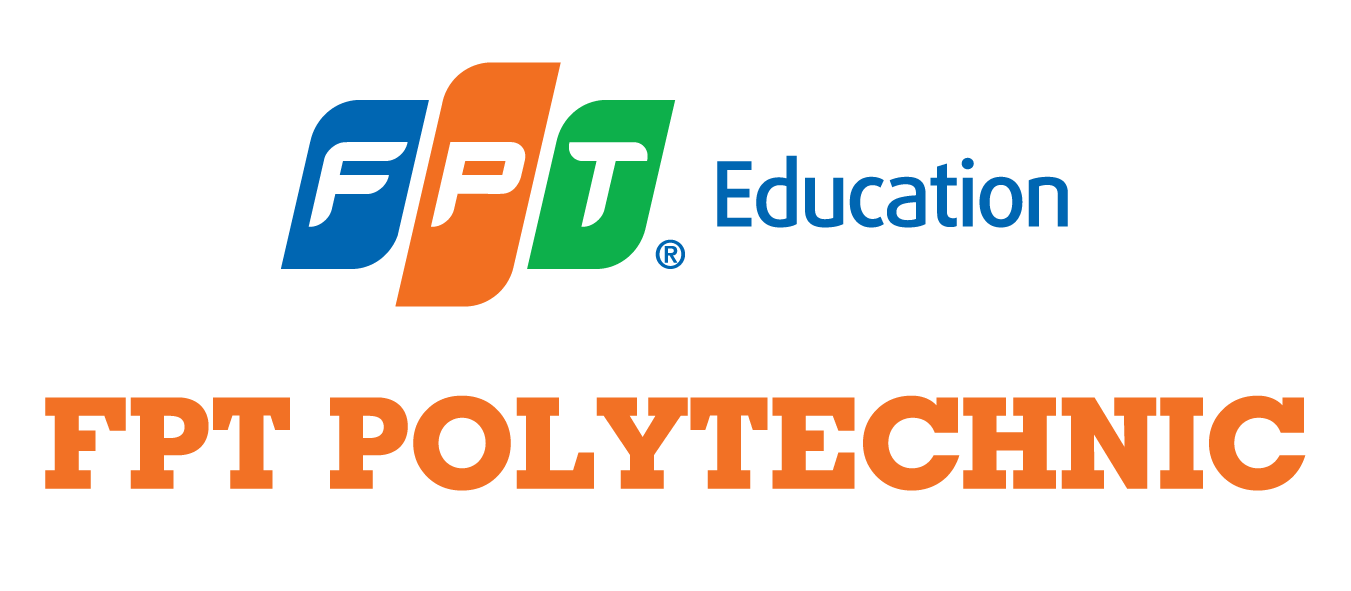
**TRƯỜNG CAO ĐẲNG FPT POLYTECHNIC**

**BỘ MÔN: Công nghệ Thông Tin**

**NGÀNH:Công Nghệ Thông Tin**

**CHUYÊN NGÀNH: Lập Trình Web**



**ASSIGNMENT**

**MÃ MÔN HỌC:** WEB503

**Giảng viên hướng dẫn: Thầy Nguyễn Văn Long**

**Sinh viên thực hiện: Nguyễn Vủ Huy - PS39702**

## **1. Giới Thiệu**

### **1.1. Mô tả Dự Án**

Website bán hàng là một hệ thống giúp người dùng dễ dàng mua sắm sản phẩm trực tuyến. Hệ thống bao gồm các chức năng cơ bản như hiển thị sản phẩm, giỏ hàng, thanh toán và quản lý tài khoản.

### **1.2. Công Nghệ Sử Dụng**

* **Ngôn ngữ làm backend**: node js, express
* **Ngôn ngữ frontend**: HTML, CSS, TypeScript, boostrap
* **Cơ sở dữ liệu**: Mongoose
* **Công nghệ khác**: Multer (để upload nhiều ảnh)
* **Xác thực & Email**: JWT, Nodemailer

## **2. Phân Tích Hệ Thống**

### **2.1. Cấu trúc dự án**

#### Backend (Thư mục xử lý phía máy chủ)

* **controllers/**: Chứa các file điều khiển logic nghiệp vụ của ứng dụng.
* **middlewares/**: Chứa các file xử lý trung gian (middleware), như xác thực, logging.
* **models/**: Chứa các file định nghĩa mô hình dữ liệu, kết nối với database.
* **node\_modules/**: Chứa các thư viện cài đặt từ npm.
* **routers/**: Chứa các định tuyến (routes) để điều hướng API.
* **services/**: Chứa các dịch vụ xử lý logic nghiệp vụ (ví dụ: xử lý thanh toán, gửi email).
* **utils/**: Chứa các hàm tiện ích dùng chung cho toàn bộ dự án.
* **.env**: Chứa các biến môi trường như thông tin database, API key.
* **package.json & package-lock.json**: Chứa thông tin về dependencies và script chạy server.
* **server.js**: File chính để khởi chạy backend.

#### Frontend (Thư mục xử lý giao diện người dùng)

* **.vscode/**: Cấu hình dành cho trình soạn thảo VS Code.
* **admin/**: Giao diện quản trị dành cho admin.
* **css/**: Chứa các file CSS để định dạng giao diện.
* **db/**: Có thể chứa file database giả lập hoặc cấu hình liên quan.
* **dist/**: Chứa các file đã được build (biên dịch) sẵn.
* **fonts/**: Chứa các font chữ sử dụng trong giao diện.
* **images/**: Chứa các hình ảnh tĩnh như logo, banner.
* **js/**: Chứa các file JavaScript xử lý tương tác phía frontend.
* **node\_modules/**: Chứa thư viện npm dành cho frontend.
* **src/**: Chứa mã nguồn chính của frontend.
* **vendor/**: Chứa các thư viện của bên thứ ba (jQuery, Bootstrap, v.v.).

### **2.2. Mô Hình Database**

Proudct:

const productSchema = new mongoose.Schema({

name: { type: String, required: true },

description: { type: String },

price: { type: Number, required: true },

categoryId: { type: mongoose.Schema.Types.ObjectId, ref: "Category", required: true },

images: { type: [String] },

tags: [{ type: String, required: true }],

variants: [

{

color: { type: String, required: true },

size: { type: String, required: true },

stock: { type: Number, required: true },

price: { type: Number }

}

],

}, { timestamps: true });

Category:

const categorySchema = new mongoose.Schema({

categoryName: { type: String, required: true },

image: { type: String },

}, { timestamps: true });

Order:

const orderSchema = new mongoose.Schema({

orderCode: { type: Number, unique: true },

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

products: [

{

\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Product', required: true },

quantity: { type: Number, required: true },

color: { type: String, required: true },

price: { type: Number, required: true },

size: { type: String, required: true },

name: { type: String, required: true },

images: [{ type: String, required: true }]

}

],

totalPrice: { type: Number, required: true },

paymentMethod: {

type: String,

enum: ['cod', 'momo', 'bank\_transfer'],

required: true

},

paymentStatus: {

type: String,

enum: ['pending', 'paid', 'failed'],

default: 'pending'

},

info: {

address: { type: String, required: true },

phone: { type: String, required: true }

},

status: { type: String, enum: ['pending', 'confirmed', 'completed', 'cancelled', 'paid'], default: 'pending' },

linkCheckout: { type: String, required: false }

}, { timestamps: true });

