**Bài thực hành 11**

**Tạo giỏ hàng**

**Mục tiêu**

* Sử dụng session trong laravel
* Xây dựng class Cart xử lý chức năng giỏ hàng
* Tạo chức năng đặt hàng và quản lý lịch sử đơn hàng

**Các bài tập**

# Bài 1: Tạo class Cart xử lý chức năng giỏ hàng

**Bước 1:** Tạo thư mục app/Helper sau đó tạo file Cart.php trong thư mục Helper này vứi nội dung sau

Tổng quan về class Cart

| **Thuộc tính/ Phương thức** | **Giải thích** |
| --- | --- |
| public $items | Thuộc tính lưu trữ mảng các sản phẩm trong giỏ hàng |
| public $totalPrice | Thuộc tính lưu giá trị tổng tiền trong giỏ hàng |
| public $totalQuantity | Thuộc tính lưu giá trị tổng số lượng trong giỏ hàng |
| public function add ($product, $quantity = 1) | Phương thức xử lý thêm mới sản phẩm vào giỏ hàng |
| public function delete($id) | Phương thức xóa sản phẩm khỏi giỏ hàng |
| public function update($id, $quantity) | Phương thức cập nhật số lượng sản phẩm trong giỏ hàng |
| public function clear() | Phương thức xóa toàn bộ sản phẩm trong giỏ hàng |
| private function getTotalPrice() | Phương thức trả về tổng tiền trong giỏ hàng |
| private function getTotalQuantity() | Phương thức trả về tổng số lượng trong giỏ hàng |

Nội dung tổng quát của class Cart

<?php

namespace App\Helper;

class Cart {

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* @var $items là mảng gồm các sản phẩm trong giỏ hàng

    \*/

    public $items = [];

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* @var $totalPrice là tổng giá trị tiền trong giỏ hàng

    \*/

    public $totalPrice = 0;

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* @var $totalQuanity là tổng giá trị số lựng sản phẩm trong giỏ hàng

    \*/

    public $totalQuantity = 0;

    public function \_\_construct()

    {

        /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

         \* giá trị khởi tạo cho các thuộc tính

         \* @var $item sẽ lấy từ session('cart')

         \* @var totalPrice được tính toán từ phương thức getTotalPrice

         \* @var totalQuanity được tính toán từ phương thức getTotalQuanity

         \*/

        $this->items = session('cart') ? session('cart') : [];

        $this->totalPrice = $this->getTotalPrice();

        $this->totalQuantity = $this->getTotalQuantity();

    }

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* phương thức add xử lý thêm mới sản phẩm và giỏ hàng

     \* @var $product là tham số dữ liệu sản phẩm đầu vào

     \* @var $quantity là tham số số lượng thêm mới, mặc định là 1

     \*/

    public function add($product, $quantity = 1)

    {

        # code...

    }

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* Phương thức xóa sản phẩm khỏi giỏ hàng

     \* @var @id là tham số id cảu sản phẩm

     \*/

    public function delete($id)

    {

        # code...

    }

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* Phương thức update cập nhật số lượng sản phẩm khỏi giỏ hàng

     \* @var @id là tham số id cảu sản phẩm

     \* @var $quantity số lượng cần cập nhật

     \*/

    public function update($id, $quantity)

    {

        # code...

    }

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* Phương thức xóa toàn bộ sản phẩm trong giỏ hàng

     \*/

    public function clear()

    {

    }

   /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* Phương thức tính toán trả về tổng tiền trong giỏ hàng

     \*/

    private function getTotalPrice()

    {

    }

    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

     \* Phương thức tính toán trả về tổng số lượng trong giỏ hàng

     \*/

    private function getTotalQuanity()

    {

    }

}

**Bước 2:** Chi tiết các phương thức

**Phương thức add**

/\*\*

 \* phương thức add xử lý thêm mới sản phẩm và giỏ hàng

 \* @var $product là tham số dữ liệu sản phẩm đầu vào

 \* @var $quantity là tham số số lượng thêm mới, mặc định là 1

 \*/

public function add($product, $quantity = 1)

{

    $id = $product->id;

    /\*\* Kiểm tra nếu đã có sản phẩm trong giỏ hàng rồi \*/

    if (isset($this->items[$id])) {

        $this->items[$id]->quantity += $quantity; // tăng số lượng lên

    } else {

        // nếu chwua thì tiến hành lưu vào sesion

        $item = [

            'id' => $product->id,

            'name' => $product->name,

            'image' => $product->image,

            'price' => $product->sale\_price ? $product->sale\_price : $product->price,

            'quantity' => $quantity,

        ];

        // thêm sản phẩm vào mảng $items, chuyển về dạng object

        $this->items[$id] = (object) $item;

    }

    /\*\* Lưu lại session giỏ hàng \*/

    session(['cart' => $this->items]);

}

Phương thức **delete**

public function delete($id)

{

    if (isset($this->items[$id])) {

        unset($this->items[$id]); // Loại bỏ sản phẩm khỏi mảng

            /\*\* Lưu lại session giỏ hàng \*/

        session(['cart' => $this->items]);

    }

}

Phương thức **update**

 /\*\*

 \* Phương thức update cập nhật số lượng sản phẩm khỏi giỏ hàng

 \* @var @id là tham số id cảu sản phẩm

 \* @var $quantity số lượng cần cập nhật

 \*/

public function update($id, $quantity)

{

    /\*\* Kiểm tra nếu đã có sản phẩm trong giỏ hàng rồi \*/

    if (isset($this->items[$id])) {

        $this->items[$id]->quantity = $quantity; // tăng số lượng lên

    }

}

Phương thức **clear**

public function clear()

{

    session(['cart' => null]);

}

Phương thức **getTotalPrice**

/\*\*

 \* Phương thức tính toán trả về tổng tiền trong giỏ hàng

 \*/

private function getTotalPrice()

