**TRƯỜNG ĐẠI HỌC NÔNG LÂM TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO BÀI TẬP LỚN

**Lập Trình Web**

ĐỀ TÀI:

**WEBSITE BÁN MỸ PHẨM**

\*\*\*

Giảng viên : Phan Đình Long

Danh sách thành viên

|  |  |  |
| --- | --- | --- |
| 22130071 | Nguyễn Ngọc Hân | DH22DTB |
| 22130068 | Lê Phước Hải | DH22DTB |
| 22130200 | Nguyễn Thị Tuyết Nhung | DH22DTB |

1. **Chương 1: GIỚI THIỆU VỀ ĐỀ TÀI**
   1. **Giới thiệu tổng quát về trang web**

Website bán mỹ phẩm online được xây dựng nhằm cung cấp một nền tảng thương mại điện tử chuyên về các sản phẩm chăm sóc da, trang điểm, và làm đẹp. Người dùng có thể duyệt qua các sản phẩm mỹ phẩm từ nhiều thương hiệu khác nhau, tìm kiếm sản phẩm theo danh mục, và xem các đánh giá của người dùng khác trước khi quyết định mua hàng. Giao diện của website được thiết kế trực quan, dễ sử dụng, nhằm mang lại trải nghiệm mua sắm thuận tiện cho người dùng. Hệ thống quản lý đơn hàng và giỏ hàng giúp người dùng dễ dàng theo dõi và quản lý các sản phẩm mình đã chọn. Ngoài ra, website tích hợp các phương thức thanh toán an toàn, đảm bảo bảo mật thông tin người dùng.

* 1. **Mục đích tạo ra trang web**
* Học được các kiến thức mới
* Hiểu sâu hơn về công nghệ Servlet
* Tìm hiểu về kiến thức tương tác CSDL(MySQL, Hibernate)
* Biết cách deploy website lên môi trường internet

1. **Chương 2: TỔNG QUAN VỀ CÁC ĐƯỢC CÔNG NGHỆ SỬ DỤNG**
   1. **Các ngôn ngữ lập trình và công cụ sử dụng**
      1. **HTML**

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu chính được sử dụng để tạo cấu trúc cho các trang web. HTML xác định cách hiển thị các thành phần cơ bản trên trang, như văn bản, hình ảnh, liên kết, và các phần tử đa phương tiện. Trong dự án website bán mỹ phẩm, HTML được sử dụng để xây dựng giao diện người dùng, giúp hiển thị thông tin sản phẩm, giỏ hàng, và các trang hỗ trợ khác trên website.

* + 1. **CSS**

CSS (Cascading Style Sheets) là ngôn ngữ dùng để tạo kiểu dáng cho các trang web, giúp điều chỉnh màu sắc, bố cục, phông chữ, và định dạng của các phần tử HTML. Trong website bán mỹ phẩm, CSS được sử dụng để thiết kế giao diện bắt mắt và thân thiện với người dùng, đảm bảo sự đồng bộ về màu sắc và phong cách, giúp website trở nên hấp dẫn và chuyên nghiệp hơn.

* + 1. **JAVASCRIPT**

JavaScript hay thường được gọi là JS là ngôn ngữ lập trình dùng để tạo ra các tính năng tương tác trên trang web, giúp tăng trải nghiệm người dùng, mang lại sự linh hoạt và sống động cho giao diện trang web.

* + 1. **JAVA**

Java là một ngôn ngữ lập trình đa năng, hướng đối tượng, nổi tiếng với tính ổn định, bảo mật và khả năng chạy trên nhiều nền tảng nhờ máy ảo Java (JVM). Java thường được sử dụng trong phát triển ứng dụng web, phần mềm doanh nghiệp, và hệ thống di động, nhờ cấu trúc rõ ràng, khả năng mở rộng tốt và cộng đồng hỗ trợ lớn.

* + 1. **StartUML**

StarUML là một công cụ mô hình hóa phần mềm hỗ trợ tạo các sơ đồ UML (Unified Modeling Language), được sử dụng rộng rãi trong thiết kế và phân tích hệ thống phần mềm. Công cụ này giúp các nhà phát triển dễ dàng biểu diễn cấu trúc và luồng hoạt động của phần mềm qua các sơ đồ như sơ đồ lớp, sơ đồ trình tự, và sơ đồ thành phần. Với giao diện trực quan và hỗ trợ nhiều chuẩn UML, StarUML là một công cụ hữu ích để phát triển và tài liệu hóa các dự án phần mềm một cách hiệu quả.

* + 1. **Mô hình MVC**

MVC (Model-View-Controller) là một mô hình kiến trúc phần mềm phân chia ứng dụng thành ba phần riêng biệt,với cấu trúc phân tách này, MVC giúp quản lý và bảo trì mã nguồn dễ dàng hơn, hỗ trợ phát triển linh hoạt và mở rộng ứng dụng nhanh chóng.

* **Model**: Quản lý dữ liệu và logic nghiệp vụ của ứng dụng, xử lý tương tác với cơ sở dữ liệu.
* **View**: Đảm nhiệm việc hiển thị dữ liệu và giao diện người dùng.
* **Controller**: Điều phối yêu cầu từ người dùng, xử lý logic và gửi dữ liệu đến View.
  + 1. **Cơ sở dữ liệu MySQL**

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, nổi tiếng với hiệu suất cao, độ tin cậy, và dễ sử dụng. MySQL sử dụng ngôn ngữ SQL (Structured Query Language) để quản lý và truy vấn dữ liệu, và thường được sử dụng trong các ứng dụng web để lưu trữ thông tin như tài khoản người dùng, sản phẩm, và đơn hàng. Với khả năng xử lý lượng dữ liệu lớn và hỗ trợ bảo mật cao, MySQL là một lựa chọn phổ biến cho các dự án từ nhỏ đến lớn.

* + 1. **Hibernate**

Hibernate là một framework mã nguồn mở cho Java, hỗ trợ lập trình viên quản lý cơ sở dữ liệu bằng cách chuyển đổi các đối tượng Java thành các bản ghi trong cơ sở dữ liệu (ORM - Object-Relational Mapping). Hibernate tự động hóa các thao tác như lưu trữ, truy vấn, cập nhật và xóa dữ liệu mà không cần viết nhiều mã SQL thủ công. Framework này giúp tối ưu hiệu suất truy xuất dữ liệu và hỗ trợ nhiều loại cơ sở dữ liệu, giúp ứng dụng dễ bảo trì, mở rộng và độc lập với hệ quản trị cơ sở dữ liệu.

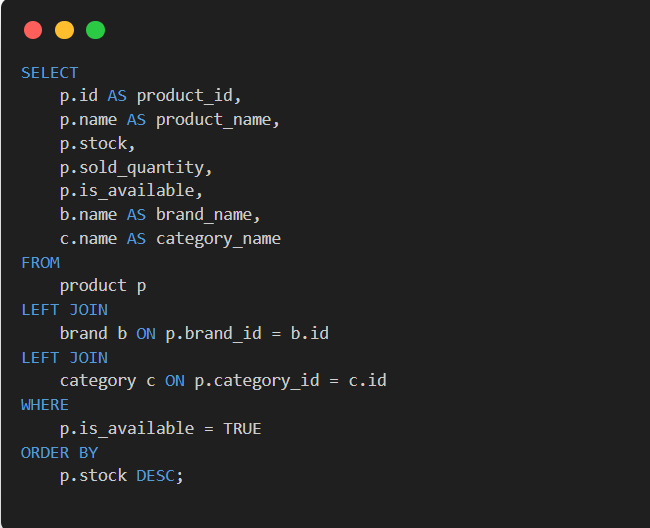
* 1. **Một số thư viện hỗ trợ**
     1. **Bootstrap**

Bootstrap là một framework CSS phổ biến, giúp thiết kế giao diện web nhanh chóng và hiệu quả. Nó cung cấp các thành phần sẵn có như lưới (grid system), biểu mẫu, nút, thanh điều hướng, và nhiều tiện ích khác. Bootstrap hỗ trợ thiết kế **responsive**, đảm bảo giao diện hiển thị đẹp trên mọi thiết bị từ máy tính đến điện thoại. Hiện tại, phiên bản mới nhất là Bootstrap 5, loại bỏ dependency với jQuery và cung cấp nhiều tính năng hiện đại hơn.

