**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MẪU THIẾT KẾ PHẦN MỀM**

**XÂY DỰNG WEBSITE BÁN GIÀY**

( MCV + Singleton + Prototype + Iterator Pattern )

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Lý Đoàn Duy Khánh**

**SINH VIÊN THỰC HIỆN:**

**Nhữ Hà Nam – 18DH110352**

**Tạ Thúy Lam – 19DH110139**

**Nguyễn Thành Đạt – 19DH111116**

**Nguyễn Đăng Khoa – 19DH111140**

**TP.HỒ CHÍ MINH – 2022**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MẪU THIẾT KẾ PHẦN MỀM**

**XÂY DỰNG WEBSITE BÁN GIÀY**

( MCV + Singleton + Prototype + Iterator Pattern )

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Lý Đoàn Duy Khánh**

**SINH VIÊN THỰC HIỆN:**

**Nhữ Hà Nam – 18DH110352**

**Tạ Thúy Lam – 19DH110139**

**Nguyễn Thành Đạt – 19DH111116**

**Nguyễn Đăng Khoa – 19DH111140**

**TP.HỒ CHÍ MINH – 2022**

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## Giới thiệu tổng quan đề tài:

### Giới thiệu đề tài:

Ngày nay, hình thức bán hàng trực tuyến qua mạng với nhiều loại mặt hàng đa dạng rất phổ biến và ưa chuộng. Cũng giống như các mặt hàng khác trên thị trường hiện nay, thời trang cũng là một vấn đề được nhiều người quan tâm kể cả nam và nữ, vì vậy giày dẹp đang là thứ mà khách hàng có nhu cầu mua hàng rất cao, tuy nhiên đa số họ vẫn yêu thích xu hướng mua hàng trực tuyến và tham khảo các hình ảnh sản phẩm, giá cả thông qua website trước khi đi đến cửa hàng, hoặc mua và thanh toán trực tuyến trên website. Nên khách hàng muốn xây dựng website bán hàng thu hút, ấn tượng sẽ giúp quảng bá được cửa hàng góp phần phát triển và thành công trong lĩnh vực thời trang.

Với xu hướng công nghệ thông tin thời đại 4.0, khi mà mọi gia đình, cá nhân đều dễ dàng truy cập internet từ thiết bị smartphone, laptop, smart TV. Nên việc bán hàng trên internet là một việc cần để bắt kịp xu thế và chiếm thị phần. Chủ cửa hàng cần một trang web để họ có thể quảng bá, bán giày dép trên đó.

Website có mục đích phân phối trực tiếp sản phẩm, dịch vụ của cửa hàng tới khách hàng thông qua hoạt động mua sắm trực tuyến. Website giúp khách hàng tiết kiệm thời gian, khách hàng có nhiều sự lựa chọn và dễ dàng so so sánh giá cả sản phẩm giữa các website bán hàng trực tuyến với nhau.

### Mục tiêu và yêu cầu sử dụng:

1. Nhu cầu bán hàng

Khi khách hàng có nhu cầu xem thông tin thì có thể sử dụng chức năng tìm kiếm thông tin, tra cứu thông tin dễ dàng. Giao diện gần gũi với khách hàng. Dễ dàng thao tác khi sử dụng website để xem và mua sản phẩm.

1. Nhu cầu quản lý

Muốn có một hệ thống bán hàng để có thể vừa bán hàng ,quảng bá sản phẩm và vừa quản lý hàng, thống kê doanh thu để có thể dễ dàng đưa ra các chiến lược kinh doanh phù hợp với xu hướng hiện nay.

## Liệt kê các chức năng

Đồ án xây dựng theo dạng Client/Server là mô hình mạng máy tính bao gồm 2 thành phần chính là máy khách (client) và máy chủ (server). Trong mô hình này, server là nơi lưu trữ tài nguyên, cài đặt các chương trình dịch vụ và thực hiện các yêu cầu của client. Client đón vai trò gửi yêu cầu đến server.

### Khách Hàng

##### Chức năng đăng ký

Khi khách hàng muốn đăng nhập vào website để đặt hàng, nếu chưa có tài khoản thì phải vào trang đăng ký tài khoản của website để đăng ký tài khoản.

##### Chức năng đăng nhập

Khi khách hàng đã có tài khoản thì có thể đăng nhập vào website để có thể đặt hàng, thanh toán, thêm yêu thích sản phẩm , xem lại các món hàng đã lưu trong giỏ, xem thông tin của đơn hàng ví dụ như tình trạng vận chuyển, chi tiết sản phẩm của đơn hàng cũng như là lịch sử các hóa đơn.

##### Thông tin khách hàng

Khách hàng có thể xem thông tin cá nhân của mình trong mục thông tin khách hàng : tên , mail , địa chỉ , số điện thoại . Và khách hàng cũng có thể thêm / sửa thông tin cá nhân của mình trong mục này.

##### Trang chủ

Trang chủ sẽ hiển thị sơ qua các mẫu , mặt hàng mà các thương hiệu đang bán chạy, các sản phẩm mới và chương trình khuyến mãi .

##### Tìm kiếm

Người dùng có thể tìm kiếm các mặt hàng cụ thể bằng cách nhập vào thanh tìm kiếm.

##### Danh Mục

Danh mục sản phẩm hiển thị danh sách các sản phẩm được phân loại theo thương hiệu, giá… Để khách hàng có thể dể dàng tìm kiếm sản phẩm theo nhu cầu mong muốn. Khách hàng có thể chọn từng sản phẩm để xem chi tiết của sản phẩm và chọn mua sản phẩm đó.

##### Chi tiết sản phẩm

Chi tiết sản phẩm của sản phẩm cụ thể, sẽ hiện thông tin của sản phẩm Như giá thành, số lượng , hình ảnh , mô tả chỉ tiết, bình luận và đánh giá của khách hàng.

##### Giỏ hàng

Khi người dùng tìm được mặt hàng mà mình ưng ý thì có thể tích vào đặt hàng, thêm số lượng nếu muốn , còn nếu không ưng ý thì có thể xóa ra khỏi giỏ hàng .

Người dùng có thể thêm / xóa / sửa các mặt hàng trong giỏ hàng theo ý của mình và bấm đặt hàng khi đã chốt được lựa chọn.

##### Mã giảm giá

Khách hàng có thể áp dụng mã giảm giá cho đơn hàng của mình. Lưu ý rằng: mỗi mã giảm giá chỉ được sử dụng 1 lần và chỉ được sử dụng cho chính sản phẩm của cửa hàng đó vì mã giảm giá là chỉ áp dụng riêng cho cửa hàng.

##### Thủ tục thanh toán ( Checkout )

Sau khi đã xác nhận món hàng, số lượng cần mua từ giỏ hàng và các thông tin vận chuyển như địa chỉ , giá tiền vận chuyển xong thì khách hàng có thể thanh toán .

Đến bước thanh toán thì sẽ hiện tổng tiền của đơn hàng gồm tiền hàng và tiền vận chuyển . Khách hàng cần phải nhập các thông tin cá nhân cần thiết như họ tên, địa chỉ , số điện thoại …

Lựa chọn hình thức thanh toán trực tiếp hoặc trực tuyến. Thanh toán trực tiếp thì khách hàng có thể thanh toán bằng tiền mặt lúc nhận hàng. Thanh toán trực tuyến thì hình thức thanh toán này sẽ thông qua paypal , sau khi thanh toán thì sẽ trừ tiền trực tiếp vào trong thẻ .

Khi đã xong các bước trên thì có thể tiến hành đặt hàng , đặt hàng thành công thì sẽ gửi thông tin của sản phẩm vào mail của khách hàng và tạo đơn hàng , tình trạng đơn hàng để khách hàng theo dõi đơn hàng của mình .

##### Đơn hàng và tình trạng của đơn hàng

Sau khi hoàn thành thủ tục thanh toán thì sẽ tạo đơn hàng để có thể giúp cho khách hàng xem thông tin và theo dõi đơn hàng của mình . Tình trạng của đơn hàng sẽ có các mốc như : Đang xử lý , Đã xác nhận , Đang vận chuyển , Hoàn thành ( đã giao ) .

Khách hàng có thể hủy đơn hàng khi tình trạng đơn hàng chưa đến bước đang vận chuyển .

Trang này cũng giúp quản lý các đơn hàng hoàn thành ( đã giao ) cho khách hàng khi khách có nhu cầu muốn xem lại ( lịch sử đặt hàng ) .

### Admin

##### Quản lý sản phẩm

Quản lý nhập xuất sản phẩm: cập nhật tình trạng, số lượng sản phẩm

Quản lý chi tiết sản phẩm: thêm / xóa / sửa thông tin các sản phẩm trên website như tên , giá thành, hình ảnh, mô tả, …

##### Quản lý loại sản phẩm

Phân chia sản phẩm theo loại : Laptop, Phụ kiện …

##### Quản lý bán hàng

Quản lý thông tin đơn hàng: hủy đơn khi khách hàng cần hủy dựa trên tình trạng đơn hàng.

Quản lý tình trạng vận chuyển: sẽ cập nhật tình trạng các mốc xử lý của đơn hàng đi theo trình tự :

* Đang xử lý.
* Đã xác nhận .
* Đang vận chuyển.
* Hoàn thành ( đã giao ) .

Tư vấn khách hàng.

##### Quản lý đơn hàng và tình trạng của đơn hàng

Chức năng này có thể nói rất là quan trọng chỉ sau quản lý sản phẩm. Vì các nhà quản lý có thể thống kê doanh thu tùy ý để có thể đưa ra các biện pháp kinh tế cho cửa hàng của mình.

Quản lý đơn hàng chưa xác nhận: xem danh sách các đơn hàng chưa xác nhận và nhân viên có thể tiến hành duyệt đơn hàng. Có thể hủy đơn nếu có yêu cầu từ khách hàng những đã thanh toán trực tiếp ( paypal ) thì không thể hủy.

Quản lý đã xác nhận và đang vận chuyển: đơn hàng ở bước này thì không thể hủy . Nhân viên sẽ theo dõi và cập nhật tình trạng cho khách hàng. Nếu nhận được tiền ( trực tiếp) và khách đã lấy hàng thì sẽ chuyển đổi tình trạng qua hoàn thành (đã giao).

