

Bài tập Capstone JS

Yêu cầu:

- Tạo website bán hàng với các chức năng mô tả ở các slide sau.
- Tạo trang quản trị để quản lý sản phẩm
- Xây dựng giao diện cho trang bán hàng và quản trị bằng SASS , có xử lý responsive.
- Xử lý call API để tương tác lấy và truyền dữ liệu với Back-End
- Tổ chức xây dựng các lớp đối tượng

Bài tập Capstone JS

Yêu cầu:

- Khi nộp bài các nhóm nhớ nộp kèm:
 - Link github chứa source code
 - Link online của dự án (đưa web lên host free: github, vercel, netlify)
 - Link youtube quay demo dự án.
 - File phân chia công việc trong nhóm (File tham khảo:
<https://tinyurl.com/3exwu438>) – lưu chung github với source code

Bài tập Capstone JS

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

1. 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: Call api lấy danh sách sản phẩm từ database BE. Viết hàm tạo giao diện cách tạo gần giống hàm tạo table đã học ở trên lớp, chỉ khác thay vì tạo `<tr></tr>` thì giờ tạo cái `<div>` cho từng sản phẩm

Bài tập lớn JS

4. Tạo một dropdown (select) cho phép người dùng filter lọc hiển thị danh sách sản phẩm theo loại sản phẩm, dropdown 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 mảng sản phẩm, lọc ra các sản phẩm có type === "samsung", lưu vào 1 mảng kết quả. Sau đó gọi hàm tạo giao diện và truyền mảng kết quả vào hàm để tạo UI

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

Bài tập lớn JS

6. Không thêm trực tiếp sản phẩm được chọn vào mảng cart vì không có số lượng , tạo một đối tượng mới dành riêng cho giỏ hàng rồi push vào mảng cart

Sơ đồ lớp gợi ý:

https://drive.google.com/file/d/1gYU_PkQ-OYycHyHkzF8J4oUzqgtaSvBS/view?usp=sharing

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.

Bài tập lớn JS

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

Bài tập lớn JS

Phần 2: Trang quản trị

1. Sử dụng Axios để call API, xây dựng các chức năng cho admin
 - Hiện danh sách sản phẩm,
 - Thêm sản phẩm
 - Xóa sản phẩm
 - Cập nhật sản phẩm
2. Kiểm tra validation
3. Tìm kiếm sản phẩm theo tên
4. Sắp xếp sản phẩm theo giá tiền (từ lớn đến bé, và ngược lại)