User:

const userSchema = new mongoose.Schema({

fullName: { type: String, required: true },

email: { type: String, required: true, unique: true },

passwordHash: { type: String, required: true },

phoneNumber: { type: String },

address: { type: String },

role: { type: String, enum: ["admin", "customer"], default: "customer" },

isVerified: { type: Boolean, default: false },

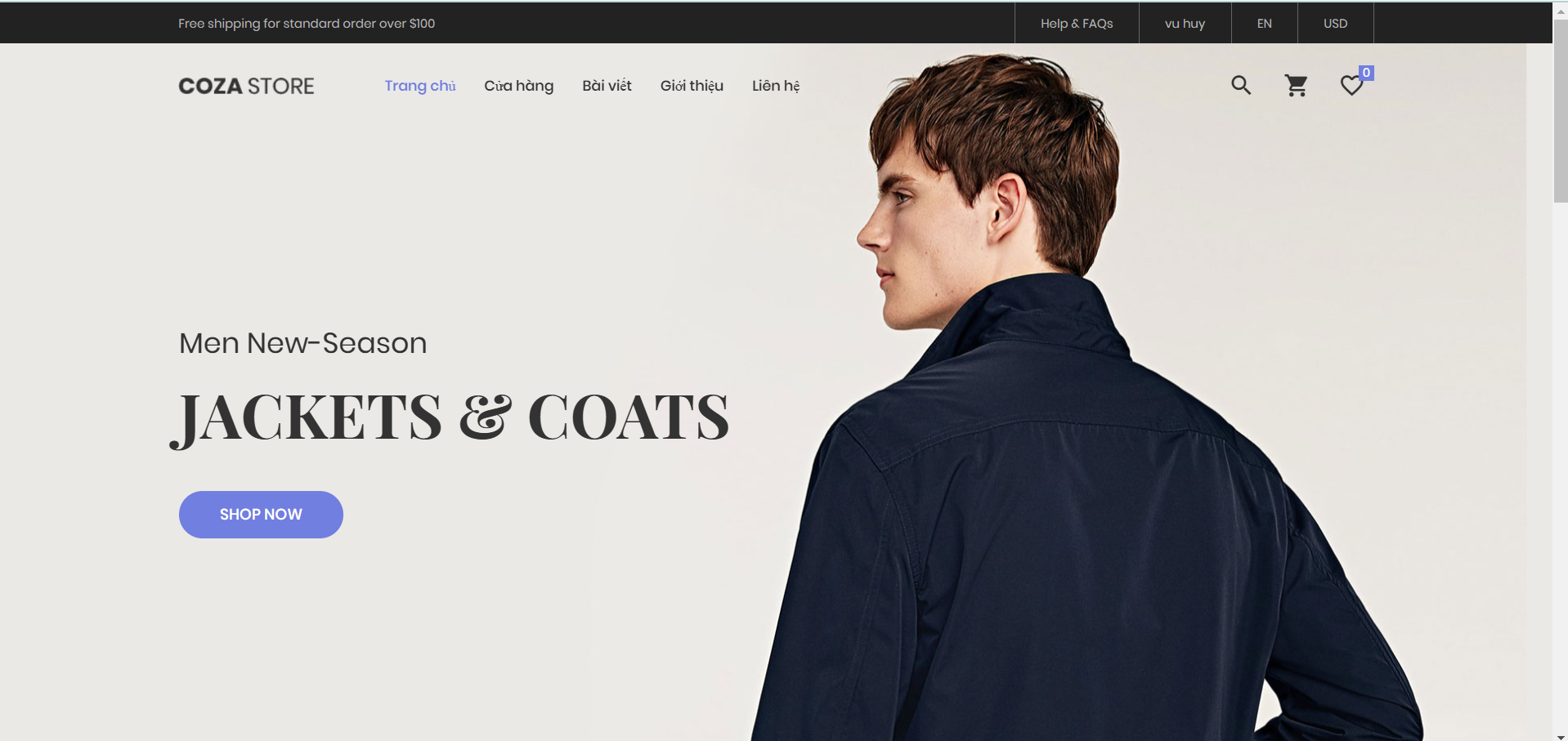
createdAt: { type: Date, default: Date.now }

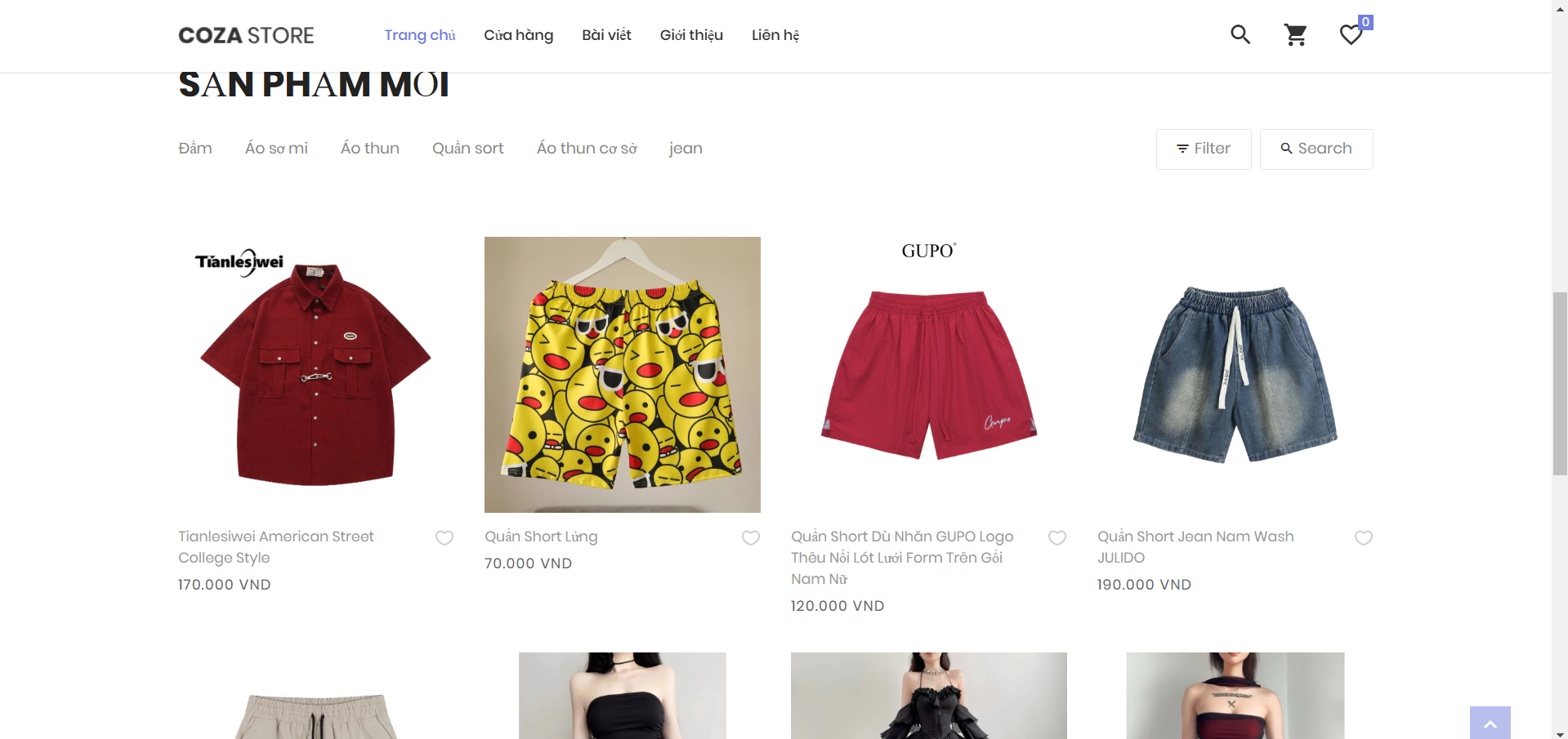
});

