

LẬP TRÌNH WEB NÂNG CAO - 503106

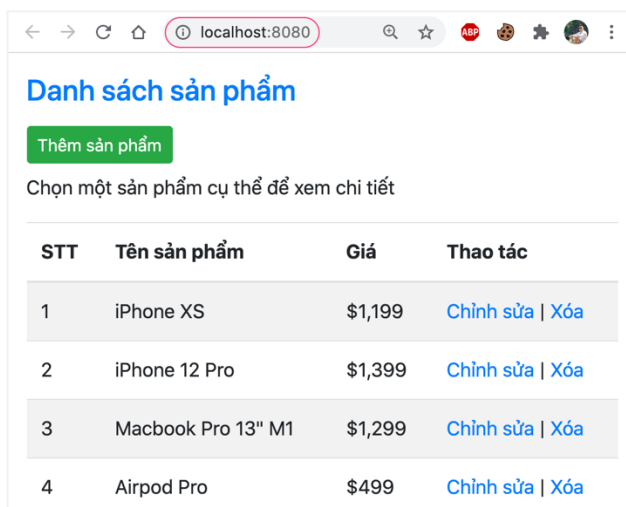
BÀI TẬP THỰC HÀNH TUẦN 3 + 4

Giảng viên biên soạn: ThS. Mai Văn Mạnh

MỤC TIÊU BÀI THỰC HÀNH:

1. Tạo trang web bằng nodejs sử dụng [express](#) framework.
2. Tìm hiểu khái niệm view engine và sử dụng các view engine trong project như [ejs](#).
3. Sử dụng công cụ [npm](#) để cài đặt thêm các module bên ngoài.
4. Tìm hiểu khái niệm [middleware](#) trong Express JS.
5. Làm quen với các modules hỗ trợ xử lý HTML form như: [body-parser](#), [multer](#), [cookie-parser](#), [express-session](#), ...
6. Xử lý dữ liệu [HTML form](#) được gửi dưới nhiều định dạng khác nhau như: [x-www-form-urlencoded](#), [multipart/form-data](#), [json](#).
7. Truyền dữ liệu cho ứng dụng web từ [environment variables](#).
8. Tổng hợp các kiến thức trên để làm một bài tập quản lý sản phẩm đơn giản.

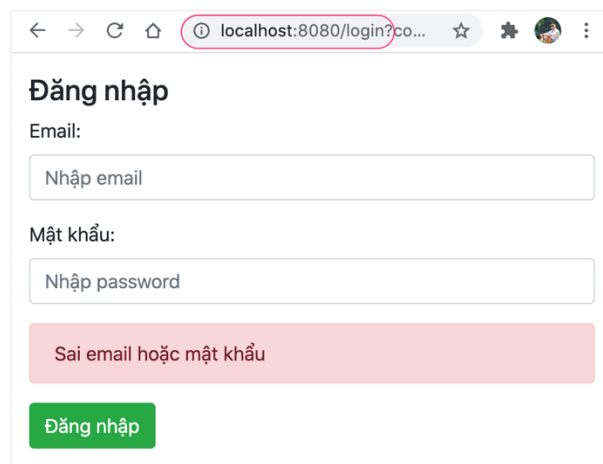
BÀI TẬP. Xây dựng trang web thực hiện chức năng quản lý sản phẩm cơ bản như sau:



STT	Tên sản phẩm	Giá	Thao tác
1	iPhone XS	\$1,199	Chỉnh sửa Xóa
2	iPhone 12 Pro	\$1,399	Chỉnh sửa Xóa
3	Macbook Pro 13" M1	\$1,299	Chỉnh sửa Xóa
4	Airpod Pro	\$499	Chỉnh sửa Xóa

Hình 1. Giao diện trang chủ hiện thị danh sách sản phẩm

- **Trang đăng nhập (/login):** Yêu cầu user phải đăng nhập thì mới có thể truy cập vào tất cả các nội dung trên website, nếu truy cập vào bất cứ nội dung nào (như mô tả bên dưới) mà chưa đăng nhập thì sẽ chuyển hướng đến trang đăng nhập
 - o **GET:** hiển thị giao diện đăng nhập.
 - o **POST:** xử lý đăng nhập và hiển thị lỗi nếu có. Thông tin tài khoản cần đọc từ **environment variables** (với username và password do sinh viên tự tạo). Nếu đăng nhập thành công thì chuyển hướng về trang chủ.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/login?co...'. The page title is 'Đăng nhập'. It contains two input fields: 'Email:' with a placeholder 'Nhập email' and 'Mật khẩu:' with a placeholder 'Nhập password'. Below these fields is a red error message box that says 'Sai email hoặc mật khẩu'. At the bottom is a green button labeled 'Đăng nhập'.