{

    $total = 0;

    foreach($this->items as $item) {

        $total += $item->quantity \* $item->price;

    }

    return $total;

}

Phương thức getTotalQuantity

/\*\*

 \* Phương thức tính toán trả về tổng số lượng trong giỏ hàng

 \*/

private function getTotalQuantity()

{

    $total = 0;

    foreach($this->items as $item) {

        $total += $item->quantity;

    }

    return $total;

}

**Tổng thể nội dung class Cart như sau**

<?php

namespace App\Helper;

class Cart {

    /\*\*

     \* @var $items là mảng gồm các sản phẩm trong giỏ hàng

    \*/

    public $items = [];

    /\*\*

     \* @var $totalPrice là tổng giá trị tiền trong giỏ hàng

    \*/

    public $totalPrice = 0;

    /\*\*

     \* @var $totalQuanity là tổng giá trị số lựng sản phẩm trong giỏ hàng

    \*/

    public $totalQuantity = 0;

    public function \_\_construct()

    {

        /\*\*

         \* giá trị khởi tạo cho các thuộc tính

         \* @var $item sẽ lấy từ session('cart')

         \* @var totalPrice được tính toán từ phương thức getTotalPrice

         \* @var totalQuanity được tính toán từ phương thức getTotalQuanity

         \*/

        $this->items = session('cart') ? session('cart') : [];

        $this->totalPrice = $this->getTotalPrice();

        $this->totalQuantity = $this->getTotalQuantity();

    }

    /\*\*

     \* phương thức add xử lý thêm mới sản phẩm và giỏ hàng

     \* @var $product là tham số dữ liệu sản phẩm đầu vào

     \* @var $quantity là tham số số lượng thêm mới, mặc định là 1

     \*/

    public function add($product, $quantity = 1)

    {

        $id = $product->id;

        /\*\* Kiểm tra nếu đã có sản phẩm trong giỏ hàng rồi \*/

        if (isset($this->items[$id])) {

            $this->items[$id]->quantity += $quantity; // tăng số lượng lên

        } else {

            // nếu chwua thì tiến hành lưu vào sesion

            $item = [

                'id' => $product->id,

                'name' => $product->name,

                'image' => $product->image,

                'price' => $product->sale\_price ? $product->sale\_price : $product->price,

                'quantity' => $quantity,

            ];

            // thêm sản phẩm vào mảng $items, chuyển về dạng object

            $this->items[$id] = (object) $item;

        }

        /\*\* Lưu lại session giỏ hàng \*/

        session(['cart' => $this->items]);

    }

    /\*\*

     \* Phương thức xóa sản phẩm khỏi giỏ hàng

     \* @var @id là tham số id cảu sản phẩm

     \*/

    public function delete($id)

    {

        if (isset($this->items[$id])) {

            unset($this->items[$id]); // Loại bỏ sản phẩm khỏi mảng

                /\*\* Lưu lại session giỏ hàng \*/

            session(['cart' => $this->items]);

        }

    }

        /\*\*

     \* Phương thức update cập nhật số lượng sản phẩm khỏi giỏ hàng

     \* @var @id là tham số id cảu sản phẩm

     \* @var $quantity số lượng cần cập nhật

     \*/

    public function update($id, $quantity)

    {

        /\*\* Kiểm tra nếu đã có sản phẩm trong giỏ hàng rồi \*/

        if (isset($this->items[$id])) {

            $this->items[$id]->quantity = $quantity; // tăng số lượng lên

        }

    }

    public function clear()

    {

        session(['cart' => null]);

    }

    /\*\*

     \* Phương thức tính toán trả về tổng tiền trong giỏ hàng

     \*/

    private function getTotalPrice()

    {

        $total = 0;

        foreach($this->items as $item) {

            $total += $item->quantity \* $item->price;

        }

        return $total;

    }

    /\*\*

     \* Phương thức tính toán trả về tổng số lượng trong giỏ hàng

     \*/

    private function getTotalQuantity()

    {

        $total = 0;

        foreach($this->items as $item) {

            $total += $item->quantity;

        }

        return $total;

    }

}

**Bước 3:** Tạo các routes cho chức năng giỏ hàng, mở file routes/web.php

<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\HomeController;

use App\Http\Controllers\CartController;

/\*

|--------------------------------------------------------------------------

| Web Routes

|--------------------------------------------------------------------------

|

| Here is where you can register web routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| contains the "web" middleware group. Now create something great!

|

\*/

Route::get('', [HomeController::class, 'index'])->name('home.index');

Route::get('product/{product}', [HomeController::class, 'product'])->name('home.product\_detail');

Route::post('product-comment/{product}', [HomeController::class, 'productComment'])->name('home.product\_comment');

Route::get('delete-comment/{comment}', [HomeController::class, 'deleteComment'])->name('home.deleteComment');

// route sẽ hiển thị form login

Route::get('login', [HomeController::class, 'login'])->name('home.login');

// route này validate dữ liệu khi submit form

Route::post('login', [HomeController::class, 'check\_login']);

/\*\*Nhóm route xử lý giỏ hàng \*/

Route::group(['prefix' => 'cart'], function() {

    Route::get('add/{id}', [CartController::class, 'add'])->name('cart.add');

    Route::get('delete/{id}', [CartController::class, 'delete'])->name('cart.delete');

    Route::get('update/{id}', [CartController::class, 'update'])->name('cart.update');

    Route::get('clear', [CartController::class, 'clear'])->name('cart.clear');

    Route::get('view', [CartController::class, 'view'])->name('cart.view');

});

**Bước 4:** Tạo controller CartController và sử dụng class Cart để thực hiện các chức năng tương ứng

Chạy lệnh cmd: **php artisan make:controller CartController**

Sau đó khai báo các phương thức tương ứng như route đã sử dụng ở trên

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Product;

use App\Helper\Cart;

class CartController extends Controller

{

    public function add(Product $product, Cart $cart)