1. **Chương 3: PHÂN TÍCH THIẾT KẾ HỆ THÔNG WEBSITE**
   1. **Đặt tả yêu cầu**

* Hệ thống hoạt động trên môi trường web nhầm phục vụ khách hàng trong việc mua sắm mỹ phẩm nói chung, giúp khách hàng có thể dễ dàng tìm kiếm, đặt hàng mà không cần ra tới cửa hàng
* Bộ phần quản trị có thể dễ dàng quản lý sản phẩm, kiểm soát và thống kê được doanh số bán hàng một cách dễ dàng
* Các chức năng phía quản trị:
* Kiểm tra tình trạng hàng hoá

Câu query:



* Cập nhật sản phẩm
* Cập nhật thông tin cơ bản của sản phẩm:

UPDATE product SET name = 'Tên mới của sản phẩm', price = 100000, cost\_price = 80000, stock = 50, is\_available = TRUE, updated\_at = NOW() WHERE id = 1;

* Cập nhật trạng thái hết hàng:

UPDATE product SET is\_available = FALSE, updated\_at = NOW() WHERE stock = 0;

* Cập nhật thương hiệu hoặc danh mục của sản phẩm:

UPDATE product

SET

brand\_id = 2, -- ID thương hiệu mới

category\_id = 5, -- ID danh mục mới

updated\_at = NOW()

WHERE

id = 3; -- ID của sản phẩm cần cập nhật

* Cập nhật hàng loạt sản phẩm(ví dụ: tăng giá tất cả sản phẩm của mọi thương hiệu:

UPDATE product SET price = price + (price \* 0.1), -- Tăng giá 10% updated\_at = NOW() WHERE brand\_id = 2; -- ID của thương hiệu

* Cập nhật ảnh đại diện thương hiệu:

UPDATE product SET thumbnail = 'https://example.com/new-image.jpg', updated\_at = NOW() WHERE id = 4;

* Quản lý loại sản phẩm
* Cập nhật loại sản phẩm:

UPDATE category SET name = 'Tên mới của loại sản phẩm', parent\_id = 2, -- ID danh mục cha mới (hoặc NULL nếu không có danh mục cha) updated\_at = NOW() WHERE id = 3; -- ID của loại sản phẩm cần cập nhật

* Thêm mới loại sản phẩm:

INSERT INTO category (name, parent\_id, created\_at, updated\_at) VALUES ('Tên loại sản phẩm', NULL, NOW(), NOW()); -- NULL nếu loại sản phẩm là cấp cao nhất

* Xóa một loại danh mục:

DELETE FROM category WHERE id = 4; -- ID của loại sản phẩm cần xóa

* Lấy danh sách tất cả danh mục:

SELECT c1.id AS category\_id, c1.name AS category\_name, c2.name AS parent\_category\_name, -- Tên danh mục cha (nếu có) c1.created\_at, c1.updated\_at FROM category c1 LEFT JOIN category c2 ON c1.parent\_id = c2.id ORDER BY c1.parent\_id, c1.name;

* Lấy danh mục con của một danh mục:

SELECT id AS subcategory\_id, name AS subcategory\_name, created\_at, updated\_at FROM category WHERE parent\_id = 1; -- ID của danh mục cha

* Kiểm tra số lượng sản phẩm trong từng danh mục:

SELECT c.id AS category\_id, c.name AS category\_name, COUNT(p.id) AS product\_count FROM category c LEFT JOIN product p ON c.id = p.category\_id GROUP BY c.id, c.name ORDER BY product\_count DESC; -- Sắp xếp theo số lượng sản phẩm giảm dần

* Quản lý sản phẩm
* Them sản phẩm mới:

INSERT INTO product ( name, price, cost\_price, stock, description, thumbnail, brand\_id, category\_id, created\_at, updated\_at ) VALUES ( 'Tên sản phẩm', 200000, -- Giá bán 150000, -- Giá vốn 100, -- Tồn kho 'Mô tả sản phẩm', 'https://example.com/image.jpg', -- Đường dẫn ảnh đại diện 1, -- ID thương hiệu 2, -- ID loại sản phẩm NOW(), NOW() );

* Cập nhật thông tin sản phẩm:

UPDATE product SET name = 'Tên mới của sản phẩm', price = 220000, cost\_price = 160000, stock = 80, description = 'Mô tả cập nhật cho sản phẩm', is\_available = TRUE, -- Đặt trạng thái sản phẩm updated\_at = NOW() WHERE id = 1; -- ID sản phẩm cần cập nhật

* Cập nhật giá hang loạt:

UPDATE product SET price = price \* 1.1, -- Tăng giá 10% updated\_at = NOW() WHERE brand\_id = 1; -- ID của thương hiệu

* Xóa sản phẩm:

DELETE FROM product WHERE id = 3; -- ID của sản phẩm cần xóa

* Lấy danh sách sản phẩm:

SELECT p.id AS product\_id, p.name AS product\_name, p.price, p.cost\_price, p.stock, p.sold\_quantity, p.is\_available, b.name AS brand\_name, c.name AS category\_name, p.created\_at, p.updated\_at FROM product p LEFT JOIN brand b ON p.brand\_id = b.id LEFT JOIN category c ON p.category\_id = c.id ORDER BY p.created\_at DESC;

* Lấy danh sách sản phẩm còn hang:

SELECT id, name, price, stock FROM product WHERE is\_available = TRUE AND stock > 0 ORDER BY stock DESC;

* Lấy sản phẩm theo thương hiệu hoặc danh mục:

SELECT id, name, price, stock FROM product WHERE brand\_id = 1 -- ID thương hiệu AND category\_id = 2; -- ID loại sản phẩm

* Kiểm tra số lượng sản phẩm đã bán và tồn kho:

SELECT id AS product\_id, name AS product\_name, stock, sold\_quantity, (stock + sold\_quantity) AS total\_quantity FROM product ORDER BY sold\_quantity DESC;

* Lấy sản phẩm đánh giá cao nhất:

SELECT p.id AS product\_id, p.name AS product\_name, AVG(r.rating) AS average\_rating, COUNT(r.id) AS total\_reviews FROM product p LEFT JOIN review r ON p.id = r.product\_id GROUP BY p.id, p.name ORDER BY average\_rating DESC, total\_reviews DESC LIMIT 10;

* Kiểm tra sản phẩm sắp hết hàng:

SELECT id AS product\_id, name AS product\_name, stock FROM product WHERE stock <= 10 -- Sản phẩm có tồn kho nhỏ hơn hoặc bằng 10 ORDER BY stock ASC;

* Quản lý nhãn hàng:
* Thêm mới nhãn hàng:

INSERT INTO brand (name, logo, created\_at, updated\_at) VALUES ('Tên nhãn hàng', 'https://example.com/logo.jpg', NOW(), NOW());

* Cập nhật thông tin nhãn hang:

UPDATE brand SET name = 'Tên mới của nhãn hàng', logo = 'https://example.com/new-logo.jpg', updated\_at = NOW() WHERE id = 1; -- ID của nhãn hàng cần cập nhật