## **3. Chức Năng Website**

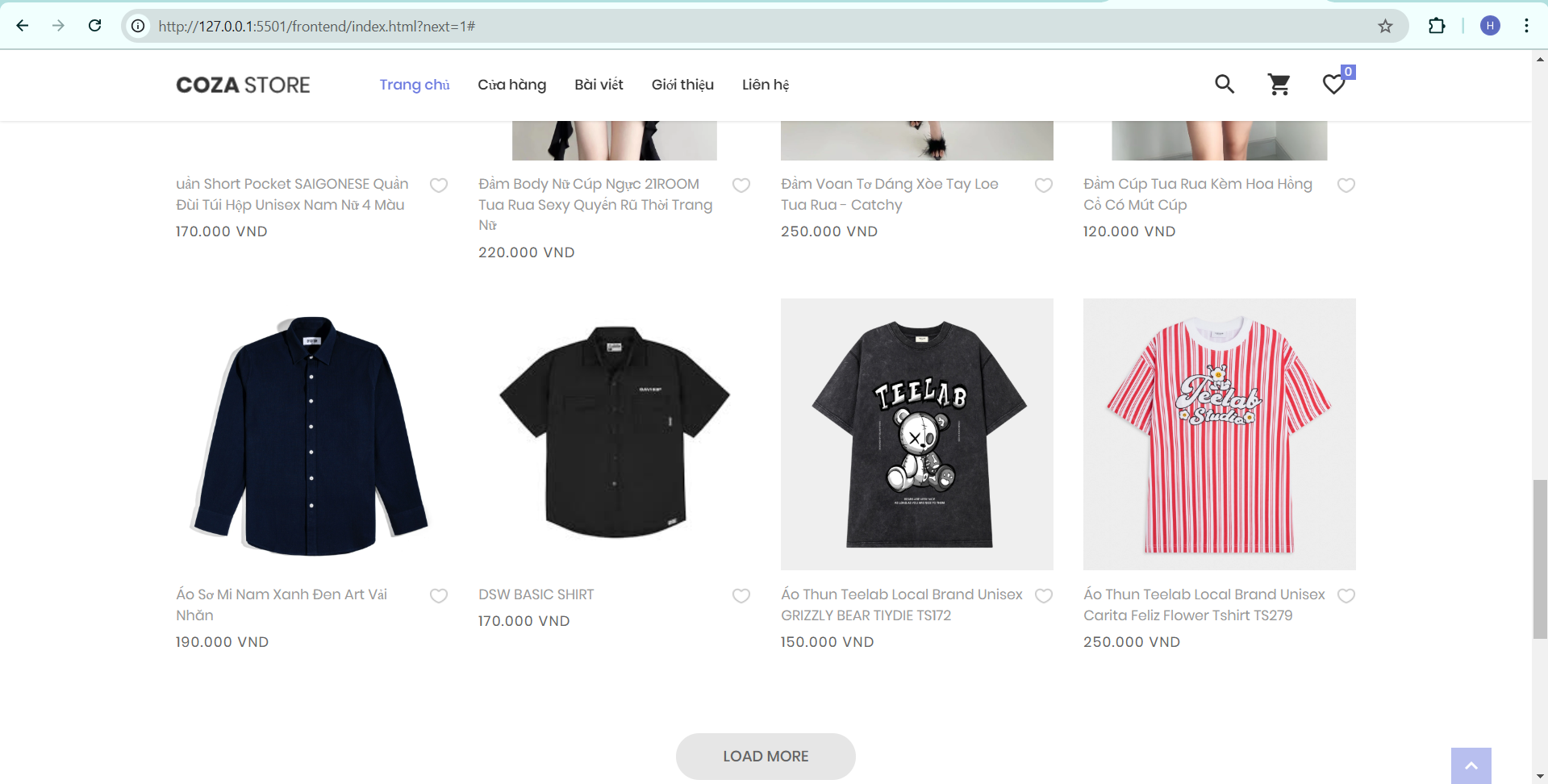
### **3.1. Trang chủ**

* **Xem danh sách sản phẩm**
* **Thêm sản phẩm vào giỏ hàng**



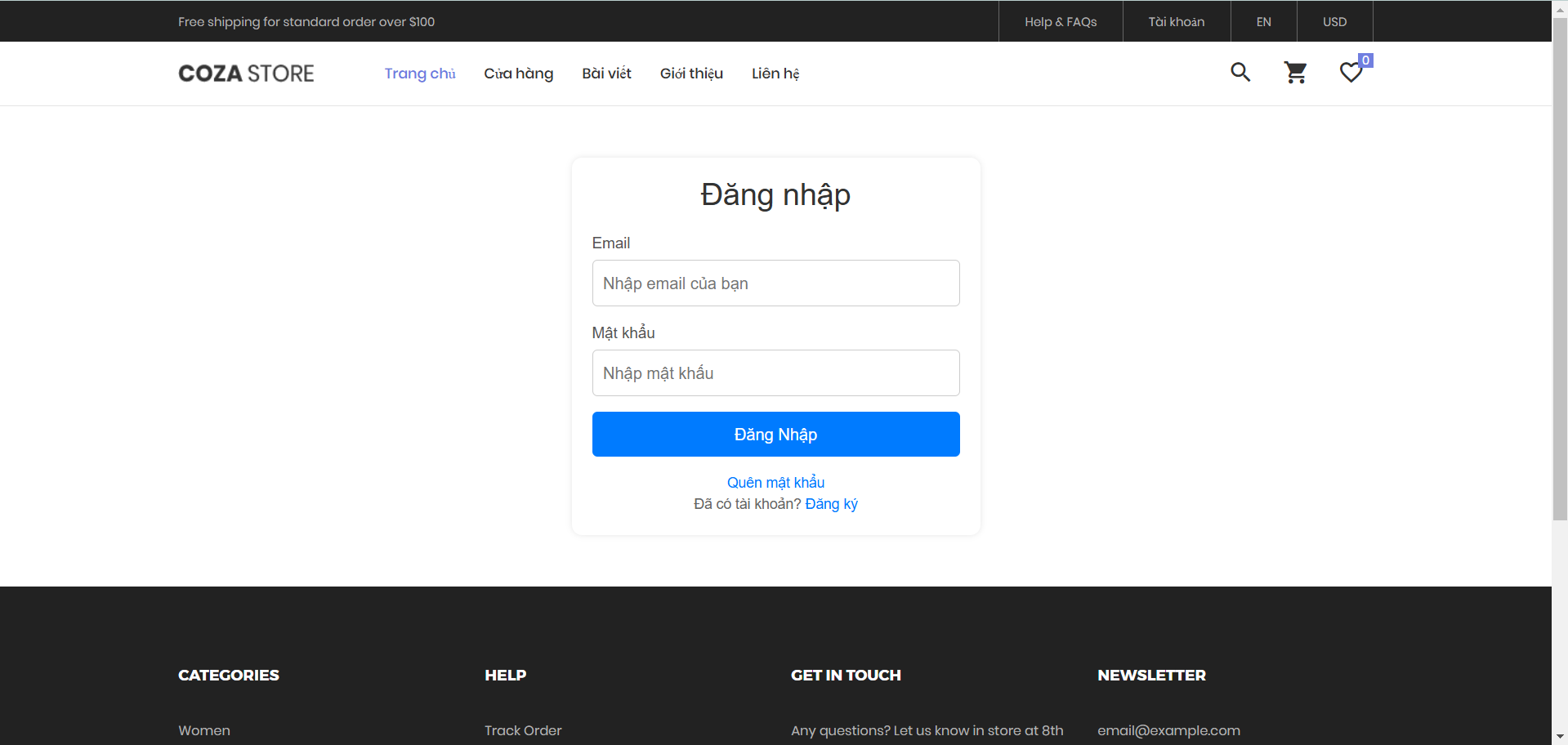


