLR-Folder-for-review](https://github.com/Hearin1/CS499/tree/TRAVLR-Folder-for-review)

Justification for Inclusion

This artifact was selected for my ePortfolio because it demonstrates critical skills in applying algorithms, managing structured data, and securing user data through encryption. Specifically:

* I implemented enhanced password encryption in util/encryption.js, replacing a static salt with random per-user salts generated using crypto.randomBytes(). This enhances password security by ensuring each hash is unique even for identical inputs.
* I added support for an optional admin bypass mechanism within the same module, which can securely override encryption requirements for testing or administrative functions. This required careful conditional logic to maintain security without compromising core integrity.
* I created a new Status Page, served at /status, which reads checkpoint data from data/status.json and renders it into a Handlebars view. This involved implementing data parsing, safe rendering, and layout integration.
* I registered the new /status route in the application router and enabled CORS support to allow the endpoint to be accessed from multiple contexts—extending the API’s interoperability.

These enhancements required thinking in terms of both algorithms (e.g., random byte generation, secure key derivation, optional logic paths) and data structures (e.g., JSON schema for checkpoint metadata, salt/IV formatting, and frontend templating).

Alignment with Course Outcomes

This work supports several Computer Science program outcomes, notably:

* Design and evaluate computing solutions using algorithmic principles. I applied cryptographic key derivation and data formatting patterns to structure encrypted outputs using salt|iv|ciphertext notation, improving clarity and reusability.
* Implement secure software design principles. By replacing static salts and introducing flexible override logic, the new encryption design anticipates security risks and mitigates vulnerabilities.
* Communicate technical work effectively. The README was updated to explain the purpose of the /status endpoint and the encryption module’s new capabilities, supporting internal and external developer audiences.

Reflection on the Enhancement Process

Enhancing the encryption and configuration features pushed me to apply foundational algorithm knowledge in a practical, high-impact way. One challenge was working within an isolated development environment that could not run npm install or download dependencies, resulting in blocked tests and server runs. These limitations were documented, and the project remains deployable in a typical Node.js setup.

I also had to decide how to securely store and load structured checkpoint data for the status page. I used a simple JSON schema for data/status.json, which was then parsed and safely rendered into a view using Handlebars. This reinforced my understanding of data serialization and the Model-View-Controller (MVC) pattern.

Overall, this milestone demonstrated my ability to write maintainable code that follows secure practices and uses algorithmic reasoning. It shows not only my technical growth in security and data modeling, but also my ability to work through constraints and communicate implementation details clearly. I will be enhancing my entire project significantly, as I will be gone during week 6. I will have everything I need turned in on week 5.

