

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MÔN HỌC: THƯƠNG MẠI ĐIỆN TỬ**

**ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN**

**NỘI THẤT NHÀ Ở**

**GVHD: NGUYỄN TRƯỜNG HẢI**

**Nhóm:**

**1. Vương Thị Thu Trang 17110385**

Mục Lục

[**Chương 1: Tổng Quan** 3](#_heading=h.gjdgxs)

[**1. Lí do chọn đề tài:** 3](#_heading=h.30j0zll)

[**2. Mô tả đề tài:** 4](#_heading=h.1fob9te)

[**2.1 Các dữ liệu cần quản lí:** 4](#_heading=h.3znysh7)

[**2.2 Các yêu cầu về nghiệp vụ:** 4](#_heading=h.2et92p0)

[**2.3 Các yêu cầu về chức năng:** 5](#_heading=h.tyjcwt)

[**Chương 2: Phân tích và thiết kế** 7](#_heading=h.3dy6vkm)

[**1 Thành phần dữ liệu:** 7](#_heading=h.1t3h5sf)

[**1.1 Lược đồ ERD của cơ sở dữ liệu:** 7](#_heading=h.4d34og8)

[7](#_heading=h.2s8eyo1)

[**1.2 Các bảng trong mô hình quan hệ:** 7](#_heading=h.17dp8vu)

[**1.3 Lược đồ quan hệ:** 12](#_heading=h.3rdcrjn)

[12](#_heading=h.26in1rg)

[**2 Thành phần chức năng:** 12](#_heading=h.lnxbz9)

[**Chương 3: Cài đặt** 18](#_heading=h.35nkun2)

[**1. Môi trường cài đặt và công nghệ sử dụng:** 18](#_heading=h.1ksv4uv)

[**2. Tổ chức các lớp trong chương trình:** 20](#_heading=h.44sinio)

[**3 Kết nối cơ sở dữ liệu:** 22](#_heading=h.2jxsxqh)

[**4 Các chức năng:** 24](#_heading=h.z337ya)

[**Chương 4: Kết luận** 86](#_heading=h.1y810tw)

[**1 Kết quả đạt được:** 86](#_heading=h.4i7ojhp)

[**2 Các hạn chế:** 86](#_heading=h.2xcytpi)

[**3 Hướng phát triển:** 86](#_heading=h.1ci93xb)

[**4 Tài liệu tham khảo:** 86](#_heading=h.3whwml4)

# **Chương 1: Tổng Quan**

## **1. Lí do chọn đề tài:**

Trong xu thế phát triển mạnh mẽ của khoa học và công nghệ hiện nay, công nghệ thông tin ngày càng được ứng dụng nhiều vào cuộc sống. Ngày nay, với sự phát triển mạnh mẽ của CNTT kết hợp với internet đã kết nối mọi người trong cuộc sống lại với nhau. Nó trở thành một công cụ đắc lực cho nhiều ngành nghề: giao thông, y học, … và đặc biệt là trong lĩnh vực thương mại.

Trước đây, trước khi việc quản trị cơ sở dữ liệu chưa được công nghệ hóa như hiện tại, việc quản lý một số lượng hàng hóa lớn ở .những cửa hàng hay trung tâm thương mại đều được làm thủ công dẫn đến việc mất thời gian và dễ xảy ra sai sót. Ngày nay, với sự giúp đỡ của công nghệ việc quản trị hàng hóa trở nên dễ dàng và chính xác hơn.

Cùng với sự phát triển của internet việc mua bán hàng hóa không còn bị gói gọn trong khuôn khổ cửa hàng mà còn cả hình thức mua hàng online. Với sự giúp đỡ của hệ quản trị cơ sở dữ liệu việc quản lí mua bán hàng hóa tại cửa hàng và online dễ dàng, thuận tiện và giảm thiểu được các sai sót.

Nhận ra nhu cầu trong việc này, cùng với sự giúp đỡ của thầy Nguyễn Thành Sơn, chúng em đã từng bước nghiên cứu và vận dụng các kiến thức đã được học, tìm hiểu, phân tích và xây dựng một website mua bán nội thất. Tuy nhiên, do kiến thức còn hạn chế nên thành phẩm không tránh khỏi những thiếu sót. Vì vậy, chúng em rất mong nhận được những sự góp ý của thầy để có thể từng bước xây dựng website ngày càng hoàn thiện và hiệu quả hơn.

## **2. Mô tả đề tài:**

### **2.1 Các dữ liệu cần quản lí:**

1.Sản phẩm: Gồm các thông tin như mã sản phẩm, tên sản phẩm, chi tiết, màu sắc, loại sản phẩm, hình ảnh minh họa, giá, mô tả sản phẩm.

2.Loại sản phẩm: Gồm các thông tin như tên loại, mã loại để quản lí. Một loại có thể có nhiều sản phẩm.

3. Chi tiết đơn hàng: Gồm các thông tin như: mã đơn hang, mã sản phẩm, tên sản phẩm, số lượng, giá.

4. Kho hàng: Gồm các thông tin như: mã sản phẩm, số lượng còn lại.

5. Đơn vị vận chuyển: Gồm các thông tin như: Mã đơn vị, tên đơn vị, giá nội thành, giá ngoại thành.

6. Đơn hàng: Gồm các thông tin như mã đơn hàng, ngày đặt, tổng tiền, mã khách hang, tên khách hàng, địa chỉ, tình trạng. Một đơn hàng có thể nhiều chi tiết đơn hàng

7. Khách hàng: Gồm các thông tin như mã khách hàng, tên, số điện thoại, email, địa chỉ, số điện thoại, tình trạng. Khách hàng có thể có đặt nhiều đơn hàng khác nhau.

8. Thành Phố: Gồm các thông tin như: mã thành phố và tên thành phố.

9. Tài khoản: Gồm các thông tin như: Tên tài khoản, mật khẩu, email, quyền truy cập.

### **2.2 Các yêu cầu về nghiệp vụ:**

#### **2.2.1 Dành cho khách hàng:**

Website được xây dựng với mục tiêu là nơi trưng bày, giới thiệu sản phẩm và các dịch vụ hỗ trợ khách hàng như: Tra cứu, đặt mua sản phẩm…

Khách hàng bắt đầu duyệt những sản phẩm trên website. Khi đã ưng ý một sản phẩm khách hàng có thể kích vào hình sản phẩm hoặc tên để xem thông tin chi tiết về sản phẩm. Sau đó khách hàng có thể chọn nút “**Add to Cart**” để đưa sản phẩm vào giỏ hàng của mình, giỏ hàng chỉ đơn giản là danh sách các sản phẩm bao gồm những thông tin như mã sản phẩm, tên sản phẩm, đơn giá, số lượng, thành tiền.

Các giỏ hàng thường cung cấp các tùy chọn để xóa, một sản phẩm, tiếp tục mua sản phẩm và cập nhật số lượng.

Ngay sau khi khách hàng có tất cả các sản phẩm cần mua, khách hàng có thể chọn nút “**Checkout**”. Trang đơn hàng này gồm có thông tin về khách hàng (họ tên, email, địa chỉ, số điện thoại, email…), thông tin về địa chỉ giao hàng (địa chỉ, quận/huyện, thành phố), thông tin về phương thức vận chuyển(dịch vụ vận chuyển, đơn giá vận chuyển,…) sau cùng là thông tin về giỏ hàng (mã sản phẩm, tên sản phẩm, đơn giá, số lượng, thành tiền).

Sau khi khách hàng đã điền đầy đủ thông tin thì chọn nút “**Checkout**” để hoàn tất quá trình mua hàng.

#### **2.2.2 Dành cho quản trị:**

Website cung cấp cho nhân viên hệ thống giao diện dùng để quản trị các thông tin như sản phẩm, đơn vị vận chuyển, đơn hàng, kho hàng, quyền truy cập, tài khoản cá nhân, thống kê doanh thu, thống kê đơn hàng theo ngày,…

### **2.3 Các yêu cầu về chức năng:**

Gồm có 2 đối tượng chính sử dụng website là: Khách hàng thông thường, Khách hàng thành viên, Người quản trị.

Chức năng của mỗi đối tượng:

- Khách hàng thông thường:

Xem danh sách sản phẩm: Khi người dùng truy cập vào website thì hệ thống sẽ tự động hiển thị một số sản phẩm. Khi người dùng chọn vào một danh mục sản phẩm cụ thể nào thì chi tiết sản phẩm của đó sẽ hiển thị lên giao diện

Xem chi tiết sản phẩm: Người dùng có thể chọn vào sản phẩm để xem thông tin chi tiết của sản phẩm đó.

Đăng ký: Khi khách hang thông thường có thể thực hiện chức năng đăng ký để trở thành thành viên của website và thực hiện các chức năng mua hàng.

Đưa sản phẩm vào giỏ hàng: Sau khi khách hàng tìm thấy sản phẩm đáp ứng nhu cầu của mình thì có thể chọn đưa sản phẩm vào giỏ hàng.

Cập nhật giỏ hàng: Khi người dùng đã lựa chọn một sản phẩm nhưng tìm được một sản phẩm hợp lý hơn thì có thể xóa sản phẩm đã có trong giỏ hàng vào chọn lại sản phẩm mới. Ngoài ra, khi cần mua sản phẩm đó với một số lượng thì người dùng cũng có thể cập nhật trong giỏ hàng.

- Khách hàng thành viên: Ngoài các chức năng của Người dùng chung còn có một số chức năng sau:

Đặt mua: Sau khi xem thông tin sản phẩm đúng với nhu cầu thì người dùng có thể chọn đặt mua để đưa sản phẩm vào giỏ hàng.

Thanh toán: Khi đã quyết định chắc chắn mua sản phẩm, người dùng có thể chọn chức năng thanh toán để thực hiện xác nhận đơn hàng cho mình. Trong quá trình thực hiện chức năng thanh toán thì người dùng có thể thấy được danh sách sản phẩm mà mình đặt mua trong giỏ hàng. Người dùng cần điền một số thông tin như người nhận (người nhận và người đặt mua là hai người khác nhau), chọn hình thức giao hàng. Sau khi thực hiện xong các bước thì người dùng xác nhận đơn hàng.

Người quản trị: Có hai loại: chỉ có chức năng quản trị và loại có tất cả các chức năng của khách hàng thành viên và có thêm chức năng quản trị:

Quản lý sản phẩm, đơn hàng, khách hàng, một số thông tin khác: Để thực hiện chức năng quản lý thì yêu cầu nhân viên phải đăng nhập hệ thống. Nhân viên chọn mục mình cần thực hiện quản lý như sản phẩm, quản lý đơn hàng, thống kê,…...

Trong quản lý có chức năng xem, thêm, sửa, xóa sản phẩm. Chọn thao tác cần thực hiện, điền đầy đủ thông tin trong quá trình thực hiện rồi xác nhận thao tác.

Trong thống kê có chức năng xem doanh thu của tháng, xem số lượng khách hàng đăng ký, biểu đồ thống kê, xem đơn hàng theo ngày.

# **Chương 2: Phân tích và thiết kế**

## **1 Thành phần dữ liệu:**

### **1.1 Lược đồ ERD của cơ sở dữ liệu:**

### 

### **1.2 Các bảng trong mô hình quan hệ:**

TaiKhoan

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **TaiKhoan** | Nvarchar(30) | Tên tài khoản |
| MatKhau | nvarchar(30) | Mật khẩu |
| QuyenTruyCap | Nvarchar(2) | Quyền truy cập |

2.KhachHang

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **MaKH** | Int | Mã Khách hàng |
| TenKh | Nvarchar(50) | Tên khách hàng |
| NamSinh | date | Ngày sinh |
| DiaChi | nvarchar(50) | Địa chỉ |
| Quan | nvarchar(30) | Quận |
| ThanhPho | nvarchar(30) | Thành phố |
| Email | nvarchar(100) | Email |
| sdt | nvarchar(11) | Số điện thoại |
| Taikhoan | nvarchar(30) | Tài khoản |
| hinhanh | nvarchar(200) | Hình ảnh |

3.Loại sản phẩm

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **maloai** | int | Mã loại sản phẩm |
| tenloai | Nvarchar(50) | Tên loại sản phẩm |

4.Sản phẩm

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **MaSp** | int | Mã hãng sản phẩm |
| tenSP | nvarchar(250) | Tên mã hãng |
| MoTa | nvarchar(MAX) | Mô tả |
| MaLoai | int | Mã loại |
| TrangThai | nvarchar(10) | Trạng thái |
| gia | real | Giá |
| Giamgia | int | Giảm giá |
| mausac | nvarchar(30) | Màu Sắc |
| SoLuongDangDat | int | Số lượng đặt hàng |
| Hinhanh | nvarchar(200) | Hình ảnh |
| LastModify | datetime | Lần chỉnh sửa cuối |

5.Khohang

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **MaSp** | Int | Mã sản phẩm |
| tenSP | Nvarchar(50) | Tên sản phẩm |
| SoLuong | Int | Số lượng còn lại |

6.DonHang

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **MaDH** | int | Mã đơn hàng |
| TenKH | nvarchar(50) | Mã sản phẩm |
| DiaChi | nvarchar(50) | Mã kích thước |
| Quan | nvarchar(30) | Quận |
| ThanhPho | nvarchar(100) | Thành phố |
| Email | nvarchar(50) | E mail |
| sdt | nvarchar(11) | Số điện thoại |
| MaDvVanChuyen | int | Mã đơn vị vận chuyển |
| TongTien | real | Tổng tiền |
| TrangThai | Nvarchar(10) | Trạng thái |
| NgayDat | date | Ngày đặt |
| MaKH | int | Mã khách hàng |

7.DvVanChuyen

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **maDvVanChuyen** | int | Mã đơn vị vẫn chuyển |
| TenDv | nvarchar(30) | Tên đơn vị vận chuyển |
| GiaNoiThanh | Real | Giá nội thành |
| GiaNgoaiThanh | Real | Giá ngoại thành |

8.ChiTietDonHang

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **MaDH** | Int | Mã đơn hàng |
| MaSP | Int | Mã sản phẩm |
| SoLuong | Int | Số lượng |
| Gia | Real | Giá |
| TenSP | Nvarchar(50) | Tên sản phẩm |

9.Taikhoan

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| **TaiKhoan** | Nvarchar(50) | Tên tài khoản |
| MatKhau | Nvarchar(50) | Mật khẩu |
| Email | Nvarchar(200) | Email |
| QuyenTruyCap | Nvarchar(2) | Quyền truy cập |

### **1.3 Lược đồ quan hệ:**

### 

## **2 Thành phần chức năng:**

**Chức năng đăng kí**:

**Tóm tắt:** Người dùng sử dụng chức năng “Đăng ký” để tạo tài khoản cho mình trên website.

**Phân quyền: Người dùng**

B1. Trên giao diện màn hình chính, Người dùng chung chọn **Sign In** rồi chọn **Sign Up**

B2. Hệ thống sẽ hiển thị giao diện đăng ký và Người dùng nhập thông tin vào giao diện để lưu vào cơ sở dữ liệu.

B3. Kết thúc.

**Trạng thái hệ thống khi bắt đầu thực hiện:** Trước khi bắt đầu thực hiện chức năng không cần điều kiện gì.

**Trạng thái hệ thống sau khi thực hiện:** Sau khi thực hiện chức năng hệ thống sẽ chuyển đến trang cập nhật thông tin cá nhân, để khách hang điền các thông tin cá nhân và lưu xuống bản KhachHang

**Chức năng đăng nhập:**

**Tóm tắt:** Khách hàng thành viên và người quản trị sử dụng chức năng “Đăng nhập” để tham gia mua hàng trực tuyến hoặc quản lý.