Load thêm sản phẩm

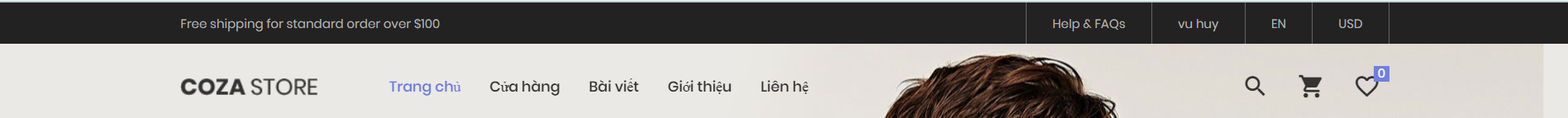


### **3.2. Trang đăng nhập/đăng ký/quên mật khẩu**

* **Đăng ký đăng nhập**
* **Quên mật khẩu**

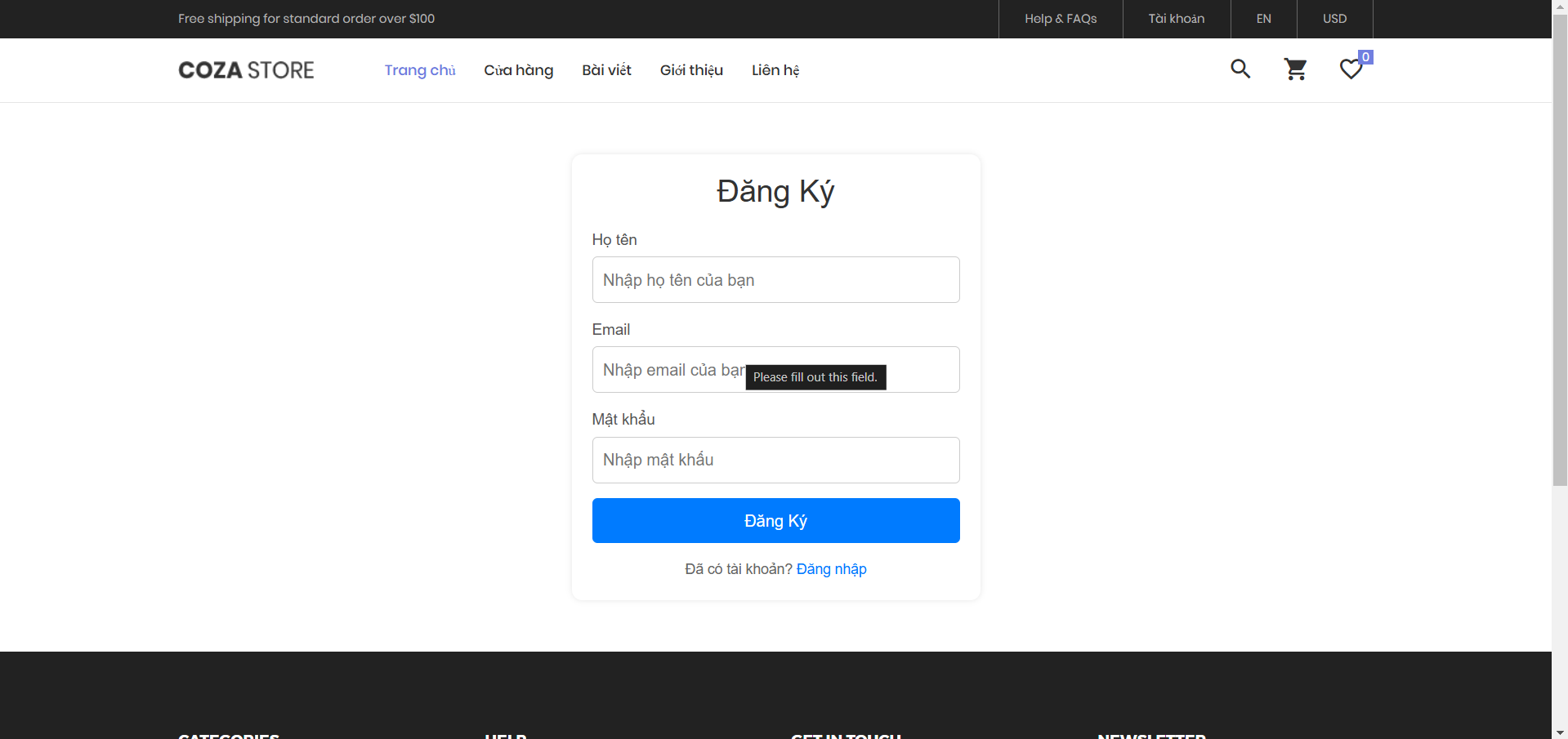