Quản lý đơn hàng hoàn thành (đã giao): các đơn hàng ở tình trạng này có thể được đưa vào để thống kê

##### Quản lý các mã giảm giá

Có thể xem danh sách, thêm, chỉnh sửa, xóa, tìm kiếm mã giảm giá.

# CHƯƠNG 2: THIẾT KẾ

## Liệt kê function và non-function

### Function

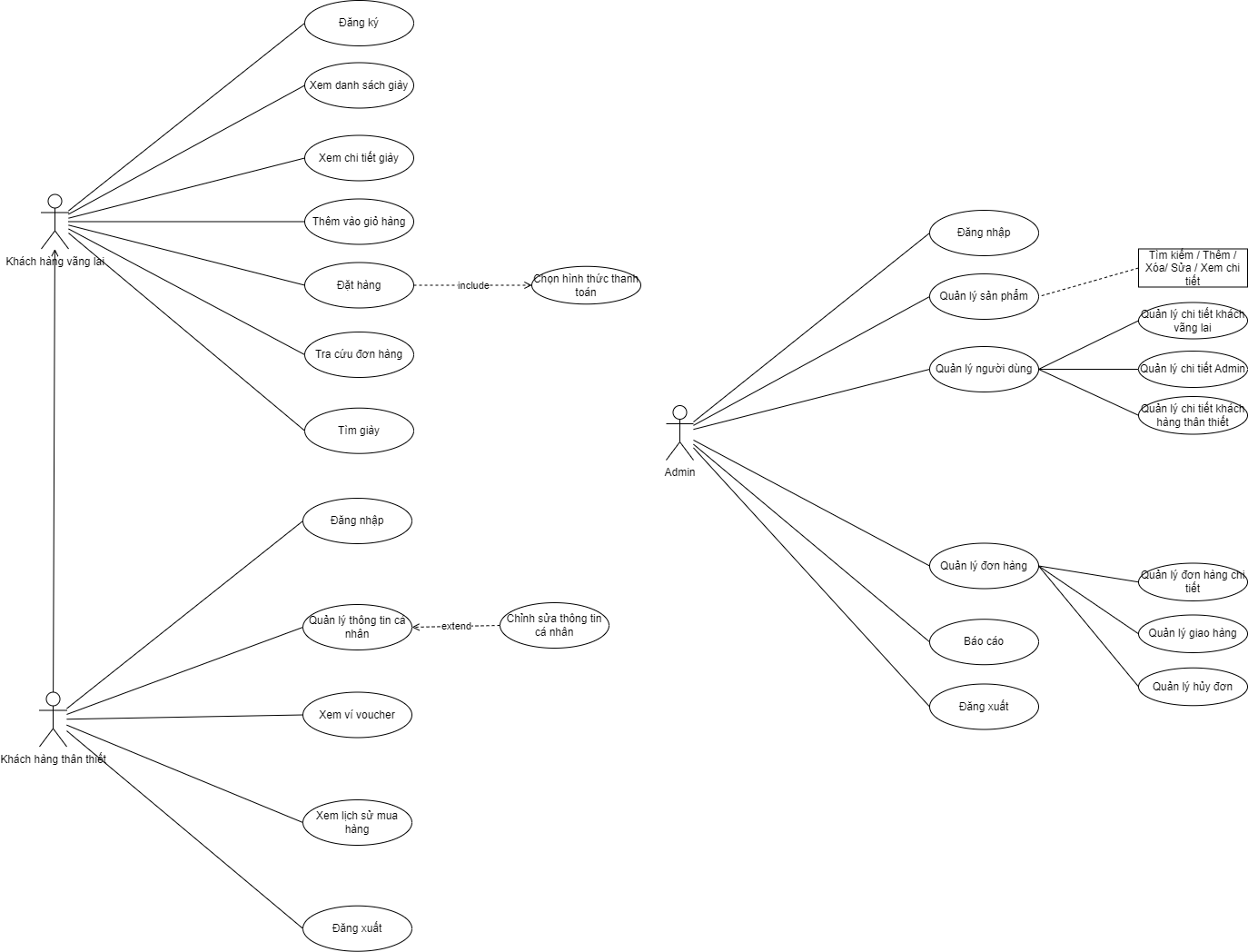
* Đăng ký, đăng nhập phân quyền đăng nhập vào hệ thống
* Phân quyền quản trị
* Quản lý Khách Hàng
* Quản lý loại sản phẩm
* Cập nhật thông tin sản phẩm
* Quản lý sản phẩm
* Quản lý đơn hàng
* Tra cứu sản phẩm
* Thanh toán trực tuyến
* Xem lịch sử mua hàng

### Non-Function

* Yêu thích.
* Feedback.
* Tra cứu theo loại sản phẩm, …

## Usecase và các sơ đồ liên quan

### Sơ đồ Usecase



Hình 1. Sơ đồ usecase

### Đặc tả các chức năng của ứng dụng(mô tả actor , mô tả usecase , đặc tả chức năng chính)

##### Mô tả Actor

|  |  |  |
| --- | --- | --- |
| **#** | **Tên Actor** | **Mô tả** |
| **1** | **Admin** | Admin website thường được gọi là quản trị viên website. Họ có quyền cho phép điều phối và kiểm soát tất cả quy trình hoạt động của một website. |
| **2** | **Khách vãng lai** | Khách hàng vãng lai là khách hàng chưa có tài khoản nhưng vẫn có thể mua hàng. |
| **3** | **Khách hàng thân thiết** | Khách hàng thân thiết là khách hàng có tài khoản đăng nhập và sử dụng đầy đủ các tính năng cửa website dành cho người mua. |

##### Mô tả usecase

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Code** | **Name** | **Brief Description** |
| **1** | **UC01** | Thêm sản phẩm | Cho phép người dùng thêm một sản phẩm vào hệ thống |
| **2** | **UC02** | Xóa sản phẩm | Cho phép người dùng xóa một sản phẩm trong hệ thống |
| **3** | **UC03** | Sửa sản phẩm | Cho phép người dùng chỉnh sửa thông tin một sản phẩm và lưu vào hệ thống |
| **4** | **UC04** | Tìm kiếm sản phẩm | Cho phép người dùng tìm kiếm sản phẩm có trong hệ thống |
| **5** | **UC05** | Tìm kiếm đơn hàng | Cho phép người dùng tìm kiếm đơn hàng bằng mã đơn hàng |
| **6** | **UC06** | Lọc đơn hàng theo trạng thái đơn hàng | Cho phép người dùng xem danh sách các đơn hàng được lọc theo trạng thái đơn hàng |
| **7** | **UC07** | Lọc đơn hàng theo ngày đặt | Cho phép người dùng xem danh sách các đơn hàng được lọc theo ngày đặt |
| **8** | **UC08** | Xác nhận đã chuẩn bị đơn | Cho phép người dùng xác nhận đơn hàng đã được chuẩn bị |
| **9** | **UC09** | Xác nhận đã giao đơn | Cho phép người dùng xác nhận đơn hàng đã được giao |
| **10** | **UC10** | Xác nhận hủy đơn hàng | Cho phép người dùng xác nhận đơn hàng đã được hủy |
| **11** | **UC11** | Thêm người dùng | Cho phép người dùng thêm một người dùng vào hệ thống |
| **12** | **UC12** | Xóa người dùng | Cho phép người dùng xóa một người dùng trong hệ thống |
| **13** | **UC13** | Sửa người dùng | Cho phép người chỉnh sửa thông tin một người dùng và lưu vào hệ thống |
| **14** | **UC14** | Tìm kiếm người dùng | Cho phép người dùng tìm kiếm người dùng có trong hệ thống |
| **15** | **UC15** | Quản lý thống kê | Cho phép người dùng xem thống kê đơn hàng và doanh thu theo khoảng thời gian. |
| **16** | **UC16** | Đăng ký (Khách hàng) | Cho phép người dùng tạo một người dùng mới vào hệ thống |
| **17** | **UC17** | Xem danh sách sách (Khách hàng) | Cho phép người dùng xem tất cả sản phẩm có trong hệ thống |
| **18** | **UC18** | Xem chi tiết cuốn sách | Cho phép người dùng xem chi tiết một cuốn sách |
| **19** | **UC19** | Thêm vào giỏ hàng | Cho phép người dùng thêm sản phẩm vào giỏ hàng |
| **20** | **UC20** | Xem giỏ hàng | Cho phép người dùng xem danh sách sản phẩm có trong giỏ hàng |
| **21** | **UC21** | Cập nhật giỏ hàng | Cho phép người dùng cập nhật số lượng sản phẩm trong giỏ hàng |
| **22** | **UC22** | Cập nhật thông tin cá nhân | Cho phép người dùng chỉnh sửa thông tin cá nhân của mình |
| **23** | **UC23** | Xem lịch sử mua hàng | Cho phép người dùng xem lịch sử mua hàng của mình |
| **24** | **UC24** | Xem chi tiết đơn hàng | Cho phép người dùng xem chi tiết đơn hàng |
| **25** | **UC25** | Hủy đơn | Cho phép người dùng hủy đơn hàng. |
| **26** | **UC26** | Đăng nhập (Khách hàng) | Cho phép người dùng đăng nhập vào hệ thống với vai trò là người mua |
| **27** | **UC27** | Hiển thị sách theo chủ đề | Cho phép người dùng lọc sách theo chủ đề |
| **28** | **UC28** | Hiển thị sach theo nhà xuất bản | Cho phép người dùng lọc sách theo nhà xuất bản |
| **29** | **UC29** | Đăng xuất | Cho phép người dùng đăng xuất khỏi hệ thống |
| **30** | **UC30** | Tìm kiếm cuốn sách | Cho phép người dùng tìm cuốn sách mong muốn |
| **31** | **UC31** | Tra cứu đơn hàng | Cho phép người dùng tra cứu đơn hàng bằng mã đơn hàng hoặc số điện thoại |
| **32** | **UC32** | Đặt hàng | Cho phép người dùng xác nhận đặt đơn hàng |
| **33** | **UC33** | Áp dụng voucher | Cho phép người dùng áp dụng voucher vào đơn hàng. |

