# CSE 340 – Outcome Mastery Report

Name: Mozart Armstrong Bandeira Soares

GitHub: URL: https://github.com/mozartsoares/cse340

Deployed website URL: https://cse340-b79d.onrender.com

## Instructions:

*For each of the course outcomes listed below, include 2-3 examples of the way you have demonstrated this outcome. Each example should include:*

* *A reference to the specific file/function you have created.*
* *A 2-3 sentence description of what you did.*

Outcome 1: Develop to current web frontend standards of validity and practice.

Example 1: Reviews Interface

File/Function: views/account/reviews.ejs, public/css/styles.css

Description: Implemented a responsive and accessible reviews interface using modern HTML5 and CSS3 standards. The interface includes semantic HTML elements and follows BEM naming conventions for CSS classes.

Example 2: Vehicle Details Page

File/Function: utilities/index.js **(buildInventoryItemPage function)**

Description: Created a modern, responsive vehicle details page that adapts to different screen sizes. Implemented proper image handling with alt texts and semantic HTML structure.

Outcome 2: Use variables, arrays, functions, and control structures in server code.

Example 1: Review Management

File/Function: controllers/reviewController.js **(getUserReviews, updateReview functions)**

Description: Implemented controller functions using async/await patterns and proper error handling. Used array methods to process review data and conditional logic to handle different user states.

Example 2: Navigation Building

File/Function: utilities/index.js **(getNav function)**

Description: Created a dynamic navigation system that uses array methods to process classification data and build HTML structure. Implements proper error handling and async operations.

Outcome 3: Develop web applications that implement common design patterns.

Example 1: MVC Architecture

File/Function: models/review-model.js, controllers/reviewController.js, views/account/reviews.ejs

Description: Implemented the MVC pattern for the reviews system. Separated data **(model)**, presentation **(view)**, and business logic **(controller)** concerns.

Example 2: Middleware Pattern

File/Function: utilities/index.js **(checkLogin, handleErrors functions)**

Description: Created middleware functions for authentication and error handling. Follows the Express.js middleware pattern for request processing.

Outcome 4: Design and use relational databases for CRUD interactions.

Example 1: Review Management

File/Function: models/review-model.js **(addReview, updateReview, deleteReview functions)**

Description: Implemented complete CRUD operations for user reviews. Used parameterized queries for security and proper database relationships between reviews, users, and vehicles.

Example 2: Vehicle Inventory

File/Function: models/inventory-model.js

Description: Created database interactions for vehicle inventory management. Implemented joins between classifications and inventory tables with proper foreign key relationships.

Outcome 5: Validate data **(client-side and server-side)** appropriate to the task.

Example 1: Review Validation

File/Function: public/js/user-reviews.js, controllers/reviewController.js

Description: Implemented both client-side validation for review text and server-side validation for user authentication and data integrity. Includes proper error handling and user feedback.

Example 2: Account Management

File/Function: utilities/account-validation.js

Description: Created comprehensive validation rules for account registration and updates. Includes both client-side and server-side validation with proper error messaging.

Outcome 6: Demonstrate the skills of a productive team member **(such as solving problems, collaborating with others, communicating clearly, fulfilling assignments, and meeting deadlines.)**

Example 1: Code Organization

File/Function: Entire project structure

Description: Maintained consistent code formatting and clear documentation throughout the project. Followed established naming conventions and project structure patterns. This made the project much easier for other team members to understand and help me when iIasked

Example 2: Helping others

File/Function: Entire Project Structure

Description: Helped other team members with a teach to learn approach specially in the first weeks of the course, I helped them to understand the backend concepts with my previous knowledge, this helped me to learn a lot more while also helping other to learn too