**Bài tập tổng hợp Express, Routing và Middleware**

## 1. Mục tiêu

* Hiểu được các kiến thức cơ bản về thư viện Express.js
* Hiểu và vận dụng được cách cài đặt môi trường phát triển Express.js
* Hiểu được khái niệm và nắm được cách routing nâng cao trong Express.js.
* Hiểu và nắm được cơ chế hoạt động của Middleware trong Express.js.
* Hiểu và nắm được cơ chế serving static file trong Express.js.
* Hiểu và nắm sự khác biệt giữa SSR (Server-side rendering) website vs CSR (Client-side rendering) website
* Hiểu được cách tạo một CSR dynamic website.

## 2. Mô tả

* Middleware là một thành phần đóng vai trò trung gian giữa request/response và xử lý logic bên trong web server.
* Middleware trong ExpressJS là các hàm được dùng để tiền xử lý, lọc các request trước khi đưa vào xử lý logic hoặc điều chỉnh các response trước khi gửi về cho người dùng.
* Việc sử dụng middleware sẽ giúp chương trình tinh gọn, dễ dàng tái sử dụng và mở rộng hơn.
* Application-level middleware (Middleware cấp ứng dụng)/Third-party middleware (middleware của bên thứ 3)
  + Nhìn vào ví dụ Bước 3/exercise 03 chúng ta đã cài đặt và require một module body-parser (Thư viện của bên thứ 3) và sử dụng (app.use()) trước khi thực hiện bất kỳ request nào.
  + Việc sử dụng này chính là sử dụng một Middleware cấp application-level để xử lý các request trước khi response về cho người dùng
* Router-level middleware (Middleware cấp router)
  + Là các callback function dạng

(req, res, next) => {

//Logic;

next();

}

* + Các callback function (middleware) sẽ xử lý các yêu cầu nhất định của bài toán về request trước khi đi vào callback function cuối cùng của route
  + Các đoạn logic trong các middleware này sẽ được thực thi, sau khi thực thi xong, hàm next sẽ được gọi để di chuyển tiếp đến các middleware tiếp theo
  + Bởi vì hàm next có thể di chuyển đến các middleware tiếp theo, vậy nên chúng ta có thể lồng bao nhiêu middleware xử lý các logic khác nhau trong route cũng được

VD: app.get(“/”, authMiddleware, checkRoleMiddleware, … (req, res) => {})

* Viết một middleware tên là **checkExist**
  + Middleware này sẽ được truyền vào toàn các Route (GET:id/POST/PUT/DELETE) ở exercise 03
  + Middleware này có tác dụng check nếu question có tồn tại hay không theo cả id và content
  + Đối với trường hợp id
    - Nếu không tìm thấy question → trả về cho người dùng dưới dạng json message {message: “Question not found” } → res.status(code).json(message)
    - Nếu tìm thấy question → gọi hàm next() để chạy đến callback function cuối cùng
  + Đối với trường hợp content
    - Nếu tìm thấy question → trả về cho người dùng dưới dạng json message {message: “Question already exists” } → res.status(code).json(message)
    - Nếu không tìm thấy question → gọi hàm next() để chạy đến callback function cuối cùng
* **NHỚ XÓA CODE CHECK QUESTION Ở CÁC ROUTE TRÊN ĐI ĐỂ SỬ DỤNG MIDDLEWARE**

## 3. Đánh giá

Để hoàn thành bài tập, học viên cần:

* Đưa mã nguồn lên GitHub.
* Dán link của repository lên phần nộp bài trên hệ thống.