**Bài thực hành 12**

**Gửi email trong laravel**

***Mục tiêu***

* Cấu hình gửi email sử dụng dịch vụ smtp
* Tạo form liên hệ (contact) gửi email liên hệ.
* Gửi email xác nhận đơn hàng

**Các bài hướng dẫn**

**Lưu ý: Xem lại và hoàn thành lab 11 (Tạo giỏ hàng và đặt hàng)**

# Bài 1: Cấu hình gửi email bằng smtp.gmail.com

**Bước 1:** Mở file config/mail.php tìm đến smpt và cấu hình kết nối tài khoản email

'smtp' => [

    'transport' => 'smtp',

    'host' => 'smtp.gmail.com',

    'port' => 587,

    'encryption' => 'tls',

    'username' => myemail@gmail.com,

    'password' => 'tmfrfyprahwvxdjz',

    'timeout' => null,

    'auth\_mode' => null,

],

**Lưu ý quan trọng**:

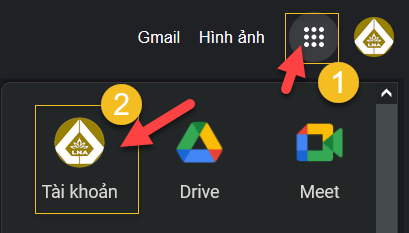
**username**: là địa chỉ email mà bạn đang sử dụng bình thường

**password** (mật khẩu) thì bạn cần tạo một mật khẩu ứng dụng cho email của bạn.

**Bước 2:** Tạo mật khẩu ứng dụng cho tài khoản gmail của bạn

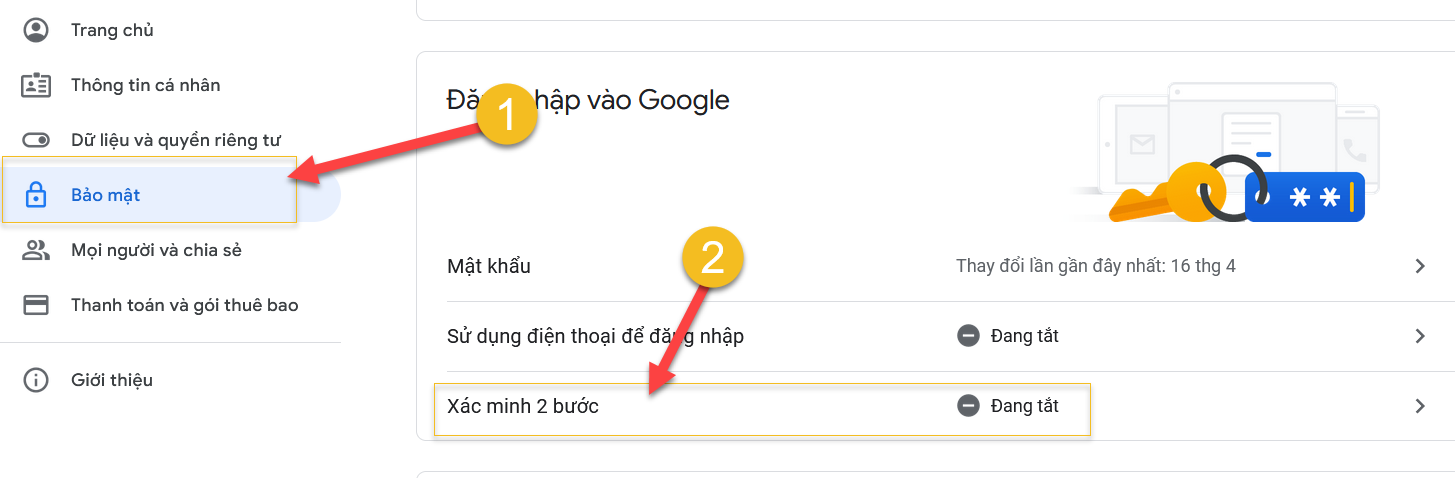
Đăng nhập vào tài khoản gmail đang sử dụng