    {

        $quantity = request('quantity', 1);

        $quantity = $quantity > 0 ? $quantity : 1;

        // gọi phương thức add và truyền tham số tương ứng

        $cart->add($product, $quantity);

        // Chuyển hướng qua danh sách giỏ hàng

        return redirect()->route('cart.view');

    }

    public function delete($id, Cart $cart)

    {

        // gọi phương thức delete và truyền tham số tương ứng

        $cart->delete($id);

        // Chuyển hướng qua danh sách giỏ hàng

        return redirect()->route('cart.view');

    }

    public function update($id, Cart $cart)

    {

        // gọi phương thức update và truyền tham số tương ứng

        $quantity = request('quantity', 1);

        $quantity = $quantity > 0 ? $quantity : 1;

        $cart->update($id, $quantity);

        // Chuyển hướng qua danh sách giỏ hàng

        return redirect()->route('cart.view');

    }

    public function clear(Cart $cart)

    {

        // gọi phương thức clear

        $cart->clear();

        // Chuyển hướng qua danh sách giỏ hàng

        return redirect()->route('cart.view');

    }

    public function view(Cart $cart)

    {

        // Truyền dữ liệu qua view 'cart-view

        return view('cart-view', compact('cart'));

    }

}

**Bước 5:** Thêm link cho các nút bấm add to cart trên giao diện

Tại danh sách sản phẩm trang chủ file views/index.blade.php

@extends('master.home')

@section('main')

<h2>Sản phẩm mới nhất</h2>

<div class="row products">

    @foreach($products as $pro)

    <div class="col-md-3">

        <div class="thumbnail">

            <img src="https://image.kacana.vn/images/product/gio-xach-dep-hang-hieu-tui-nu-fsm-2303-4013216754.jpg"

                alt="">

            <div class="caption">

                <h3>{{ $pro->name }}</h3>

                <p>

                    <s>Giá gốc: {{ $pro->price }}</s>

                    <b>Giá mới: {{ $pro->sale\_price }}</b>

                </p>

                <p>

                    <a href="{{ route('home.product\_detail', $pro->id) }}" class="btn btn-primary">View</a>

                    <a href="{{ route('cart.add', $pro->id) }}" class="btn btn-default">Add To Cart</a>

                </p>

            </div>

        </div>

    </div>

    @endforeach

</div>

@stop()

Tại file chi tiết sản phẩm: views/product-detail.blade.php

TÌm đến form thêm mới giỏ hàng

@extends('master.home')

@section('main')

<h2>Chi tiết sản phẩm</h2>

<div class="row products">

    <div class="col-md-5">

        <img src="https://image.kacana.vn/images/product/gio-xach-dep-hang-hieu-tui-nu-fsm-2303-4013216754.jpg" alt=""

            style="width:100%">

    </div>

    <div class="col-md-7">

        <h3>{{ $product->name }}</h3>

        <p style="font-size: 18px">

            <s>Giá gốc: {{ $product->price }}</s>

            <b>Giá mới: {{ $product->sale\_price }}</b>

        </p>

        <p>

        <form action="{{ route('cart.add', $product->id) }}" method="GET" class="form-inline" role="form">

            <div class="form-group">

                <input type="number" class="form-control" name="quantity" placeholder="Input quantity">

            </div>

            <button type="submit" class="btn btn-success">Add To Cart</button>

        </form>

        </p>

    </div>

</div>

<hr>

<!-- Kiểm tra xem người dùng đã đăng nhập chưa  -->

@if (auth('customer')->check())

<form action="{{ route('home.product\_comment', $product->id) }}" method="POST" role="form">

    <input type="hidden" name="product\_id" value="{{$product->id}}">

    <legend>Bình luận sản phẩm</legend>

    @csrf

    @if (Session::has('ok'))

    <div class="alert alert-success">

        <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

        <strong>{{ Session::get('ok') }}!</strong>

    </div>

    @endif

    <div class="form-group">

        <label for="">Nội dung bình luận</label>

        <textarea name="content" class="form-control" placeholder="Nội dung bình luận"></textarea>

        @error('content') {{ $message }} @enderror

    </div>

    <button type="submit" class="btn btn-primary">Gửi bình luận</button>

</form>

@else

<!-- Nếu chwua đăng nhập thì thông báo  -->

<div class="alert">

    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

    <strong>Chưa đăng nhập</strong> bạn hãy <a href="{{ route('home.login') }}">click vào đây</a> để đăng nhập

</div>

@endif

<hr>

<div class="comments">

    @foreach($product->comments as $comment)

    <div class="media">

        <a class="pull-left" href="#">

            <img width="60" class="media-object" src="https://haycafe.vn/wp-content/uploads/2022/03/Avatar-TikTok-anh-dai-dien-TikTok.jpg" alt="Image">

        </a>

        <div class="media-body">

            <h4 class="media-heading">{{$comment->user->name}}</h4>

            <p>{{$comment->content}}</p>

        @if (auth('customer')->check() && auth('customer')->user()->can('change-comment', $comment))

            <a href="{{ route('home.deleteComment', $comment->id) }}" class="label label-danger">Xóa</a>

            <a href="" class="label label-default">Sửa</a>

        @endif

        </div>

    </div>

    @endforeach

</div>

<hr>

@stop()

Bước 6: Tạo file hiển thị danh sách giỏ hàng views/cart-view.blade.php với nội dung sau

@extends('master.home')

@section('main')

<div class="row">

    <div class="col-md-6">

        <h2>Thông tin giỏ hàng </h2>

    </div>

    <div class="col-md-6">

        <h2>Số lượng: {{ $cart->totalQuantity }}, Tổng tiền: {{ number\_format($cart->totalPrice) }} đ </h2>

    </div>

</div>

@if (count($cart->items) > 0)