**Phân quyền: Khách hàng thành viên hoặc Người quản trị**

B1. Trên giao diện màn hình chính, Khách hàng thành viên chọn **Sign In**.

B2. Hệ thống sẽ hiển thị giao diện đăng nhập và Khách hàng thành viên, Người quản trị nhập thông tin vào giao diện để kiểm tra tài khoản và mật khẩu.

B3. Kết thúc.

Nếu Khách hàng thành viên không muốn đăng nhập thì chọn **nút thoát trên góc phải màn hình**

**Trạng thái hệ thống khi bắt đầu thực hiện:** Trước khi bắt đầu thực hiện chức năng yêu cầu phải Đăng ký(tài khaonr phải tồn tại trong cơ sở dữ liệu).

**Trạng thái hệ thống sau khi thực hiện:** Sau khi thực hiện chức năng hệ thống sẽ chuyển tới trang tương ứng với Khách hang thành viên hoặc người quản trị

**Chức năng đưa sản phẩm vào giỏ hàng:**

**Tóm tắt:** Người dùng chung sử dụng chức năng “Đưa sản phẩm vào giỏ hàng” để đặt những sản phẩm mình cần mua vào không gian lưu trữ tạm thời trên web.

**Phân quyền: Người dùng**

B1. Trên giao diện màn hình chi tiết sản phẩm, Người dùng chọn **Add to Cart**

B2. Hệ thống sẽ lưu trữ thông tin sản phẩm mà Người dùng đưa vào giỏ

B3. Kết thúc.

**Trạng thái hệ thống khi bắt đầu thực hiện:** Sau khi khách hàng xem thông tin sản phẩm.

**Trạng thái hệ thống sau khi thực hiện:** Sau khi thực hiện chức năng hệ thống sẽ xuất thông tin của sản phẩm ra giao diện Giỏ hàng.

**Chức năng đặt mua:**

**Tóm tắt:** Khách hàng sử dụng chức năng “Đặt mua” để tham gia mua hàng trực tuyến.

**Phân quyền: Khách hàng**

B1. Trên giao diện giỏ hàng, Khách hàng chọn **Đặt mua**.

B2. Hệ thống sẽ hiển thị giao diện chứa thông tin khách hàng và danh sách các sản phẩm khách hàng đặt mua. Sau khi nhập đầy đủ thông tin thì khách hàng xác nhận đặt mua.

B3. Kết thúc.

**Trạng thái hệ thống khi bắt đầu thực hiện chức năng:** Trong giỏ hàng của khách hàng phải có tối thiểu 1 sản phẩm.

**Trạng thái hệ thống sau khi thực hiện chức năng:** Sau khi thực hiện chức năng hệ thống sẽ thông báo mua hang thành công

**Chức năng quản lí tài khoản admin:**

Khi đăng nhập vào trang admin với tài khoản nào thì chỉ xem được thông tin của tài khoản đóvà chỉ được sửa những thông tin của tài khoản đó, ngoài ra còn có chức năng cấp quyền đăng nhập cho một tài khoản người dùng

**Tóm tắt:** Admin dùng chức năng “quản lí tài khoản admin” này để quản lí tài khoản của mình

**Phân quyền: Người quản trị**

B1. Trên giao diện của trang admin, admin chọn **Account**

B2. Hệ thống sẽ hiển thị giao diện của trang thông tin tài khoản chọn chức năng **Edit trên giao diện**

B3. Sau đó người quản trị chỉnh sửa các thông tài khoản.

B4. Kết thúc.

**Các yêu cầu đặc biệt:** Để thực hiện được chức năng này yêu cầu quản trị viên phải thực hiện đăng nhập vào hệ thống.

**Chức năng quản lí sản phẩm:**

Chức năng này cho phép người quản trị thêm sản phẩm mới, cập nhật sản phẩm và xóa sản phẩm không còn bán nữa.

**Tóm tắt:** Admin dùng chức năng “quản lí sản phẩm” này để quản lí sản phẩm trên giao diện quản trị viên

**Phân quyền: Người quản trị**

B1. Chọn vào Nút **Product** để chuyển đến trang xem danh sách các sản phẩm.

B2. Trên giao diện của trang Product, admin chọn **Create New** để tạo mới. Chọn vào Detail để xem chi tiết sản phẩm.

B3. Hệ thống sẽ hiển thị giao diện của trang tạo một sản phẩm mới nếu chọn **Create New**, hiển thị giao diện của trang Chi tiết sản phẩm. Chọn **Edit /Delete** để chuyển đến trang **Edit/Delete.**

B3. Sau đó người quản trị nhập các thông tin nếu tạo sản phẩm mới đối với tạo mới sản phẩm. Chỉnh sửa các thông tin nếu cập nhật sản phẩm.

B4. Kết thúc.

**Các yêu cầu đặc biệt:** Để thực hiện được chức năng này yêu cầu quản trị viên phải thực hiện đăng nhập vào hệ thống.

**Chức năng quản lí đơn hàng:**

Chức năng này cho phép người quản trị xem các đơn hàng mà khách hàng đã đặt, xem các thông tin của khách hàng như địa chỉ, số điện thoại và sau đó duyệt đơn hàng cho khách.

**Chức năng quản lí loại sản phẩm:**

Chức năng này cho phép người quản trị thêm loại sản phẩm mới, cập nhật loại sản phẩm và xóa loại sản phẩm không còn bán nữa.

**Tóm tắt:** Admin dùng chức năng “quản lí loại sản phẩm” này để quản lí loại sản phẩm trên giao diện quản trị viên

**Phân quyền: Người quản trị**

B1. Trên giao diện của trang Category để xem danh sách các loại sản phẩm.

B2. Có thể chọn nút Add new để chuyển đến trang Thêm loại sản phẩm và nhập đầy đủ các thông tin và them mới loại sản phẩm vào Cơ sở dữ liệu. Nếu chọn nút Chi tiết hệ thống sẽ chuyển đến trang Chi tiết loại sản phẩm

B3. Có thể chọn nút **edit** hoặc **delete** trên trang để có thể chỉnh sửa hoặc xóa loại sản phẩm.

B3. Sau đó người quản trị nhập các thông chỉnh sửa nếu cập nhật loại sản phẩm.

B4. Kết thúc.

**Các yêu cầu đặc biệt:** Để thực hiện được chức năng này yêu cầu quản trị viên phải thực hiện đăng nhập vào hệ thống.

**Chức năng quản lí vận chuyển:**

Chức năng này cho phép người quản trị them đơn vị vận chuyển mới, cập nhật đơn vị vận chuyển và xóa đơn vị vận chuyển.

**Tóm tắt:** Admin dùng chức năng “quản lí đơn vị vận chuyển” này để quản lí các đơn vị vận chuyển trên giao diện quản trị viên.

**Phân quyền: Người quản trị**

B1. Trên giao diện nhấn vào nút Delivery để xem danh sách các loại sản phẩm.

B2. Có thể chọn nút Add new để chuyển đến trang thêm đơn vị vận chuyển và nhập đầy đủ các thông tin để thêm mới loại sản phẩm vào Cơ sở dữ liệu. Nếu chọn nút Chi tiết hệ thống sẽ chuyển đến trang Chi tiết loại sản phẩm

B3. Có thể chọn nút **edit** hoặc **delete** trên trang để có thể chỉnh sửa hoặc xóa loại sản phẩm.

B3. Sau đó người quản trị nhập các thông chỉnh sửa nếu cập nhật loại sản phẩm.

B4. Kết thúc.

**Các yêu cầu đặc biệt:** Để thực hiện được chức năng này yêu cầu quản trị viên phải thực hiện đăng nhập vào hệ thống.

**Chức năng quản lý người dùng và quyền truy cập:**

**Tóm tắt:** Admin dùng chức năng “Quản lý người dùng và phân quyền truy cập” này để xem danh sách người dùng, xem danh sách người quản trị, cấp hoặc bỏ quyền truy cập đối với một người dùng nào đó.

**Phân quyền: Người quản trị**

B1. Trên giao diện nhấn vào nút Authorize để xem danh sách các người quản trị.

B2. Có thể chọn nút View User để xem dánh sách khách hàng.

B3. Có thể chọn nút **set hoặc reset** trên trang để có thể chỉnh sửa quyền truy cập của một tài khoản.

B4. Kết thúc.

**Các yêu cầu đặc biệt:** Để thực hiện được chức năng này yêu cầu quản trị viên phải thực hiện đăng nhập vào hệ thống.

**Chức năng xem thống kê:**

**Tóm tắt:** Admin dùng chức năng “thống kế” này để xem các thống kê về doanh thu, số lượng khách hàng, số lượng đơn hàng, số lượng đơn hàng trong ngày,…..

**Phân quyền: Người quản trị**

B1. Trên giao diện nhấn vào nút Statictical để xem thống kê

B2. Có thể chọn ngày và chọn nút View để xem thống kê cho ngày tương ứng

B3. Kết thúc.

**Các yêu cầu đặc biệt:** Để thực hiện được chức năng này yêu cầu quản trị viên phải thực hiện đăng nhập vào hệ thống.

# **Chương 3: Cài đặt**

## **1. Môi trường cài đặt và công nghệ sử dụng:**

Môi trường phát triển ứng dụng:

Lập trình bằng ngôn ngữ C#

Sử dụng ASP.net core với mô hình MVC

Giao diện với Html, Css và Javascript, Boostrap

Lập trình cơ sở dữ liệu (SQL Server)

Công cụ xây dựng cơ sở dữ liệu SQL Server 2017

Kết nối Cơ sở dữ liệu bằng ADO.NET

Công cụ đã dùng để xây dựng ứng dụng: Visual Studio 2019

Môi trường triển khai ứng dụng:

+ Hệ điều hành Windows 10

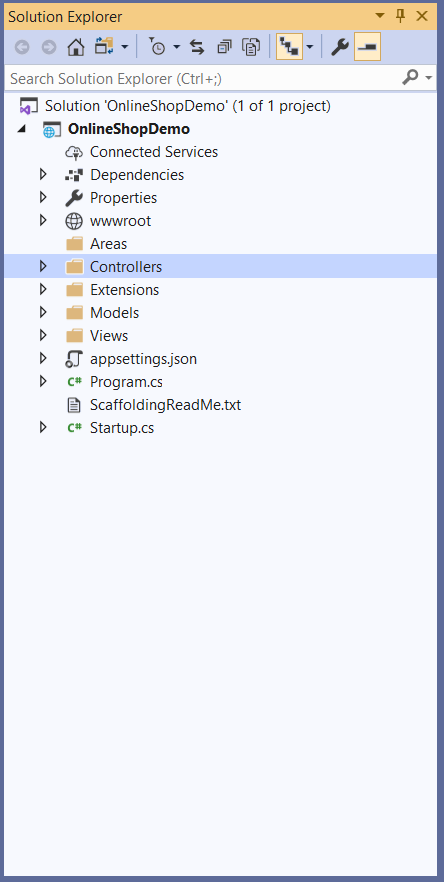
+ Để chạy được Website yêu cầu máy phải cài ít nhất một trình duyệt web.

+ Yêu cầu ASP.NET core 2.1

<https://dotnet.microsoft.com/download/dotnet-core/2.1>

<https://blogs.msdn.microsoft.com/benjaminperkins/2018/06/04/how-to-install-asp-net-core-2-1-for-development/>

## **2. Tổ chức các lớp trong chương trình:**



1. Properties: là file chứa cấu hình chung của project.

2. References: là nơi chứa các DLL (Dynamic Link Library) của ứng dụng.

3. Strartup.cs: Chứa các khởi tạo ban đầu, đường dẫn cho trang đầu tiên, đăng ký session(session có thể hiểu là: một nơi để lưu trữ các dữ liệu theo dạng từ điển(Key,Value), khi một người dùng đăng nhập thành công, người dùng được cấp một session dùng để lưu các thao tác của người dùng đó ví dụ lưu thông tin tài khoản, hoặc lưu các sản phẩm trong giỏ hàng khi người dùng mua hàng)

4. Controllers: chứa các controller nhận yêu cầu, xử lý và trả các view về cho phía người dùng.

5. Extentions: Chứa các lớp mở rộng chứa các phương thức dùng để lấy ra hoặc thêm dữ liệu vào session

6. Model: Chứa các model

7. Views: Chứa các views của giao diện của chương trình.

8. appsettings.json: chứa ConnectionString

## **3 Kết nối cơ sở dữ liệu:**

Sử dụng các thư viện của ADO.NET để thao tác với cơ sở dữ liệu có sẵn trong sql server:

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Mvc;

ConectionString trong appsettings.json

"AllowedHosts": "\*",

"ConnectionStrings": {

"DefaultConnection": "Server=DESKTOP-GISFHHL;Database=FurnitureShop;Trusted\_Connection=True;MultipleActiveResultSets=true"

}

Lấy ConnectionString lưu trong appsettings.json

public IConfiguration Configuration { get; }

private readonly IHostingEnvironment \_appEnvironment;

public

DeliveryController(IHostingEnvironment hostingEnvironment, IConfiguration configuration)

{

\_appEnvironment = hostingEnvironment;

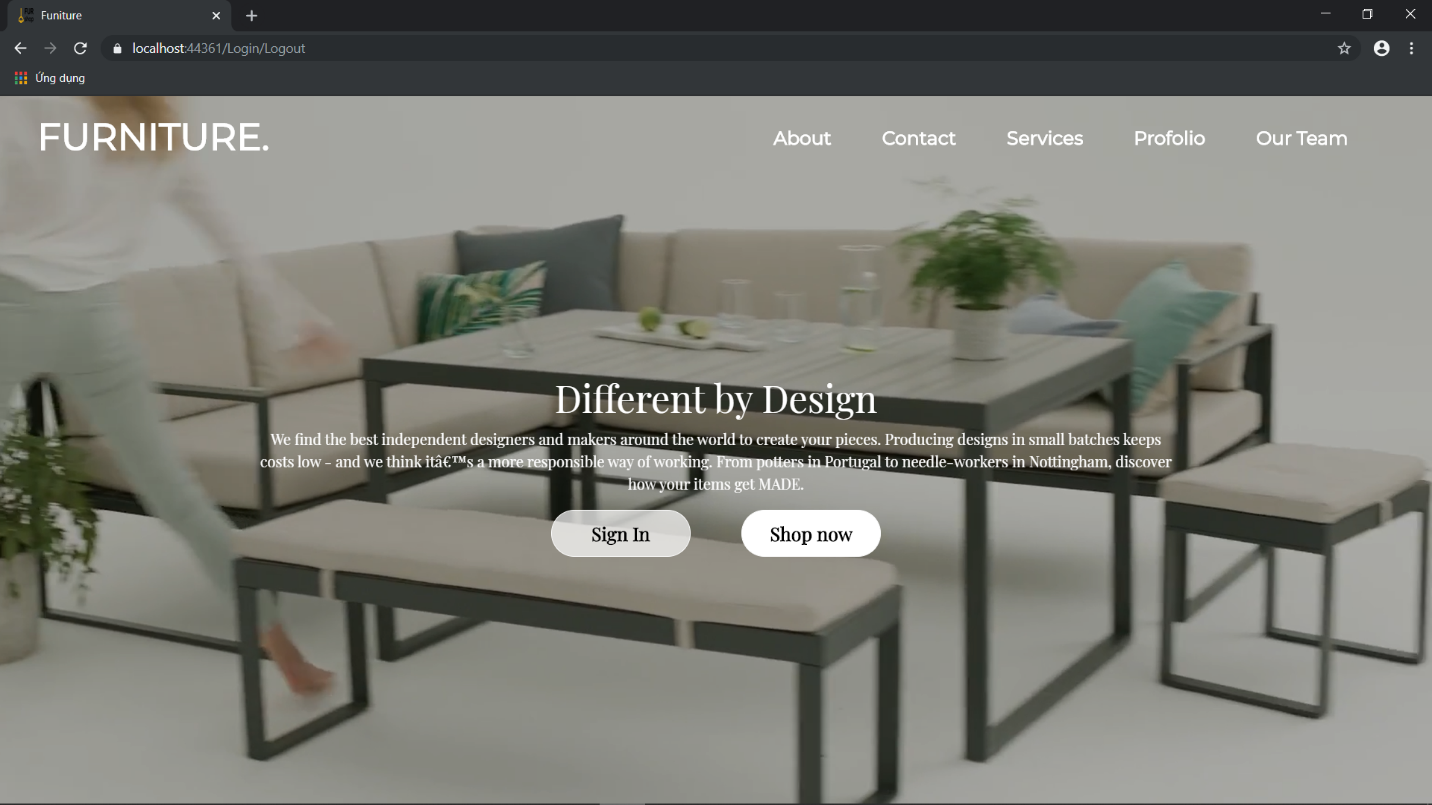
Configuration = configuration;

