# LẬP TRÌNH WEB NÂNG CAO - 503106

# BÀI TẬP THỰC HÀNH TUẦN 8 + 9

Giảng viên biên soạn: ThS. Mai Văn Mạnh

## MỤC TIÊU BÀI THỰC HÀNH

1. Tạo Rest Api với ExpressJS.
2. Tìm hiểu khái niệm Cross-Origin Resource Sharing (CORS).
3. Tìm hiểu khái niệm Json Web Token (JWT) và ứng dụng để xác thực người dùng.
4. Lưu trữ dữ liệu vào MongoDB sử dụng Mongoose.
5. Tìm hiểu khái niệm Router trong ExpressJS.
6. Ôn lại kiến thức về mô hình MVC.

## ĐỀ BÀI

Xây dựng rest api dùng để quản lý thông tin tài khoản, quản lý sản phẩm và các đơn hàng. Thông tin chi tiết như sau:

- Tài khoản người dùng có các thông tin: email, mật khẩu.
- Sản phẩm có các thông tin: mã số, tên sản phẩm, giá, ảnh minh họa, mô tả.
- Đơn hàng có các thông tin: mã đơn hàng, tổng giá bán, và danh sách sản phẩm. Mỗi phần tử trong danh sách sản phẩm của một đơn hàng đều có các thông tin như mã sản phẩm, số lượng, giá.

Rest api cung cấp hai endpoints /products và /orders hỗ trợ nhiều phương thức HTTP khác nhau để thực hiện các chức năng thêm, cập nhật, xóa, lấy danh sách và lấy chi tiết một sản phẩm/đơn hàng. Ngoài ra còn có api /account/register và /account/login để quản lý việc đăng ký tài khoản và đăng nhập. Một số api endpoint đòi hỏi người dùng cần phải đăng nhập thì

mới có truy cập được. Tham số đầu vào cho tất cả các endpoints đều có định dạng là JSON.

Mô tả chi tiết về các api endpoints như sau:

- http://localhost/api/account/register:
    - **POST**: đăng ký một tài khoản mới
- http://localhost/api/account/login:
    - **POST**: người dùng đăng nhập
- http://localhost/api/products:
    - **GET**: Trả về danh sách tất cả sản phẩm trong hệ thống
    - **POST**: Thêm một sản phẩm mới
- http://localhost/api/products/{id}
    - **GET**: Trả về thông tin chi tiết của một sản phẩm dựa theo id.
    - **PUT**: Cập nhật thông tin của một sản phẩm dựa theo id.
    - **DELETE**: Xóa sản phẩm dựa theo id.
- http://localhost/api/orders:
    - **GET**: Trả về danh sách tất cả các đơn hàng.
    - **POST**: Thêm một đơn hàng mới.
- http://localhost/api/orders/{id}:
    - **GET**: Trả về thông tin chi tiết của một đơn hàng dựa theo id.
    - **PUT**: Cập nhật thông tin của một đơn hàng dựa theo id.
    - **DELETE**: Xóa đơn hàng dựa theo id.

Với các api endpoint có method được tô màu đỏ thì người dùng cần phải đăng nhập mới có thể truy cập được.

## CÁC YÊU CẦU KHÁC

- Tất cả dữ liệu về tài khoản, sản phẩm, đơn hàng cần được lưu trong MongoDB và được kết nối từ NodeJS thông qua Mongoose modules.
- Sử dụng Express Router để hiện thực chức năng cho các route riêng biệt (ví dụ AccountRouter, ProductRouter, OrderRouter).
- Thiết lập Cross-Origin Resource Sharing để cho phép mọi web client dù khác tên miền vẫn có thể truy cập và tương tác được với rest api.
- Sử dụng bcrypt để hash mật khẩu, sử dụng JWT để xác thực người dùng.
- Tiếp tục sử dụng các modules đã giới thiệu trong các bài học trước vào bài tập, chẳng hạn như:
    - Sử dụng express-form, express-validator để kiểm tra dữ liệu từ HTML form.
    - Sử dụng multer để xử lý upload tập tin.
    - Sử dụng tính năng rate limit trong express để phòng chống tấn công DDOS.
- Cần có cơ chế xử lý lỗi và trả về thông báo lỗi phù hợp: ví dụ lỗi thiếu thông tin, thông tin không đúng định dạng, tập tin upload có kích thước quá lớn, lỗi endpoint không hợp lệ, phương thức (http method) không được hỗ trợ…