1. Thiết kế cơ sở dữ liệu

3.1 Sơ đồ erd:

3.2 Mô tả CSDL

#### AppUser

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaDonHang **( Primary Key)** | int |
| VoucherID | int |
| MaKH | int |
| TongTien | int |
| TinhTrang | nvarchar(MAX) |
| NgayXacNhan | nvarchar(MAX) |
| DiaChi | nvarchar(MAX) |
| SDT | nvarchar(MAX) |
| HoTen | nvarchar(MAX) |

#### Vouchers

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaDonHang **( Primary Key)** | int |
| VoucherID | int |
| MaKH | int |
| TongTien | int |
| TinhTrang | nvarchar(MAX) |
| NgayXacNhan | nvarchar(MAX) |
| DiaChi | nvarchar(MAX) |
| SDT | nvarchar(MAX) |
| HoTen | nvarchar(MAX) |

#### TruyCaps

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaTruyCap **( Primary Key)** | int |
| SoLanTruyCap | int |
| MaSanPham | int |
| KhachHang\_MaKH | int |

#### ThuongHieus

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaThuongHieu **( Primary Key)** | int |
| TenThuongHieu | nvarchar(MAX) |
| SoLuong | int |
| MaDanhMuc | int |

#### SanPhams

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaSanPham **(Primary Key)** | int |
| TenSanPham | nvarchar(MAX) |
| GiaBan | int |
| MoTa | nvarchar(MAX) |
| TomTat | nvarchar(MAX) |
| Hinh | nvarchar(MAX) |
| TinhTrang | nvarchar(MAX) |
| SoLuong | int |
| MaDanhMuc | int |
| MaThuongHieu | int |

#### Quyens

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaQuyen **(Primary Key)** | int |
| TenQuyen | nvarchar(MAX) |

#### Orders

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| OrderID (Primary Key) | int |
| MakH | int |
| VoucherID | int |
| TongTien | int |
| NgayDatHang | nvarchar(MAX) |
| DiaChi | nvarchar(MAX) |
| HoTen | nvarchar(MAX) |
| SDT | nvarchar(MAX) |

#### OrderComfirmeds

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaDonHang (Primary Key) | int |
| MakH | int |
| MaSanPham | int |
| VoucherID | int |
| TongTien | int |
| NgayDatHang | nvarchar(MAX) |
| SoLuong | int |
| TinhTrang | nvarchar(MAX) |

#### Reviews

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| ReviewID ( Primary Key) | int |
| NoiDung | nvarchar(MAX) |
| MaKH | int |
| MaSanPham | int |
| NgayDang | Nvarchar(MAX) |

#### ChiTietDonHangHuys

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaChiTietDonHangHuy **(Primary Key)** | int (Not null) |
| MaSanPham | int |
| MaHuyDon | int |
| SoLuong | int |
| TinhTrang | nvarchar(MAX) |

#### DanhMucs

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaDanhMuc **( Primary Key)** | int (Not null) |
| TenDanhMuc | nvarchar(max) |
| SoLuong | int |

#### ChiTietTruyCaps

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaChiTietTruyCap | int (Not null) |
| MaKH | Int |
| MaSanPham | Int |
| NgayTruyCap | nvarchar(MAX) |

#### ChiTietGioHang

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaChiTietGioHang  **(Primary Key)** | int (Not null) |
| MaSanPham | int |
| OrderID | int |
| SoLuong | int |

#### GioHangs

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaGioHang **(Primary Key)** | int (Not null) |
| MaKh | int |
| MaSanPham | int |
| VoucherID | int |
| TongTien | int (Not null) |
| NgayDatHang | nvarchar(MAX) |
| SoLuong | int |

#### HuyDonHangs

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaHuyDon **(Primary Key)** | int (Not null) |
| MaDonHang | nvarchar(MAX) |
| MaKH | int |
| TongTien | int |
| NgayXacNhan | nvarchar(MAX) |
| TinhTrang | nvarchar(MAX) |

#### KhachHangs

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaKH **(Primary Key)** | int (Not null) |
| TenKH | nvarchar(MAX) |
| DiaChi | nvarchar(MAX) |
| SDT | nvarchar(MAX) |
| DiemTichLuy | int (Not null) |
| Email | nvarchar(MAX) |
| NgaySinh | nvarchar(MAX) |
| UserName | nvarchar(25) |
| Password | nvarchar(30) |
| MaQuyen | int |
| MaLoaiKH | int |

#### LoaiKhachHangs

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaLoaiKH **(Primary Key)** | int (Not null) |
| TenLoaiKH | nvarchar(MAX) |
| GhiChu | nvarchar(MAX) |

#### LoginManages

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| LoginManageKey **( Primary Key)** | int (Not null) |
| UserName | nvarchar(MAX) |
| TimeLogin | nvarchar(MAX) |
| TimeLogout | nvarchar(MAX) |

#### ComfirmedDetails

|  |  |
| --- | --- |
| **Column Name** | **Data Type** |
| MaChiTietDonHang **( Primary Key)** | int (not null) |
| MaSanPham | int |
| MaDonHang | int |
| SoLuong | int |
| TinhTrang | nvarchar(MAX) |

# CHƯƠNG 3: KỸ THUẬT THỰC HIỆN

## Giới thiệu các mẫu thiết kế

1. **MVC**

Mô hình model-view-controller (MVC)và các mô hình liên quan như HMVC và MVVM cho phép bạn chia code thành các đối tượng logic thực hiện các mục đích khác nhau. Model hoạt động như là lớp truy cập dữ liệu , nơi dữ liệu được fetch và trả về trong định dạng có thể sử dụng trong ứng dụng. Controller quản lý các request, tiến trình dữ liệ được trả về từ Model và load View. View hiển thị template (markup, xml, …).

1. **Singleton Pattern**

Một class thuộc dạng Singleton có nghĩa là: nó chỉ có một instance duy nhất, bất kỳ ở đâu đều có thể truy cập tới instance của class singleton đó.

Điển hình nhất mà ta vẫn thấy khi lập trình game đó là GameManager, AudioManager, RoundManager, GameCenterManager, AdsManager,…

Khởi tạo just-in-time hoặc khởi tạo ở lần gọi đầu tiên

1. **Prototype Pattern**

Prototype Pattern là một trong những pattern phổ biến trong lập trình hướng đối tượng, là một pattern thuộc nhóm Creational Patterns. Ý tưởng này là một design pattern đặc biệt có liên quan đến việc khởi tạo đối tượng (Object), thay vì tạo ra Object, Prototype pattern sử dụng việc cloning (copy nguyên mẫu của Object).

1. **Iterator pattern**

Là một Design Pattern được xếp vào nhóm hành vi (Behavioral Patterns). Như đã trình bày ở phần giới thiệu, một Iterator được thiết kế cho phép xử lý nhiều loại tập hợp khác nhau bằng cách truy cập các phần tử với cùng một phương thức định sẵn, mà không cần hiểu rõ về tập hợp gốc.

Thông thường, Iterator đã được cài đặt sẵn trong các ngôn ngữ lập trình hiện đại dưới dạng 1 interface, với 2 phương thức chính:

next(): trả về phần tử kế tiếp trong tập hợp.

hasNext(): trả về giá trị True nếu vẫn còn phần tử trong tập hợp, và False trong trường hợp ngược lại.

## Đặc điểm

### MCV

**Ưu điểm của mô hình MVC:**

Mô hình MVC phù hợp với các dự án với các đội ngũ làm việc độc lập với nhau, như vậy các công việc đều có thể được thực hiện một cách linh hoạt.

Như vậy, mô hình này nổi bật nhờ các ưu điểm sau:

* **Hỗ trợ quá trình phát triển nhanh chóng:** Với đặc điểm hoạt động độc lập của từng thành phần, các lập trình viên có thể làm việc đồng thời trên từng bộ phận khác nhau của mô hình này. MVC giúp bạn tiết kiệm rất nhiều thời gian.
* **Khả năng cung cấp đồng thời nhiều khung View:** Với mô hình MVC, bạn có thể tạo ra đồng thời nhiều khung View cho Model.
* **Hỗ trợ các kỹ thuật không đồng bộ:** MVC có thể hoạt động trên nền tảng JavaScript. Điều này có nghĩa là các ứng dụng MVC có thể hoạt động với các file PDF, các trình duyệt web cụ thể, và cả các widget máy tính.
* **Dễ dàng thao tác chỉnh sửa:** Bộ phận Model hoạt động tách biệt với View đồng nghĩa với việc bạn có thể đưa ra các thay đổi, chỉnh sửa hoặc cập nhật dễ dàng ở từng bộ phận.
* **Giữ nguyên trạng thái data:** Mô hình MVC truyền lại dữ liệu nhưng không định dạng lại dữ liệu. Do đó, các dữ liệu này có thể được dùng lại cho các thay đổi sau này.
* **Hỗ trợ các nền tảng phát triển SEO:** Với mô hình MVC, bạn có thể dễ dàng tạo ra các mã SEO URL để thu hút lượng truy cập đối với ứng dụng bất kỳ.

1. **Nhược điểm của mô hình MVC:**

MVC khuyến khích việc thực hành mã hóa và hỗ trợ khả năng mở rộng nhanh chóng. Nếu mối quan tâm lớn nhất của bạn là bạn có thể xây dựng ứng dụng một cách nhanh chóng, dễ cập nhật, thì mô hình MVC chính là lựa chọn tốt nhất của bạn. Nếu bạn quan tâm nhiều hơn đến hiệu suất và giữ những cấu trúc tệp ít phức tạp hơn, thì MVC có thể không phải là lựa chọn phù hợp.

* **Khó khăn trong quá trình điều hướng code**: Điều hướng khung có thể phức tạp vì mô hình này bao gồm nhiều lớp và yêu cầu người dùng thích ứng với các tiêu chí phân tách của MVC. Không thích hợp việc phát triển các ứng dụng nhỏ vì mô hình này yêu cầu bạn lưu trữ một số lượng lớn các file.
* **Nhiều khung hoạt động đồng thời**: Việc phân tách một tính năng thành ba bộ phận khác nhau dễ dẫn đến hiện tượng phân tán. Do đó, đòi hỏi các nhà phát triển phải duy trì tính nhất quán của nhiều bộ phận cùng một lúc.