* Xóa nhãn hàng:

DELETE FROM brand WHERE id = 2 AND NOT EXISTS ( SELECT 1 FROM product WHERE brand\_id = 2 );

* Lấy danh sách tất cả nhãn hàng:

SELECT id AS brand\_id, name AS brand\_name, logo, created\_at, updated\_at FROM brand ORDER BY name ASC; -- Sắp xếp theo tên nhãn hàng

* Lấy nhãn hàng kèm số lượng sản phẩm:

SELECT b.id AS brand\_id, b.name AS brand\_name, b.logo, COUNT(p.id) AS product\_count FROM brand b LEFT JOIN product p ON b.id = p.brand\_id GROUP BY b.id, b.name, b.logo ORDER BY product\_count DESC; -- Sắp xếp theo số lượng sản phẩm giảm dần

* Tìm nhãn hàng theo tên hoặc từ khóa:

SELECT id AS brand\_id, name AS brand\_name, logo FROM brand WHERE name LIKE '%Nike%' -- Tìm nhãn hàng chứa từ "Nike" ORDER BY name ASC;

* Quản lý đơn hàng
* Thêm mới đơn hàng:

INSERT INTO `order` ( address\_id, user\_id, status, payment\_method, shipping\_fee, note, total\_price, order\_date, created\_at ) VALUES ( 1, -- ID địa chỉ nhận hàng 2, -- ID người dùng 'PENDING', -- Trạng thái đơn hàng 'COD', -- Phương thức thanh toán (COD, Online) 30000, -- Phí vận chuyển 'Giao hàng nhanh nhất', -- Ghi chú 500000, -- Tổng giá trị đơn hàng NOW(), NOW() );

* Them chi tiết đơn hàng:

INSERT INTO order\_detail (order\_id, product\_id, quantity, total\_price) VALUES (1, 3, 2, 200000), (1, 4, 1, 300000);

* Cập nhật trạng thái đơn hàng:

UPDATE `order` SET status = 'CONFIRMED', -- Trạng thái mới confirm\_at = NOW() -- Thời gian xác nhận WHERE id = 1;

* Lấy danh sách tất cả đơn hàng:

SELECT o.id AS order\_id, o.status, o.payment\_method, o.total\_price, o.shipping\_fee, u.full\_name AS customer\_name, a.recipient\_name, a.recipient\_phone, a.city, o.order\_date, o.confirm\_at FROM `order` o LEFT JOIN user u ON o.user\_id = u.id LEFT JOIN address a ON o.address\_id = a.id ORDER BY o.order\_date DESC;

* Lấy chi tiết đơn hàng:

SELECT od.id AS order\_detail\_id, p.name AS product\_name, p.price AS product\_price, od.quantity, od.total\_price FROM order\_detail od LEFT JOIN product p ON od.product\_id = p.id WHERE od.order\_id = 1;

* Thống kê số đơn hàng theo trạng thái:

SELECT status, COUNT(\*) AS total\_orders FROM `order` GROUP BY status;

* Tìm đơn hàng của khách hàng cụ thể:

SELECT o.id AS order\_id, o.status, o.total\_price, o.order\_date, o.confirm\_at FROM `order` o WHERE o.user\_id = 2 -- ID người dùng ORDER BY o.order\_date DESC;

* Lấy danh sách đơn hàng đang giao hoặc chờ giao hàng:

SELECT o.id AS order\_id, o.status, o.total\_price, a.recipient\_name, a.recipient\_phone, a.city, o.order\_date FROM `order` o LEFT JOIN address a ON o.address\_id = a.id WHERE o.status IN ('SHIPPING', 'PREPARING') ORDER BY o.order\_date ASC;

* Lấy đơn hàng có giá trị cao nhất:

SELECT o.id AS order\_id, o.total\_price, u.full\_name AS customer\_name, a.recipient\_name, a.city, o.order\_date FROM `order` o LEFT JOIN user u ON o.user\_id = u.id LEFT JOIN address a ON o.address\_id = a.id ORDER BY o.total\_price DESC LIMIT 1;

* Quản lý người dùng:
* Thêm người dùng mới:

INSERT INTO user ( full\_name, email, password, phone, date\_of\_birth, gender, avatar, status, created\_at, updated\_at ) VALUES ( 'Nguyen Van A', -- Họ tên 'example@example.com', -- Email 'hashed\_password', -- Mật khẩu (hashed) '0123456789', -- Số điện thoại '1990-01-01', -- Ngày sinh 'MALE', -- Giới tính NULL, -- Avatar 'ACTIVE', -- Trạng thái NOW(), NOW() );

* Cập nhật thông tin người dùng:

UPDATE user SET full\_name = 'Nguyen Van B', phone = '0987654321', gender = 'OTHER', updated\_at = NOW() WHERE id = 1; -- ID người dùng cần cập nhật

* Cập nhật trạng thái người dùng:

UPDATE user SET status = 'INACTIVE', updated\_at = NOW() WHERE id = 1; -- ID người dùng

* Lấy danh sách tất cả người dùng:

SELECT id AS user\_id, full\_name, email, phone, gender, status, created\_at, updated\_at FROM user ORDER BY created\_at DESC;

* Lấy thông tin chi tiết của người dùng:

SELECT u.id AS user\_id, u.full\_name, u.email, u.phone, u.date\_of\_birth, u.gender, u.avatar, u.status, u.last\_login, u.created\_at, u.updated\_at, GROUP\_CONCAT(r.name) AS roles -- Lấy các vai trò của người dùng FROM user u LEFT JOIN user\_has\_role ur ON u.id = ur.user\_id LEFT JOIN role r ON ur.role\_id = r.id WHERE u.id = 1 -- ID người dùng cần lấy chi tiết GROUP BY u.id;

* Lấy danh sách người dùng theo trạng thái:

SELECT id AS user\_id, full\_name, email, status FROM user WHERE status = 'ACTIVE' ORDER BY full\_name ASC;

* Thêm vai trò người dùng:

INSERT INTO user\_has\_role (user\_id, role\_id) VALUES (1, 2);

* Lấy danh sách người dùng và vai trò:

SELECT u.id AS user\_id, u.full\_name, u.email, GROUP\_CONCAT(r.name) AS roles FROM user u LEFT JOIN user\_has\_role ur ON u.id = ur.user\_id LEFT JOIN role r ON ur.role\_id = r.id GROUP BY u.id ORDER BY u.full\_name ASC;

* Cập nhật thời gian đăng nhập cuối của người dùng:

UPDATE user SET last\_login = NOW() WHERE id = 1;

* Thống kê doanh số
* Thống kê doanh thu:

SELECT

SUM(total\_price) AS total\_revenue,

COUNT(\*) AS total\_completed\_orders

FROM

`order`

WHERE

status = 'DELIVERED';

* Thống kê số lượng người dùng theo trạng thái:

SELECT status, COUNT(\*) AS total\_users FROM user GROUP BY status;

* Thống kê số lượng sản phẩm đã bán theo từng danh mục:

SELECT c.name AS category\_name, SUM(od.quantity) AS total\_sold FROM order\_detail od JOIN product p ON od.product\_id = p.id JOIN category c ON p.category\_id = c.id GROUP BY c.id ORDER BY total\_sold DESC;

* Thống kê sản phẩm bán chạy nhất:

SELECT p.id AS product\_id, p.name AS product\_name, SUM(od.quantity) AS total\_sold FROM order\_detail od JOIN product p ON od.product\_id = p.id GROUP BY p.id ORDER BY total\_sold DESC LIMIT 1;