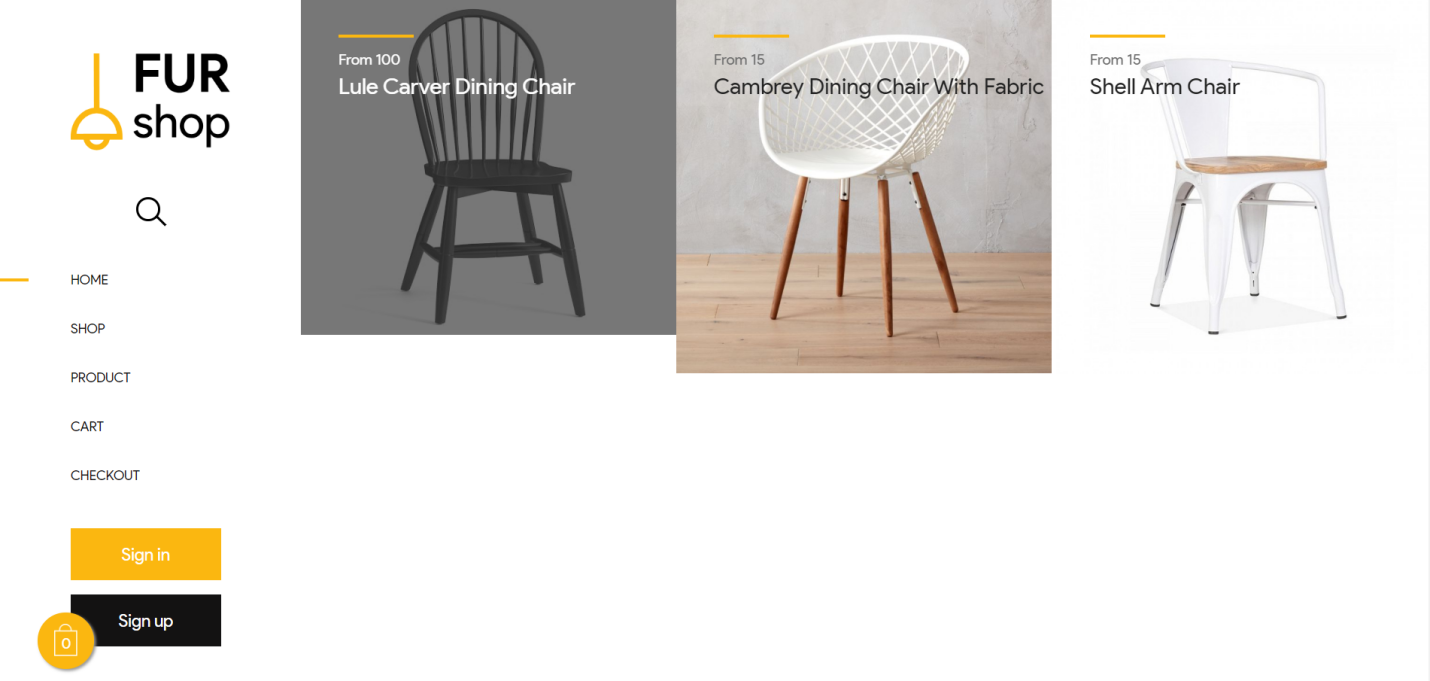
}

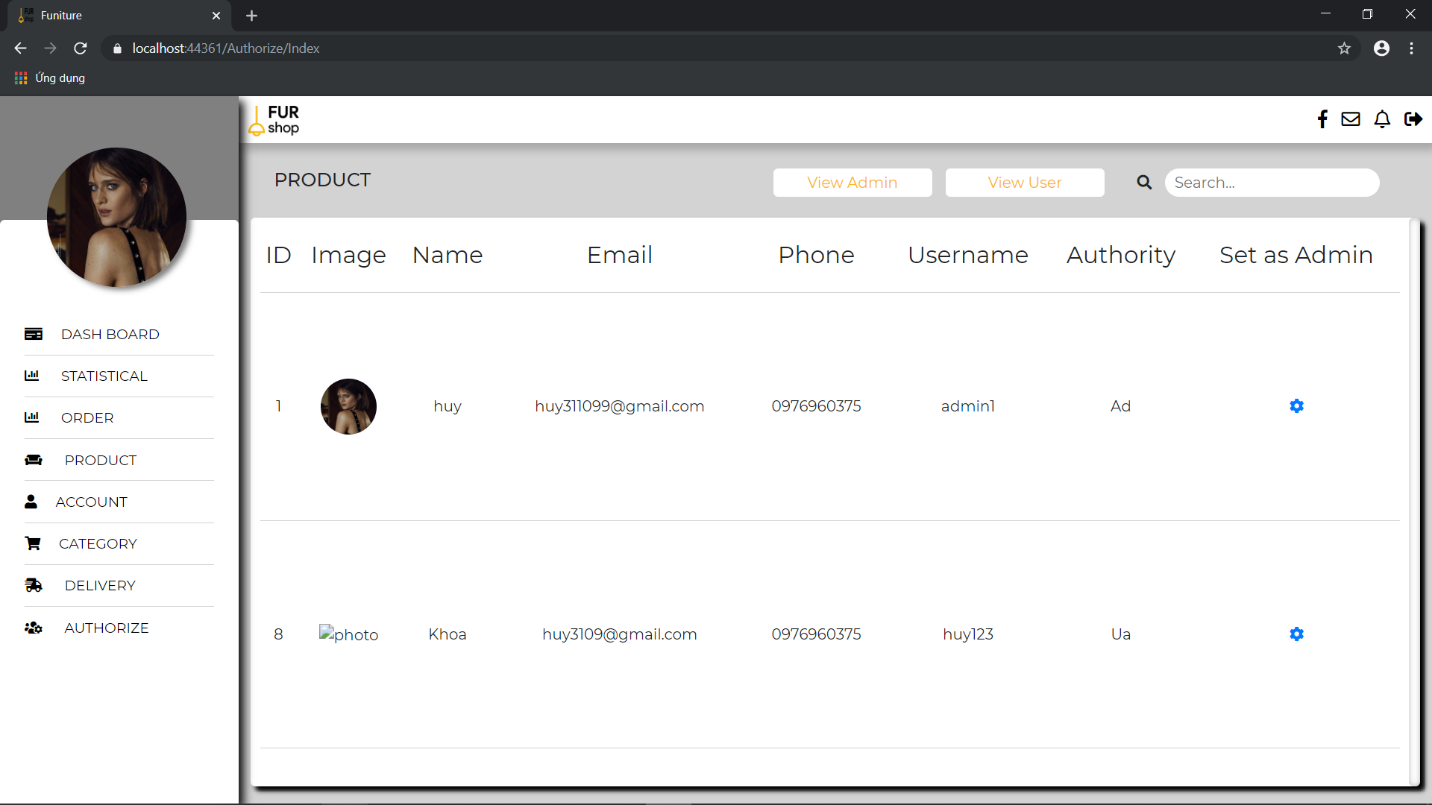
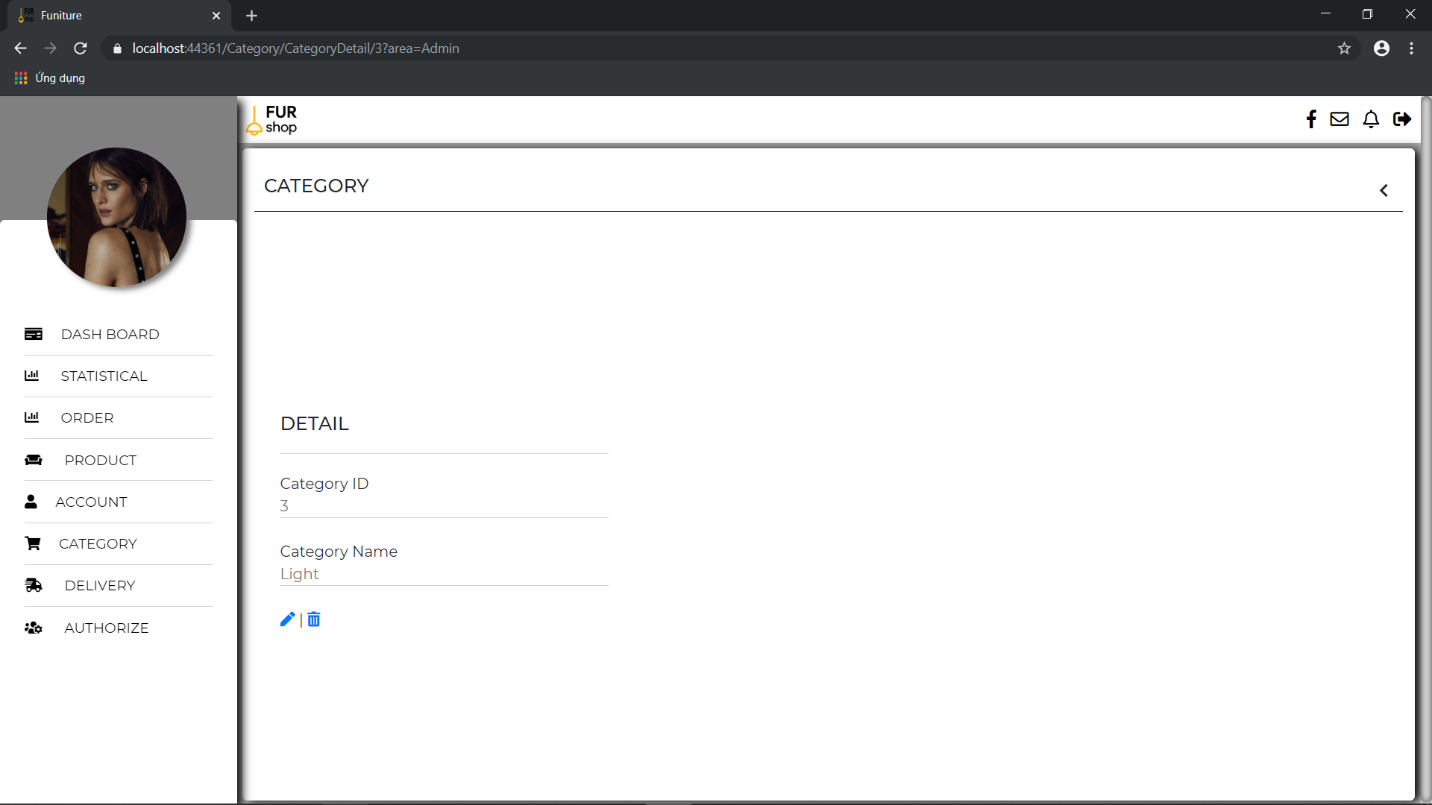
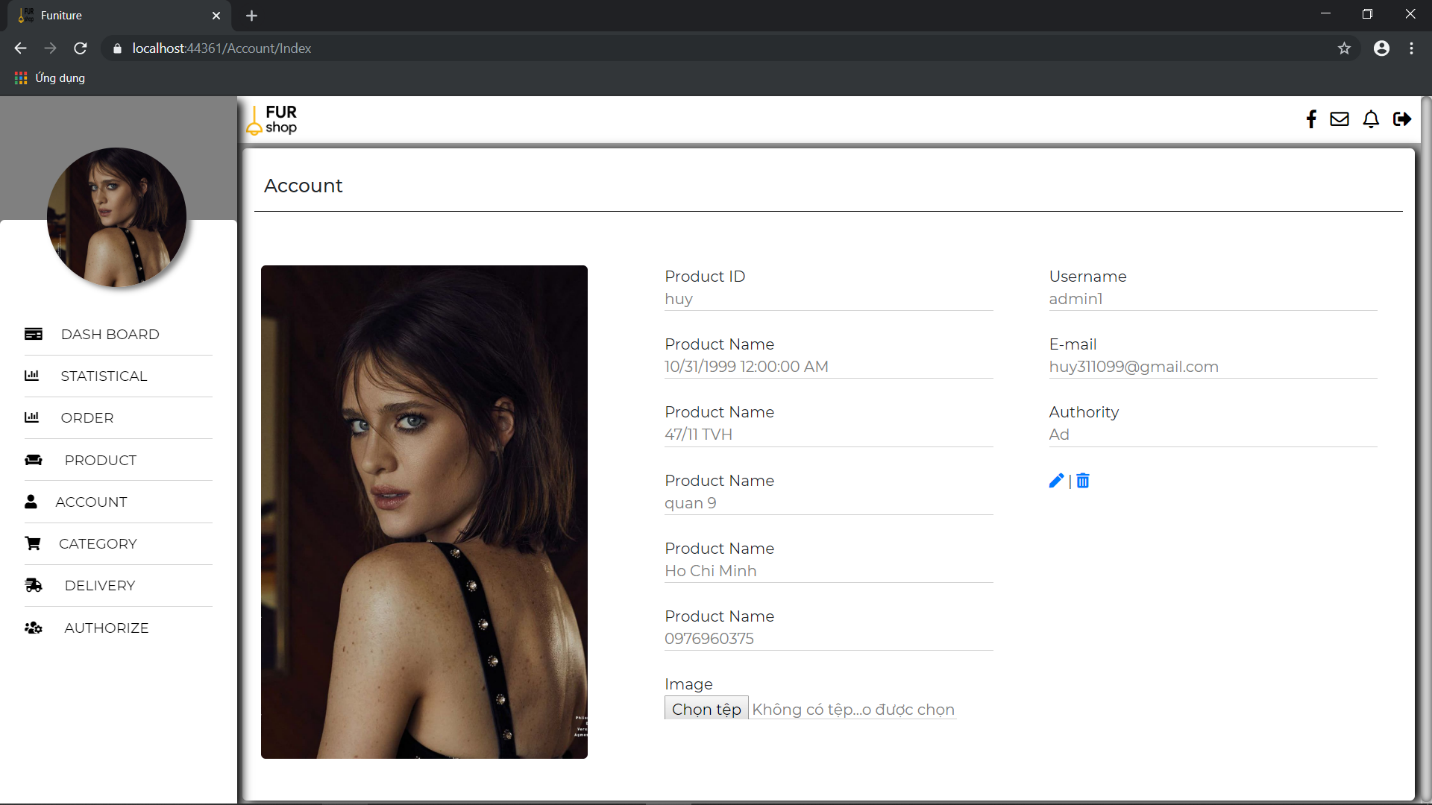
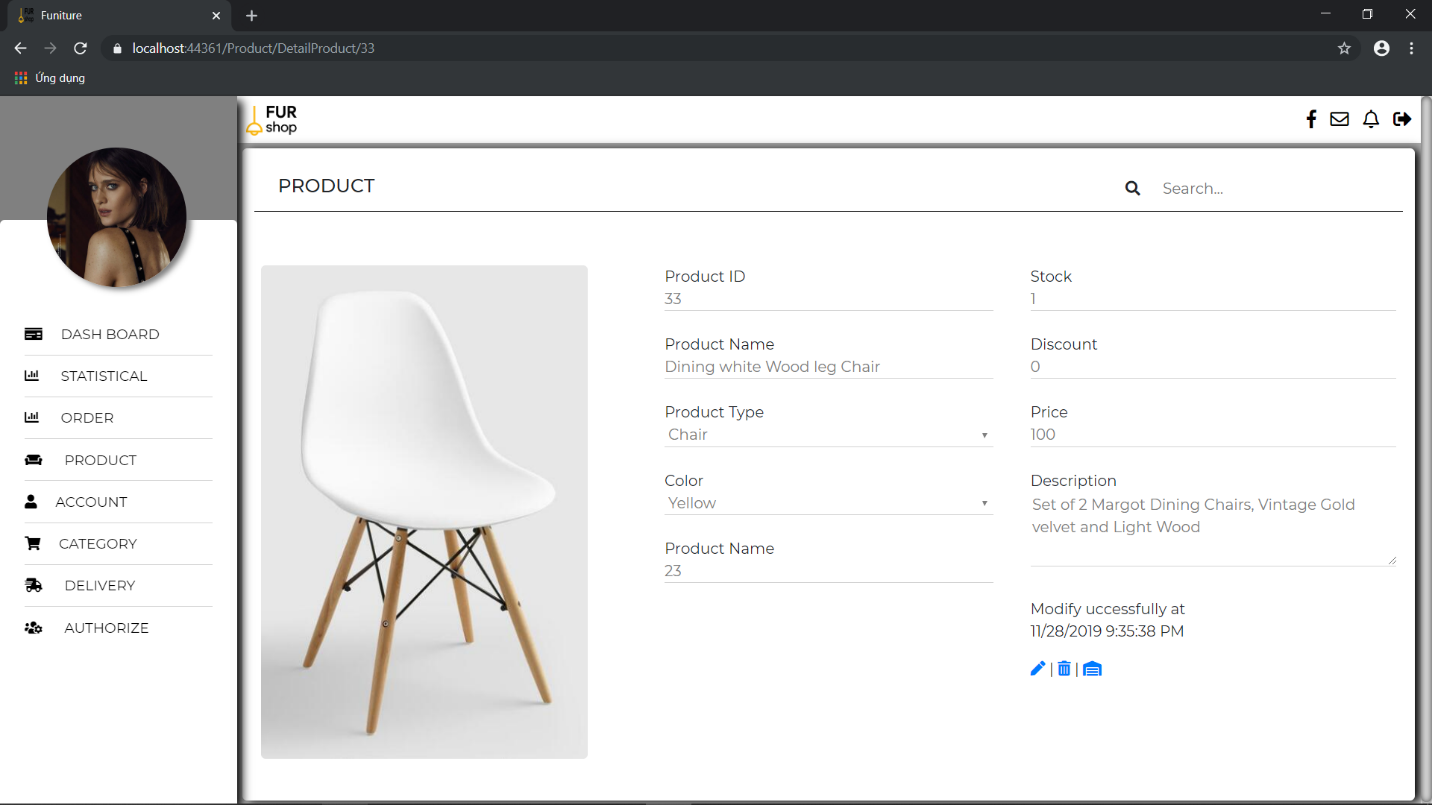
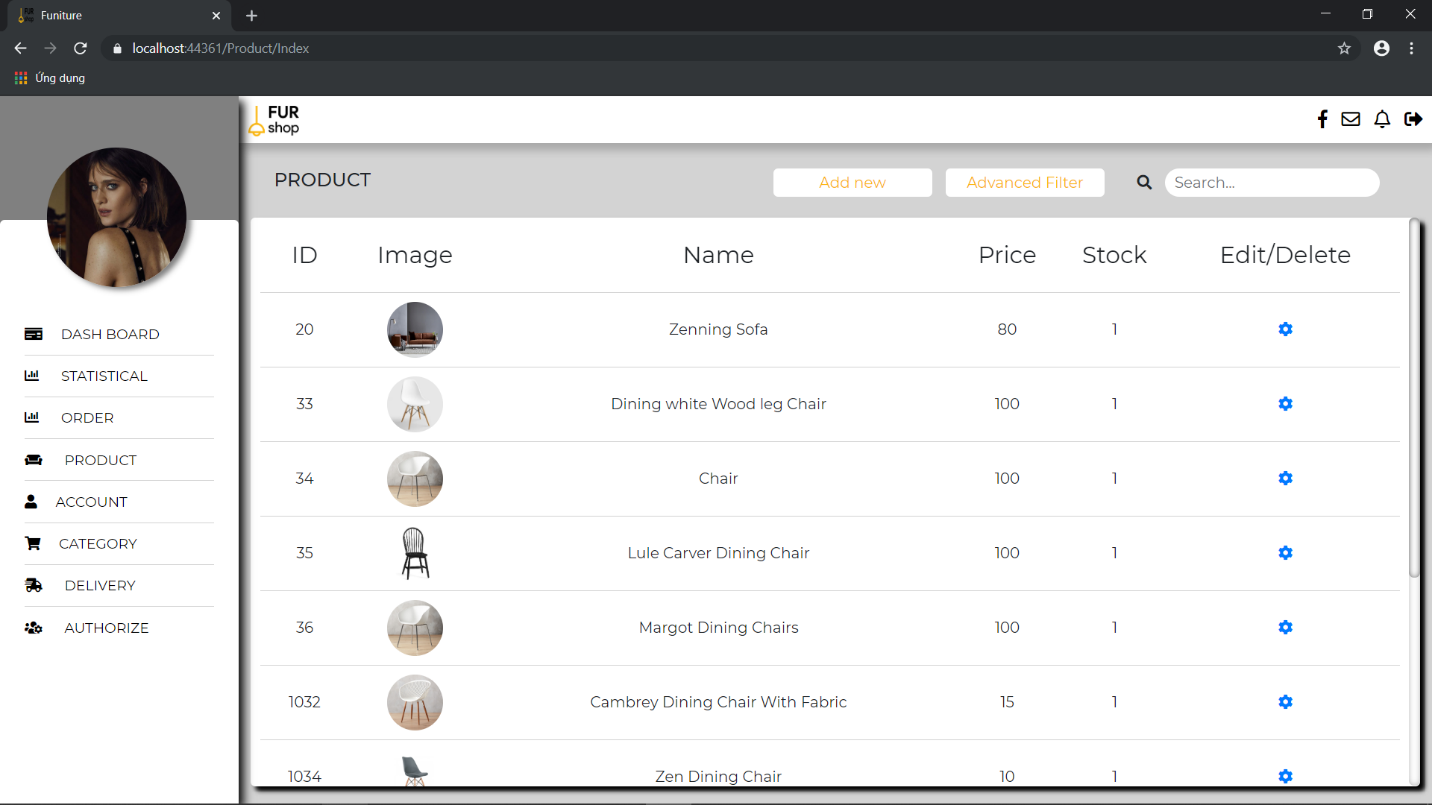