<table class="table table-bordered table-hover">

    <thead>

        <tr>

            <th class="text-center">STT</th>

            <th>Name</th>

            <th>Price</th>

            <th>Quantity</th>

            <th>Sub Total</th>

            <th></th>

        </tr>

    </thead>

    <tbody>

        @foreach($cart->items as $item)

        <tr>

            <td class="text-center">{{$loop->index + 1}}</td>

            <td>{{$item->name}}</td>

            <td>{{ number\_format($item->price) }} đ</td>

            <td>

                <form action="{{ route('cart.update', $item->id) }}">

                    <input type="number" name="quantity" value="{{$item->quantity}}"

                        style="width:80px; text-align: center">

                    <button class="btn btn-sm btn-success">Update</button>

                </form>

            </td>

            <td>{{ number\_format($item->quantity \* $item->price) }} đ</td>

            <td>

                <a href="{{ route('cart.delete', $item->id) }}" class="btn btn-sm btn-danger" onclick="return confirm('Bạn có chắc không?')">Delete</a>

            </td>

        </tr>

        @endforeach

    </tbody>

</table>

<div class="text-center">

    <a href="{{ route('home.index') }}" class="btn btn-sm btn-primary">Tiếp tục mua hàng</a>

    <a href="{{ route('cart.clear') }}" class="btn btn-sm btn-danger" onclick="return confirm('Bạn có chắc không?')">Xóa

        hết</a>

    <a href="{{ route('home.index') }}" class="btn btn-sm btn-success">Đặt hàng</a>

</div>

@else

<div class="alert alert-danger">

    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

    <strong>Giỏ hàng rỗng</strong> Giỏ hàng đang rỗng, <a href="{{ route('home.index') }}">hãy click vào đây</a> để tiếp

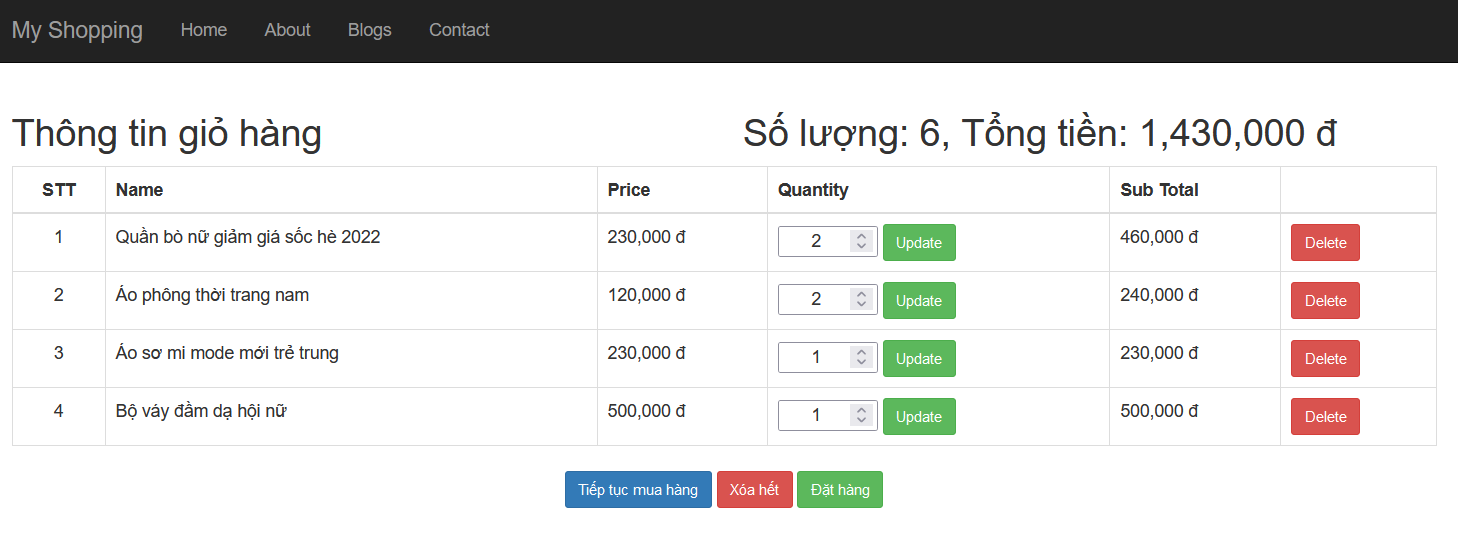
    tục mua hàng

</div>

@endif

@stop()

Bước 7: Kết quả ta có giỏ hàng như sau



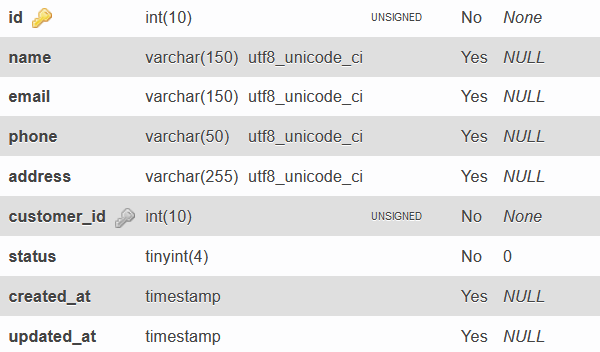
# Bài 2: Chức năng đặt hàng

Bước 1: Review lại CSDL, ngoài các thông tin liên quan đến sản phẩm, thì CSDL cần có các bảng sau

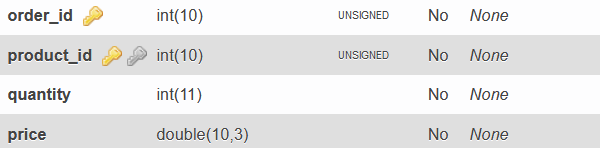
Bảng **customers, orders, order\_details**

Xem lại bài lab 5 – migration

Bảng **orders**



Bảng **order\_details**



**Bước 2:** Model Order và model OrderDetail

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Order extends Model

{

    use HasFactory;