1. **Singleton**
2. **Ưu điểm của Singleton Pattern:**

Ai cũng có thể truy cập vào instance của singleton class, gọi nó ở bất cứ đâu, không cần phải search FindObjectOfType<T> hay GetComponent, …

Dữ liệu không thay đổi bởi chỉ có một instance duy nhất, có thể duy trì dữ liệu sau khi đổi scene (DontDestroyOnLoad(object)).

Singleton class hỗ trợ interface trong khi static class thì lại không, đây là sự khác biệt cần phải xác định để tạo static class hay singleton class.

Hỗ trợ kế thừa, static class thì không hỗ trợ kế thừa.

1. **Nhược điểm của Singleton:**

Đầu tiên, các Singletons không được khai báo dưới dạng biến hay dạng tường minh, mà được sử dụng một cách globally, thẳng tay sử dụng instance ngay trong code mà không cần khai báo.

Lợi ích này đồng thời sẽ làm code trở nên bí ẩn hơn, bạn sẽ không biết được code mình đã sử dụng các singletons (dependencies) nào sau 1 – 2 tháng đọc lại code, hay người khác (đồng nghiệp) đọc hiểu code của mình, việc duy nhất lúc này là ngồi soi lại từng đoạn code.

Thứ hai, Singleton không áp dụng đa hình được nên chỉ có thể sử dụng đúng tên của class Singleton đó.

Ví dụ ban đầu bạn có Player Singleton, về sau lại sinh ra XaydaPlayer và NamekPlayer, … kế thừa từ Player Singleton nhưng bạn lại không sử dụng được bởi không áp dụng được đa hình, đồng nghĩa với việc code của bạn đã bị kết dính với class Player.

Phải sử dụng tới keyword trung gian là .instance (có thể cached khi sử dụng tần suất nhiều).

Chỉ có một instance duy nhất.

Làm tăng kết nối giữa các script và điều này là không tốt, các scripts con phụ thuộc quá nhiều vào các singletons và khi singletons thay đổi có thể gây ra bug hoặc lỗi

Không sử dụng được đa hình

1. **Prototype Pattern**

**Ưu điểm của Prototype Pattern:**

Tránh việc tạo nhiều lớp con cho mỗi đối tượng tạo như của Abstract Factory Pattern.

Giảm chi phí để tạo ra một đối tượng mới theo "chuẩn", điều này sẽ làm tăng hiệu suất so với việc sử dụng từ khóa new để tạo đối tượng mới.

**Khởi tạo object mới bằng cách thay đổi một vài thuộc tính của object (các object có ít điểm khác biệt nhau):** 1 hệ thống linh động có khả năng cho phép tự định nghĩa một hành động nào đó thông qua sự kết hợp với một object (nghĩa là một phương thức của một class) hơn là định nghĩa một class mới. **Client** có thể thể hiện một tác động khác bằng cách ủy quyền cho lớp prototype. Đồng thời cách thức khởi tạo này cũng khai báo một "lớp mới" mà không phải lập trình gì cả. Thực tế thì việc copy một nguyên mẫu giống như việc khởi tạo một object từ một class mới. Prototype pattern giúp giảm số lớp mà hệ thống cần dùng.

**Khởi tạo object mới bằng cách thay đổi cấu trúc:** Rất nhiều ứng dụng xây dựng hệ thống từ nhiều phần và các phần con. Các phần con lại khởi tạo từ nhiều phần con khác (chia nhỏ bài toán). Prototype pattern cũng hỗ trợ điều này. Nghĩa là các phần đó có thể được khởi tạo từ việc copy một nguyên mẫu từ một "cấu trúc" khác. Miễn là các phần kết hợp đều thể hiện Clone() và được sử dụng với cấu trúc khác nhau làm nguyên mẫu.

1. **Nhược điểm của Prototype Pattern:**

**Giảm việc phân lớp:** Đôi khi hệ thống quá phức tạp vì có quá nhiều class, và cây thừa kế của lớp khởi tạo có quá nhiều lớp song song cùng mức. Prototype pattern rõ ràng làm giảm số lớp và sự phức tạp của cây thừa kế (class hierarchy).

1. **Iterator Pattern**
2. **Ưu điểm của Iterator Pattern:**

**Đảm bảo nguyên tắc Single responsibility principle (SRP)**: chúng ta có thể tách phần cài đặt các phương thức của tập hợp và phần duyệt qua các phần tử (iterator) theo từng class riêng lẻ.

**Đảm bảo nguyên tắc Open/Closed Principle (OCP):** chúng ta có thể implement các loại collection mới và iterator mới, sau đó chuyển chúng vào code hiện có mà không vi phạm bất cứ nguyên tắc gì.

Chúng ta có thể truy cập song song trên cùng một tập hợp vì mỗi đối tượng iterator có chứa trạng thái riêng của nó.

Trong một vài trường hợp, bạn có thể trì hoãn một lần lặp lại và tiếp tục nó khi cần thiết.

1. **Nhược điểm của Iterator Pattern:**

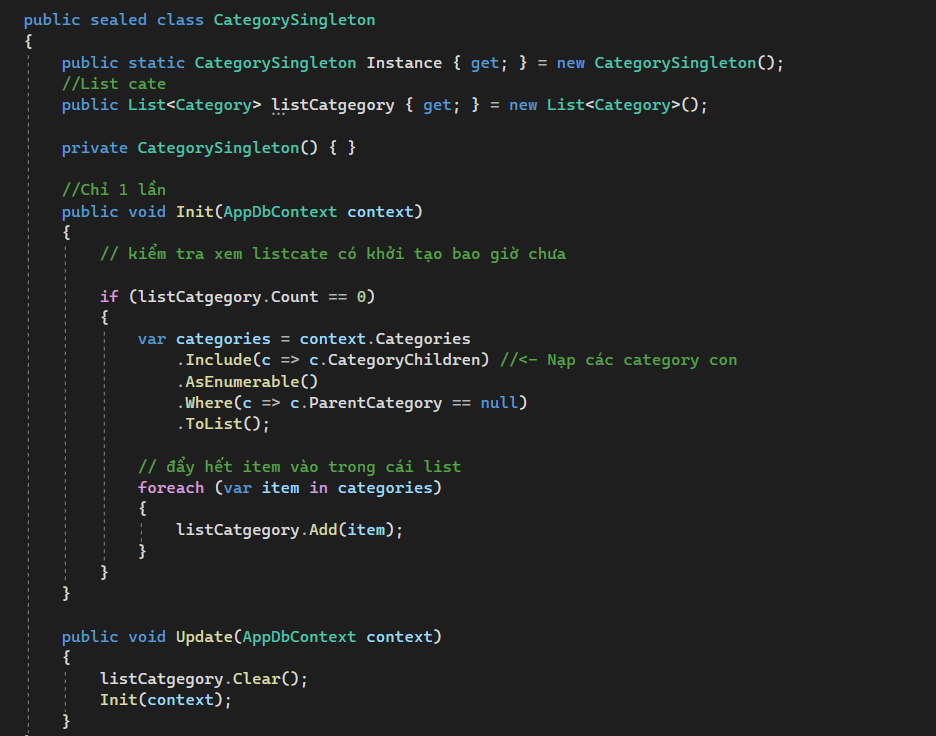
Sử dụng iterator có thể kém hiệu quả hơn so với việc duyệt qua các phần tử của bộ sưu tập một cách trực tiếp.

Có thể không cần thiết nếu ứng dụng chỉ hoạt động với các collection đơn giản.

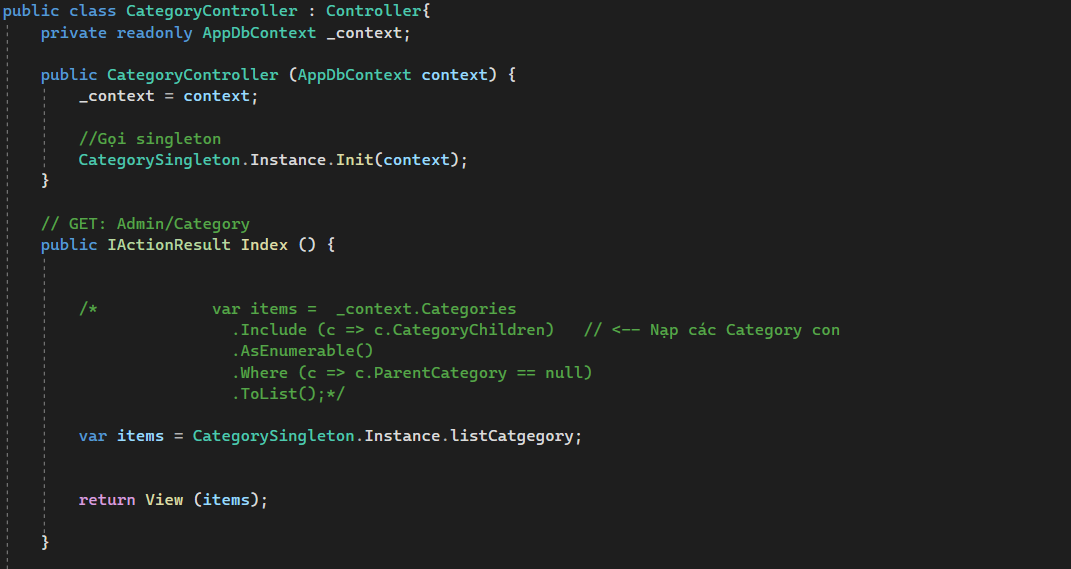
## Áp dụng vào phần mềm

### Áp dụng mẫu Singleton

Tạo class Mẫu Singleton có 1 instance duy nhất dùng hàm init để khởi tạo danh sách Danh mục từ database, hàm update giúp đồng bộ dữ liệu giữa class và phần mềm khi có thay đổi.

