Đoạn mã này chứa hai hàm JavaScript bất đồng bộ (**addCart** và **updateCart**) để tương tác với API của server thông qua các yêu cầu HTTP (**fetch**). Dưới đây là phân tích chi tiết từng phần:

**1. Hàm addCart(id, quantity)**

**Mục đích**

* Gửi yêu cầu thêm sản phẩm vào giỏ hàng.

**Cách hoạt động**

let res = await fetch('/products//cart', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

'Accept': 'application/json'

},

body: JSON.stringify({id, quantity})

});

1. **fetch**:
   * Gửi yêu cầu HTTP **POST** đến API /products//cart (có lỗi dư / sau products cần được sửa thành /products/cart).
   * **headers**:
     + **Content-Type: application/json**: Định nghĩa kiểu dữ liệu gửi lên server là JSON.
     + **Accept: application/json**: Yêu cầu server trả về dữ liệu dạng JSON.
   * **body**: Chứa dữ liệu id và quantity của sản phẩm, được chuyển thành chuỗi JSON.
2. **Kết quả trả về**:
3. let json = await res.json();
4. document.getElementById('cart-quantity').innerText = `(${json.quantity})`;
   * Chuyển đổi phản hồi từ server thành JSON.
   * Cập nhật giao diện:
     + **cart-quantity**: Hiển thị tổng số lượng sản phẩm trong giỏ hàng bằng giá trị json.quantity.

**2. Hàm updateCart(id, quantity)**

**Mục đích**

* Gửi yêu cầu cập nhật số lượng sản phẩm trong giỏ hàng.

**Cách hoạt động**

if(quantity > 0){

let res = await fetch('/products/cart', {

method: 'PUT',

headers: {

'Content-Type': 'application/json',

'Accept': 'application/json'

},

body: JSON.stringify({id, quantity})

});

1. **Kiểm tra số lượng (if (quantity > 0))**:
   * Chỉ thực hiện cập nhật nếu quantity lớn hơn 0 (tránh cập nhật số lượng âm hoặc bằng 0).
2. **Gửi yêu cầu HTTP PUT**:
   * **URL**: /products/cart.
   * **method: 'PUT'**: Chỉ định đây là một yêu cầu cập nhật.
   * **body**: Gửi id và quantity của sản phẩm trong giỏ hàng.
3. **Xử lý phản hồi**
4. if(res.status == 200){
5. let json = await res.json();
6. document.getElementById('cart-quantity').innerText = `(${json.quantity})`;
7. document.getElementById('subtotal').innerText = `(${json.subtotal})`;
8. document.getElementById('total${id}').innerText = `($${json.item.total})`;
9. }
   * **Kiểm tra trạng thái phản hồi (status == 200)**:
     + Nếu yêu cầu thành công:
       - **json.quantity**: Tổng số lượng sản phẩm trong giỏ hàng.
       - **json.subtotal**: Tổng giá trị tạm tính của giỏ hàng.
       - **json.item.total**: Giá trị tổng cho sản phẩm có id được cập nhật.
     + Cập nhật giao diện:
       - **cart-quantity**: Hiển thị số lượng sản phẩm trong giỏ hàng.
       - **subtotal**: Hiển thị tổng giá trị tạm tính.
       - **total${id}**: Hiển thị tổng giá trị của sản phẩm có id.

**3. Điểm cần cải thiện**

**Đường dẫn /products//cart**

* Có lỗi dư / trong URL:
* '/products//cart' -> '/products/cart'

**Xử lý lỗi**

* Hiện tại không có xử lý cho các trạng thái lỗi. Có thể thêm:
* if (!res.ok) {
* console.error('Failed to update cart:', res.statusText);
* return;
* }

**Hiển thị định dạng tiền tệ**

* json.subtotal và json.item.total nên được định dạng:
* document.getElementById('subtotal').innerText = `$${json.subtotal.toFixed(2)}`;

**Kiểm tra dữ liệu trước khi cập nhật giao diện**

* Cần kiểm tra json có dữ liệu hợp lệ trước khi truy cập các thuộc tính.

**4. Tóm tắt**

* **addCart**: Gửi yêu cầu thêm sản phẩm vào giỏ hàng và cập nhật số lượng sản phẩm trong giao diện.
* **updateCart**: Gửi yêu cầu cập nhật số lượng sản phẩm và cập nhật giao diện, bao gồm tổng giá trị giỏ hàng.
* **Tối ưu**: Sửa lỗi URL, thêm xử lý lỗi, và định dạng dữ liệu tiền tệ khi hiển thị.