# CSE 340 – Outcome Mastery Report

Name: [ANYALECHI CHIDIEBERE PRINCEDANIEL ]

GitHub: URL: [https://github.com/qwes3d/cse-motors]

Deployed website URL: [https://cse-motors-qboi.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. Example: Navigation Bar with Dynamic User

File/Function: views/partials/header.ejs

Description: I created a dynamic navigation bar that displays login or logout options based on user authentication status. It uses EJS templating to inject user data and follows semantic HTML5 structure with accessible tags and responsive design.

2. Example: Support Ticket Form Page

**File/Function**: views/my-tickets.ejs

**Description:** I implemented a valid HTML form using semantic tags and best practices for accessibility. The form includes proper labeling, a textarea for message input, and a submit button, meeting current frontend standards.

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

**Example:**

1. Example: JWT Authentication Middleware

**File/Function**: middleware/authMiddleware.js

**Description:** I used control structures (try/catch, if statements) and variables to verify JWT tokens, check account types, and handle errors. The middleware prevents unauthorized access to client-only routes.

2. Example: Fetching Support Tickets

**File/Function**: routes/supportRoute.js, GET /support/my-tickets

**Description:** I used query variables and conditional logic to retrieve support tickets from the database based on the logged-in user's ID. Results are returned as an array and passed into the EJS view for render

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

**Example:**

1. Example: MVC Pattern Implementation

**File/Function**: controllers/supportController.js, routes/supportRoute.js, views/my-tickets.ejs

**Description**: I followed the MVC architecture pattern by separating business logic into controllers, routing logic into route files, and the UI into EJS views. This improves maintainability and scalability.

2. Example: DRY Navigation Bar

**File/Function: views/partials/header.ejs**

**Description**: I used a partial layout for the navigation bar that is included in every page. This follows the DRY (Don't Repeat Yourself) design principle, allowing consistent updates across pages.

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

**Example:**

1. Example: Submit Support Ticket (Create)

**File/Function**: routes/supportRoute.js, POST /support/my-tickets

**Description**: I created a form submission route that inserts a new ticket into the support\_ticket table. It uses parameterized queries to prevent SQL injection and handle data properly.

2. Example: Fetching Tickets (Read)

**File/Function**: routes/supportRoute.js, GET /support/my-tickets

**Description:** I queried the support\_ticket table to retrieve all tickets for the logged-in user. The data is ordered by created\_at to show the most recent tickets first.

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

**Example:**

1. Example: HTML5 Form Validation

**File/Function:** views/my-tickets.ejs

**Description:** I used the required attribute in the support ticket form fields to ensure the user does not submit empty messages. This provides instant client-side feedback.

2. Example: Server-side Ticket Validation

**File/Function**: routes/supportRoute.js, POST /support/my-tickets

**Description**: I used destructuring and conditionals to validate input data before saving it to the database. This ensures integrity and prevents null or malicious inputs.

Outcome 6: Demonstrate the skills of a productive team member.

**Example:**

1. Example: GitHub Collaboration and Trello Planning

**File/Function:** GitHub repo + team/trello-board

**Description**: I collaborated using GitHub branches and Trello task cards to divide work among team members. I clearly communicated changes during code reviews and helped debug teammate issues.

2. Example: Support Ticket Feature Assignment

**File/Function:** routes/supportRoute.js, views/my-tickets.ejs

**Description**: I was responsible for implementing the entire support ticket submission and viewing feature. I met the project deadline and kept documentation updated for teammates to review.