khi đăng nhập thành công

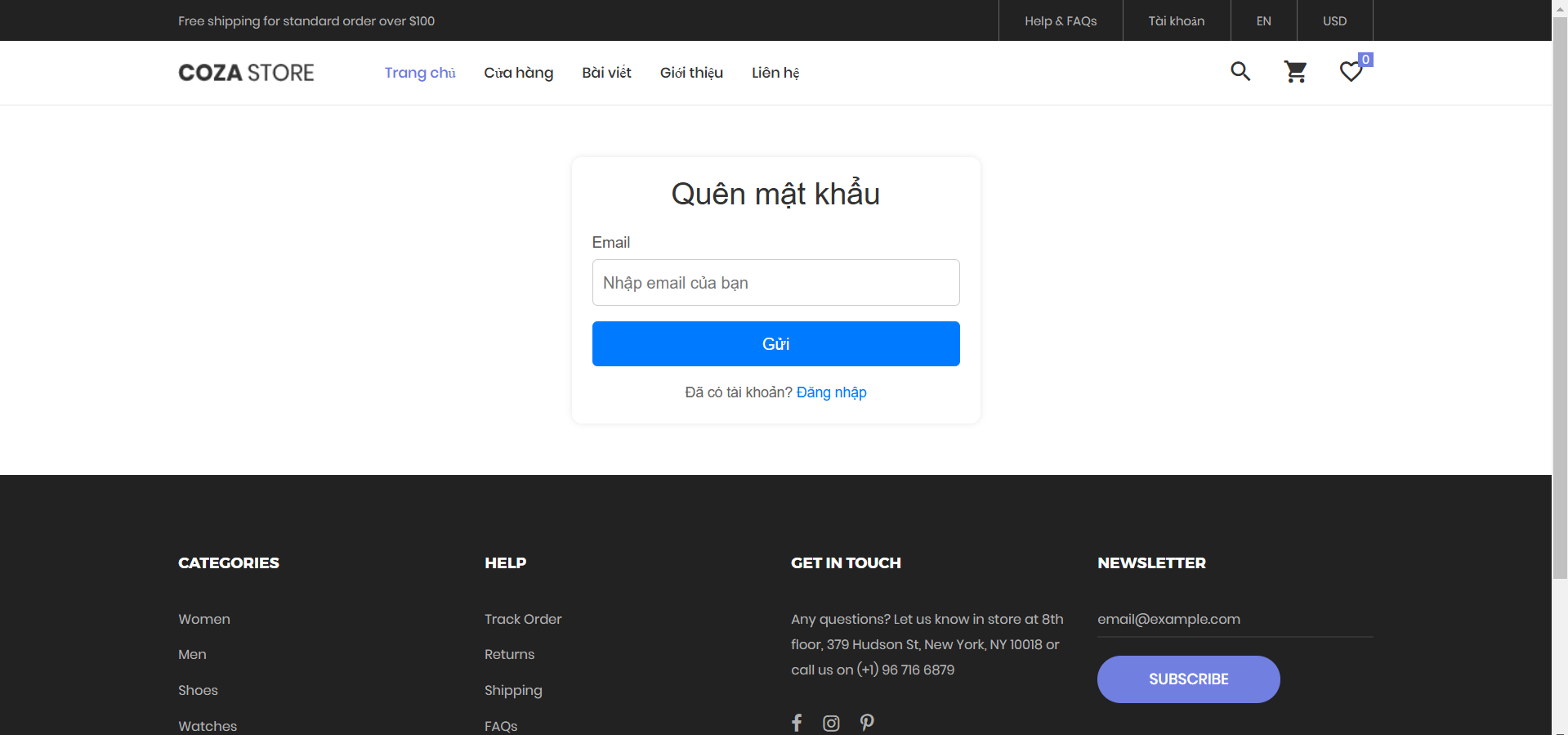


Sẽ hiển thị tên người dùng

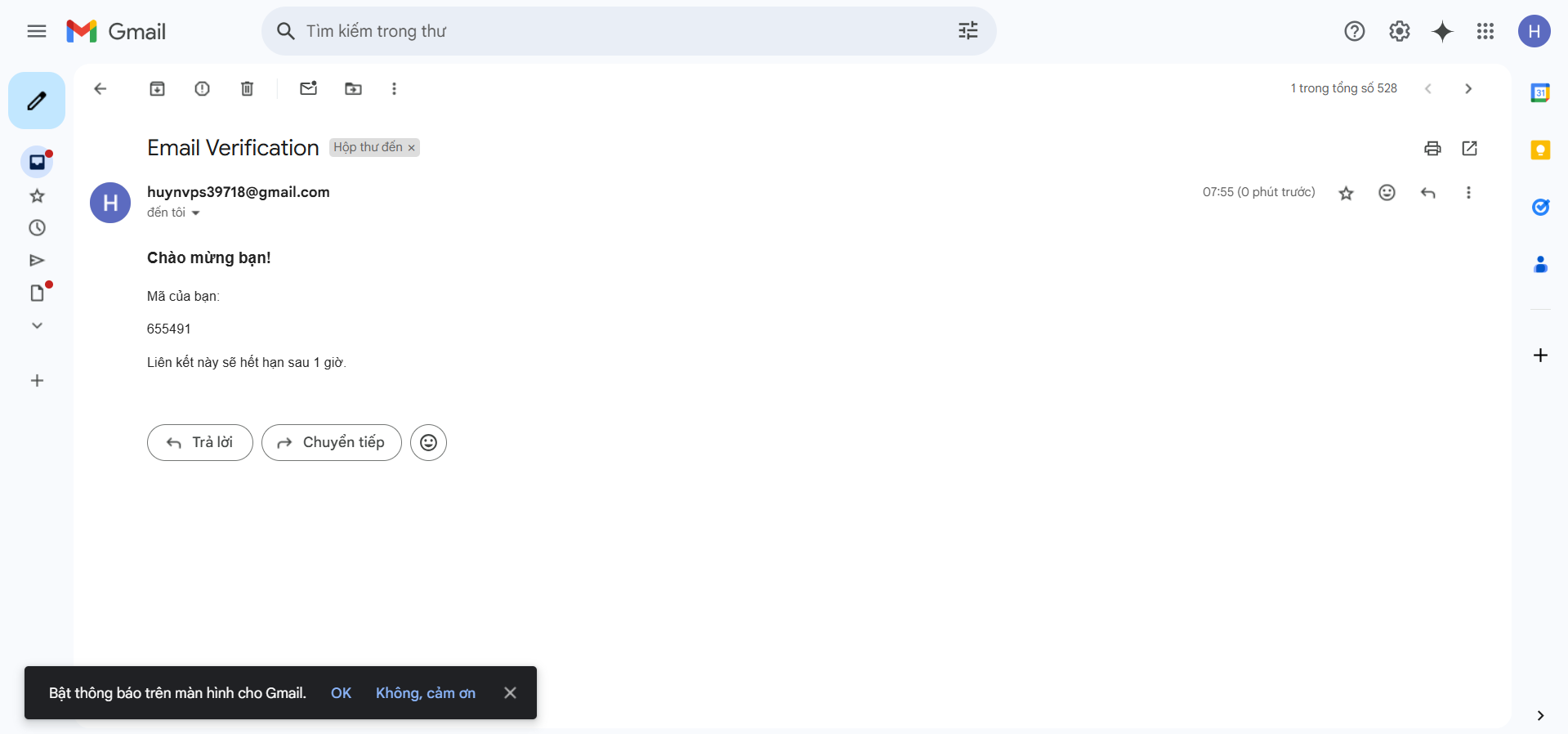
1. Đăng ký

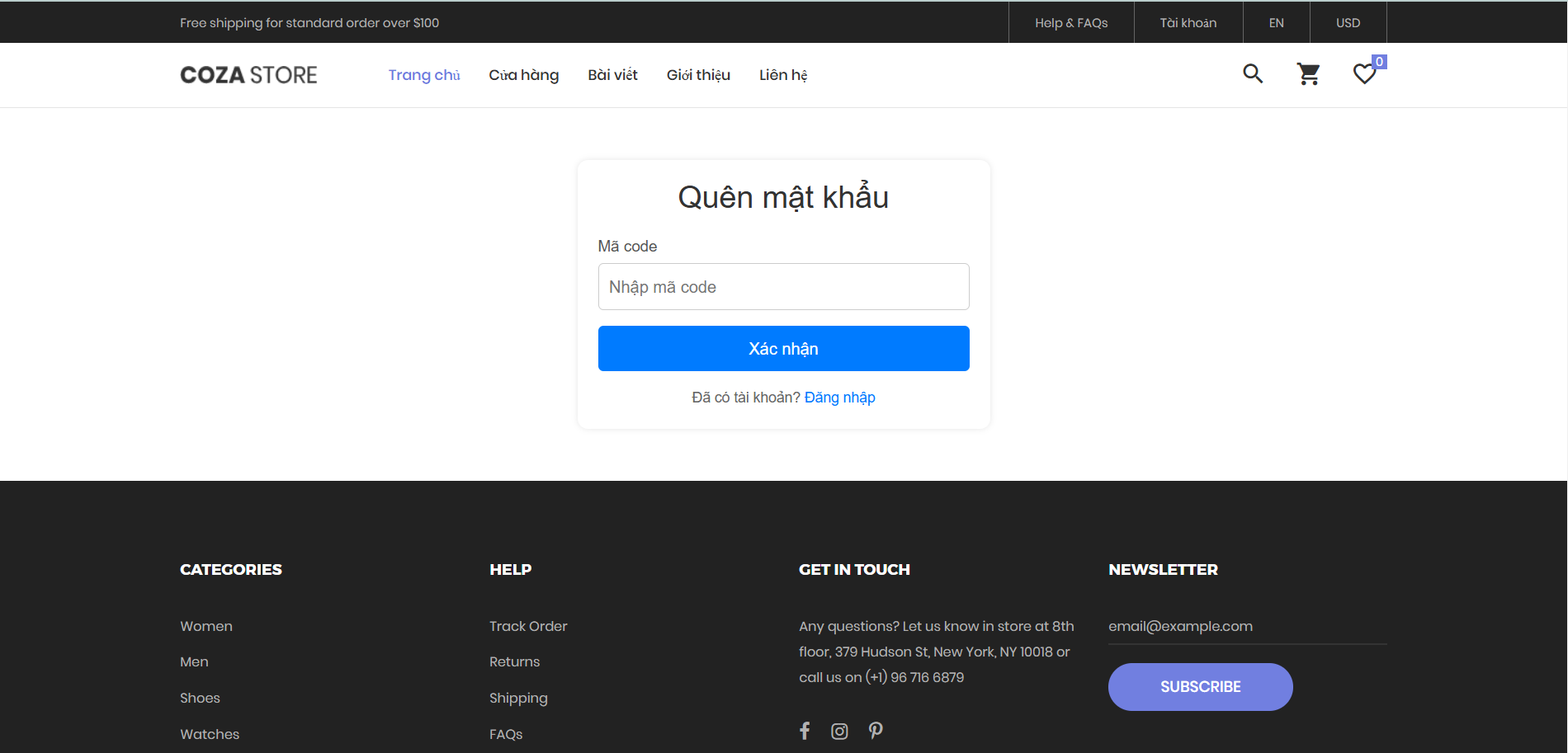


1. Quên mật khẩu