* Thống kê doanh thu theo tháng:

SELECT DATE\_FORMAT(order\_date, '%Y-%m') AS month, SUM(total\_price) AS monthly\_revenue FROM `order` WHERE status = 'DELIVERED' GROUP BY DATE\_FORMAT(order\_date, '%Y-%m') ORDER BY month ASC;

* Thống kê khách hàng mua nhiều hàng nhất:

SELECT u.id AS user\_id, u.full\_name AS customer\_name, COUNT(o.id) AS total\_orders, SUM(o.total\_price) AS total\_spent FROM `order` o JOIN user u ON o.user\_id = u.id WHERE o.status = 'DELIVERED' GROUP BY u.id ORDER BY total\_spent DESC LIMIT 1;

* Thống kê sản phẩm tồn kho theo thương hiệu:

SELECT b.name AS brand\_name, SUM(p.stock) AS total\_stock FROM product p JOIN brand b ON p.brand\_id = b.id GROUP BY b.id ORDER BY total\_stock DESC;

* Thống kê số lượng sản phẩm đã bán theo từng khu vực:

SELECT a.region AS region\_name, SUM(od.quantity) AS total\_sold FROM order\_detail od JOIN `order` o ON od.order\_id = o.id JOIN address a ON o.address\_id = a.id GROUP BY a.region;

* Thống kê đánh giá sản phẩm:

SELECT p.id AS product\_id, p.name AS product\_name, COUNT(r.id) AS total\_reviews, AVG(r.rating) AS average\_rating FROM review r JOIN product p ON r.product\_id = p.id GROUP BY p.id ORDER BY average\_rating DESC;

* Các chức năng phía khách hàng:
* Xem sản phẩm
* Cơ bản nhất:

SELECT p.id AS product\_id, p.name AS product\_name, p.price, p.stock, p.thumbnail, b.name AS brand\_name, c.name AS category\_name FROM product p LEFT JOIN brand b ON p.brand\_id = b.id LEFT JOIN category c ON p.category\_id = c.id WHERE p.is\_available = TRUE -- Chỉ hiển thị sản phẩm còn bán ORDER BY p.created\_at DESC;

* Hiển thị sản phẩm yêu thích của khách hàng:

SELECT p.id AS product\_id, p.name AS product\_name, p.price, p.thumbnail FROM wishlist w JOIN product p ON w.product\_id = p.id WHERE w.user\_id = 1 -- ID khách hàng hiện tại ORDER BY w.created\_at DESC;

* Tìm kiếm sản phẩm
* Theo tên hoặc thương hiệu:

SELECT p.id AS product\_id, p.name AS product\_name, p.price, p.thumbnail, b.name AS brand\_name FROM product p LEFT JOIN brand b ON p.brand\_id = b.id WHERE p.is\_available = TRUE AND (p.name LIKE '%từ\_khóa%' OR b.name LIKE '%từ\_khóa%') -- Tìm theo từ khóa ORDER BY p.name ASC;

* Lọc theo danh mục:

SELECT p.id AS product\_id, p.name AS product\_name, p.price, p.thumbnail, p.stock FROM product p LEFT JOIN category c ON p.category\_id = c.id WHERE p.is\_available = TRUE AND c.id = 1 -- ID của danh mục cần lọc ORDER BY p.price ASC;

* Đặt mua sản phẩm
* Tạo đơn hàng mới:

INSERT INTO `order` ( user\_id, address\_id, payment\_method, shipping\_fee, note, total\_price, order\_date, status ) VALUES ( 1, -- ID khách hàng 2, -- ID địa chỉ giao hàng 'CASH\_ON\_DELIVERY', -- Phương thức thanh toán 50000, -- Phí vận chuyển 'Giao hàng trong giờ hành chính.', -- Ghi chú 1000000, -- Tổng giá trị đơn hàng NOW(), -- Ngày đặt hàng 'PENDING' -- Trạng thái đơn hàng );

* Thêm chi tiết sản phẩm vào đơn hàng:

INSERT INTO `order\_detail` ( order\_id, product\_id, quantity, total\_price ) VALUES (1, 3, 2, 400000), -- Sản phẩm 3, số lượng 2, tổng giá 400,000 (1, 5, 1, 600000);

* Cập nhật số lượng sản phẩm trong kho:

UPDATE `product` SET stock = stock - 2 WHERE id = 3; UPDATE `product` SET stock = stock - 1 WHERE id = 5;

* Xác nhận đơn hàng:

UPDATE `order` SET status = 'CONFIRMED', confirm\_at = NOW() WHERE id = 1;

* Hiển thị thông tin đơn hàng:

SELECT o.id AS order\_id, o.order\_date, o.status, o.payment\_method, o.shipping\_fee, o.total\_price, od.product\_id, p.name AS product\_name, od.quantity, od.total\_price AS product\_total\_price, a.recipient\_name, a.recipient\_phone, a.city, a.district, a.ward, a.note AS address\_note FROM `order` o JOIN order\_detail od ON o.id = od.order\_id JOIN product p ON od.product\_id = p.id JOIN address a ON o.address\_id = a.id WHERE o.id = 1;

* Thanh toán trực tuyến
* Xác nhận thanh toán:

INSERT INTO `payment` ( order\_id, user\_id, method, status, amount, transaction\_id ) VALUES ( 1, -- ID đơn hàng 1, -- ID khách hàng 'CASH\_ON\_DELIVERY', -- Phương thức thanh toán 'COMPLETED', -- Trạng thái thanh toán 1000000, -- Tổng giá trị thanh toán NULL -- Mã giao dịch (nếu không có để NULL) );

* Đăng ký tài khoản
* Thêm thông tin người dùng:

INSERT INTO `user` ( full\_name, email, password, phone, date\_of\_birth, gender, avatar, status, created\_at, updated\_at ) VALUES ( 'Nguyen Van A', -- Tên đầy đủ 'nguyenvana@example.com', -- Email 'hashed\_password', -- Mật khẩu đã mã hóa '0123456789', -- Số điện thoại '1990-01-01', -- Ngày sinh 'MALE', -- Giới tính (MALE, FEMALE, OTHER) NULL, -- Avatar (nếu có thể để null) 'NONE', -- Trạng thái (NONE, ACTIVE, INACTIVE) NOW(), -- Thời gian tạo tài khoản NOW() -- Thời gian cập nhật );

* Gắn vai trò người dùng:

INSERT INTO `user\_has\_role` ( user\_id, role\_id ) VALUES ( (SELECT LAST\_INSERT\_ID()), -- ID của người dùng vừa tạo 2 -- ID vai trò "USER" trong bảng `role` );

* Đăng nhập tài khoản
* Kiểm tra tài khoản tồn tại:

SELECT id, full\_name, password, -- Mật khẩu đã mã hóa status -- Trạng thái tài khoản (ACTIVE, INACTIVE, NONE) FROM `user` WHERE email = 'nguyenvana@example.com';

* Kiểm tra trạng thái tài khoản:

SELECT id, full\_name, password, status FROM `user` WHERE email = 'nguyenvana@example.com' AND status = 'ACTIVE';

* Cập nhật thời gian đăng nhập:

UPDATE `user` SET last\_login = NOW() WHERE id = 1;

* Trả về thông tin người dùng sau khi đăng nhập:

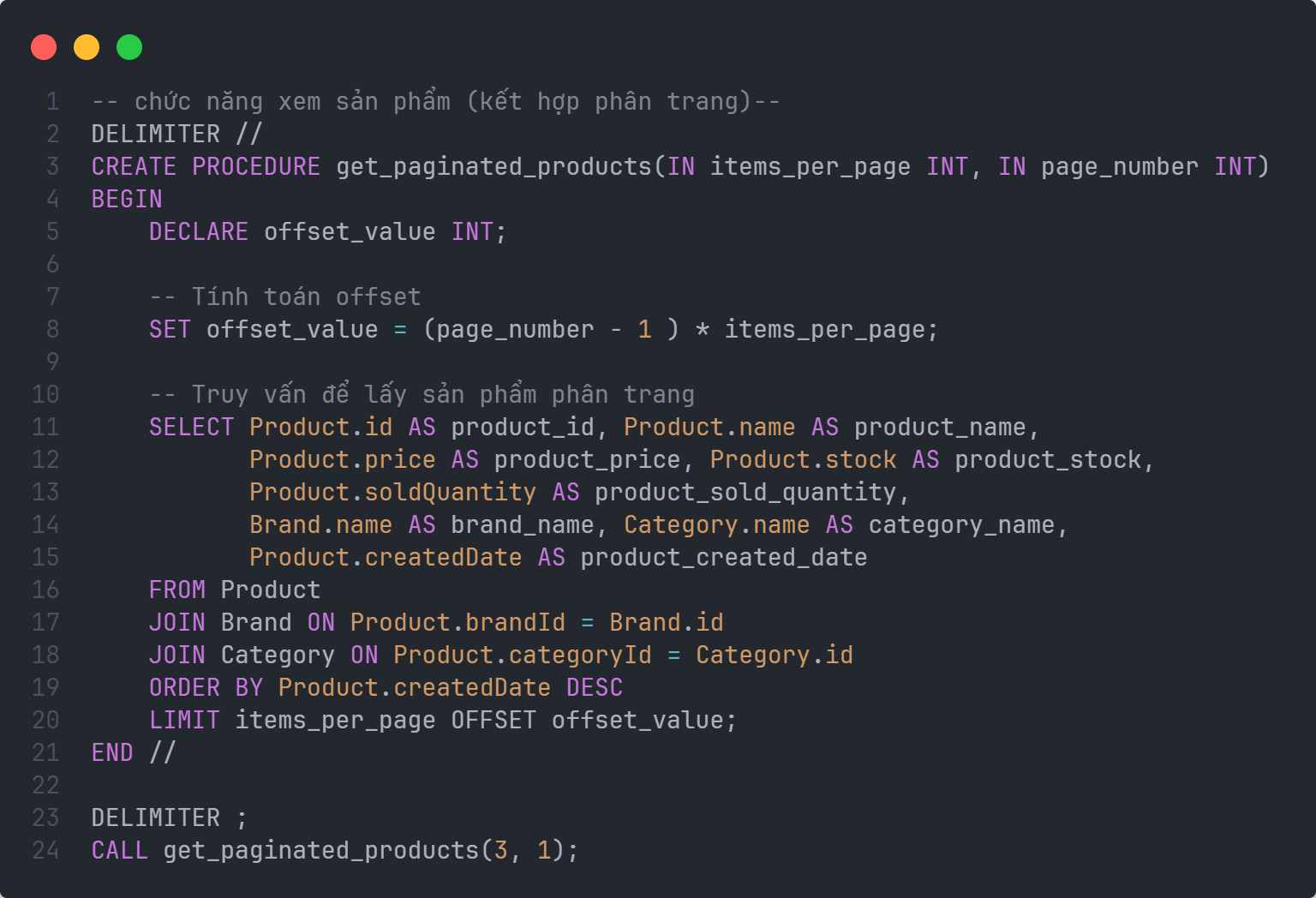
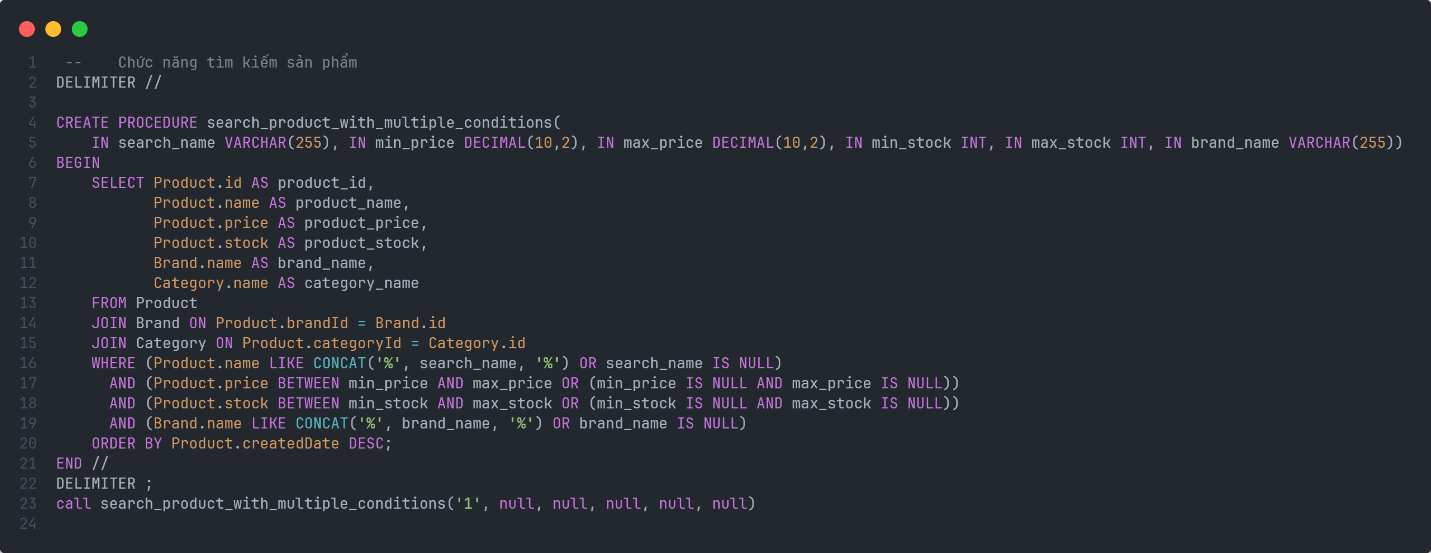
SELECT id, full\_name, email, status, last\_login FROM `user` WHERE id = 1;