Click vào biểu tượng 9 chấm, chọn tài khoản

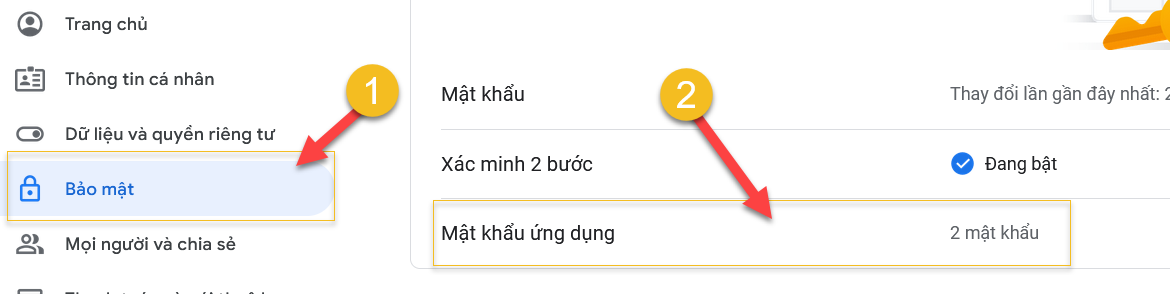


Sau đó tìm đến mục Bảo mật

, lăn chuột xuống dưới tìm đến mục “Xác minh 2 bước”, tại đây hãy làm theo hướng dẫn để bật “Xác minh 2 bước”



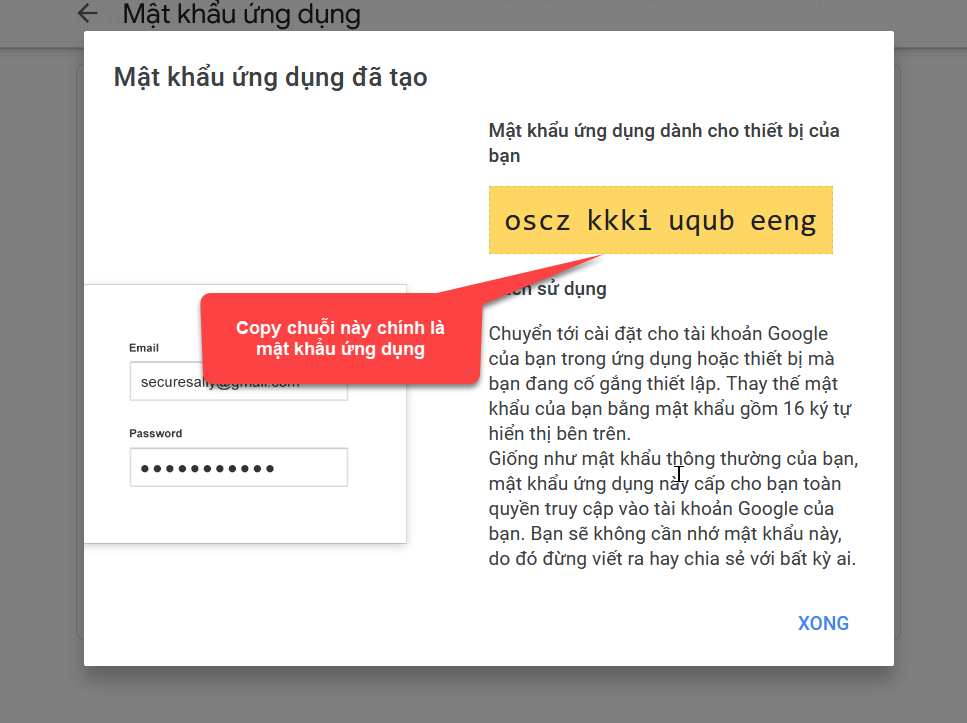
Sau khi bật “Xác minh 2 bước” quay lại màn hình b”Bảo mật” chúng ta thấy có mục “Mật khẩu ứng dụng”, chọn vào đó



Sau khi vào màn hình tạo “Mật khẩu ứng dụng” chọn mục “Khác” và nhập tên tùy ý muốn sau dó nhấn “Tạo”



Sau khi nhấn “Tạo”, màn hình xuất hiện thông báo “Mật Khẩu ứng dụng” hãy copy nó và dán vào mật khẩu chỗ cấu hình gửi email ở bước 1



Lưu ý: Các mật khẩu sau khi tạo, không thể xem lại được, cho nên bạn cần lưu lại nếu muốn sử dụng cho nhiều lần sau hoặc mỗi lần cấu hình thì tạo một mật khẩu mới tương tự các bước ở trên

# Bài 2: Tạo form gửi email liên hệ

**Bước 1**: Tạo Mailable (class ContactMailable) bằng lệnh sau

**php artisan make:mail ContactMailable**

Sau khi chạy lệnh trên, file được tạo trong thư mục App\Mail

Khai báo cấu hình gửi mail cho file này như sau

<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Mail\Mailable;

use Illuminate\Queue\SerializesModels;

class ContactMailable extends Mailable

{

    use Queueable, SerializesModels;

    /\*\*

     \* Create a new message instance.