sau khi nhập gmail thì mã sẽ được gửi qua gmail

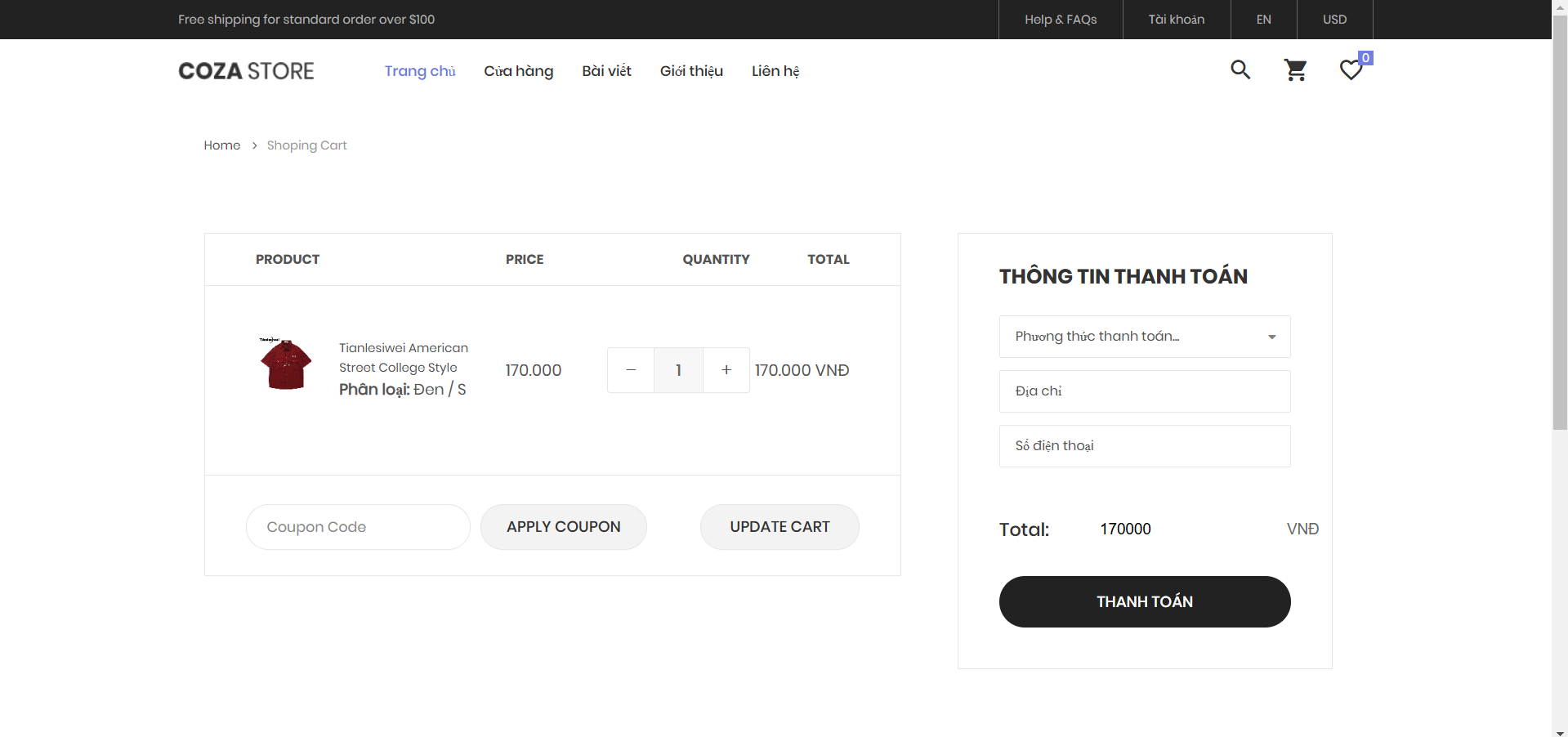




Sau khi nhập mã thì sẽ được update mật khẩu mới

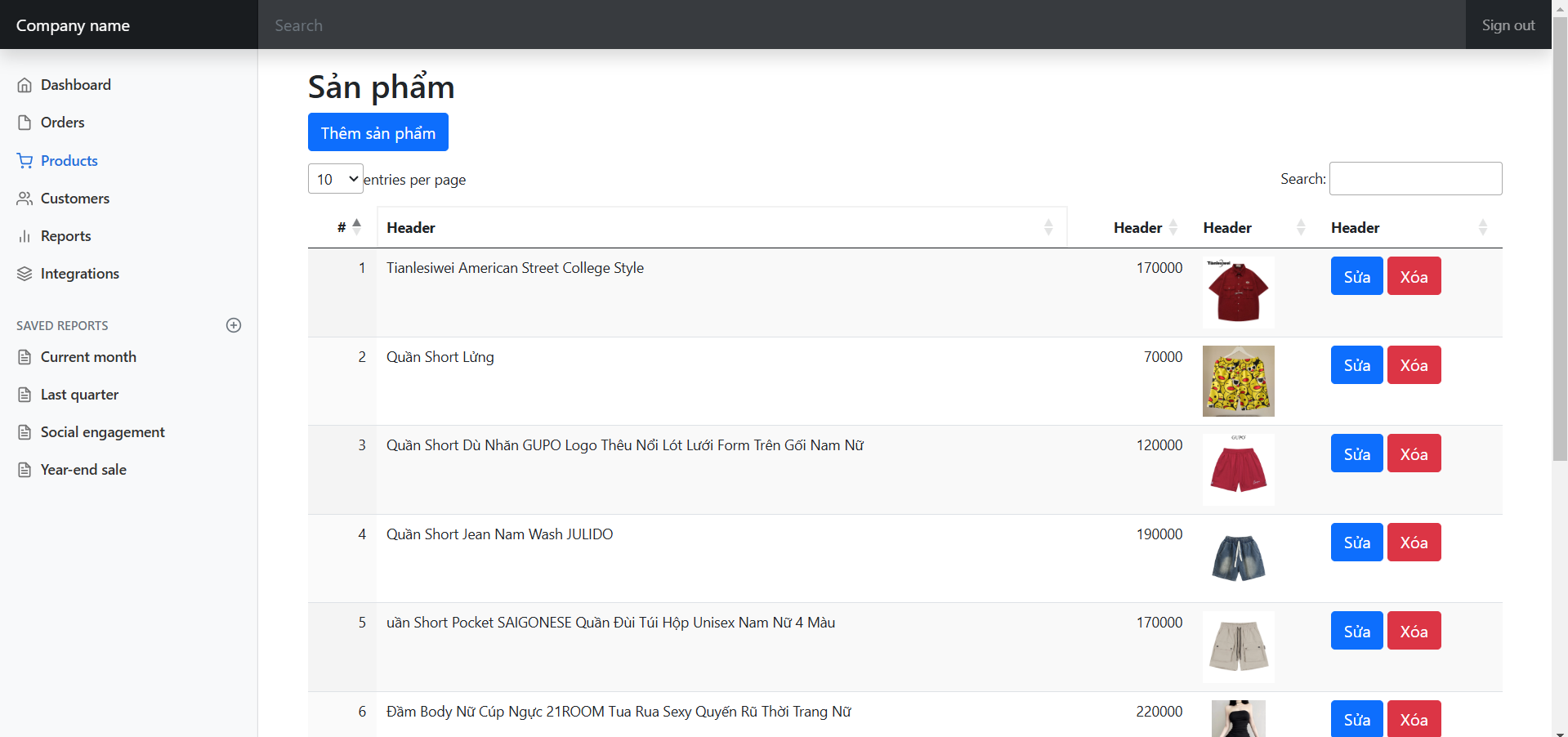