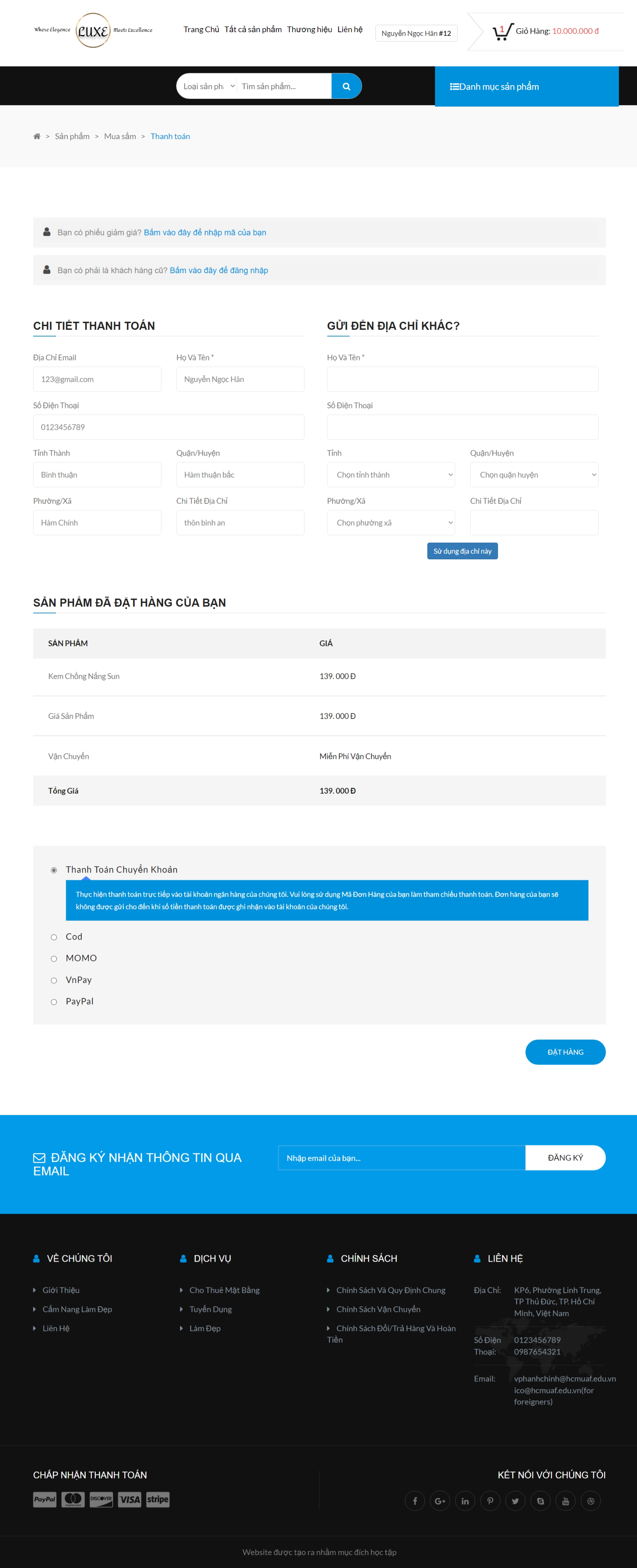
* Xem thông tin cá nhân

SELECT id, full\_name, email, phone, date\_of\_birth, gender, avatar, status, last\_login, created\_at, updated\_at FROM `user` WHERE email = 'nguyenvana@example.com';

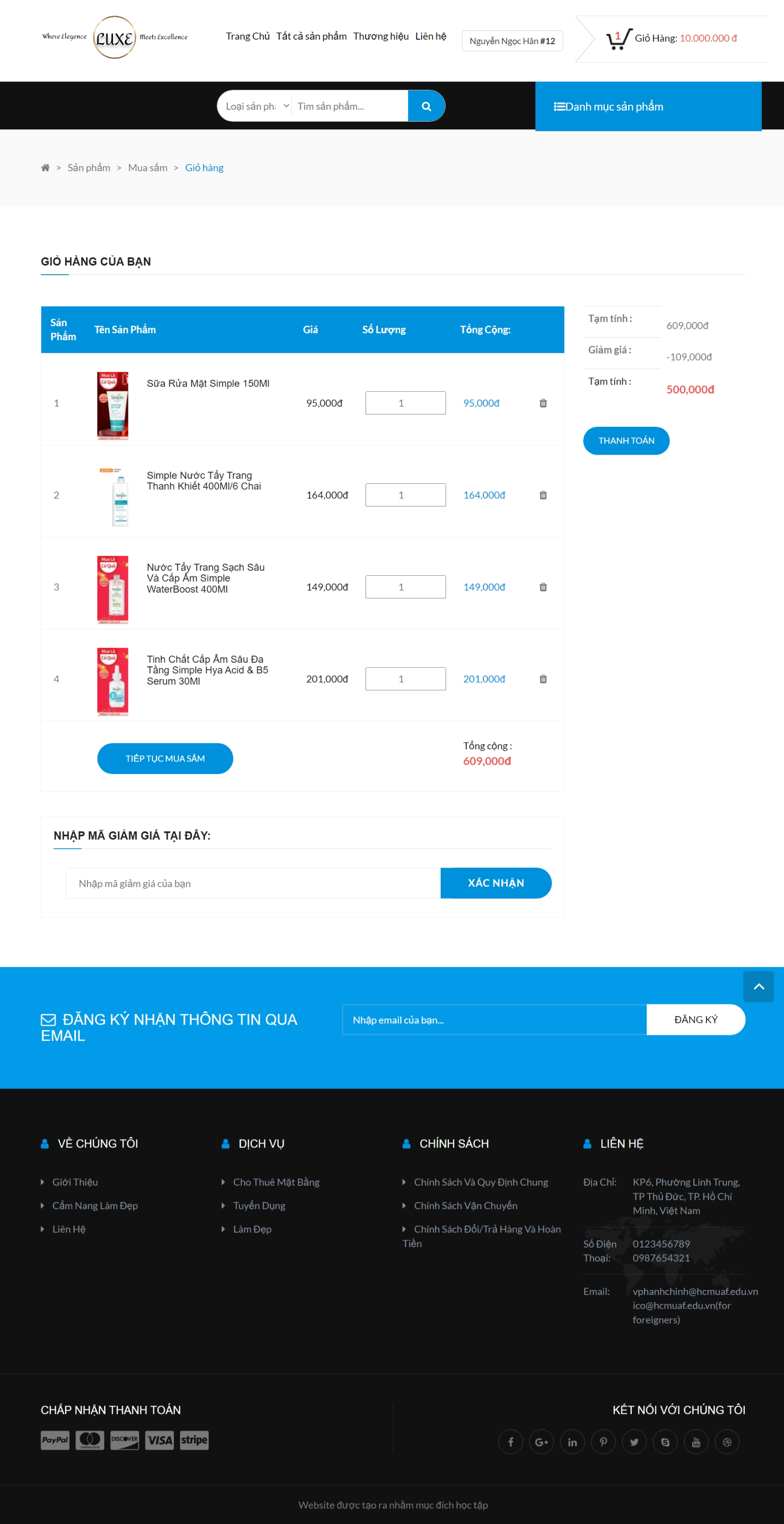
* Đổi mật khẩu và quên mật khẩu:

UPDATE `user` SET password = 'new\_hashed\_password', updated\_at = NOW()WHERE id = 1 AND password = 'old\_hashed\_password';