    // khai báo các trường dữ liệu

    protected $fillable = ['customer\_id','name','email','phone','address','status'];

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class OrderDetail extends Model

{

    use HasFactory;

    protected $fillable = ['order\_id','product\_id','price','quantity'];

}

**Bước 3:** Tạo các routes cho chức năng đơn hàng

Mở file **routes/web.php,** các route này sử dụng middleware customer để kiểm tra đăng nhập trước khi vào đặt hàng

***Để có middleware xem lại lab 08.2***

/\*\*Nhóm route xử lý dơn hàng, sử dụng middleware là customer check đăng nhập trước khi đặt hàng \*/

Route::group(['prefix' => 'order','middleware' => 'customer'], function() {

    // Hiển thị form nhập thông tin mua hàng

    Route::get('checkout', [OrderController::class, 'checkout'])->name('order.checkout');

    // Tạo đơn hàng, lưu thông tin vào CSDL

    Route::post('checkout', [OrderController::class, 'post\_checkout']);

    // khi đặt hàng thành công thì chuyển hướng về daay

    Route::get('order-success', [OrderController::class, 'order\_success'])->name('order.success');

    // danh sách đơn hàng đã giao dịch

    Route::get('history', [OrderController::class, 'history'])->name('order.history');

    // chi tiết các sản phẩm trong đơn hàng

    Route::get('detail/{order}', [OrderController::class, 'detail'])->name('order.detail');

});

**Bước 4:** Tạo controller OrderController

**php artisan make:controller OrderController**

Nội dung OrderController

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Helper\Cart;

use App\Models\Order;

use App\Models\OrderDetail;

class OrderController extends Controller

{

    /\*\* Phương thức hiển thị form đặt hàng  \*/

    public function checkout(Cart $cart)

    {

        $customer = auth('customer')->user();

        return view('checkout', compact('cart','customer'));

    }

    /\*\*

     \* Phương thức lưu đơn ahngf

     \*/

    public function post\_checkout(Request $req, Cart $cart)

    {

        $data = $req->only('name','email','phone','address');

        $data['customer\_id'] = auth('customer')->id();

        // lưu thông tin vào bagnr orders

        if ($order = Order::create($data)) {

            // duyệt gỏ hàng lưu vào order\_details

            foreach($cart->items as $item) {

                $detail = [

                    'order\_id' => $order->id,

                    'product\_id' => $item->id,

                    'price' => $item->price,

                    'quantity' => $item->quantity,

                ];

                OrderDetail::create($detail);

            }

            // hủy giỏ hàng, cho về rỗng

            $cart->clear();

            return redirect()->route('order.success');

        }

        return redirect()->back()->with('no','Có lỗi, vui lòng thử lại');

    }

    // phương thức hiển thị danh sách đơn hàng

    public function history()

    {

        $customer = auth('customer')->user();

        return view('order\_history', compact('customer'));

    }

    // phương thức hiển thị chi tiết đơn hàng

    public function detail(Order $order)

    {

        $customer = auth('customer')->user();

        return view('order\_detail', compact('order','customer'));

    }

    public function order\_success()

    {

        return view('order\_success');

    }

}

**Bước 5:** Tạ file hiển thị form đặt hàng views/checkout.blade.php

@extends('master.home')

@section('main')

<div class="row">

    <div class="col-md-4">

        <h2>Thông tin đặt hàng </h2>

        @if (count($cart->items) > 0)

        <form action="" method="POST" role="form">

        @csrf

            <div class="form-group">

                <label for="">Họ và tên</label>

                <input type="text" class="form-control" name="name" value="{{$customer->name}}">

            </div>

            <div class="form-group">

                <label for="">Địa chỉ email</label>

                <input type="text" class="form-control" name="email" value="{{$customer->email}}">

            </div>

            <div class="form-group">

                <label for="">Số điện thoại</label>

                <input type="text" class="form-control" name="phone" value="{{$customer->phone}}">

            </div>

            <div class="form-group">

                <label for="">Địa chỉ giao hàng</label>

                <input type="text" class="form-control" name="address" value="{{$customer->address}}">

            </div>

            <button type="submit" class="btn btn-primary">Đặt hàng</button>

        </form>

        @endif

    </div>

    <div class="col-md-8">

        <h2>Giỏ hàng, Số lượng: {{ $cart->totalQuantity }}, Tổng tiền: {{ number\_format($cart->totalPrice) }} đ </h2>

        @if (count($cart->items) > 0)

        <table class="table table-bordered table-hover">

            <thead>

                <tr>

                    <th class="text-center">STT</th>

                    <th>Name</th>

                    <th>Price</th>

                    <th>Quantity</th>

                    <th>Sub Total</th>

                    <th></th>

                </tr>

            </thead>

            <tbody>

                @foreach($cart->items as $item)

                <tr>

                    <td class="text-center">{{$loop->index + 1}}</td>

                    <td>{{$item->name}}</td>

                    <td>{{ number\_format($item->price) }} đ</td>

                    <td>

                        <form action="{{ route('cart.update', $item->id) }}">

                            <input type="number" name="quantity" value="{{$item->quantity}}"

                                style="width:80px; text-align: center">

                            <button class="btn btn-sm btn-success">Update</button>

                        </form>

                    </td>

                    <td>{{ number\_format($item->quantity \* $item->price) }} đ</td>

                    <td>

                        <a href="{{ route('cart.delete', $item->id) }}" class="btn btn-sm btn-danger"

                            onclick="return confirm('Bạn có chắc không?')">Delete</a>

                    </td>

                </tr>

                @endforeach

            </tbody>

        </table>

        @else

        <div class="alert alert-danger">

            <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

            <strong>Giỏ hàng rỗng</strong> Giỏ hàng đang rỗng, <a href="{{ route('home.index') }}">hãy click vào đây</a> để tiếp tục mua hàng