Hình 2. Giao diện đăng nhập

- **Trang chủ:** Hiển thị danh sách sản phẩm trên một bảng. Nếu người dùng chưa đăng nhập thì tự động chuyển hướng sang trang đăng nhập.
- **Trang chi tiết sản phẩm (/id):** hiển thị thông tin chi tiết của một sản phẩm. Một sản phẩm có các thông tin: mã sản phẩm, tên sản phẩm, giá bán, ảnh minh họa và thông tin mô tả. Nếu **id** của sản phẩm không hợp lệ thì cần chuyển trang hoặc hiển thị thông báo lỗi phù hợp.
- **Trang thêm sản phẩm (/add):**
 - o **GET:** hiển thị giao diện HTML form để nhập một sản phẩm mới. Cần có chức năng upload ảnh minh họa sản phẩm.
 - o **POST:** xử lý chức năng thêm sản phẩm và hiển thị lỗi nếu có. Nếu thêm thành công thì chuyển hướng về trang chủ đồng thời hiển thị **toast message** cho biết sản phẩm đã được thêm thành công.

Quay lại

Thêm sản phẩm mới

Tên sản phẩm

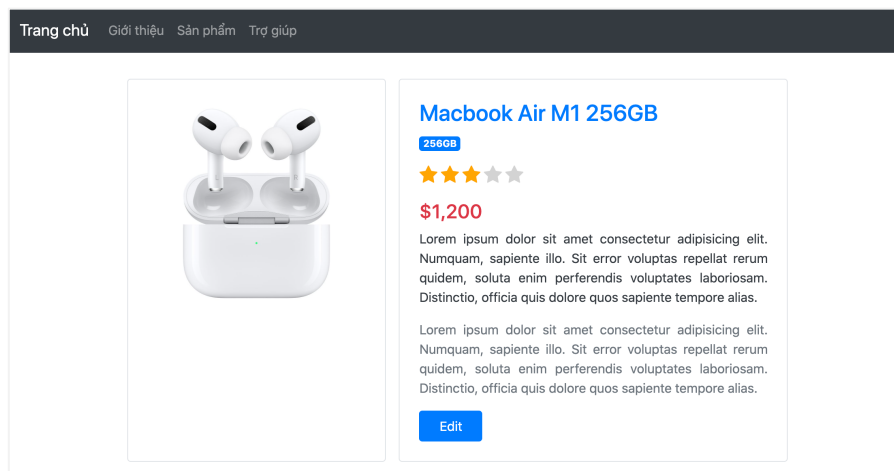
Giá sản phẩm

Ảnh sản phẩm

Mô tả

Vui lòng nhập đầy đủ thông tin

Hình 3. Giao diện thêm một sản phẩm mới



Hình 4. Giao diện trang chi tiết của một sản phẩm

- **Trang chỉnh sửa sản phẩm (/edit/{id}):**
 - o **GET:** hiển thị thông tin sản phẩm có **id** tương ứng để điều chỉnh/cập nhật các thông tin cần thiết. Nếu **id** của sản phẩm không hợp lệ thì cần chuyển trang hoặc hiển thị thông báo lỗi phù hợp.

- **POST**: xử lý cập nhật sản phẩm và hiển thị lỗi nếu có. Nếu cập nhật sản phẩm thành công thì chuyển hướng về trang chủ kèm theo toast message để cho biết sản phẩm đã được cập nhật thành công.
- **Trang xóa sản phẩm (/delete)**:
 - **POST**: Xóa một sản phẩm dựa trên **id** của sản phẩm. Trước khi xóa cần hiển thị một **modal dialog** để người dùng xác nhận. Sau khi xóa xong thì hiển thị toast message để cho người dùng biết kết quả xóa (thành công/thất bại). Nên sử dụng ajax hoặc fetch API để xóa sản phẩm mà không cần tải lại trang web.



Hình 5. Confirm dialog hiển thị trước khi xóa một sản phẩm

Các yêu cầu khác:

- Sử dụng tính năng **rate limit** để phòng chống tấn công DDOS.
- Nếu người dùng truy cập vào bất cứ path nào khác mô tả ở trên thì chuyển hướng đến đường dẫn **/error** và hiển thị giao diện thông báo lỗi 404 not found.
- Sử dụng **express js** để tạo ứng dụng web.
- Chọn một view engine để tạo giao diện cho trang web, gợi ý dùng **ejs**.
- Tất cả cá dữ liệu form cần được kiểm tra bằng cách sử dụng các middleware như **express-form**, **express-validator**...
- Dữ liệu chỉ cần lưu trong bộ nhớ, chưa cần sử dụng các kỹ thuật lưu trữ như file, db.
- Tài khoản đăng nhập cần được cấu hình thông qua các **environment variables**.
- Cần giới hạn định dạng và kích thước của tập tin khi upload.

- Đề bài chỉ cung cấp một vài giao diện mẫu, các giao diện còn lại sinh viên tự thiết kế dựa theo yêu cầu, nên sử dụng [bootstrap](#) hoặc các library tương tự để làm giao diện đẹp và responsive.