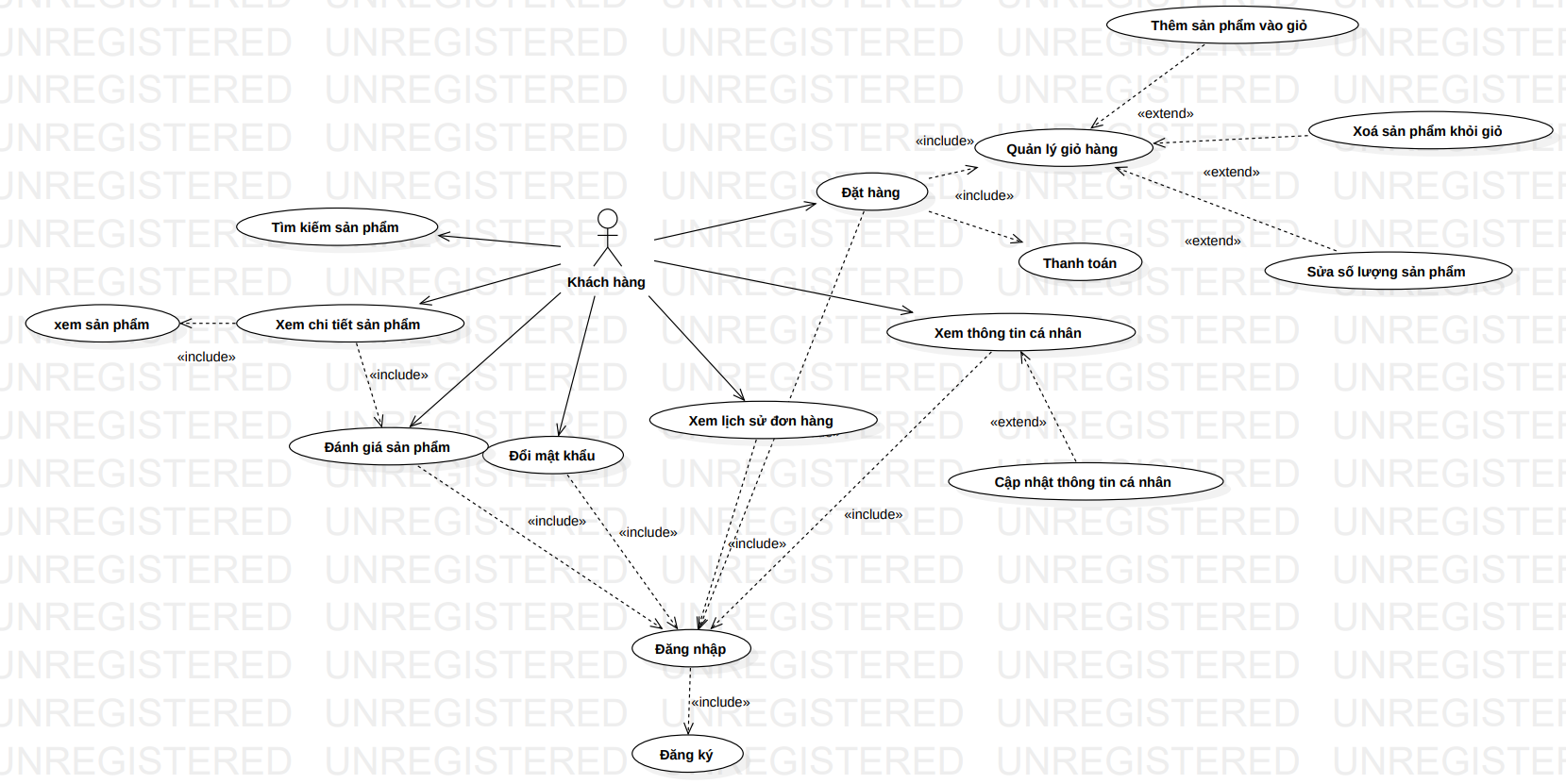
* 1. **Các yêu cầu phi chức năng**
* **Hiệu suất**: Website phải tải trang trong vòng dưới 3 giây, đảm bảo người dùng có trải nghiệm nhanh chóng khi duyệt sản phẩm và thực hiện các giao dịch.
* **Bảo mật**: Hệ thống phải bảo vệ thông tin người dùng, bao gồm thông tin cá nhân và dữ liệu thanh toán. Các giao dịch phải được mã hóa, và chỉ người dùng đã xác thực mới có quyền truy cập vào thông tin tài khoản.
* **Khả năng mở rộng**: Website phải có khả năng mở rộng để xử lý số lượng lớn người truy cập và giao dịch cùng lúc, đặc biệt vào các dịp cao điểm như khuyến mãi hoặc lễ hội mua sắm.
* **Tính thân thiện với người dùng**: Giao diện phải trực quan, dễ sử dụng trên cả máy tính và thiết bị di động, với các thao tác mượt mà giúp người dùng dễ dàng tìm kiếm và mua sản phẩm.
  1. **. Phân tích chi tiết từng chức năng**
     1. ***Chức năng xem sản phẩm (kết hợp phân trang)***
* Khi người dùng load trang, ta sẽ hiển thị các sản phẩm được bán chạy nhất của shop cho người dùng thấy
* Ta sẽ hiển thị 3 loại, mỗi loại 6 sản phẩm mới nhất
* ******Câu lệnh SQL:
  + 1. ***Tìm kiếm sản phẩm***
* Khi người dùng nhập vô thanh tìm kiếm, ta sẽ tìm theo tất cả sản phẩm có tên và loại(nếu người dùng chọn)
* Câu lệnh SQL:
  + 1. Đặt mua sản phẩm
* Người dùng chọn vào chức năng “Thêm vào giỏ hàng” ở mỗi sản phẩm để đưa sản phẩm vào danh sách muốn đặt mua.
* Tại giao diện “Giỏ hàng”, người dùng sẽ chọn những sản phẩm chắc chắn sẽ mua và thực hiện các thao tác để đặt hàng. Người dùng cũng có thể xóa những sản phẩm không còn muốn mua khỏi giỏ hàng.
  + 1. Thanh toán trực tuyến
* Sau khi thực hiện các thao tác đặt hàng, người dùng cần chọn phương thức thanh toán để hoàn thiện bước đặt hàng cuối cùng.
* Có thể thanh toán bằng các phương thức như: Thanh toán khi nhận hàng, thanh toán trực tuyến bằng tài khoản ngân hàng, momo, vnpay,...