        </div>

        @endif

    </div>

</div>

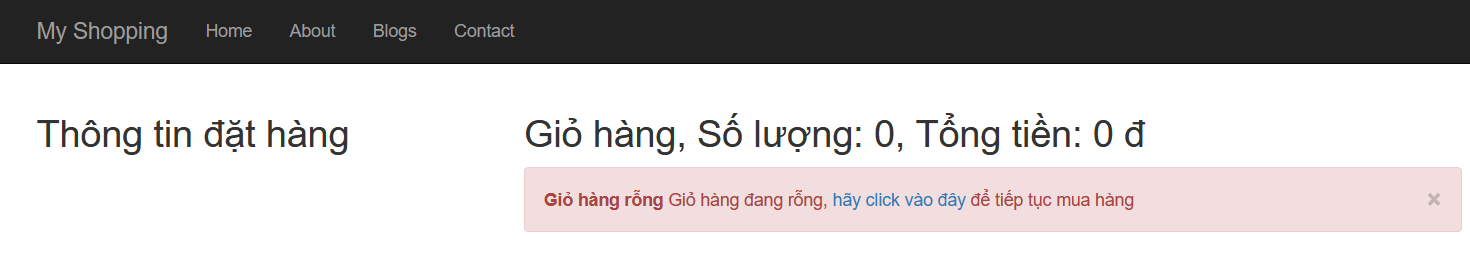
@stop()

**Bước 6:** Truy cập link <http://127.0.0.1:8000/order/checkout>

Chú ý: khi vào link này, nếu

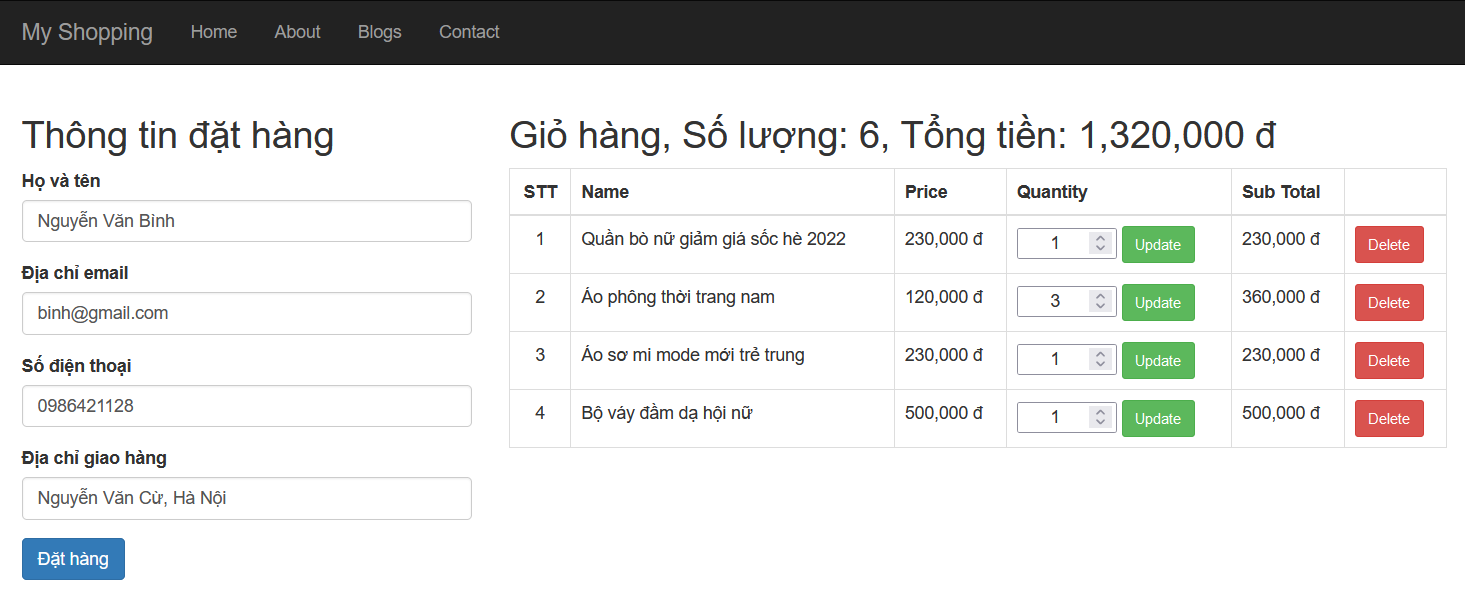
Chưa đăng nhập sẽ tự động chuyển về trang đăng nhập