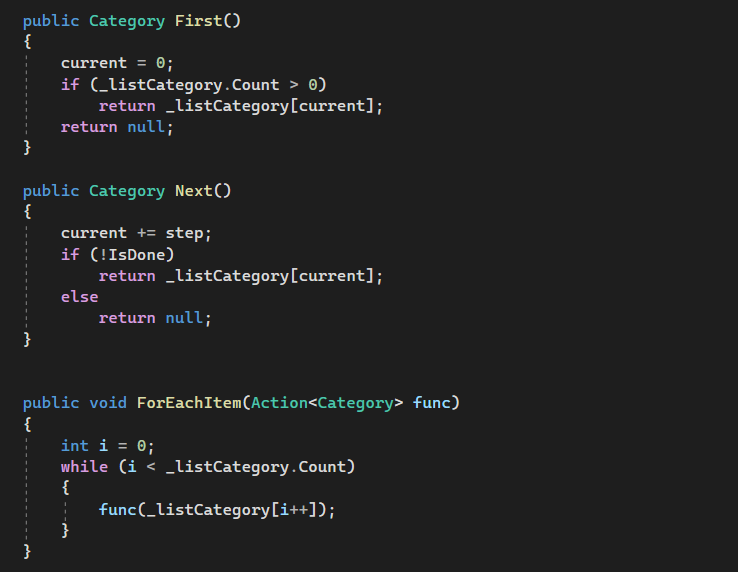
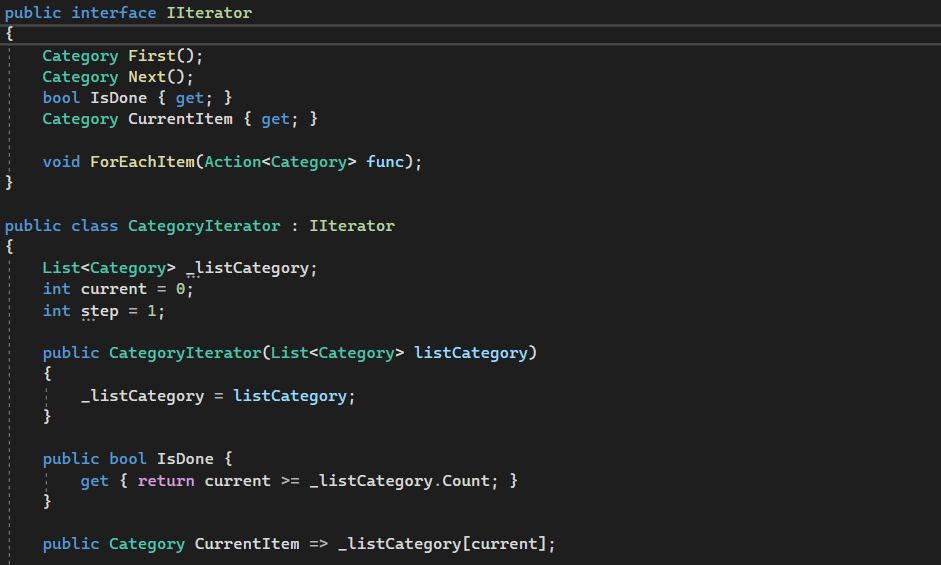


Thực hiện khai báo và truy cập vào instance của class từ CategoryController

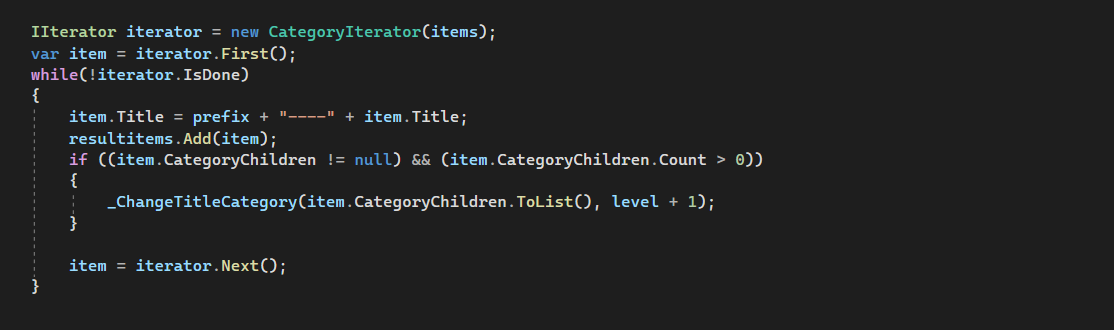


### Áp dụng mẫu Iiterator

Tạo class Iiterator thực hiện hành vi để xử lý tập hợp các sản phẩm.