     \*

     \* @return void

     \*/

    public $contact\_name;

    public $contact\_email;

    public $contact\_message;

    public function \_\_construct($name, $email, $message)

    {

        $this->contact\_name = $name;

        $this->contact\_email = $email;

        $this->contact\_message = $message;

    }

    public function build()

    {

        // rescources/views/mails/contact.blade.php

        return $this->from($this->contact\_email, $this->contact\_name)

                    ->subject('Email contact from '. $this->contact\_name)

                    ->view('mails.contact');

    }

}

Bước 2: Như file ContactMailable gọi tới views/mails/contact.blade.php

Tạo file **views/mails/contact.blade.php** với nội dung sx gửi qua mail như dưới đây

<h3>Xin chào, tôi là {{$contact\_name}}</h3>

<h3>Nội dung tin nhắn:</h3>

<p>

    {{$contact\_message}}

</p>

**Bước 3:** Tạo route để hiển thi form liên hệ và gửi email

Mở file **routes/web.php** khai báo route như dưới đây

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

Route::post('/contact', [HomeController::class, 'send\_contact']);

Bước 4: Mở file HomeController.php và khai báo 2 phương thức tương ứng cho 2 route trên như dưới đây.

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use Mail;

use App\Mail\ContactMailable;

class HomeController extends Controller

{

    public function index()

    {

        $products = Product::orderBy('id','desc')->limit(4)->get();

        return view('index', compact('products'));

    }

    public function product(Product $product)

    {

        return view('product-detail', compact('product'));

    }

    public function contact()

    {

        return view('contact');

    }

    public function send\_contact(Request $req)

    {

        $req->validate([

            'name' => 'required',

            'email' => 'required|email',

            'message' => 'required',

        ]);

         $name = $req->name;

        $message = $req->message;

        $email = $req->email;

        Mail::to('luongna@bachkhoa-aptech.edu.vn')

            ->send(new ContactMailable($name, $email, $message));

        return redirect()->back()->with('ok','Gửi email thành công');

    }

}

Bước 5: Theo như phương thức trên, chúng ta cần tạo file **views/contact.blade.php** để thiết kế form liên hệ như sau

@extends('master.home')

@section('main')

<h2>Form liên hệ</h2>

@if (Session::has('ok'))

<div class="alert alert-success">

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

    <strong>Successfully!</strong> gửi email thành công

</div>

@endif

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

    @csrf

    <div class="form-group">

        <label for="">Họ tên</label>

        <input type="text" class="form-control" name="name" placeholder="Tên liên hệ">

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

    </div>

    <div class="form-group">

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

        <input type="email" class="form-control" name="email" placeholder="Email liên hệ">

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

    </div>

    <div class="form-group">

        <label for="">Nội dung liên hệ</label>

        <textarea name="message" class="form-control" placeholder="Nội dung..."></textarea>