Nếu không có sản phẩm trong giỏ hàng thì có thông báo lỗi

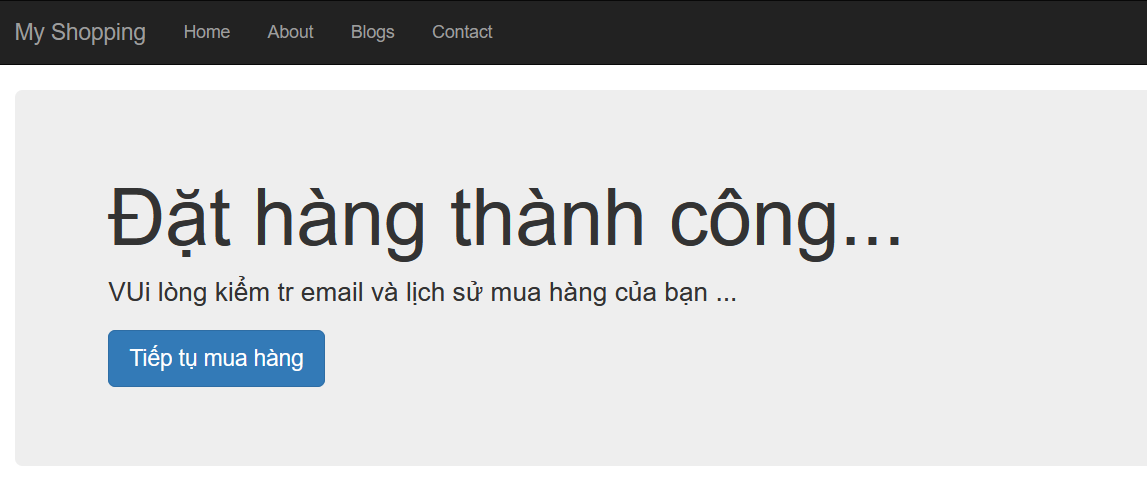


Khi thỏa mãn tất cả các điều kiện đặt hàng thì sẽ hiển thị như sau

Thông tin khách hàng tự động load lên form, và có thể chỉnh sửa



Khi nhấn nút đặt hàng và xử lý thành công thì có thông báo



Dữ liệu được lưu thành công vào 2 bảng: orders và order\_detail

**Bước 7:** Mở model Customer thêm phương thức truy vấn hasMany lấy tất cả đơn hàng của khách hàng

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

class Customer extends Authenticatable

{

    use HasFactory, Notifiable;

    // khai báo các trường dữ liệu

    protected $fillable = ['name','email','phone','password','address'];

    public function orders()

    {

        return $this->hasMany(Order::class,'customer\_id','id');

    }

}

Trong model Order truy vấn lấy chi tiết đơn hàng và tổng tiền của từng đơn hàng

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Order extends Model

{

    use HasFactory;

    // khai báo các trường dữ liệu

    protected $fillable = ['customer\_id','name','email','phone','address','status'];

    public function details()

    {

        return $this->hasMany(OrderDetail::class, 'order\_id','id');

    }

    public function totalPrice()

    {

        $total = 0;

        foreach ($this->details as $item) {

            $total += $item->quantity \* $item->price;

        }

        return $total;

    }

}

**Bước 8:** Tạo file **views/order\_history.blade.php**

@extends('master.home')

@section('main')

<h2>Danh sách đơn hàng đã giao dịch</h2>

<table class="table table-bordered table-hover">

    <thead>

        <tr>

            <th class="text-center">STT</th>

            <th>Ngày mua</th>

            <th>Tổng tiền</th>

            <th>Trạng thái</th>

            <th></th>

        </tr>

    </thead>

    <tbody>

        @foreach($customer->orders as $order)

        <tr>

            <td class="text-center">{{$loop->index + 1}}</td>

            <td>{{$order->created\_at->format('d/m/yy')}}</td>

            <td{{ number\_format( $order->totalPrice() ) }} đ</td>

            <td>

                @if ($order->status == 0)

                <span class="label label-warning">Chờ duyệt</span>

                @elseif ($order->status == 1)

                <span class="label label-primary">Đã duyệt</span>

                @elseif ($order->status == 2)

                <span class="label label-success">Đã hoàn thành</span>

                @else

                <span class="label label-danger">Đã hủy</span>

                @endif

            </td>

            <td>

                <a href="{{ route('order.detail', $order->id) }}" class="btn btn-sm btn-primary">Chi tiết</a>

            </td>

        </tr>

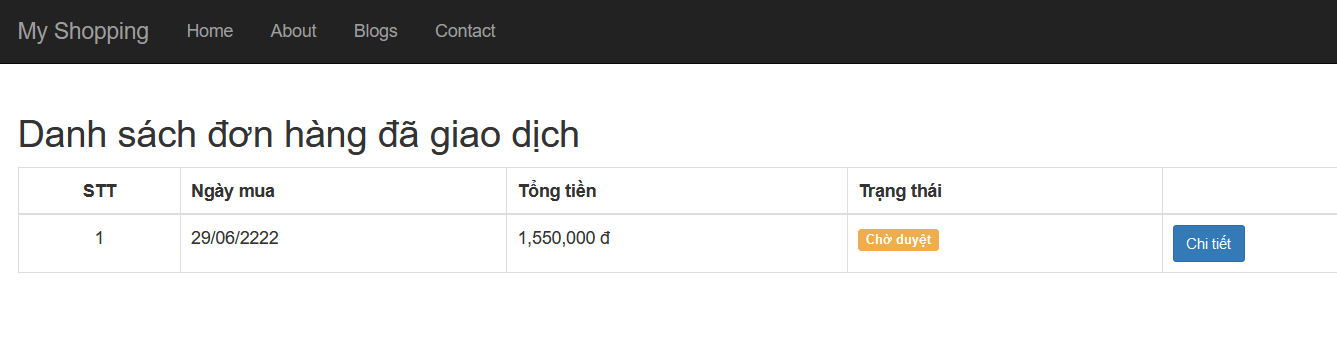
        @endforeach

    </tbody>

</table>

@stop()

Truy cập link: <http://127.0.0.1:8000/order/history>



**Bước 9:** Chi tiết đơn hàng

Mở model OrderDetail và truy vấn hasOne để lấy thông tin sản phẩm của chi tiết đơn hàng

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class OrderDetail extends Model

{

    use HasFactory;

    public $timestamps = false;

    protected $fillable = ['order\_id','product\_id','price','quantity'];

    public function product()

    {

        return $this->hasOne(Product::class,'id','product\_id');

    }

}

Tạo file **views/order\_detail.blade.php**

Soạn nội dung code như sau

@extends('master.home')

@section('main')

<h2>Chi tiết đơn hàng</h2>

<div class="row">

    <div class="col-md-6">

        <h3>Thông tin đơn hàng</h3>

        <table class="table table-bordered table-hover">

            <thead>

                <tr>

                    <th>Mã</th>

                    <td>#{{$order->id}}</td>

                </tr>

                <tr>

                    <th>Ngày đặt</th>

                    <td>{{$order->created\_at->format('d/m/yy')}}</td>

                </tr>

                <tr>

                    <th>Tổng tiền</th>

                    <td>{{number\_format( $order->totalPrice() )}}</td>

                </tr>

                <tr>

                    <th>Trạng thái</th>

                    <td>

                        @if ($order->status == 0)