### **3.3. Trang giỏ hàng**

* **Xem và chỉnh sửa giỏ hàng**
* **Đặt hàng**

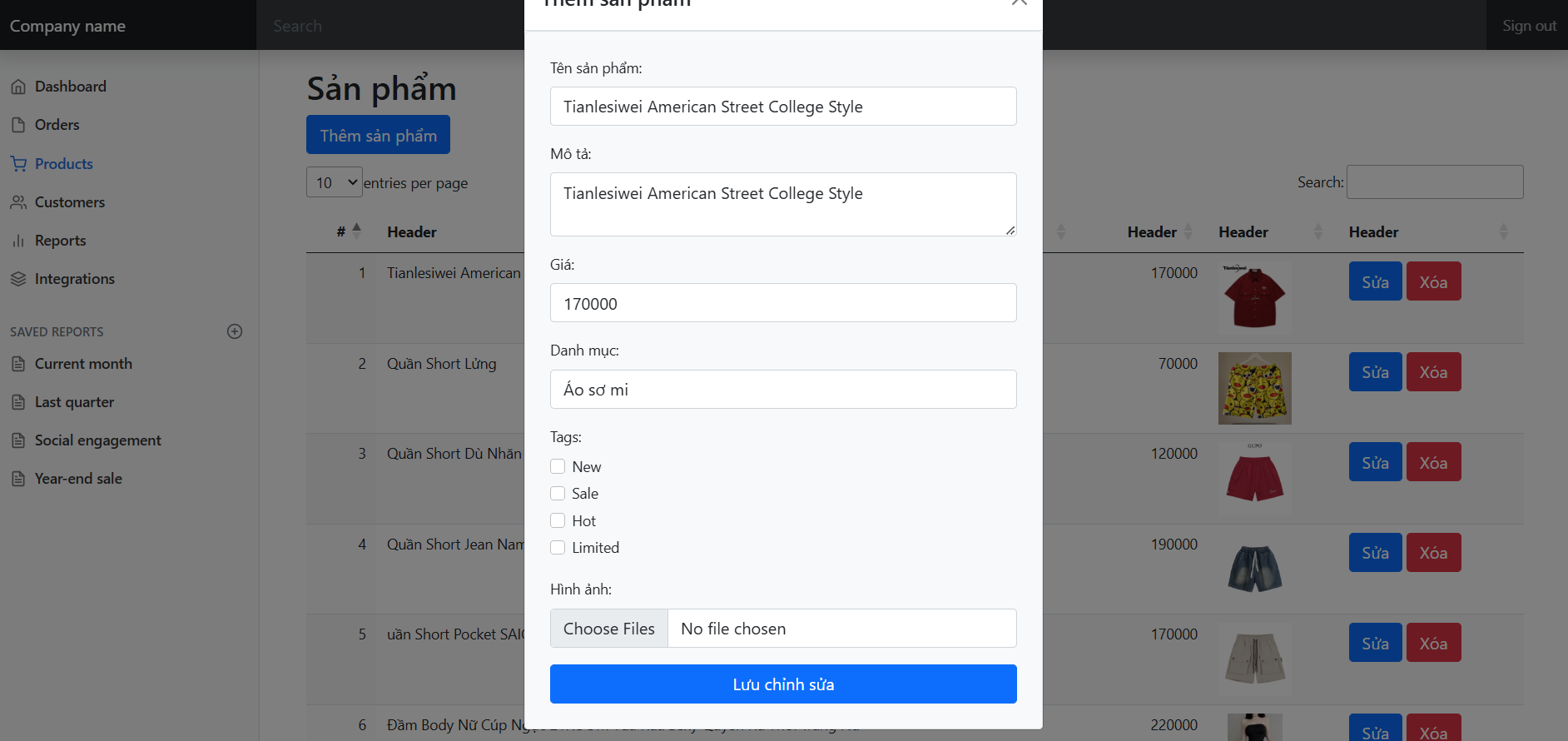


### **3.4. Trang quản lí sản phẩm**

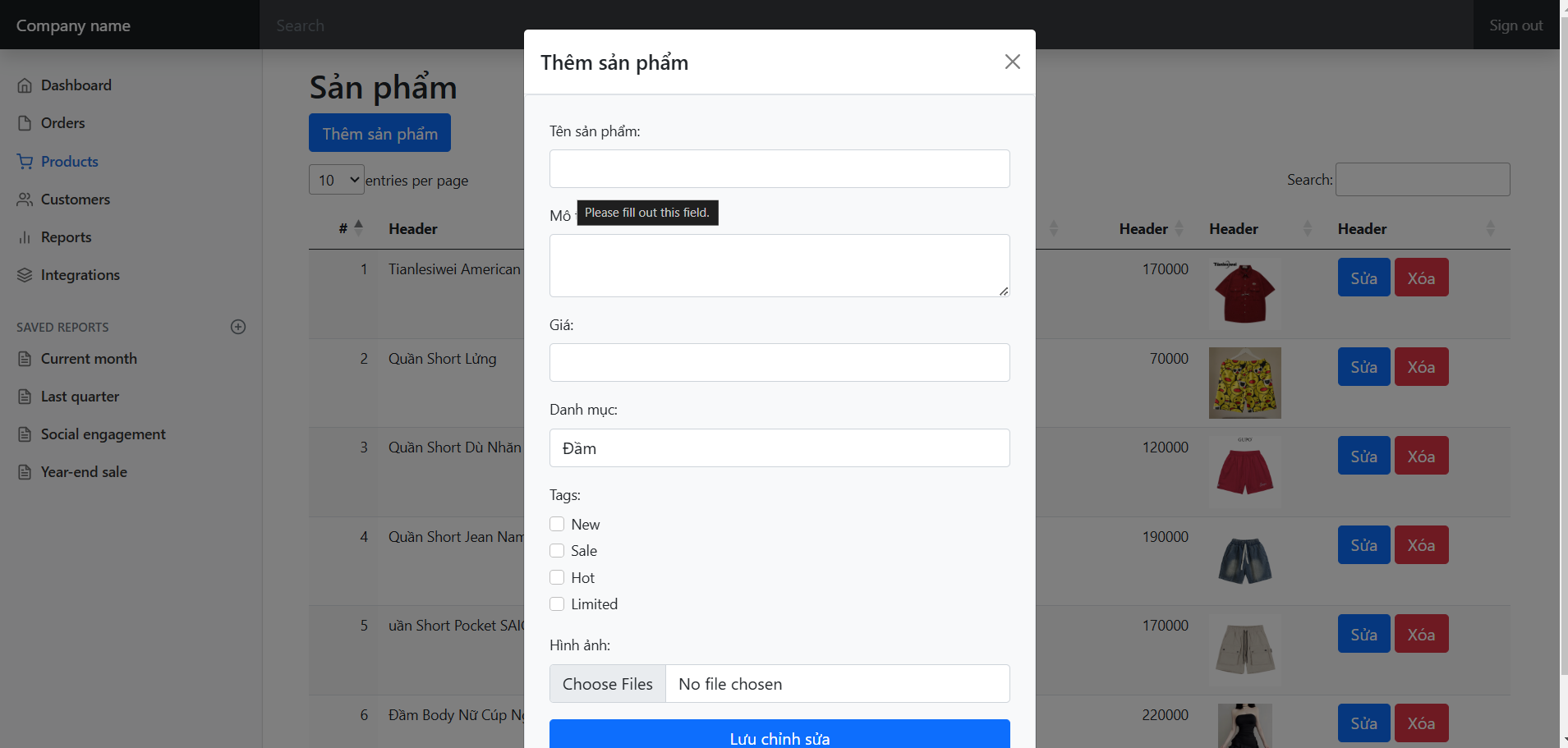
* **Xem,thêm và chỉnh sửa sản phẩm**