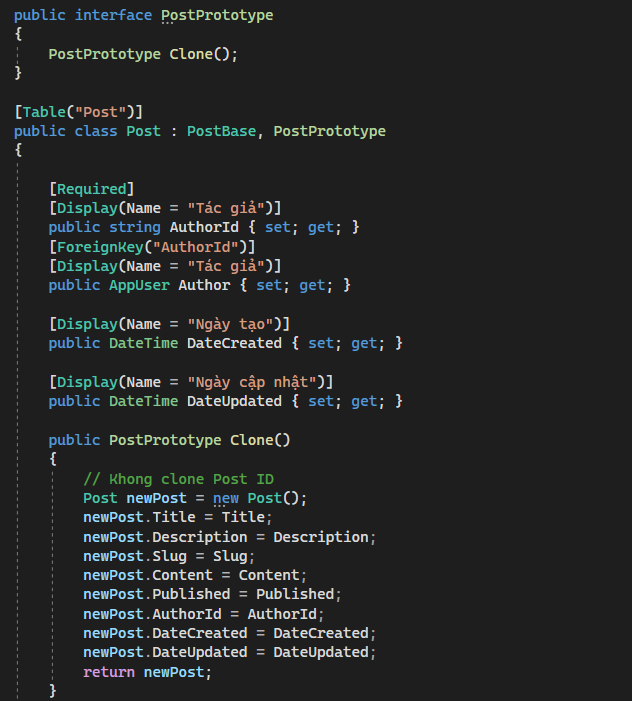


Áp dụng vào CatagoryController để hiển thị các danh mục

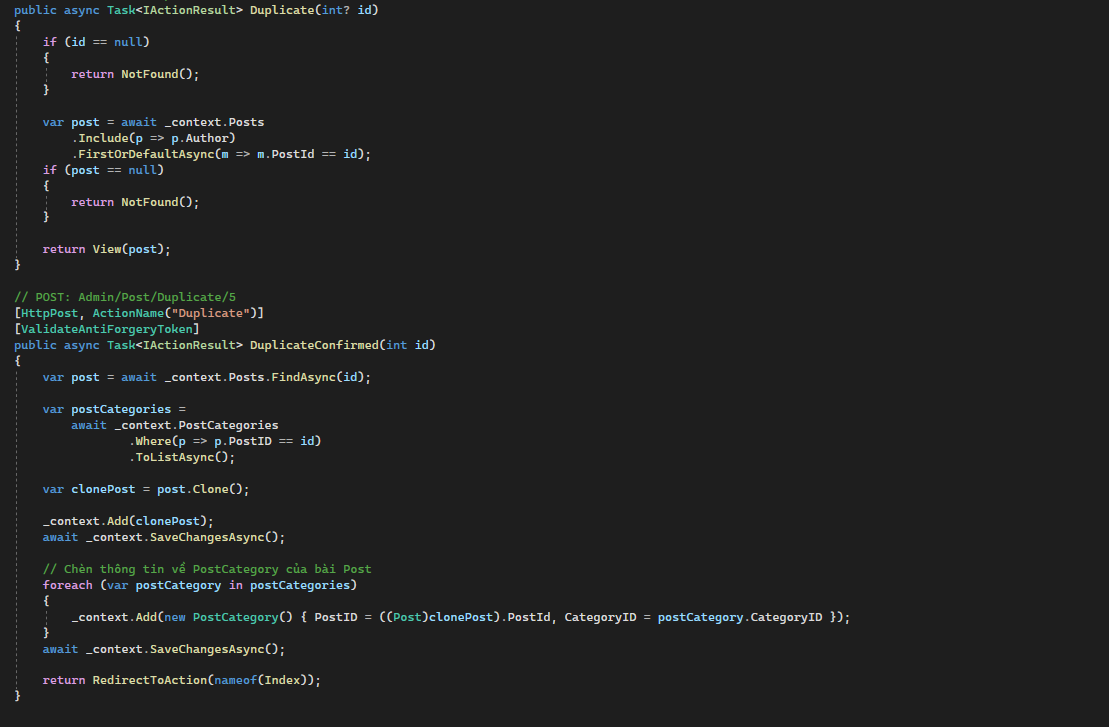


### Áp dụng mẫu Prototype

Tạo interface CategoryPrototype để tạo một clone Object để nhân bản Object

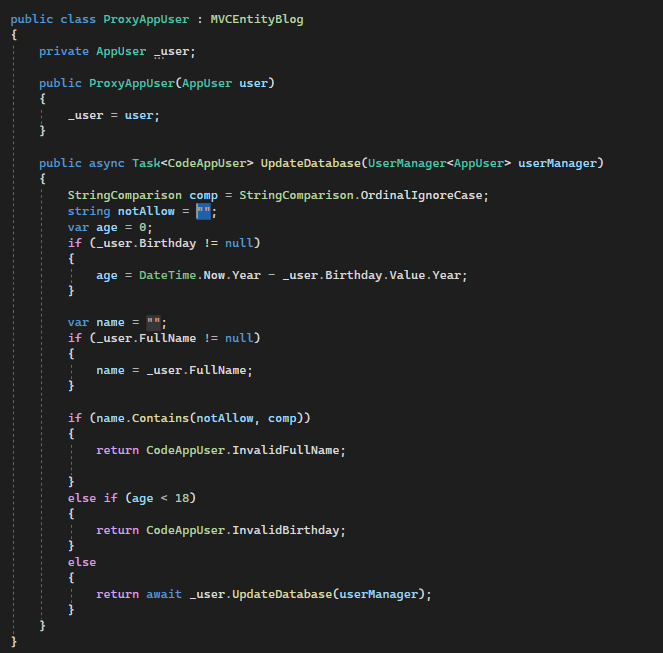


Áp dụng mẫu Prototype để nhân bản sản phẩm đã có không trùng ID



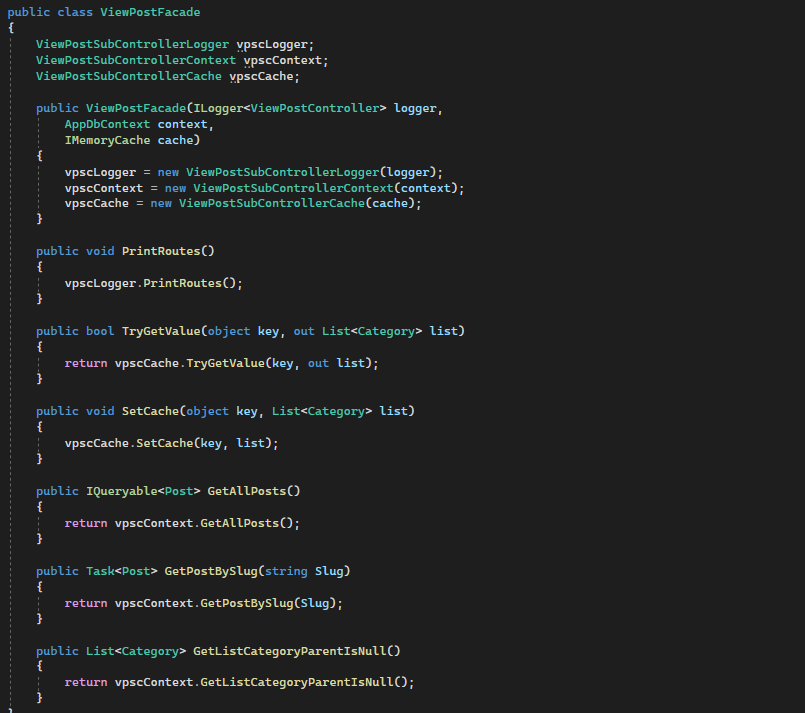
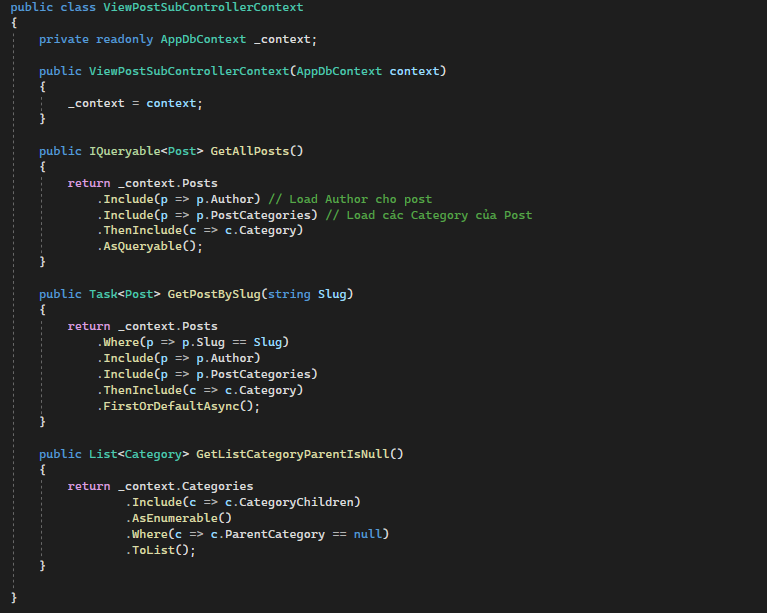
### Áp dụng mẫu Proxy

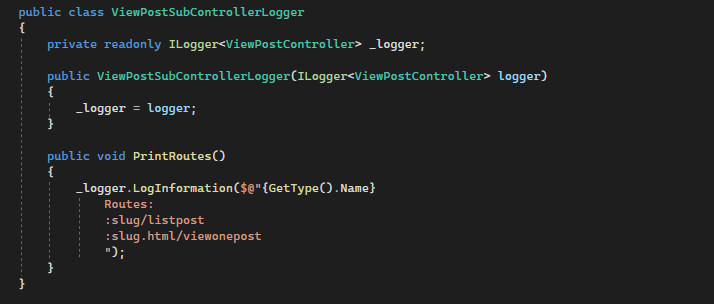
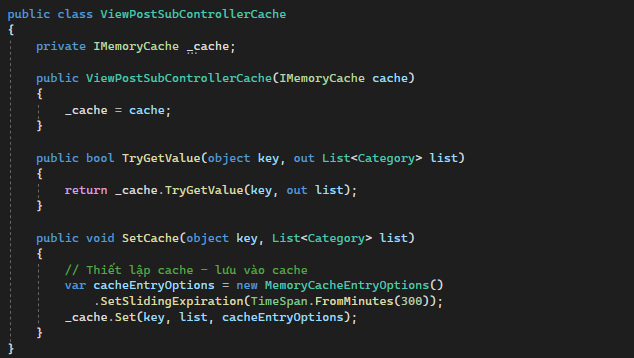
Tạo interface AppUser để kiểm tra thông tin khi user muốn thay đổi



### Áp dụng mẫu Façade

Facade phân rã 1 cái hệ thống lớn ra 1 cái tập chứa tất cả các hàm khi mà hệ thống nhỏ kiếm thì cái interface đó có mặt, khi mà bên ngoài gọi thì chỉ cần nắm thằng interface đó mà gọi ko cần tìm mấy thằng con, 1 cái project nắm giữ 3 cái là logger, context(database), cache

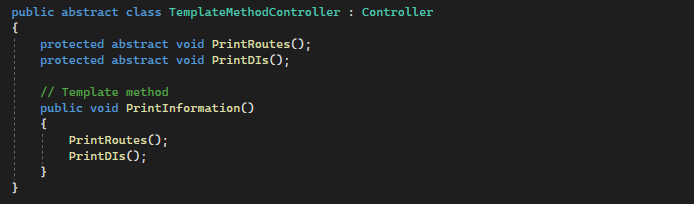
 

Áp dụng Façade

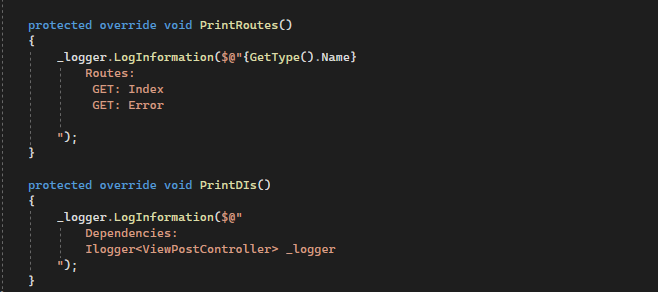


### Áp dụng mẫu TempleMethod

Để kiểm tra log debug của code



Áp dụng



# CHƯƠNG 4: TRIỂN KHAI – KẾT QUẢ

## Giao diện Khách hàng

##### Trang chủ

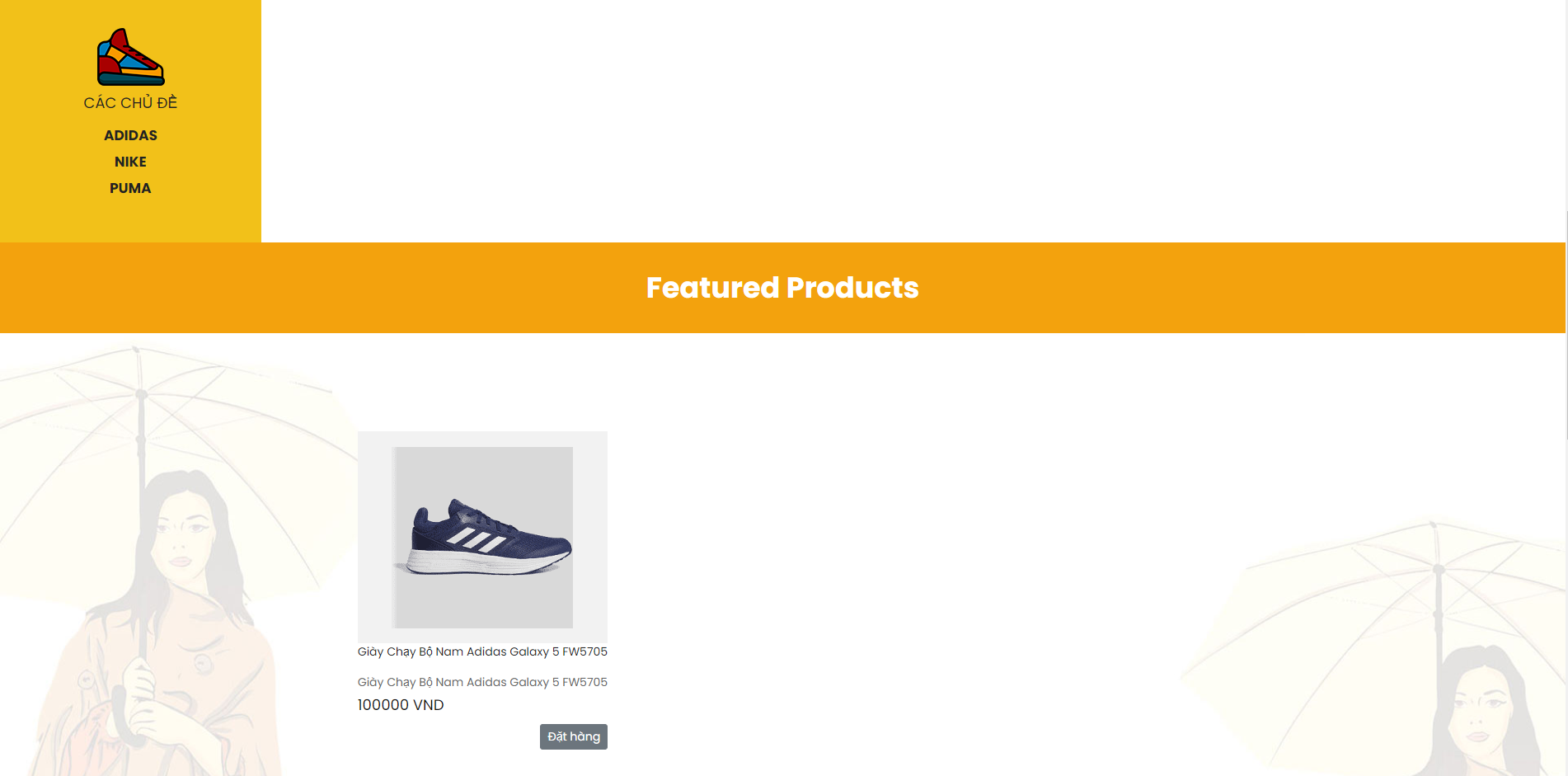
Hiển thị các danh mục sản phẩm và gợi ý các sản phẩm



