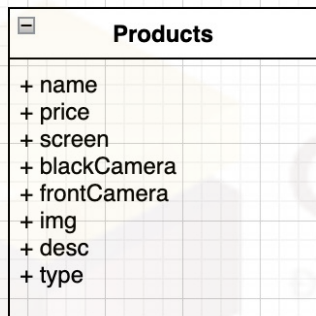


# Bài tập lớn JS

Phần 1: Tạo 1 website bán hàng với các chức năng sau đây

1. Tự tạo api Product bằng MockAPI.
  - Link data <https://tinyurl.com/yy8thsk4>
  - Hướng dẫn tạo MockAPI: <https://tinyurl.com/azdnjs3j>
2. Xây dựng lớp đối tượng Products.



Products
+ name
+ price
+ screen
+ blackCamera
+ frontCamera
+ img
+ desc
+ type

3. Hiển thị danh sách sản phẩm cho khách hàng.

**Gợi ý:** tạo một mảng ProductList ngoài global, call axios lấy danh sách sản phẩm từ database, gán mảng sản phẩm trả về vào biến ProductList, viết hàm tạo giao diện. Hàm tạo giao diện giống y chang hàm tạo bảng, chỉ khác thay vì tạo <tr></tr> thì giờ tạo cái <div> cho từng sản phẩm

# Bài tập lớn JS

4. Tạo một ô select cho phép người dùng filter theo loại sản phẩm, ô select có 2 option là samsung và iphone, viết hàm gắn vào sự kiện onChange của select

**Gợi ý:** dựa vào thuộc tính **type** của đối tượng sản phẩm, có 2 loại: samsung hoặc iphone. ví dụ: nếu người dùng chọn samsung, tiến hành duyệt vòng for, lọc ra các sản phẩm có type === "samsung", bỏ vào 1 mảng, sau đó gọi hàm tạo giao diện động dựa theo mảng vừa tìm được

5. Cho phép người dùng chọn sản phẩm bỏ vào giỏ hàng

Gợi ý: - tạo một mảng giỏ hàng - cart (biến global), mảng cart sẽ chứa các đối tượng cartItem

6. Không push trực tiếp sản phẩm được chọn vào mảng cart vì không có số lượng, tạo một đối tượng mới có định dạng sau rồi push vào mảng cart

var cartItem = {product: {đối tượng sản phẩm cần push}, quantity: 'bla bla'}.

Vd : var cartItem = {  
    product: {id: 1, price: 10, name : 'samsung a10' },  
    quantity: 1  
}

Khi người dùng click nút ADD TO CART, tiến hành truyền vào id, tìm sản phẩm trong ProductList, ví dụ tìm được index = 2, thì lưu đối tượng ProductList[2] vào thuộc tính product của cartItem. Sau đó, push cartItem vào mảng giỏ hàng.

# Bài tập lớn JS

7. Khi thêm sản phẩm vào giỏ hàng, nếu sản phẩm chưa có trong giỏ hàng thì push vào cart với quantity là 1, nếu đã có rồi thì ko push nữa mà chỉ tăng quantity lên 1 đơn vị
8. In giỏ hàng ra màn hình, viết hàm renderCart và duyệt mảng giỏ hàng, có bao nhiêu sản phẩm thì tạo ra bấy nhiêu <tr> tương ứng.
9. Trong giao diện giỏ hàng, cho phép người dùng chỉnh sửa số lượng (gợi ý: cho 2 nút tăng giảm), viết hàm gắn vào 2 nút đó, khi nhấn vào thì truyền vào id, tìm trong mảng giỏ hàng xem sản phẩm đó nằm ở đâu, lấy quantity ra tăng hoặc giảm , sau đó render lại giao diện
10. In tổng tiền ra giao diện. Trong hàm renderCart, tính tổng giá tiền của tất cả sản phẩm và hiện ra (giá tiền \* số lượng)
11. Lưu giỏ hàng vào localStorage , lần sau khi vào trang sẽ load lên lại.
12. Người dùng nhấn nút thanh toán, clear giỏ hàng, set mảng giỏ hàng về mảng rỗng []
13. Cho phép người dùng remove sản phẩm ra khỏi giỏ hàng

**Giao diện gợi ý:** học viên tự làm giao diện riêng dựa trên giao diện tham khảo

<https://cyber-shopping-cart-js.vercel.app/>

Trang web tham khảo theme: <https://themeforest.net/category/ecommerce>

# Bài tập lớn JS

## Phần 2: Phần quản trị (tạo 1 source code riêng để làm)

1. Sử dụng Axios để xây dựng các chức năng cho admin (hiện danh sách, thêm, xóa, cập nhật sản phẩm)
2. Kiểm tra validation

**Giao diện gợi ý:** học viên tự làm giao diện riêng dựa trên giao diện tham khảo

<https://cyber-qlsp.vercel.app/>

Trang web tham khảo theme: <https://themeforest.net/category/site-templates/admin-templates>

## Phần 3: Khi nộp bài tập lớn và project cuối khoá, học viên cần nộp:

- Link github chứa source code
- Link online của dự án (đưa web lên host free)
- Link youtube quay demo dự án.