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

    </div>

    <button type="submit" class="btn btn-primary">Gửi liên hệ</button>

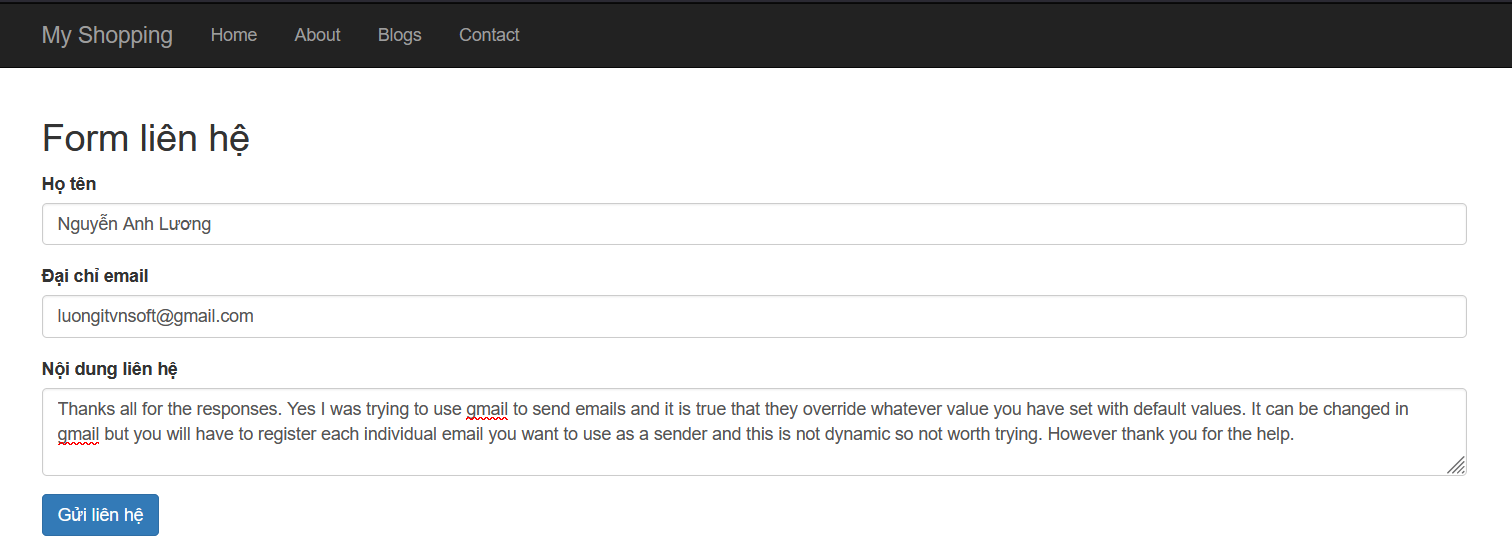
</form>

@stop()

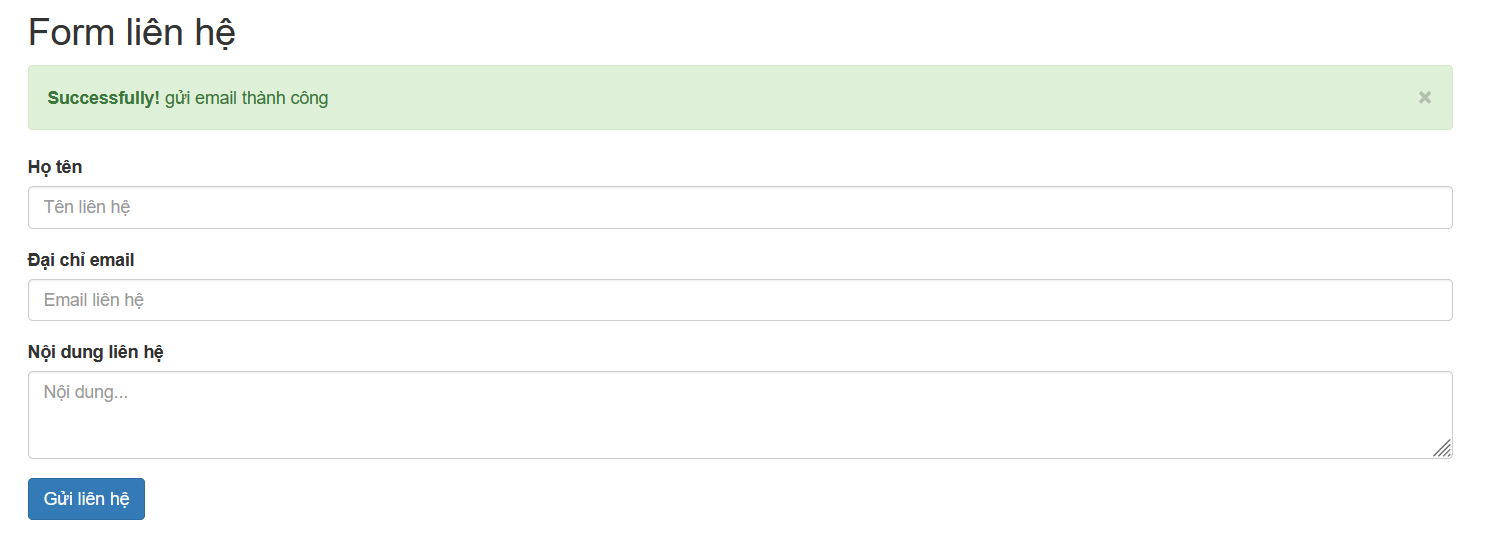
Bước 6: Test gửi email bằng cách truy cập vào link

