EE 547-QL Inter/Intranet App Development

# SUMMER 2023 Project #4

# MySQL and the Express Server

In this report, I will outline the steps I followed to create a Node.js Express web server that performs CRUD operations on a MySQL database. I will also discuss the challenges I encountered during the process and how I overcame them. Finally, I will demonstrate the functionality of the server by performing CRUD operations through the web interface.

Step 1: Set up the Development Environment

To begin the project, I first ensured that I had the necessary tools installed:

• Installed Docker to facilitate containerization and environment management.

• Installed Node.js and npm to enable JavaScript runtime and package management.

Step 2: Creating a Node.js Express Web Server

a. Initializing the Project: I initiated a new Node.js project using the npm init command. This allowed me to set up the project's basic configuration.

b. Installing Dependencies: I used npm to install the required dependencies: Express for building the web server and mysql2 for MySQL database connectivity.

c. Setting Up the Web Server: In the index.js file, I set up an Express web server. I defined routes for each CRUD operation (Create, Read, Update, Delete) to handle user data.

d. Challenges and Solutions:

Challenge: One challenge I encountered was handling asynchronous database operations within the Express routes. I needed to ensure that data operations completed before sending responses.

Solution: I used JavaScript's async/await syntax to handle asynchronous database calls. This allowed me to await the completion of database operations before proceeding.

Step 3: Setting Up a MySQL Database

a. Installing MySQL Client Packages: I installed the mysql2 package using npm to enable communication between Node.js and the MySQL database.

b. Configuring Database Connection: Within the index.js file, I configured the MySQL database connection by specifying the host, port, username, password, and database name.

c. Creating the Database and Table: I used SQL queries to create the necessary database and table for storing user data.

CREATE DATABASE myappdb;

USE myappdb;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

d. Challenges and Solutions: Challenge: One challenge I faced was setting up the database schema and handling potential errors during database operations.

Solution: I carefully designed the database schema and implemented error handling using try-catch blocks to gracefully manage database-related errors.