                        <span class="label label-warning">Chờ duyệt</span>

                        @elseif ($order->status == 1)

                        <span class="label label-primary">Đã duyệt</span>

                        @elseif ($order->status == 2)

                        <span class="label label-success">Đã hoàn thành</span>

                        @else

                        <span class="label label-danger">Đã hủy</span>

                        @endif

                    </td>

                </tr>

            </thead>

        </table>

    </div>

    <div class="col-md-6">

        <h3>Thông tin giao hàng</h3>

        <table class="table table-bordered table-hover">

            <thead>

                <tr>

                    <th>Họ tên</th>

                    <td>{{$order->name}}</td>

                </tr>

                <tr>

                    <th>ĐỊa chỉ</th>

                    <td>{{$order->address}}</td>

                </tr>

                <tr>

                    <th>Email</th>

                    <td>{{$order->email}}</td>

                </tr>

                <tr>

                    <th>Điện thoại</th>

                    <td>{{$order->phone}}</td>

                </tr>

            </thead>

        </table>

    </div>

</div>

<h3>Chi tiết sản phẩm của đơn hàng</h3>

<table class="table table-bordered table-hover">

    <thead>

        <tr>

            <th class="text-center">STT</th>

            <th>Tên sản phẩm</th>

            <th>Số lượng</th>

            <th>Đơn giá</th>

            <th>Thành tiền</th>

        </tr>

    </thead>

    <tbody>

        @foreach($order->details as $detail)

        <tr>

            <td class="text-center">{{$loop->index + 1}}</td>

            <td>{{$detail->product->name}}</td>

            <td>{{$detail->quantity}}</td>

            <td>{{ number\_format( $detail->price ) }} đ</td>

            <td>{{ number\_format( $detail->price\*$detail->quantity ) }} đ</td>

        </tr>

        @endforeach

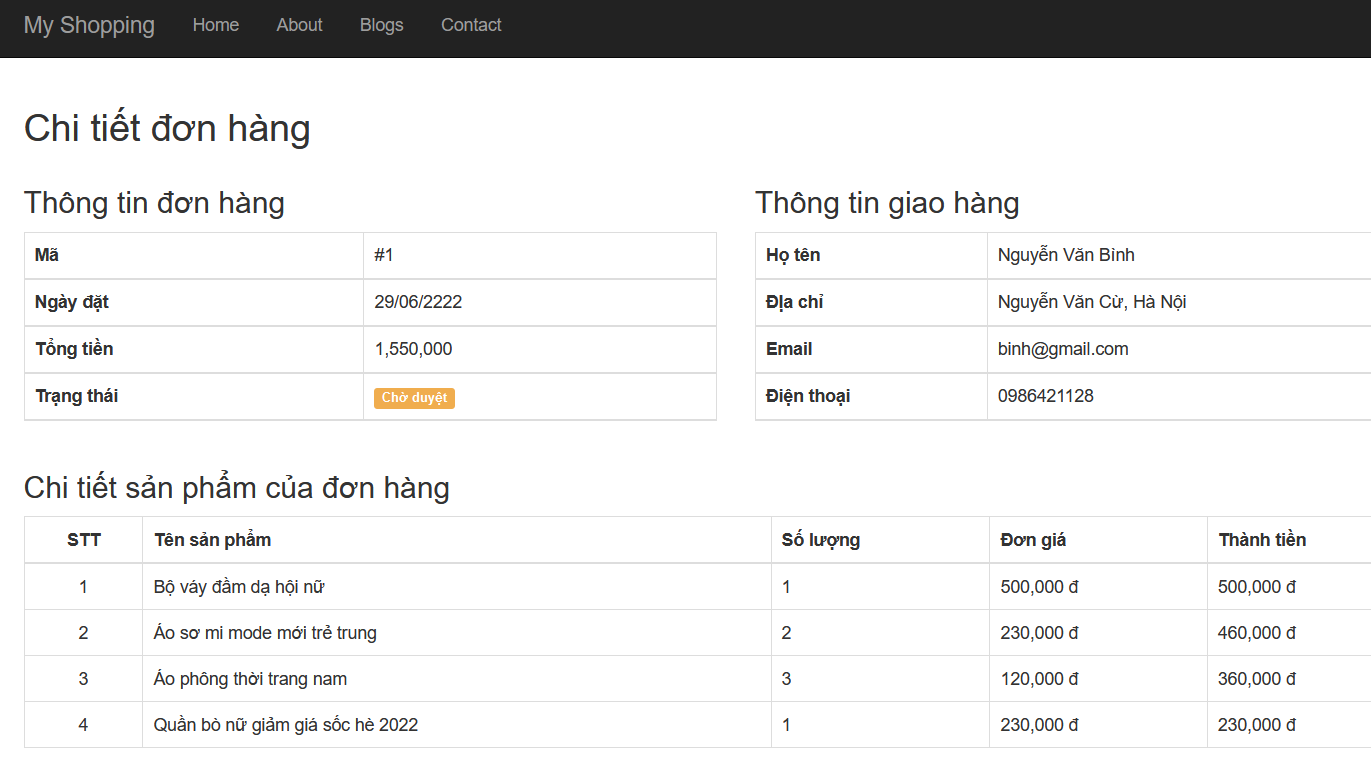
    </tbody>

</table>

@stop()

**Truy cập một chi tiết đơn hàng có link**

<http://127.0.0.1:8000/order/detail/1>



# Bài tập

Sinh vien hãy hoàn thành chức năng giỏ hàng và đặt hàng ở trang người dùng

Hoàn thiện chức năng quản lý đơn hàng ở trang quản trị