<http://localhost:8000/contact>

Form có dạng



Sau khi gửi email thành công sẽ có thông báo trên giao diện



Kiểm tra email nhận (Chính là email trong cấu hình to(……)

public function send\_contact(Request $req)

{

    $req->validate([

        'name' => 'required',

        'email' => 'required|email',

        'message' => 'required',

    ]);

    $name = $req->name;

    $message = $req->message;

    $email = $req->email;

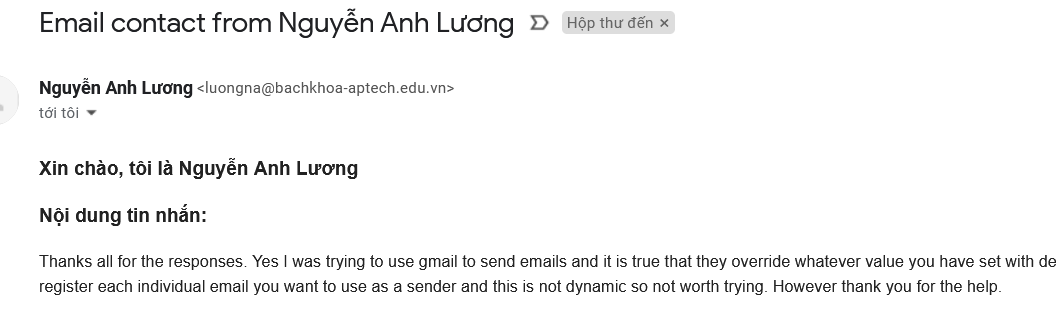
    Mail::to('luongna@bachkhoa-aptech.edu.vn')

        ->send(new ContactMailable($name, $email, $message));

    return redirect()->back()->with('ok','Gửi email thành công');

}

Nội dung email



# Bài 2: Cấu hhinhf gửi email xác nhận đơn hàng

**Bước 1:** Tạo thêm route để request xác nhận đơn hàng

Khi khách hàng click vào nút xác nhận đơn hàng trong email gửi về, sẽ gửi kèm một mã token

/\*\*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');

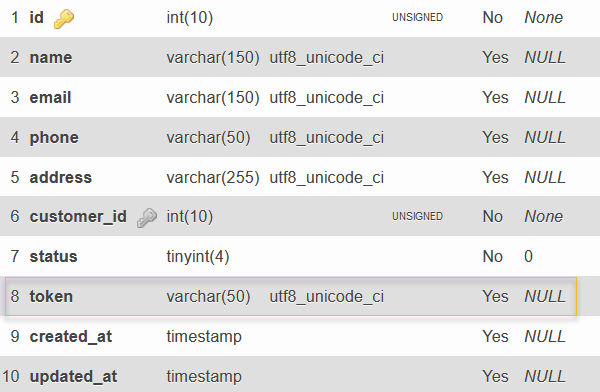
    // chi tiết các sản phẩm trong đơn hàng

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

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

});

**Bước 2:** Tạo thêm cột token kiểu varchar(50) null trong bảng orders



Và thêm token trong fillable của model Order

<?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','token'];

    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 3:** Mở file OrderController thêm phương thức **verify\_order** mà route đã trỏ tới với nội dung sau

**Bước 4:** Tạo OrderMailable với lệnh sau

**php artisan make:mail OrderMailable**

Soạn nội dung cho OrderMailable

<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;

use Illuminate\Contracts\Queue\ShouldQueue;

use Illuminate\Mail\Mailable;

use Illuminate\Queue\SerializesModels;

class OrderMailable extends Mailable

{

    use Queueable, SerializesModels;

    /\*\*

     \* Create a new message instance.

     \*

     \* @return void

     \*/

    public $order;

    public $customer;

    public function \_\_construct($order, $customer)

    {

        $this->order = $order;

        $this->customer = $customer;

    }

    /\*\*

     \* Build the message.