*Hình 1: Trang chủ*

##### Trang danh mục sản phẩm

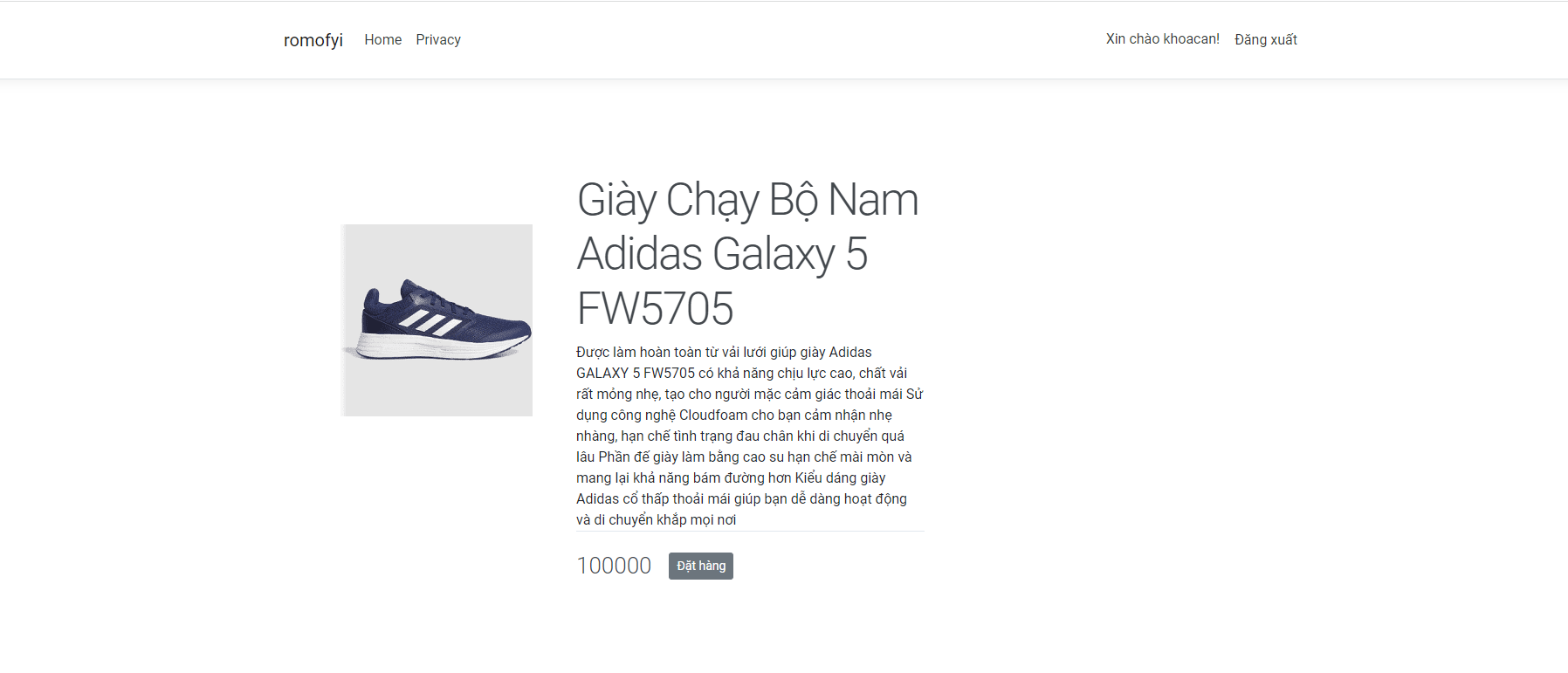
Hiển thị các sản phẩm theo danh mục



*Hình 2: Danh mục sản phẩm*

##### Trang chi tiết sản phẩm

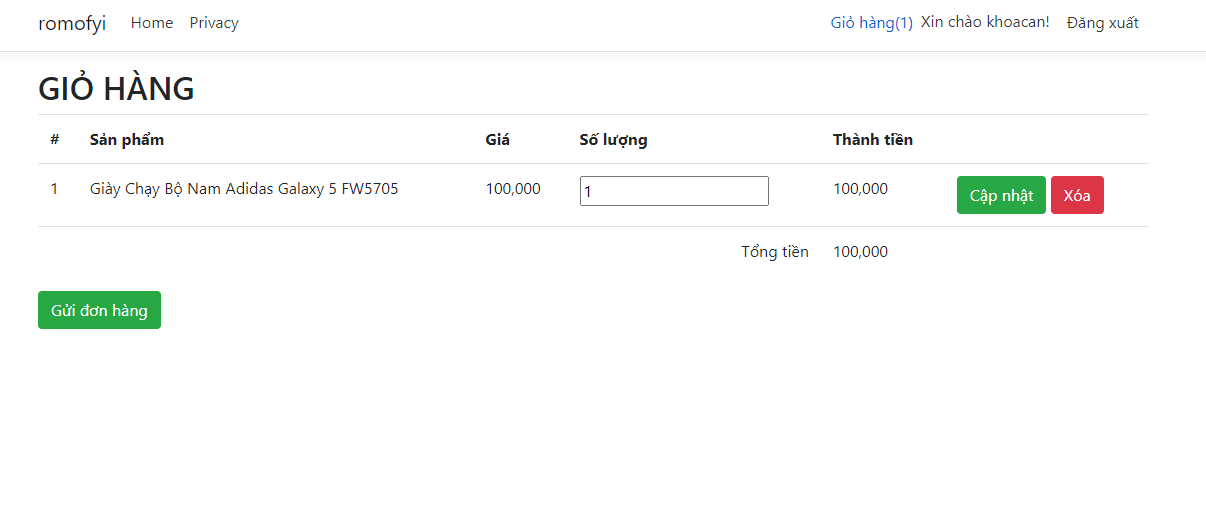
User có thể xem chi tiết sản phẩm lựa chọn số lượng cũng như size để cho vào giỏ hàng. Có thể đánh giá sản phẩm



*Hình 3: Trang chi tiết sản phẩm*

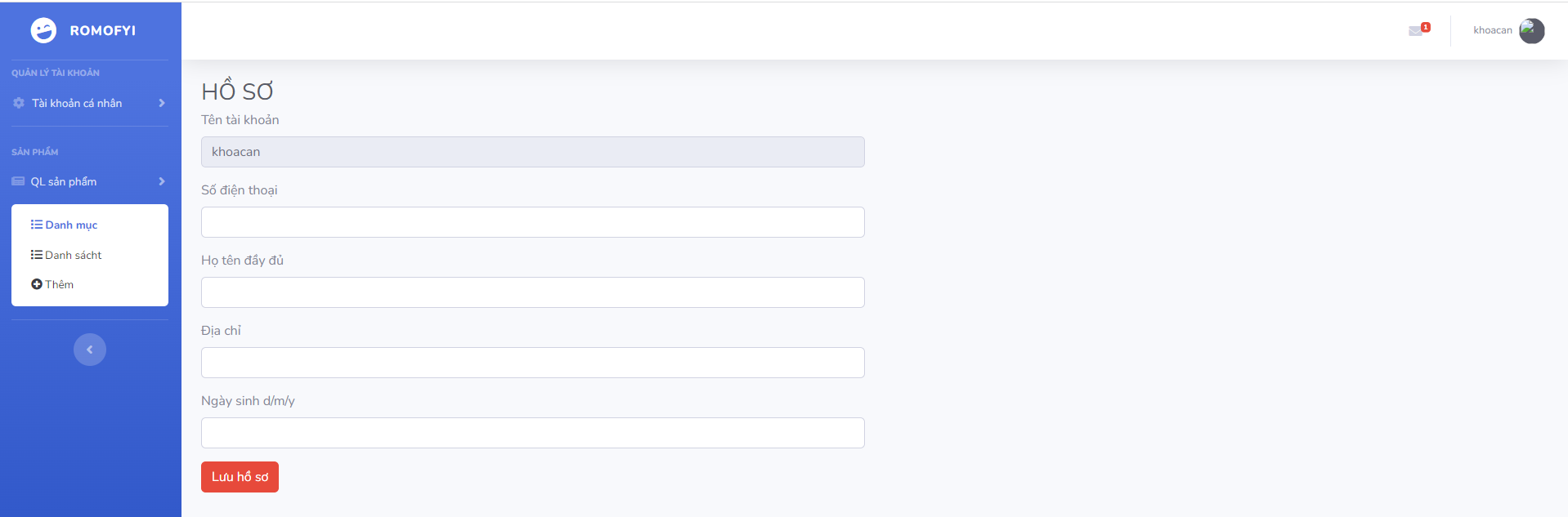
##### Trang giỏ hàng

Xem danh sách sản phẩm đã thêm vào giỏ. Có thể nhập mã giảm giá sau đó nhấn vào nút đặt hàng để tiến hành checkout