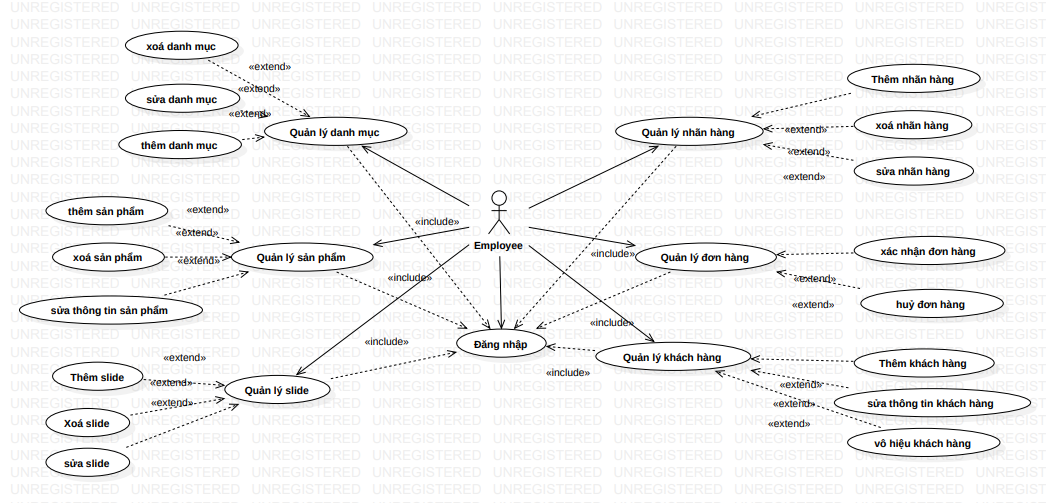


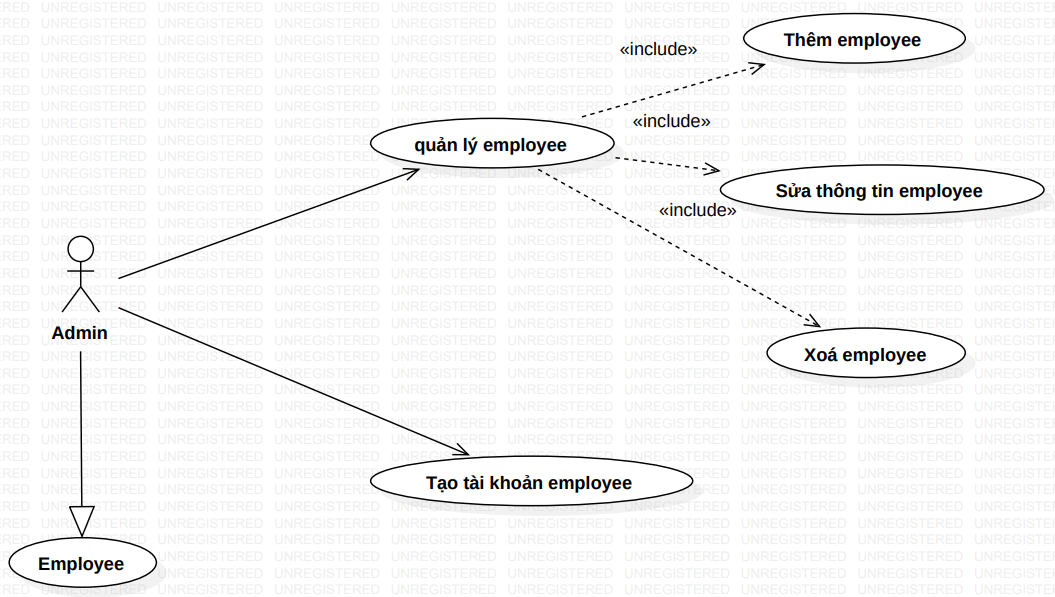
* + 1. Cập nhật giỏ hàng
* Giỏ hàng là nơi người dùng lưu lại các sản phẩm mà mình có thể mua.
* Tại đây người dùng có thể chỉnh sửa lại số lượng sản phẩm hoặc xóa sản phẩm khỏi danh sách. Người dùng cũng có thể thêm các sản phẩm mới vào giỏ hàng tại chức năng “Tiếp tục mua sắm”.



* + 1. Đăng ký tài khoản
* Nếu lần đầu tiên truy cập vào cửa hàng, người dùng cần thực hiện các bước đăng ký tài khoản để lưu trữ lại thông tin khách hàng trên hệ thống của cửa hàng.
* Người dùng cũng có thể đăng ký thêm nhiều tài khoản khác với email khác nhau
* Chỉ khi có tài khoản, người dùng mới có thể đặt mua sản phẩm.
  + 1. Đăng nhập tài khoản
* Nếu từng đăng ký tài khoản trên hệ thống của cửa hàng nhưng đã bị đăng xuất khỏi tài khoản, người dùng có thể đăng nhập lại chỉ bằng email và mật khẩu đúng với thông tin mà khách hàng từng đăng ký.
* Đối với trường hợp quên mật khẩu, người dùng có thể yêu cầu đổi lại mật khẩu mới bằng cách nhập địa chỉ email và đợi mã OTP gửi về.
  + 1. Xem thông tin cá nhân
* Thông tin cá nhân của khách hàng sẽ có hai trạng thái: công khai và riêng tư.
* Thông tin công khai bao gồm: Ảnh đại diện, tên biệt danh. Các khách hàng khác sẽ thấy thông tin này khi bạn xuất hiện ở một bài đánh giá sản phẩm nào đó.
* Các thông tin còn lại như: họ tên, giới tính, địa chỉ giao hàng, địa chỉ email, số điện thoại và ngày sinh. Thông tin này được lưu
* Thông
* Đây là những thông tin phục vụ cho việc đặt mua hàng
  1. **. Sơ đồ use case (Use Case Diagram)**
     1. **Use case Customer**



* + 1. **Use case Employee**
    2. **Use case Admin**



* 1. **Sơ đồ quan hệ cơ sở dữ liệu (ERD)**



* 1. **Mô tả chi tiết các bảng**
     1. **User**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | Int | pk, increment |
| full\_name | nvarchar(255) | not null |
| email | varchar(255) | unique, not null |
| password | varchar(255) | not null |
| phone | varchar(255) | not null |
| date\_of\_birth | date | not null |
| avatar | varchar(500) | null |
| status | enum | default: ‘NONE’ |
| last\_login | datetime | null |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Role**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| name | varchar(255) | not null |
| description | text | not null |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **UserHasRole**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| user\_id | int | foreign key(user.id) |
| role\_id | int | foreign key(role.id) |

* + 1. **Address**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| user\_id | int | null, foreign key(user.id) |
| recipient\_name | nvarchar(255) | not null |
| recipient\_phone | varchar(15) | not null |
| city | nvarchar(255) | not null |
| district | nvarchar(255) | not null |
| ward | nvarchar(255) | not null |
| note | nvarchar(255) | null |
| is\_default | boolean | default: false |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Brand**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| name | nvarchar(255) | not null |
| logo | varchar(500) | not null |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Category**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| parent\_id | int | foreign key(category.id) |
| logo | varchar(500) | not null |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Product**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| name | nvarchar(255) | not null |
| price | int | not null |
| cost\_price | int | not null |
| stock | int | default: 0 |
| sold\_quantity | int | default: 0 |
| description | text | default: ‘’ |
| is\_available | boolean | default: true |
| thumbnail | varchar(500) | not null |
| brand\_id | int | foreign key(brand.id) |
| category\_id | int | foreign key(category.id) |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **ProductImage**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| url | varchar(500) | not null |
| product\_id | int | foreign key (product.id) |

* + 1. **Review**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| user\_id | int | foreign key(user.id) |
| product\_id | int | foreign key(product.id) |
| rating | int | not null |
| comment | text | default: ‘’ |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Order**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| address\_id | int | foreign key (address.id) |
| user\_id | int | foreign key(user.id) |
| status | enum | default: ‘PENDING’ |
| payment\_method | varchar(255) | not null |
| shipping\_fee | int | default:0 |
| note | varchar(500) | default: ‘’ |
| total\_price | int | not null |
| order\_date | datetime | default: now() |
| confirm\_at | datetime | null |

* + 1. **OrderDetail**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| order\_id | int | foreign key(order.id) |
| product\_id | int | foreign key(product.id) |
| quantity | int | not null |
| total\_price | int | not null |

* + 1. **Slide**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| image | varchar(500) | not null |
| url | varchar(500) | not null |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Contact**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| email | varchar(255) | not null |
| title | nvarchar(255) | not null |
| content | text | not null |
| status | enum | default: ‘PENDING’ |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Payment**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| order\_id | int | foreign key(order.id) |
| user\_id | int | foreign key(user.id) |
| method | varchar(255) | not null |
| status | enum | default: ‘PENDING’ |
| amount | int | not null |
| transaction\_id | varchar(255) | null |
| created\_at | datetime | default: now() |
| updated\_at | datetime | default: now() |

* + 1. **Coupon**

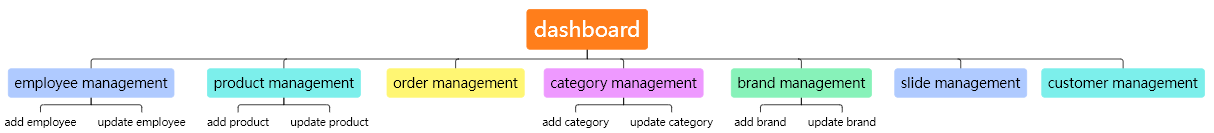
|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| code | varchar(255) | unique, not null |
| min\_order\_value | int | default:0 |
| discount\_type | enum | default: ‘PERCENTAGE’ |
| discount\_value | int | not null |
| max\_discount\_value | int | null |
| start\_date | datetime | not null |
| end\_date | datetime | not null |
| current\_usage | int | default:0 |
| max\_usage | int | null |
| created\_at | datetime | default: now() |

* + 1. **Wishlist**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc** |
| id | int | pk, increment |
| user\_id | int | foreign key(user.id) |
| product\_id | int | foreign key(product.id) |
| created\_at | datetime | default: now() |

1. **Chương 4: THIẾT KẾ VÀ XÂY DỰNG WEBSITE**
   1. **Giao diện User**

* A diagram of a home

  Description automatically generated**Tổng quan về giao diện phía user**
  1.  **Giao diện Admin**