Sửa sản phẩm

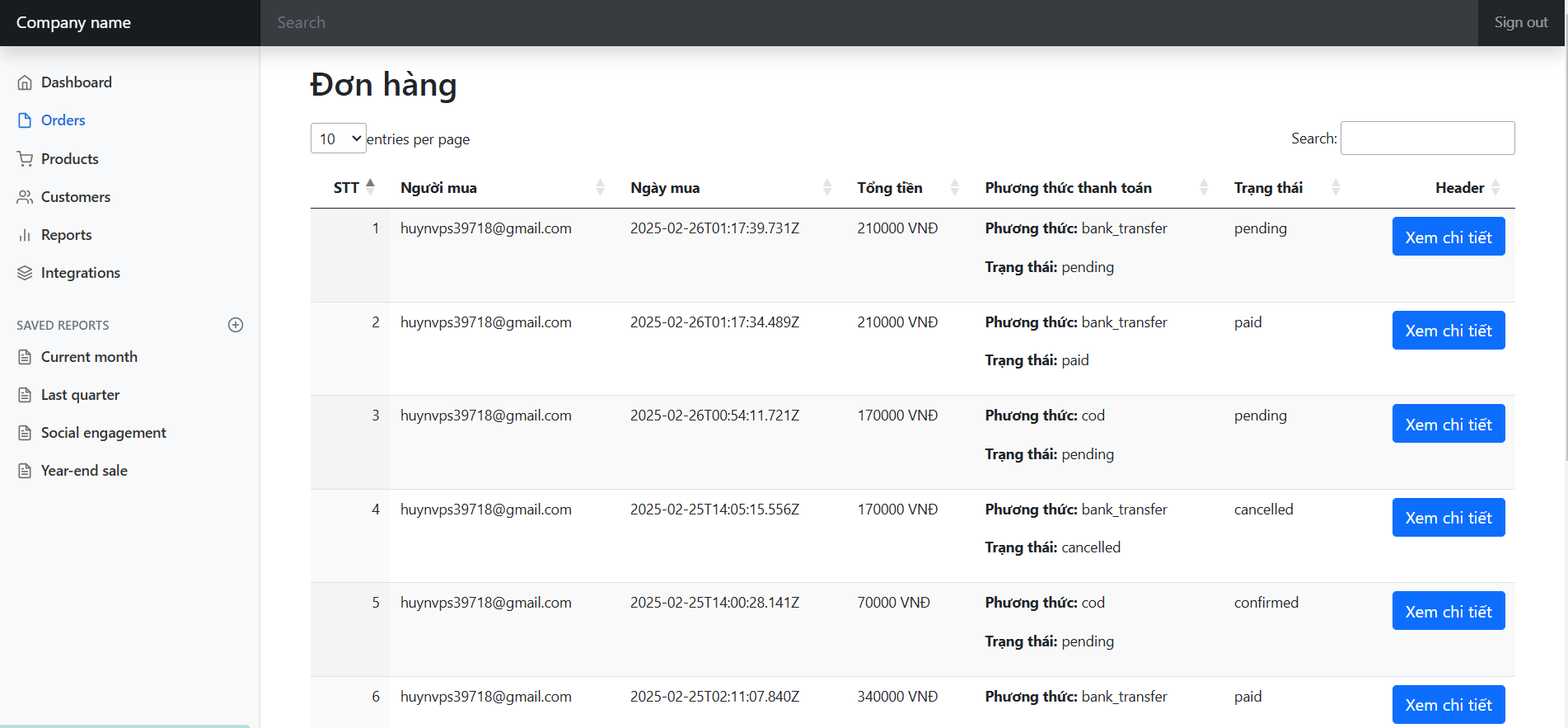


Thêm sản phẩm



### **3.4. Trang quản lí đơn hàng**

* **Xác nhận đơn hàng**

****

Xem chi tiết đơn hàng và xác nhận đơn hàng