*Hình 4: Trang giỏ hàng*

Trang profile



*Hình 5: Trang profile*

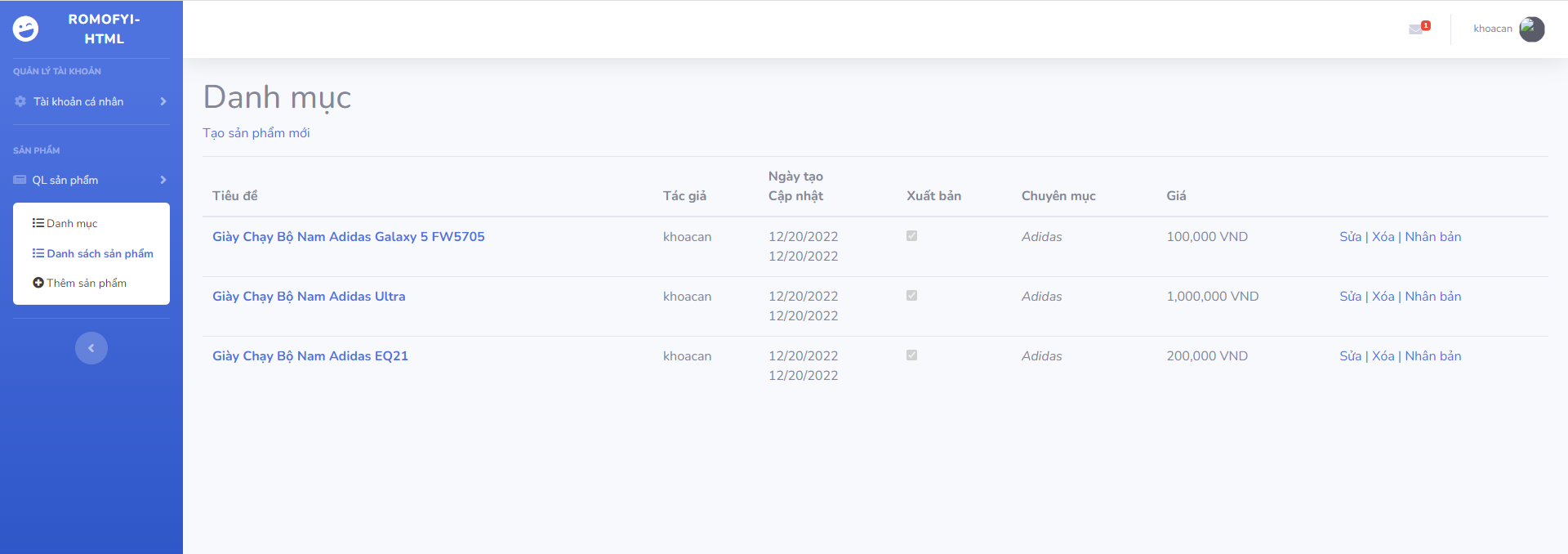
## Giao diện QL

##### Trang quản lý thông tin



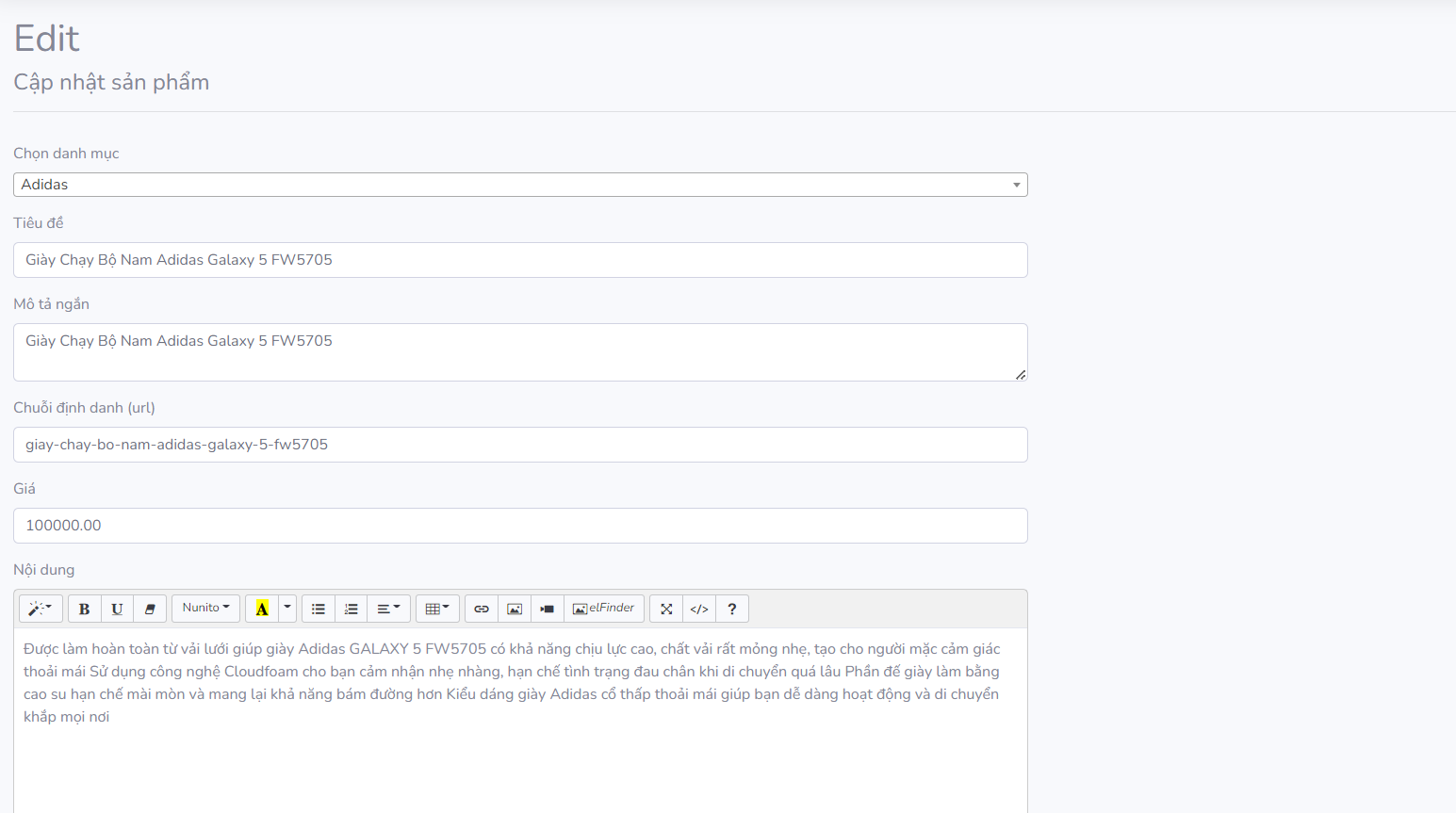
*Hình 6: Trang quản lý thông tin*

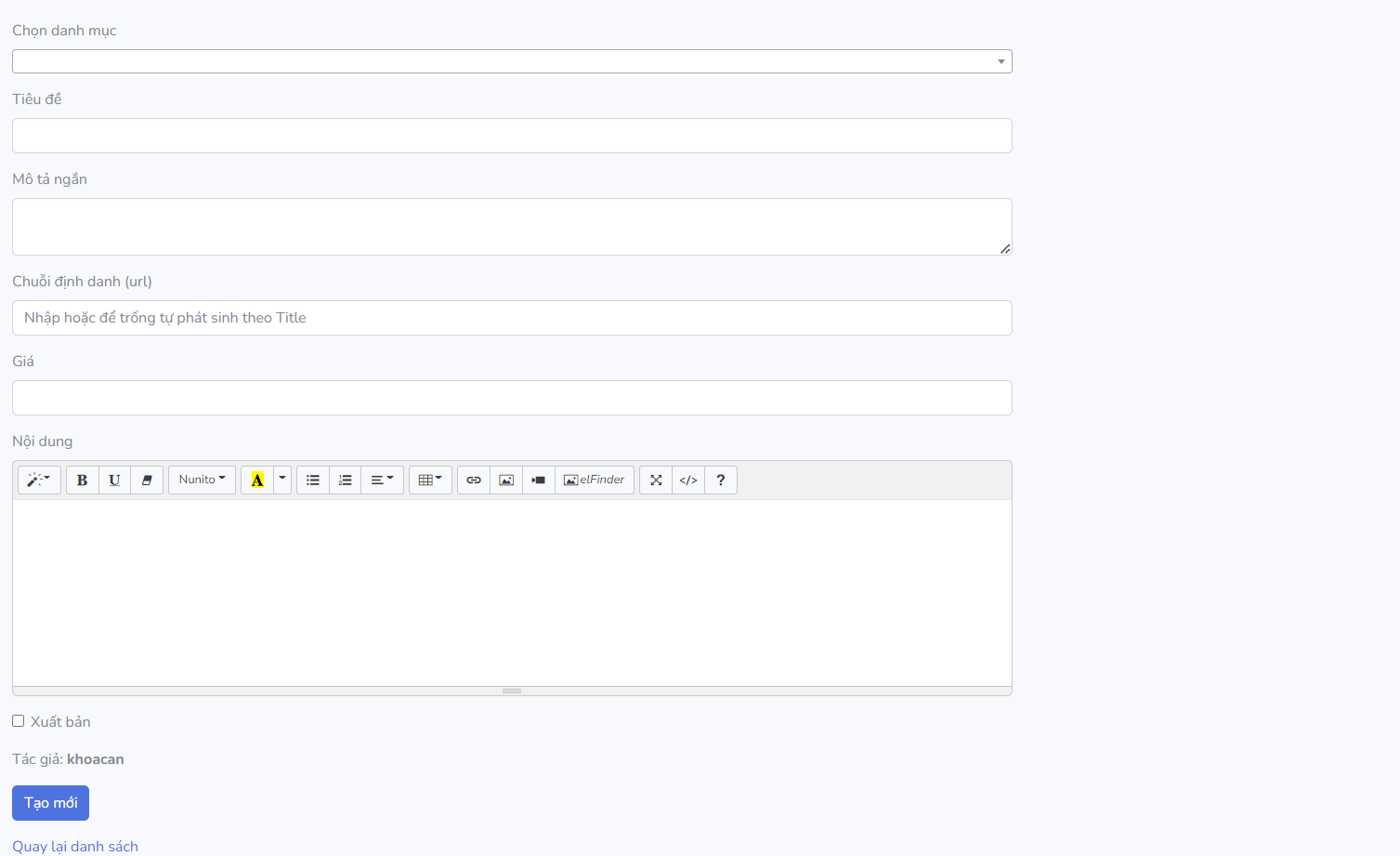
##### Trang quản lý sản phẩm



*Hình 7: Trang quản lý sản phẩm*

##### Trang thêm, sửa sản phẩm



*Hình 8: Trang thêm, sửa sản phẩm*

# KẾT LUẬN

Trang web bán giày online Romofyi của nhóm đã đạt được những yêu cầu của trang web e-commerce cơ bản như đăng ký, đăng nhập, tìm kiếm sản phẩm, đặt hàng, thanh toán.

Tuy nhiên, website cũng tồn tại một số khuyết điểm như dựa vào đường truyền internet khá nhiều, truyền dữ liệu còn chậm từ server lên client. Do nhóm mới tiếp xúc với các công nghệ mới trong lĩnh vực web nên còn thiếu sót về kiến thức và kỹ năng xử lý vấn đề vì vậy cách viết code chưa được gọn gàng và tối ưu nên trang web truyền dữ liệu chậm.

Sau một khoảng thời gian học và làm đồ án thức nghiệm nhóm đã nắm được ở mức độ cơ bản của ngôn ngữ JavaScript, React, SQL, Docker, Microservices và cách lập trình theo phương pháp lập trình hướng đối tương.