**GIÁO TRÌNH LARAVEL - BUỔI 20**

**THỰC HÀNH**

1. **Nội dung**

* Hiển thị chi tiết sản phẩm
* Xây dựng chức năng thêm sản phẩm, sửa số lượng sản phẩm trong giỏ hàng
* Hiển thị danh sách giỏ hàng

1. **Hiển thị chi tiết sản phẩm**

Để hiển thị chi tiết sản phẩm chúng ta cần phải truyền slug\_cate hoặc id trên route . ở đây chúng ta truyền slug giúp tối ưu hoá seo

|  |
| --- |
| Route::get('{slug\_cate}.html','\productController@GetDetail'); |

Phía controller chúng ta cần phải tìm sản phẩm có slug được truyền trên route và truyền sản phẩm đó sang views bằng cách sử dụng *where* *('slug',$slug\_cate )*

Lúc này sang bên view chỉ việc hiển thị thông tin chi tiết sản phẩm bằng biến $data

1. **xây dựng chức năng giỏ hàng**

Ta tạo route xử lý việc thêm, xóa, sửa số lượng 1 sản phẩm trong giỏ và các function xử lý tương ứng bên cartController:

* **GetCart** : Hiển thị giỏ hàng chúng ta làm tương tư như những trang hiển thị đã làm

Chúng ta sử dụng *Cart::Total(0,'',',')* để lấy tổng tiền của giỏ hàng

* **AddCart** : Sử dụng tham số id\_prd để truyền id của sản phẩm sau đó làm tương tự như các trang thêm .

Riêng phần số lượng ($qty) chúng ta sể để nếu có quantity tức *$quantity>0* thì $qty=$r->$quantity, còn *$quantity=0* thì chúng ta cho $qty =1

Sau đó sử dụng *Cart::add()* thêm dữ liệu giỏ hàng

* **UpdateCart** : Ở đây chúng ta cần 2 tham số là *rowId* và *qty* để lấy dữ liệu số lượng sản phảm thay đổi của sản phẩm

Sử dụng *Cart::update()* để sửa dữ liệu giỏ hàng

* **DelCart** : Sử dụng tham số rowId để lấy dữ liệu sản phẩm muốn xóa ,sau đó dùng *Cart::remove()*  để xóa sản phẩm. Cuối cùng bổ sung script hỏi trước khi xóa sản phẩm

Bây giờ ở views detail chúng ta cần tạo form chứa số lượng sản phẩm và action đến route thêm sản phẩm (kèm với đó là id sản phẩm muốn thêm vào giỏ)

|  |
| --- |
| <form action="cart/add/{{ $product->id }}" method="get"> |

Lúc này thì id sản phẩm muốn thêm vào giỏ chúng ta có thể lấy ra được và thực hiện lấy thông tin sản phẩm trong database dựa vào id . từ những thông tin đó chúng ta sẽ thêm vào trong giỏ của shopping cart. (Tham khảo hàm AddCart() bên trên nhé)

1. **Hiển thị danh sách giỏ hàng**

Chúng ta chỉ việc truyền tất cả các sản phẩm có trong giỏ của shopping cart sang View và hiển thị ra . Ở phương thức GetCart() phía trên mình đã chú thích rồi.

Để sử dụng được chức năng xoá,sửa chúng ta cần sử dụng ajax

|  |
| --- |
| @section('script')  @parent  <script>  function update\_qty(rowId,qty) // giúp cập nhập số lượng sản phẩm trong giỏ  {  $.get('cart/update/'+rowId+'/'+qty,  function()  {  window.location.reload();  }  );  }  </script>  @stop |

Và ở lúc duyệt dữ liệu chúng ta cần quan tâm cho mình 2 thẻ a :

Xoá sản phẩm

|  |
| --- |
| <div class="display-tc">  <a onclick="return del\_cart('{{ $product->name }}')" //sử dụng javascript hỏi trước khi truy xuất vào href  href="cart/del/{{ $product->rowId }}" class="closed"></a>  </div> |

Sửa sản phẩm

|  |
| --- |
| <div class="display-tc">  <input onchange="update\_qty('{{ $product->rowId }}',this.value)"  type="number" id="quantity" name="quantity"  class="form-control input-number text-center" value="{{ $product->qty }}">  </div> |

1. **Xây dựng chức năng thanh toán**

a. Hiển thị danh sách đơn hàng đã thanh toán

Đơn hàng thanh toán là phần *processed* trong group *order* được xử lý bởi function GetProcessed trong orderController.

Để hiển thị danh sách đơn hàng :

* **B1** : Tạo model *order.php* để kết nối đến bảng order
* **B2** : Sử dùng $data để nhận dữ liệu các bản ghi từ database

Đơn hàng đã thanh toán có state =1 nên ta sử dụng *where* để lấy bản ghi có state = 1

Sử dụng *orderby* để sắp xếp thứ tự đơn hàng từ mới -> cũ và chúng ta phân trang luôn bằng *paginate*

* **B3 :** Hiển thị danh sách đơn hàng đã thanh toán ở processed.blade.php bằng foreach

b. Hiển thị danh sách đơn hàng

Hiển thị danh sách đơn hàng *order* làm tương tự như trang đơn hàng đã thanh toán nhưng ở đây ta cho hiển thị đơn hàng chưa thanh toán có state=2

c. Hiển thị chi tiết đơn hàng

Vì chi tiết đơn hàng (detail) chứa thông tin sản phẩm của đơn hàng (product\_order) nên chúng ta cần phải liên kết 2 bảng *order* và *product\_order* , 1 order có thể có nhiều sản phẩm nên là liên kết One – Many

Sau đó ta hiển thị chi tiết đơn hàng theo tham số *idOrder* chính là id của đơn hàng tương tự như các phần hiển thị đã làm

d. Xây dựng chức năng xử lý đơn hàng

Chức năng thanh toán là khi chúng ta nhấn vào nút ‘Đã xử lý’ ở phần detail thì đơn hàng chưa xử lý sẽ được chuyển thành đơn hàng đã xử lý (processed)

Vì vậy chúng ta cần 1 route chứa đường dẫn đến hành động xử lý đơn hàng theo tham số *idOrder.*

Hành động xử lý đơn hàng chúng ta sẽ thực thi trong 1 function là paid() , chuyển state của đơn hàng từ 2 thành 1 và sau đó redirect đến trang processed .