     \*

     \* @return $this

     \*/

    public function build()

    {

        return $this->view('mails.order');

    }

}

**Bước 5:** Tạo **views/mails/order.blade.php** nội dung xác nhận đơn hàng

<div style="width:600px; margin: auto">

    <h4 style="text-align: center">Xin chào quý khách hàng: {{$customer->name}}</h4>

    <p style="text-align: center">

        Dưới đây là email thông tin đơn hàng của quý khách, quý khách hàng vui lòng xác nhận đơn hàng bằng cách click và

        nút bấm <b>"Xác nhận đơn hàng"</b> dưới đây để chúng ôi biết là đơn hàng của quý khách

    </p>

    <p style="text-align: center">

        <a href="{{ route('order.verify\_order', $order->token) }}" target="\_blank"

            style="display: inline-block; text-decoration: none; padding:10px 25px; color: #fff; background: green">Xác

            nhậ đơn hàng</a>

    </p>

    <h4>Thông tài khoản đặt hàng:</h4>

    <p>

        <ul>

            <li><b>Họ tên:</b> {{$customer->name}}</li>

            <li><b>Email:</b> {{$customer->email}}</li>

            <li><b>SỐ ĐT:</b> {{$customer->phone}}</li>

            <li><b>Địa chỉ:</b> {{$customer->Address}}</li>

        </ul>

    </p>

    <div style="width:48%; float: left">

        <h4>Thông tin đơn hàng</h4>

        <table border="1" cellspacing="0" cellpadding="5" style="width:100%">

            <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>Chờ duyệt</td>

                </tr>

            </thead>

        </table>

    </div>

    <div style="width:48%; float: right">

        <h4>Thông tin giao hàng</h4>

        <table border="1" cellspacing="0" cellpadding="5" style="width:100%">

            <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>

    <br>

    <div style="clear: borth"></div>

    <h4>Chi tiết sản phẩm của đơn hàng</h4>

    <table border="1" cellspacing="0" cellpadding="5" style="width:100%">

        <thead>

            <tr>

                <th>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 style="text-align: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>

</div>

**Bước 6:** Mở lại OrderController, cập nhật code gửi email khi đặt hàng

Tìm tới phương thức **post\_checkout** và sửa code như sau

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();

        // Update TOken cho đơn hàng

        $order->update(['token' => \Str::random(40)]);

        // Gửi email

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

        Mail::to($customer->email)->send(new OrderMailable($order, $customer));

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

    }

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

}

Tham khảo toàn bộ code của OrderController dưới đây

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Helper\Cart;

use App\Models\Order;

use App\Models\OrderDetail;

use App\Mail\OrderMailable;

use Mail;

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();

            // Update TOken cho đơn hàng

            $order->update(['token' => \Str::random(40)]);

            // Gửi email

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

            Mail::to($customer->email)->send(new OrderMailable($order, $customer));

            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');

    }

    public function verify\_order($token)

    {

        $order = Order::where('token', $token)->first();

        if ($order) {

            $order->update(['status' => 1, 'token' => null]);

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

        }

        return abort(404); // nếu không có thì báo 404

    }

}

**Bước 7:** Tiến hành đặt hàng và xem kết quả trong địa chỉ email của bạn

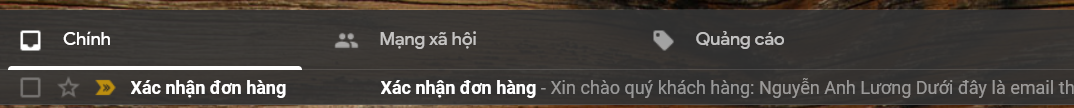
**Trước khi xác nhận email:** Xem lại lịch sử đơn hàng để xem trạng thái khi chưa xác nhân

Link xem lịch sử đơn hàng của kahcsh ahngf

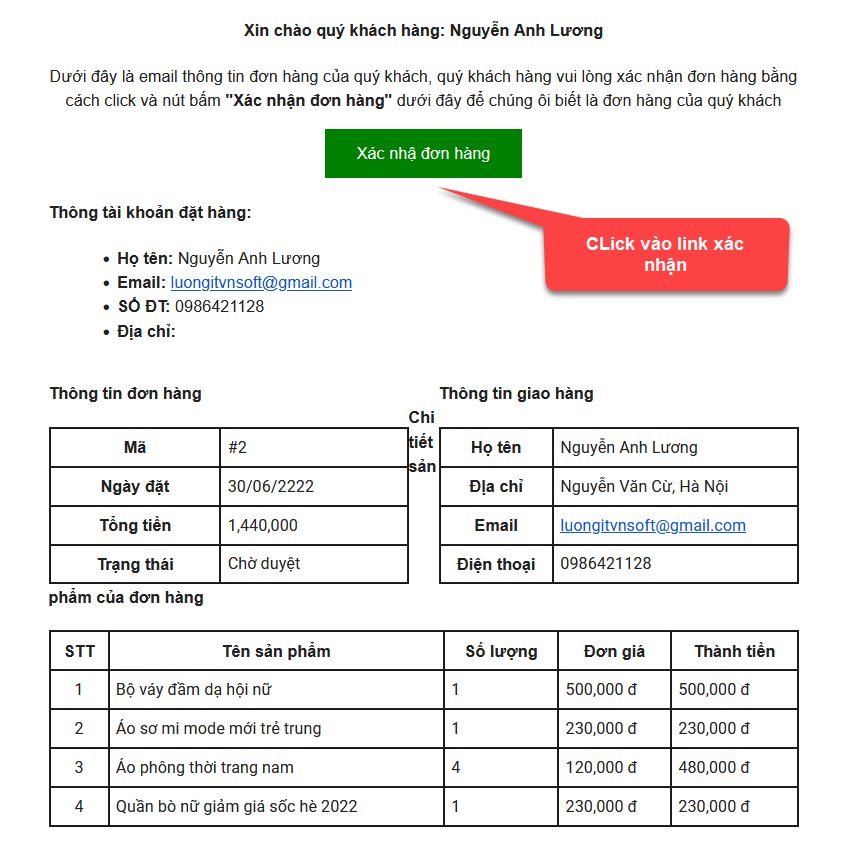
<http://localhost:8000/order/history>



Vào xác nhận email: Click lên link có sẵn trong nội dung email

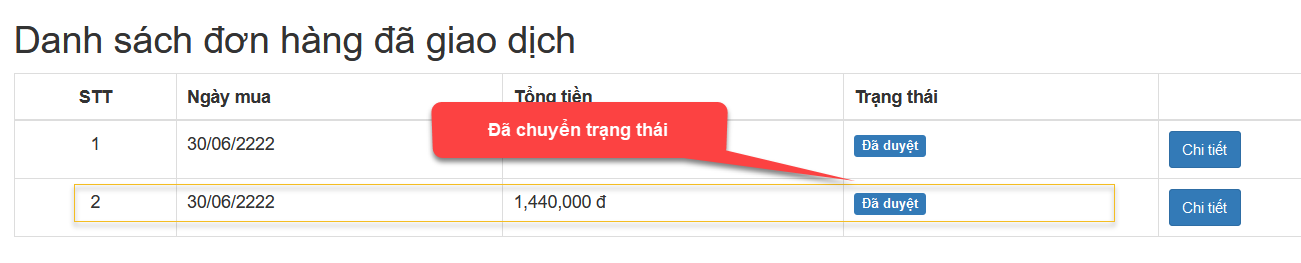


Nội dung email

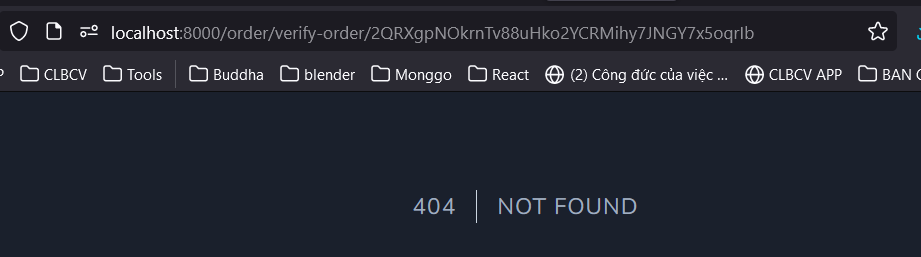


Xem lại lịch sử đơn hàng sau khi click xác nhận email

<http://localhost:8000/order/history>



Sau khi xác nhận đơn hàng, nếu cố tình click vào link xác nhận trong email thì sẽ có thông báo 404



# Bài tập làm thêm

Sinh viên hoàn thiện chức năng xác nhận đơn hàng

Sinh viên thực hiện chức năng lấy lại mật khẩu cho tài khoản khách hàng