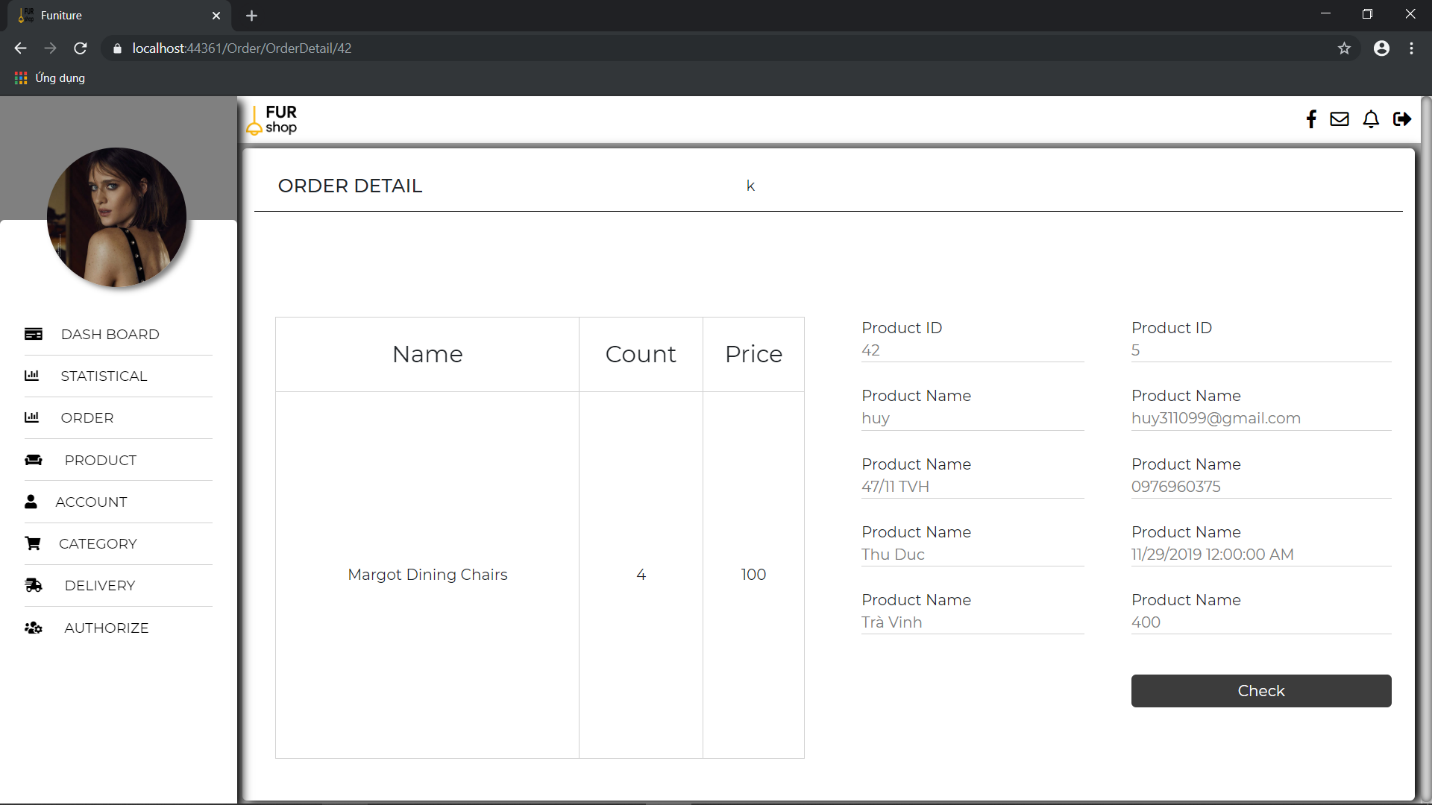
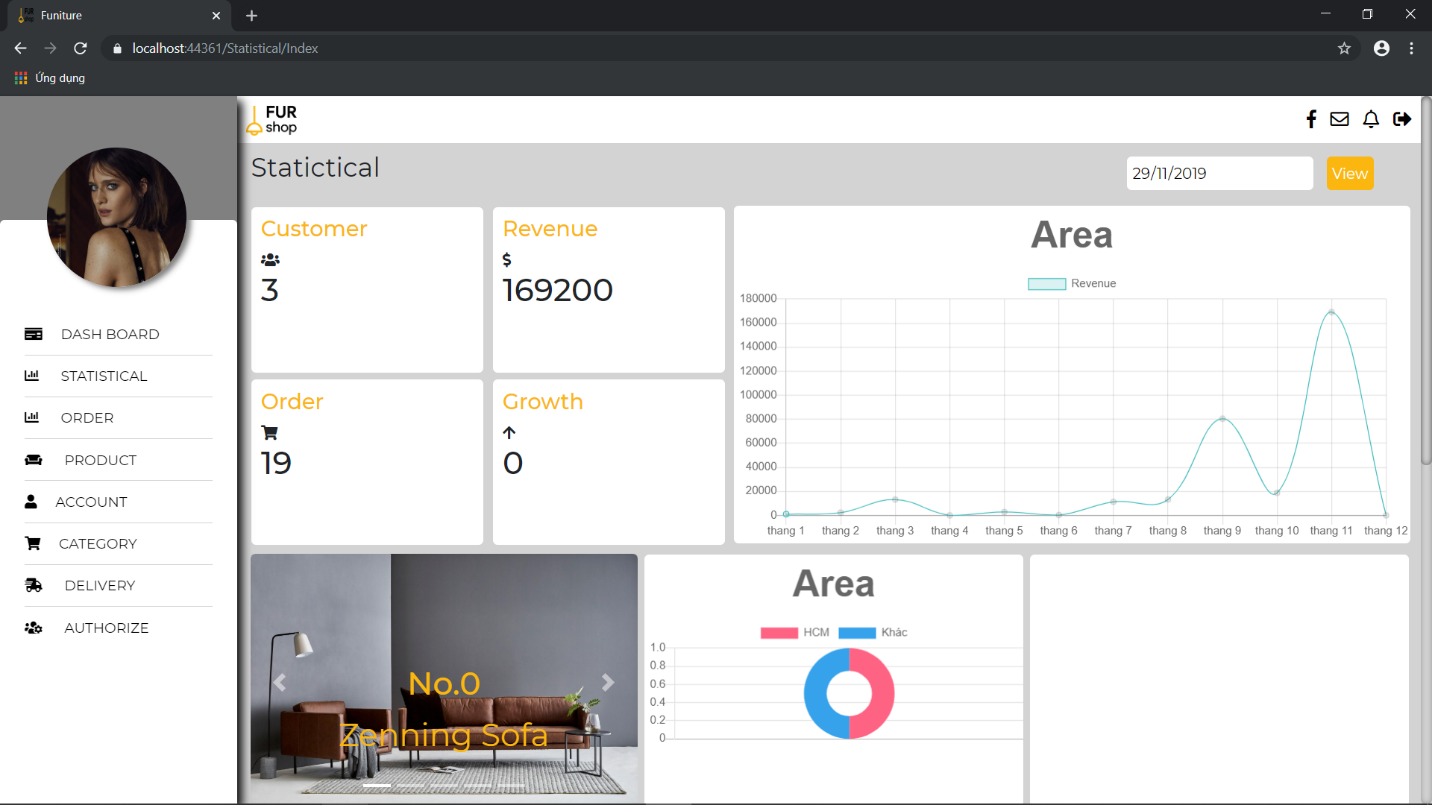
string connectionString = Configuration["ConnectionStrings:DefaultConnection"];

## **4 Các chức năng:**

Giao diện chính của người dùng:

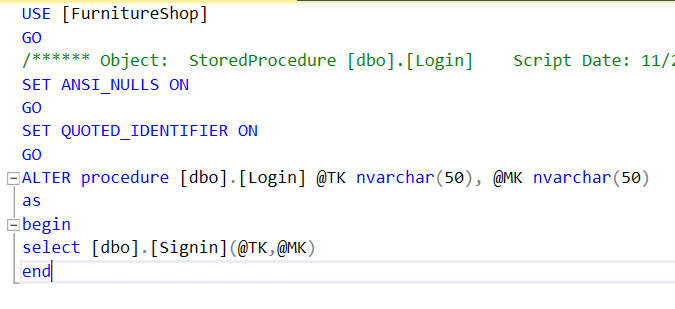




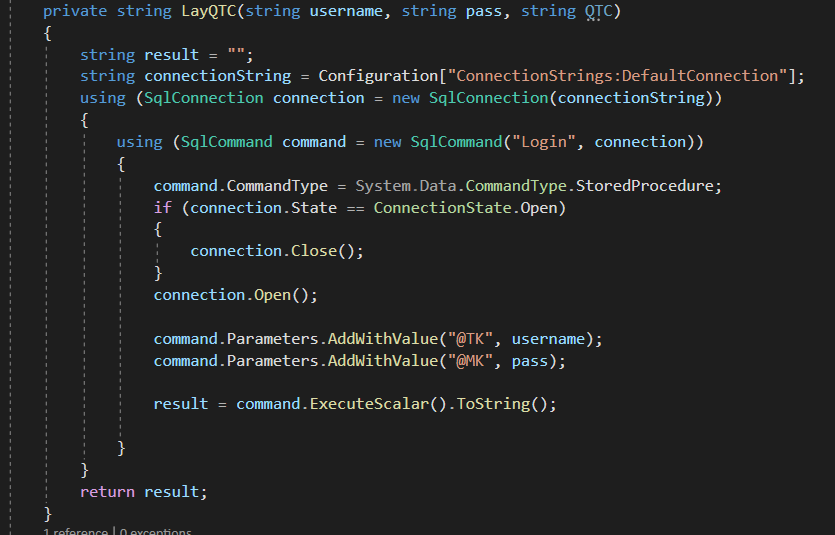


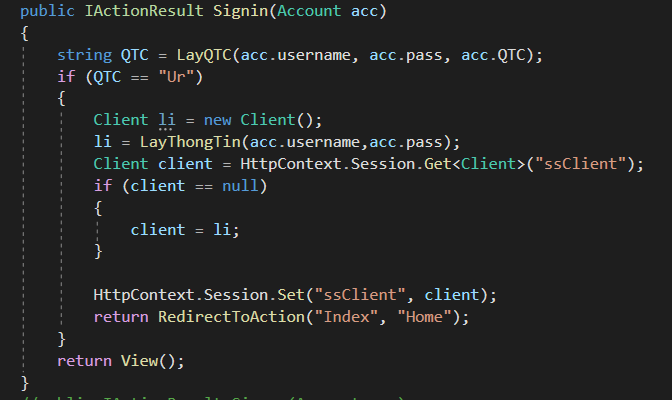
**Chức năng đăng nhập của Admin:**

+ Code sql:



+ Gọi thực thi hàm:

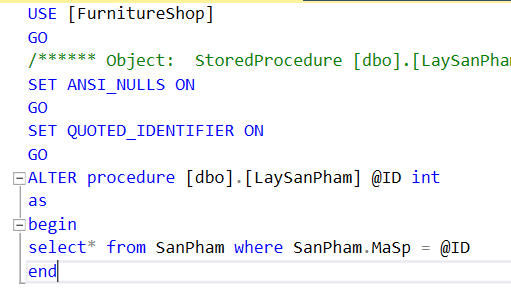




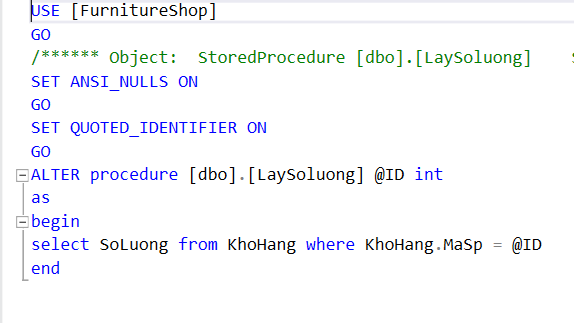
**Chức năng thêm vào giỏ hàng:**

+Code sql

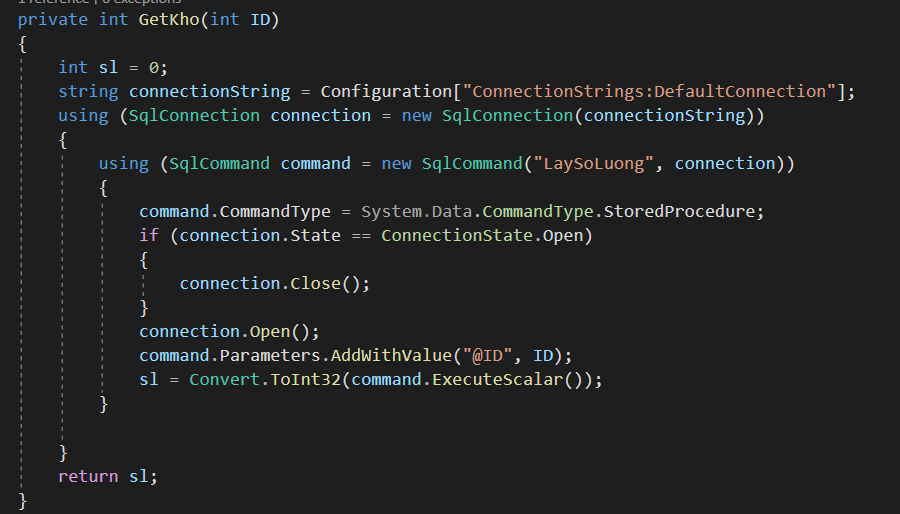
Hàm lấy sản phẩm:



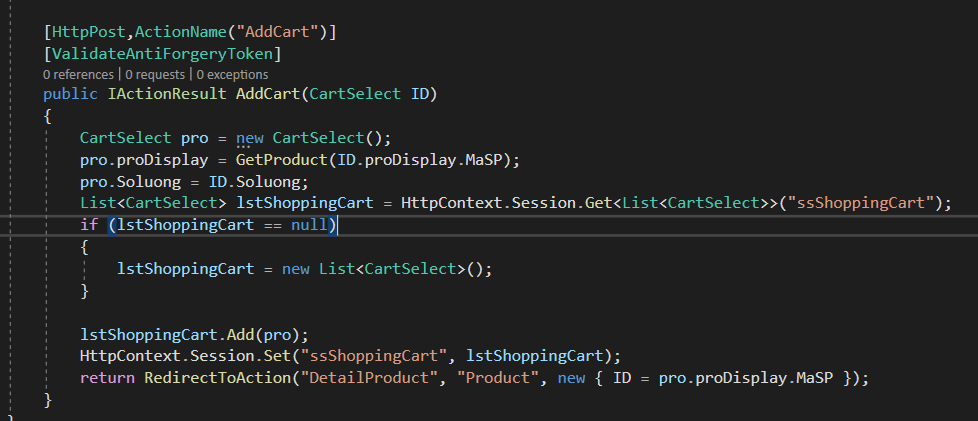
Hàm tìm số lượng còn lại:



+Gọi hàm thực thi:







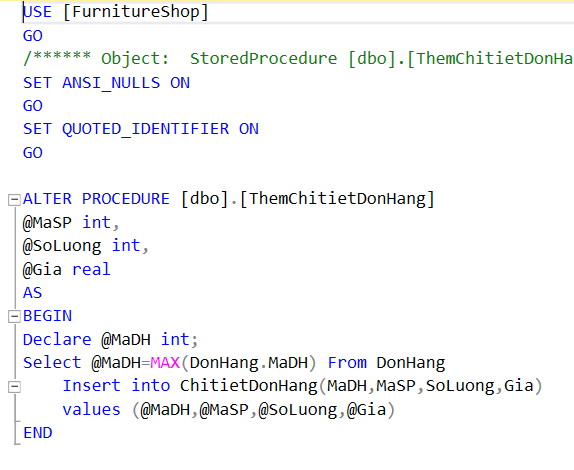
**Chức năng đặt mua:**

+ Code sql:

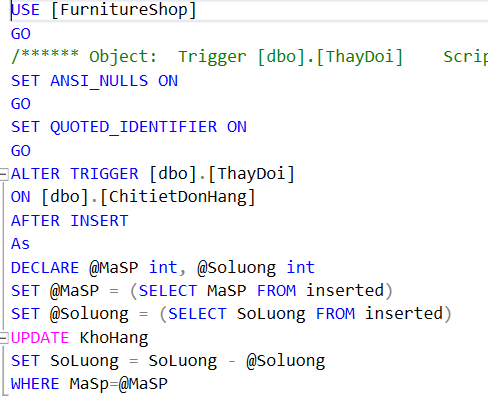
Thêm đơn hàng



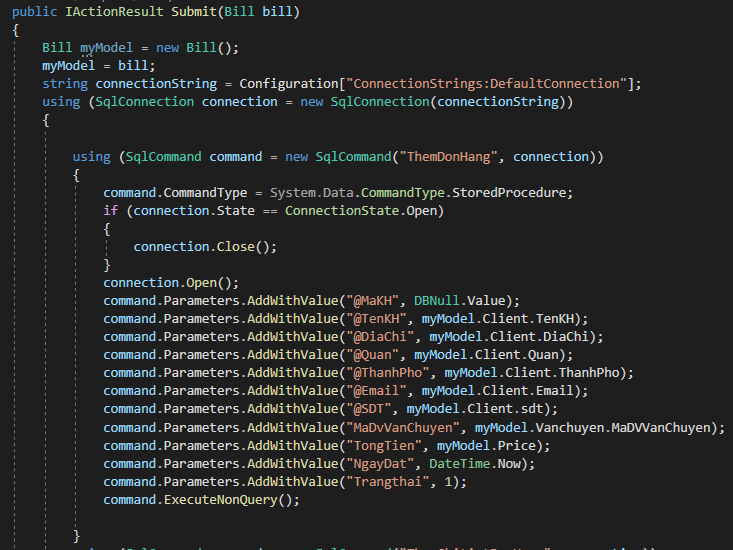
Thêm chi tiết đơn hàng

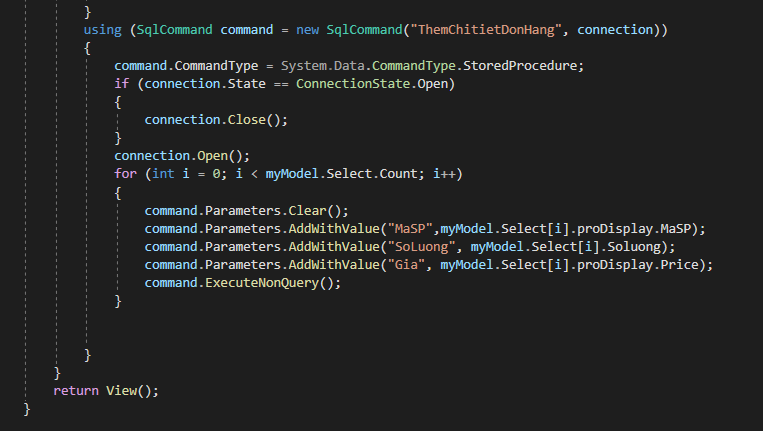


Trừ số sản phẩm trong kho



+ Gọi thực thi thủ tục và hàm:



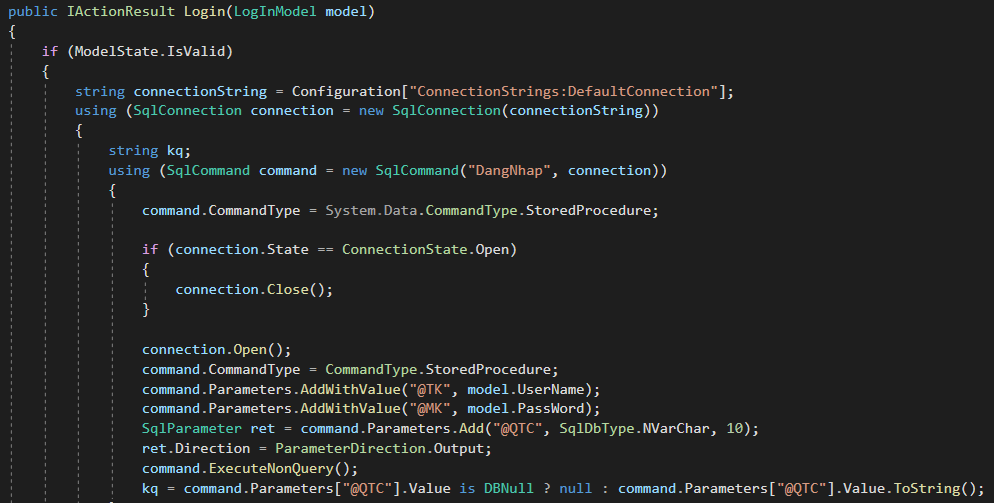


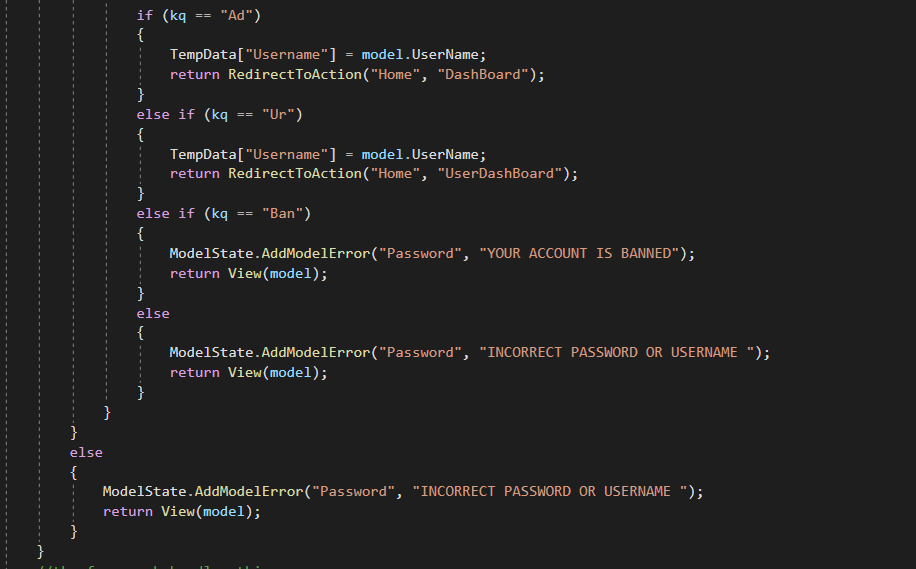
**Chức năng đăng nhập của người quản trị:**

+ Code sql:



+ Gọi thực thi hàm:



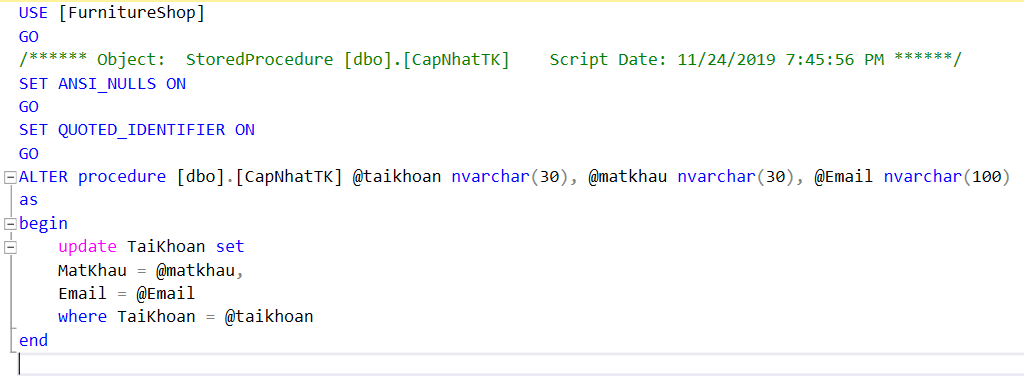


**Chức năng quản lí tài khoản admin:**

Giao diện của quản trị viên:

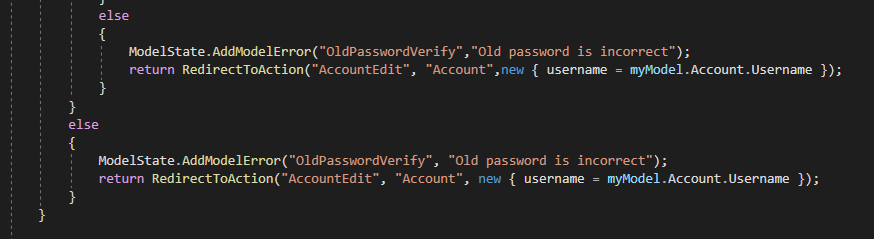
**+ Cập nhật tài khoản:**

+ Code sql:



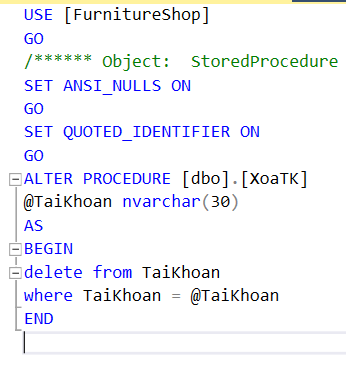
+ Gọi thực thi thủ tục:



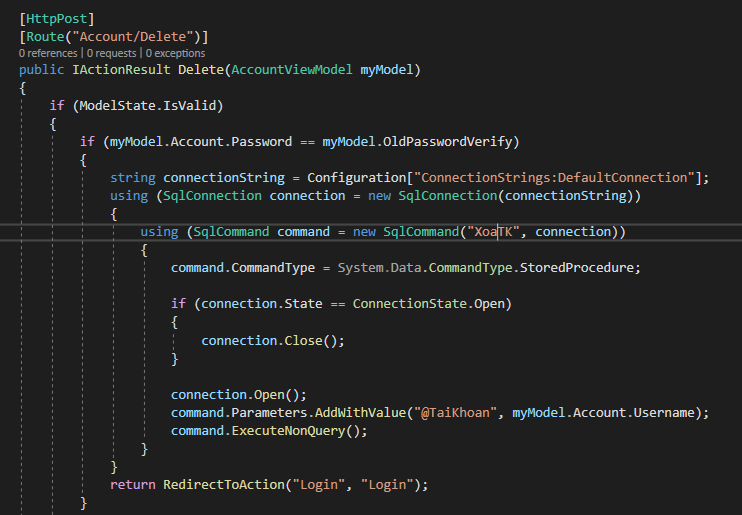


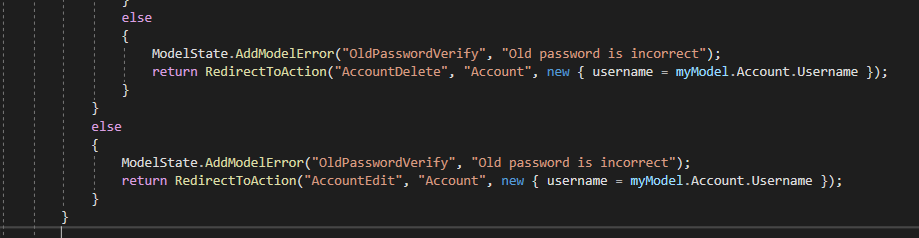
Chức năng xóa tài khoản:

+ Code SQL:



+ Gọi thực thi thủ tục:





**Chức năng quản lí sản phẩm:**

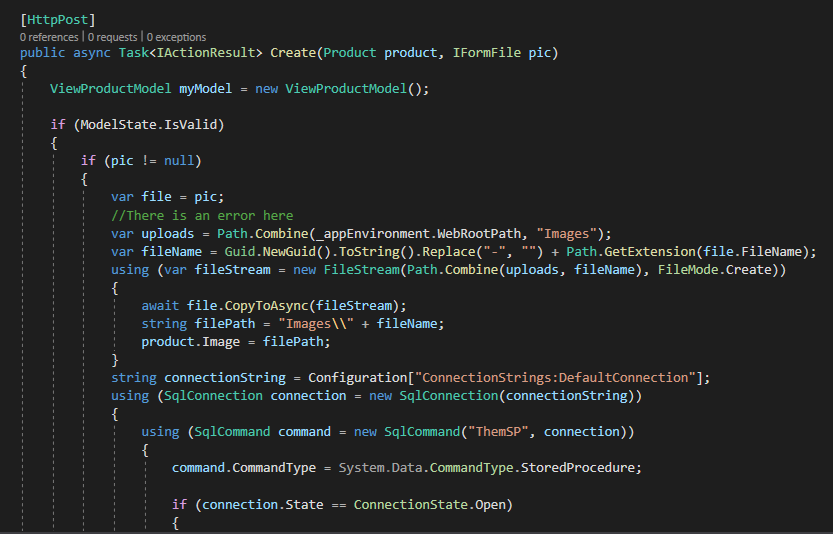
Giao diện của quản lí sản phẩm:

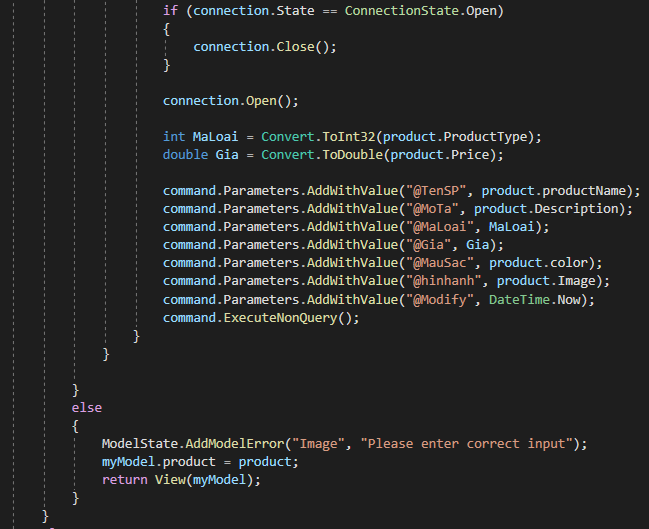
**+ Thêm sản phẩm:**

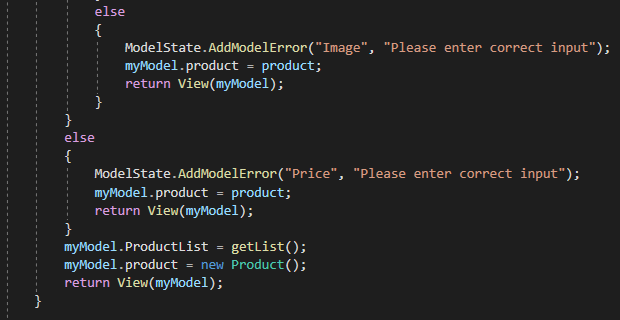
+ Code sql:



+ Gọi thực thi thủ tục:

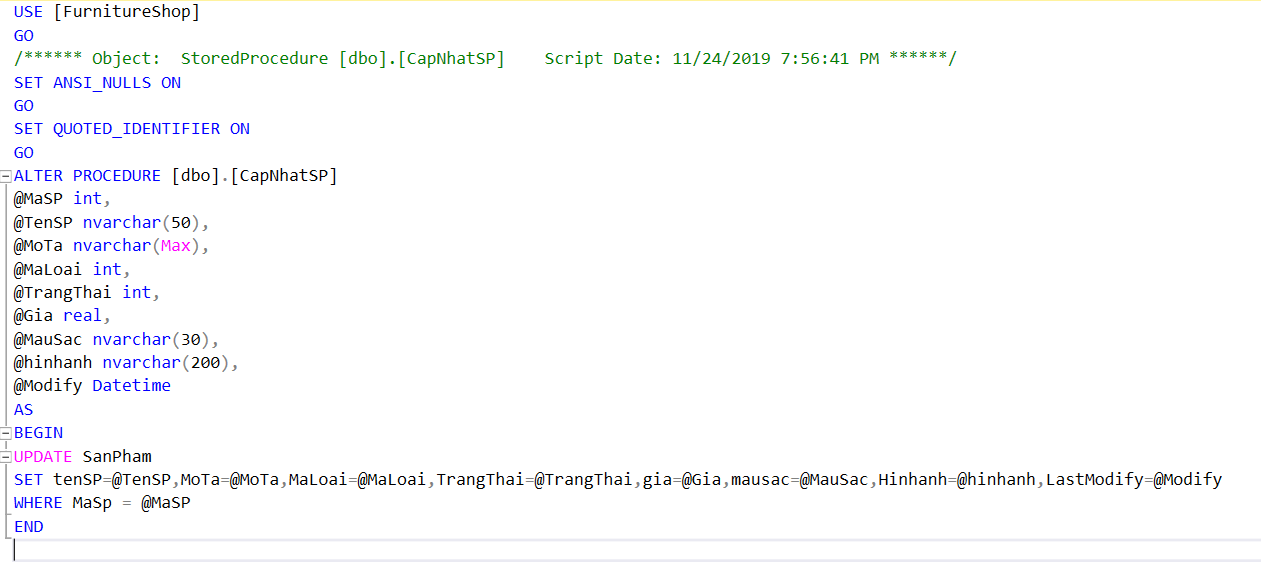




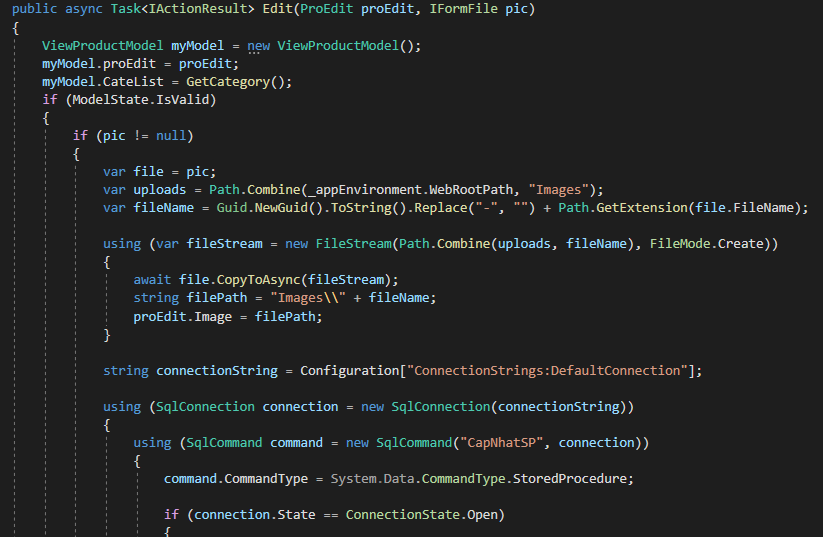


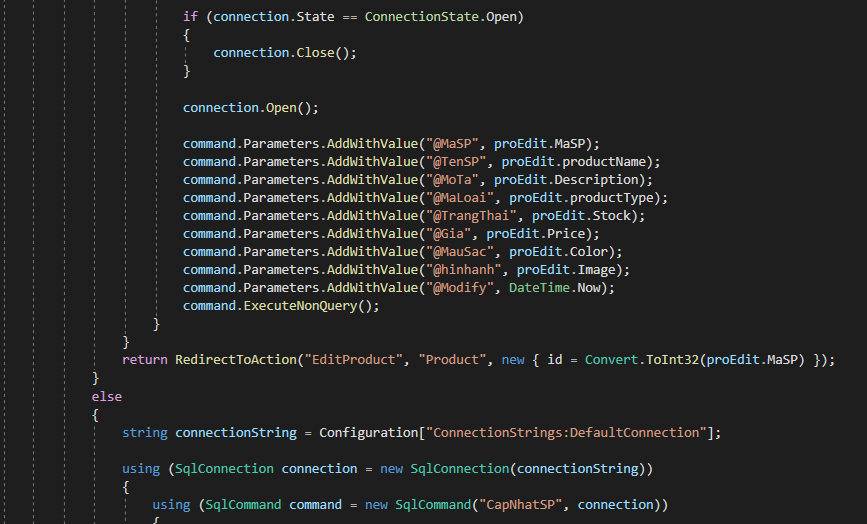
**- Cập nhật sản phẩm:**

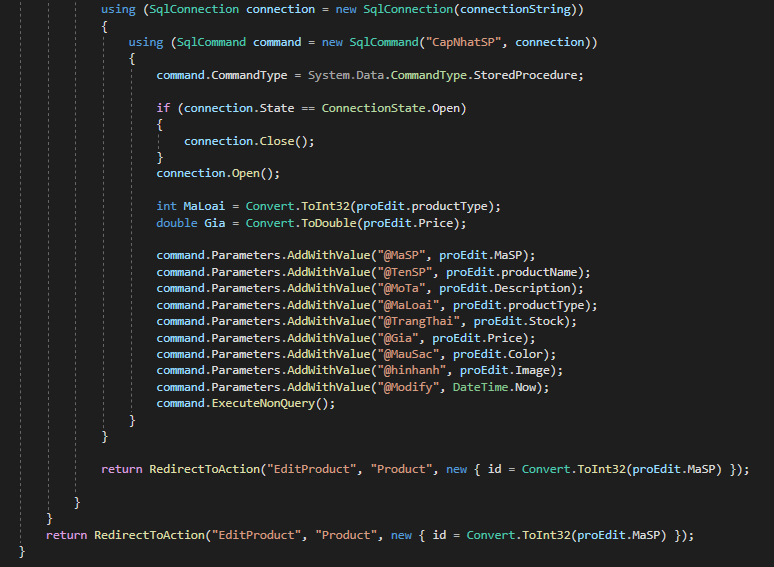
+Code sql:



+ Gọi thực thi thủ tục:

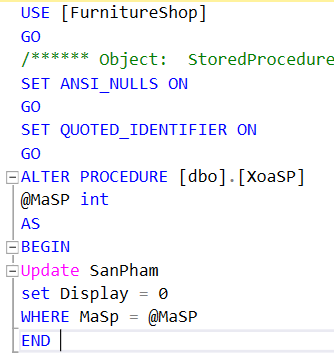




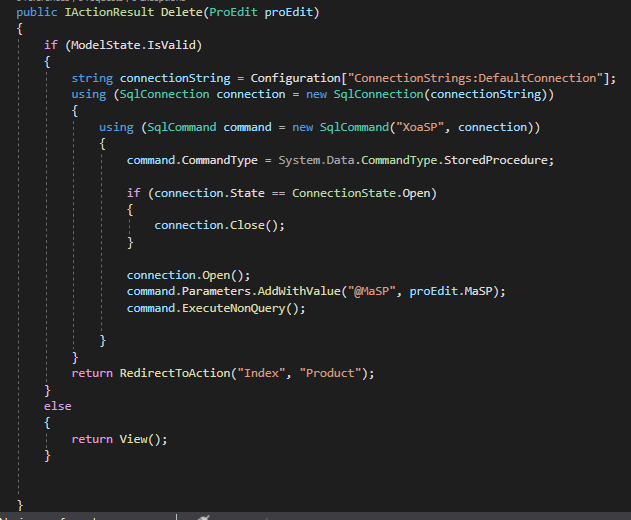


**+ Xóa sản phẩm:**

+Code sql:

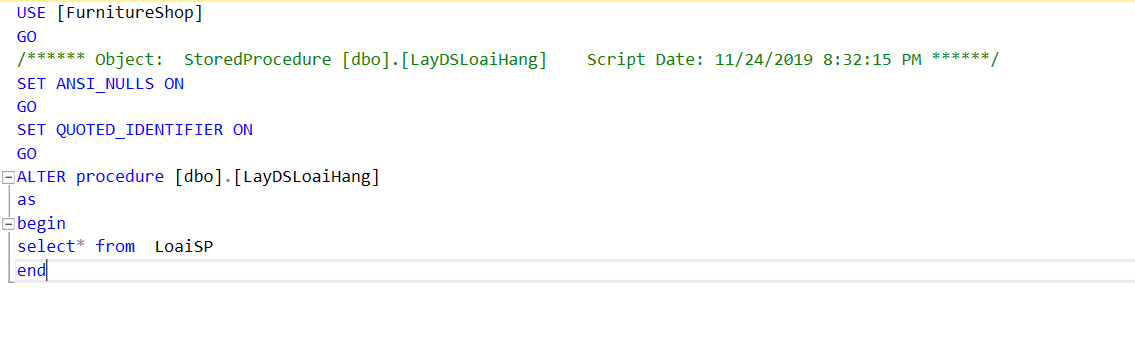


+ Gọi thực thi thủ tục:



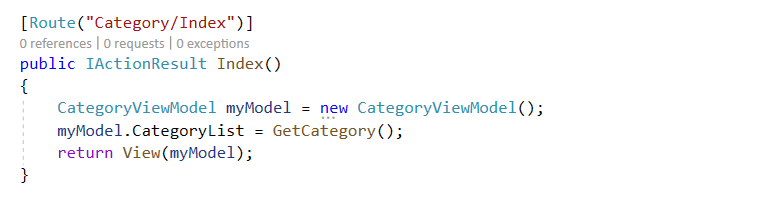
**Chức năng quản lí loại sản phẩm:**

+ Code Sql:



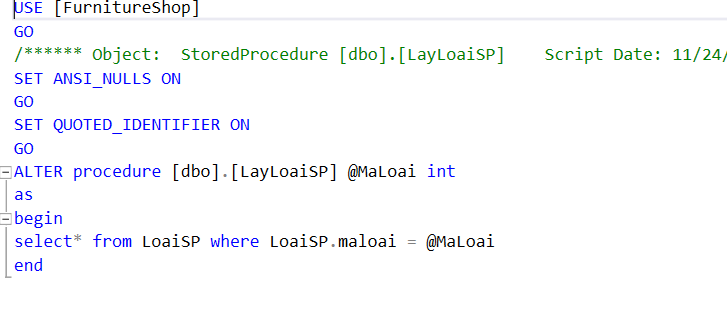
+ Gọi thực thi:





**Chức năng xem chi tiết loại sản phẩm:**

+code Sql:



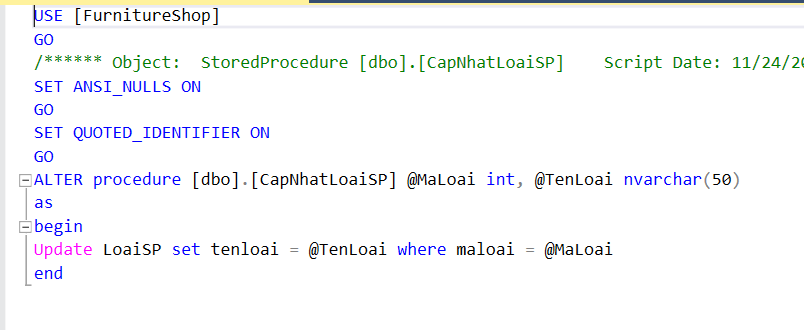
+gọi thủ tuc thực thi:



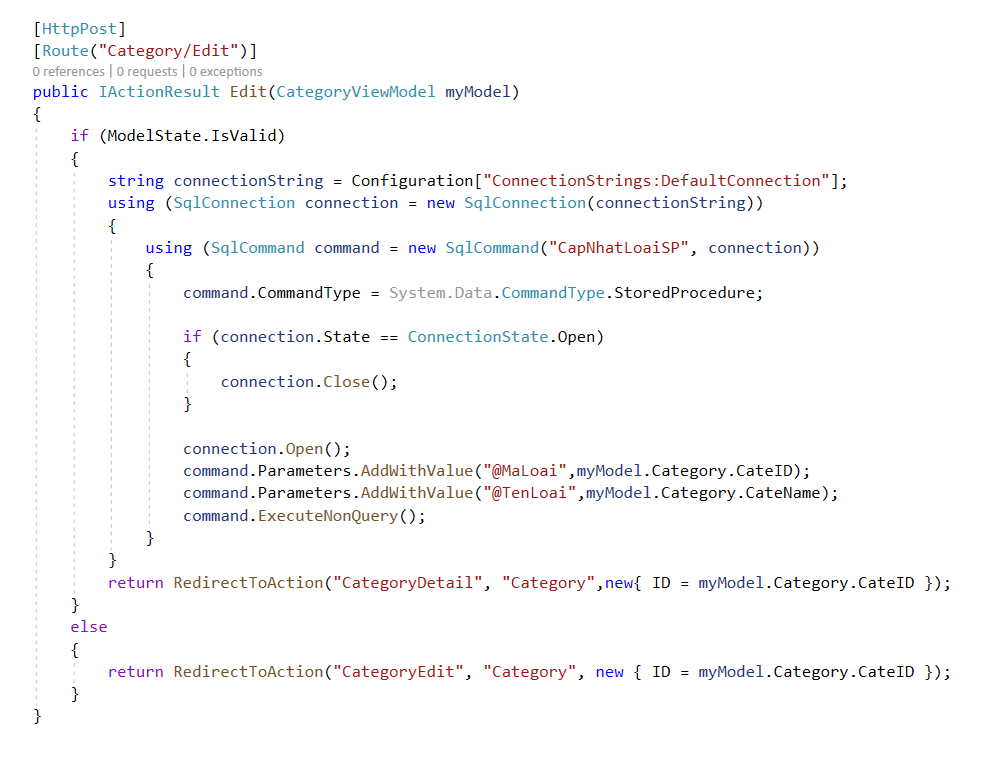


**Chức năng Chỉnh sửa loại sản phẩm**

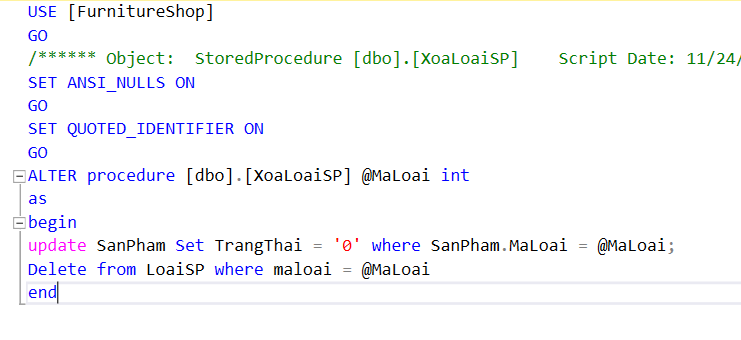
+Code SQL:



+gọi thủ tục thực thi:



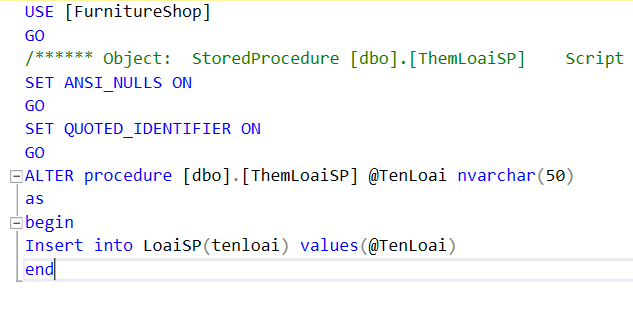
**Chức năng Xóa một loại sản phẩm:**

+ Code SQL:

+ Gọi thủ tục thực thi:

**Chức năng tạo mới loại sản phẩm:**

+ Code SQL :

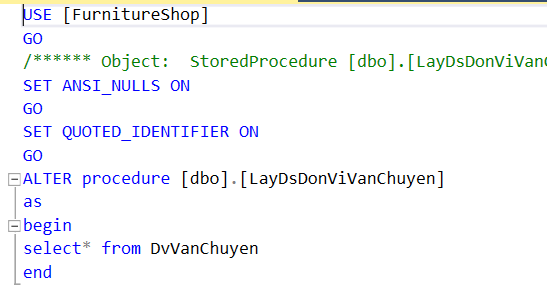


+ Gọi thực thi:



**Chức năng Xem các dịch vụ vận chuyển:**

+ Code SQL:



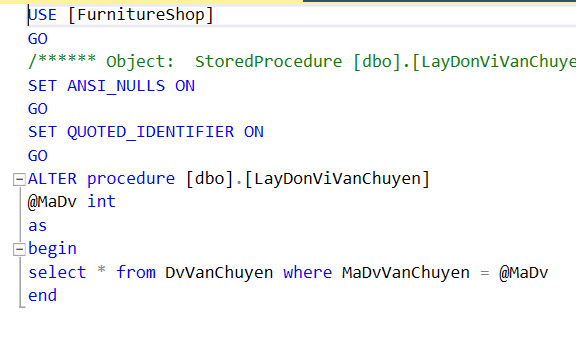
+ Gọi thực thi:





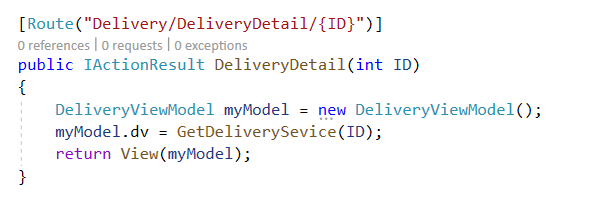
**Chức năng xem chi tiết dịch vụ vận chuyển:**

+ Code SQL:



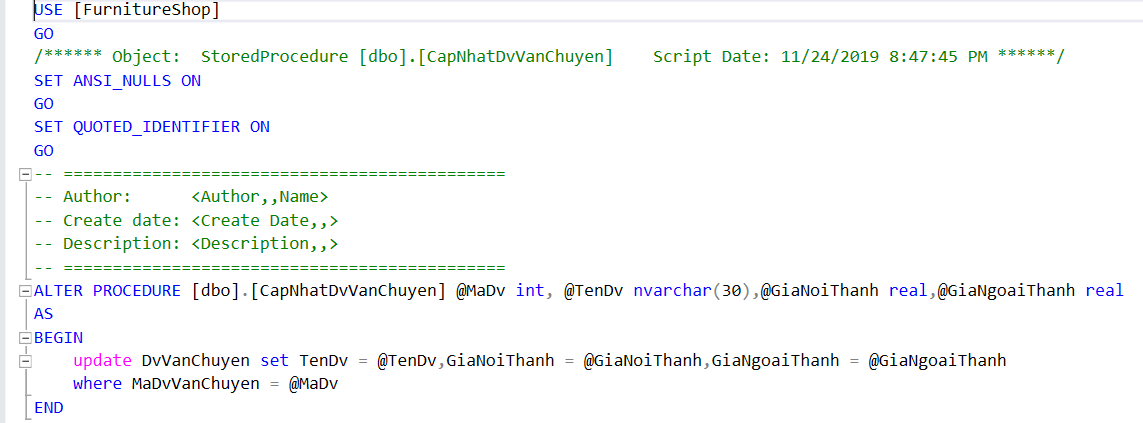
+ Gọi thực thi:





**Chức năng sửa thông tin đơn vị vận chuyển :**

+ Code SQL:

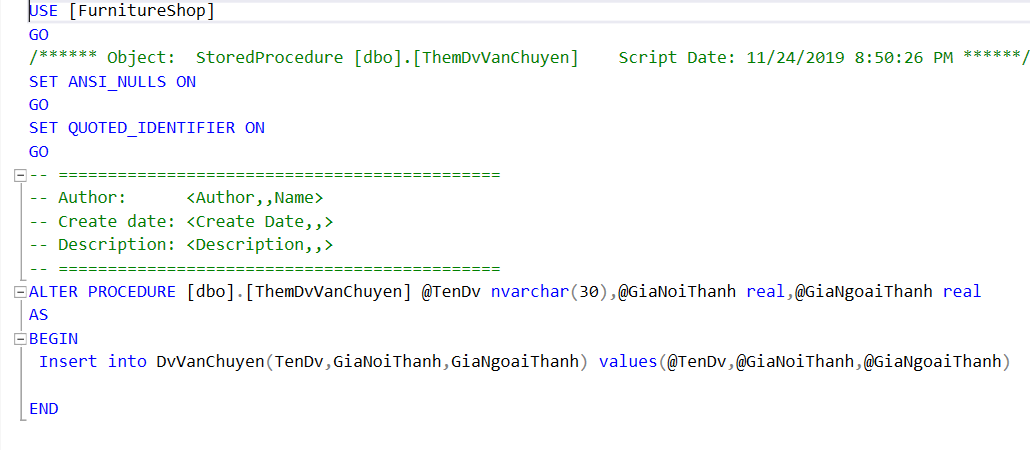


+Gọi thực thi:



**-chức năng Thêm mới một đơn vị vận chuyển:**

+ Code SQL:



+ Gọi thực thi:



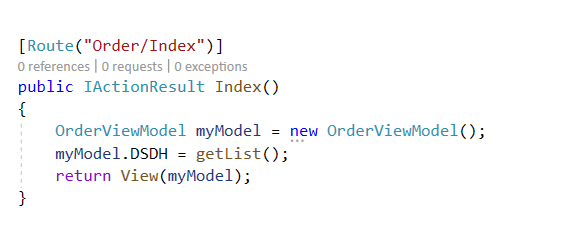
**-Chức năng xem đơn hàng**

+ Code SQL:



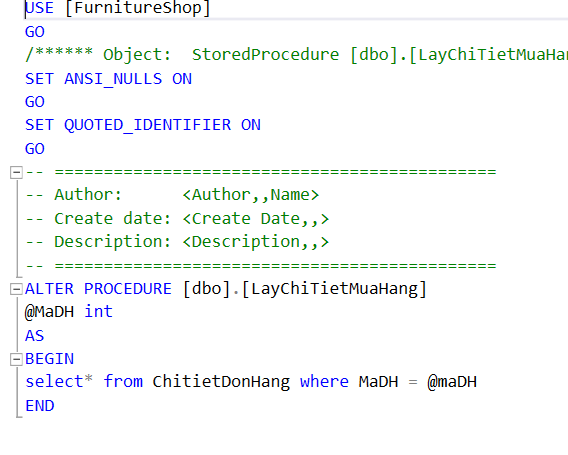
+ Gọi thực thi:

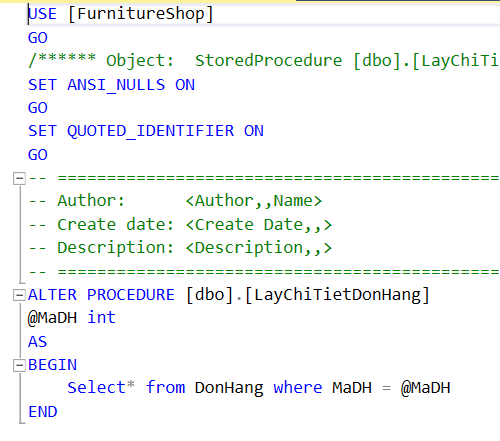




**-Xem chi tiết đơn hàng:**

+Code SQL:

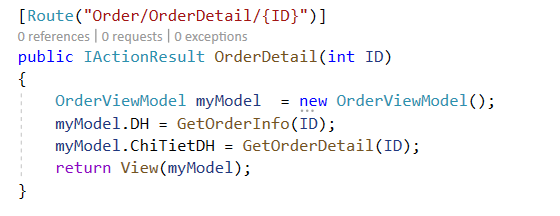




+Gọi thực thi:

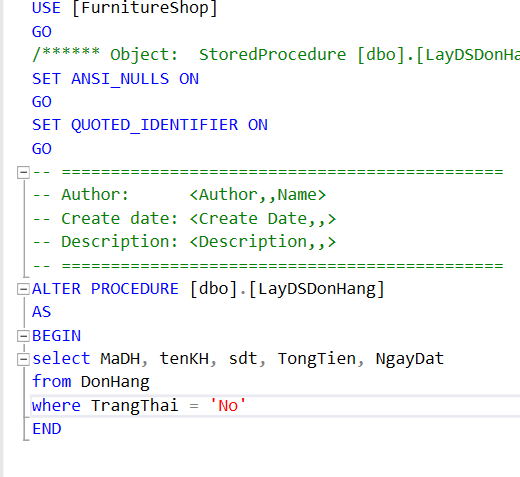






**Chức năng duyệt đơn hàng**

+ Code SQL:

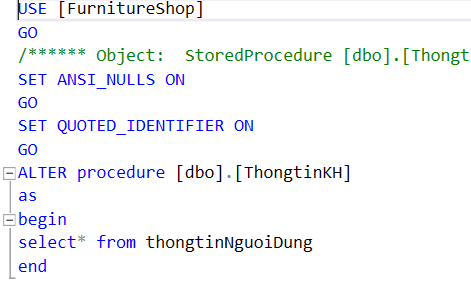


+ Gọi thực thi:

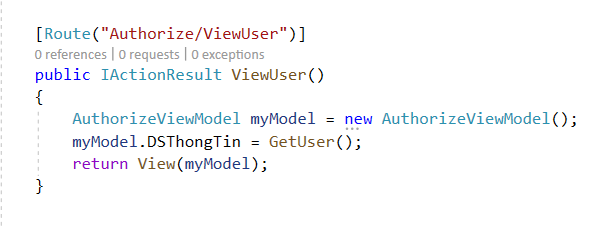


**Chức năng xem danh sách khách hàng**

+ Code SQL:



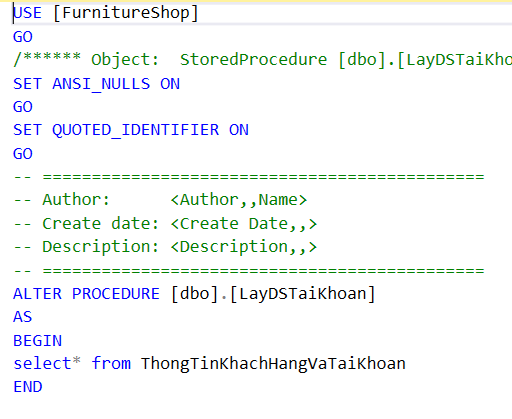
+ Gọi thực thi:



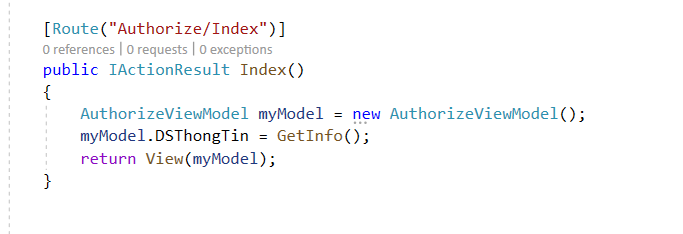


**Chức năng xem danh sách Người quản trị:**

+Code SQL:



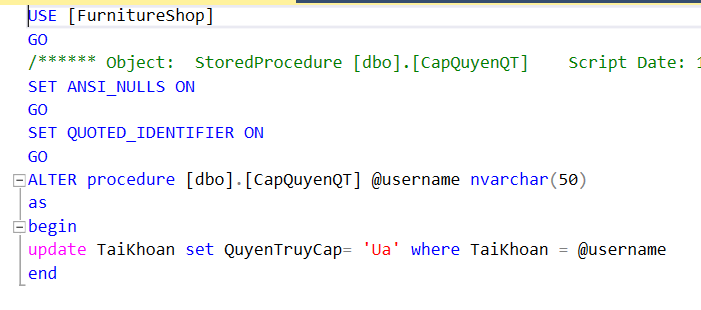
+Gọi thực thi:





**Chức năng bỏ quyền quản trị**

+ Code SQL:

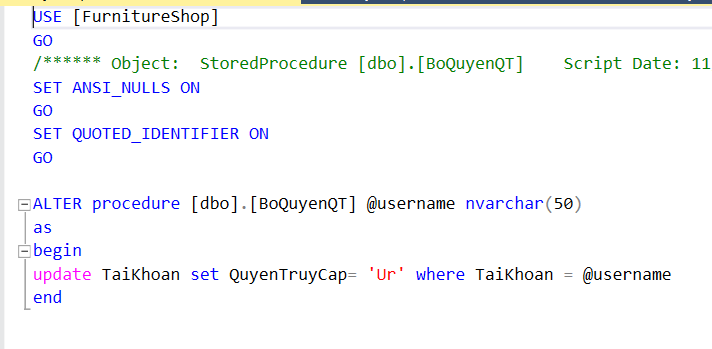


+ Gọi thực thi



**Chức năng cấp quyền quản trị:**

+ Code SQL:



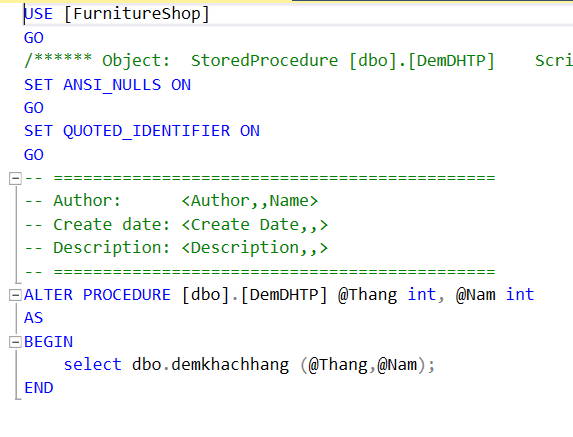
+ Gọi thực thi:

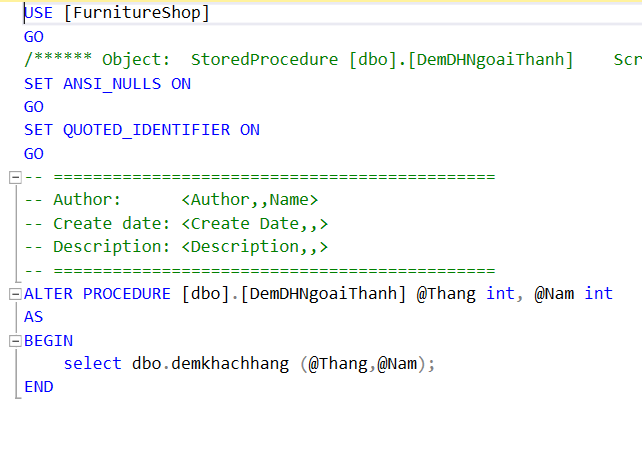


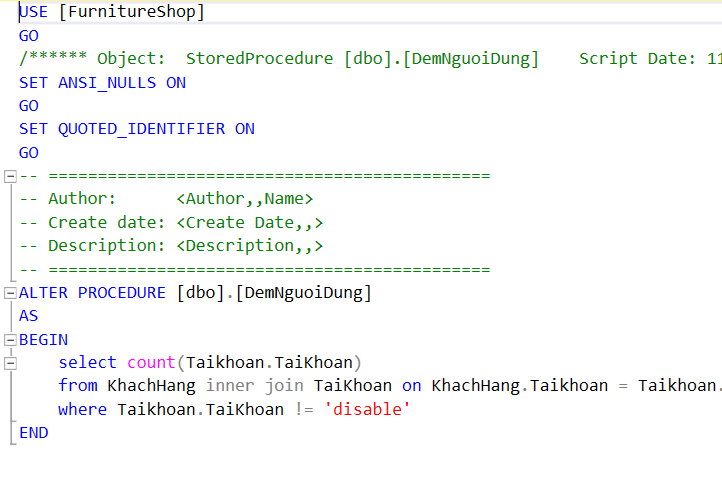
**Chức năng xem thống kê:**

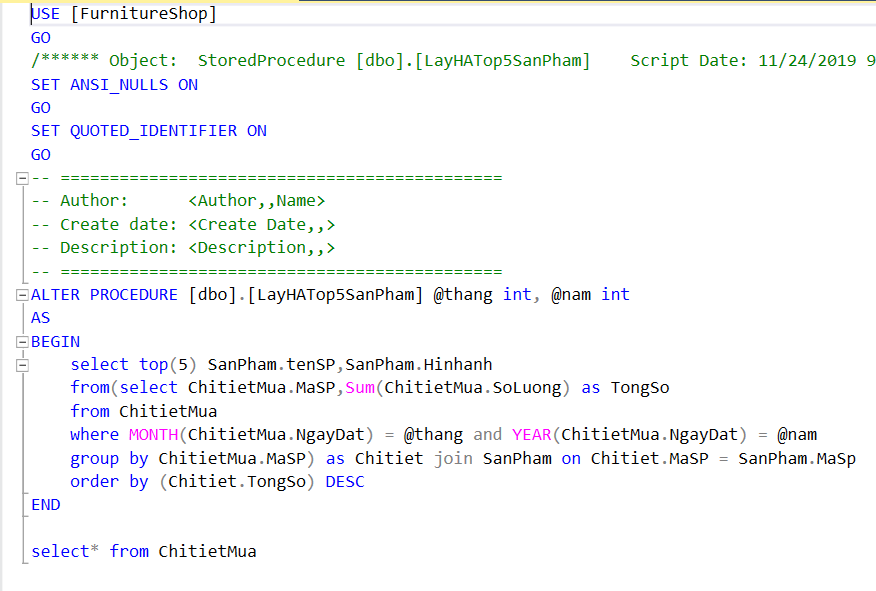
+ Code Sql:

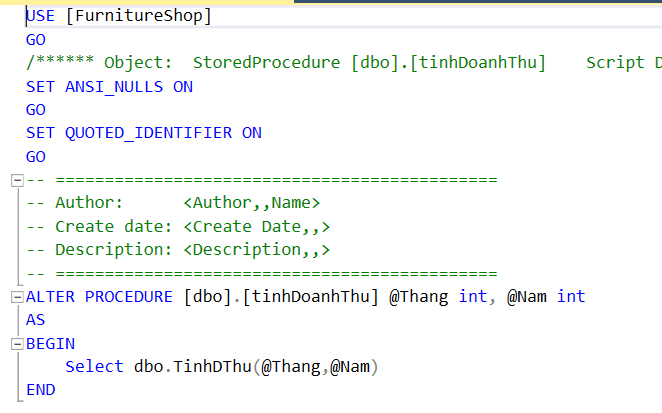


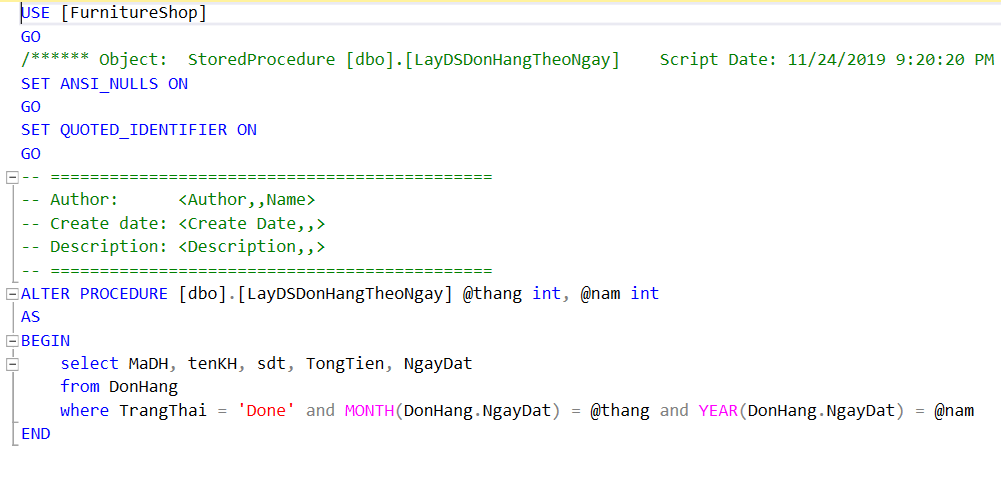










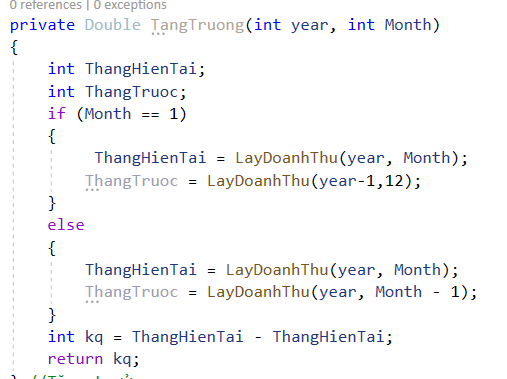


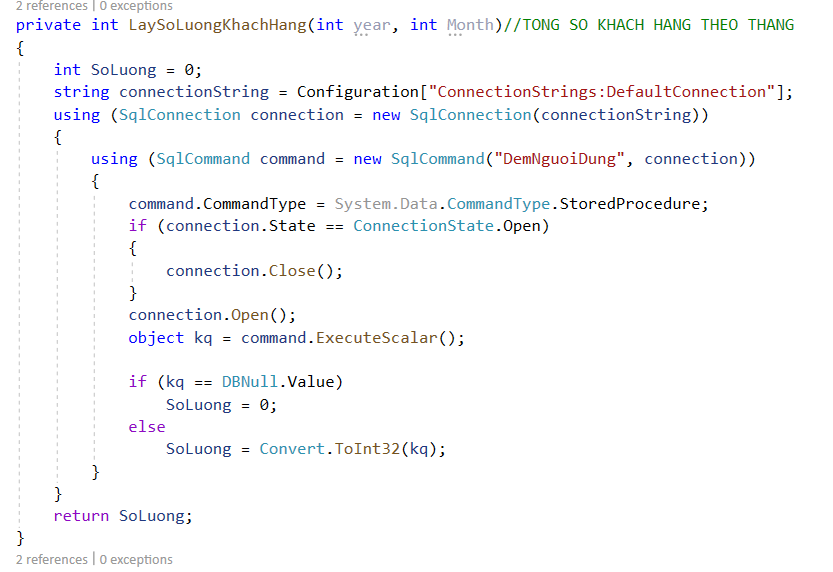
+ Gọi thực thi:





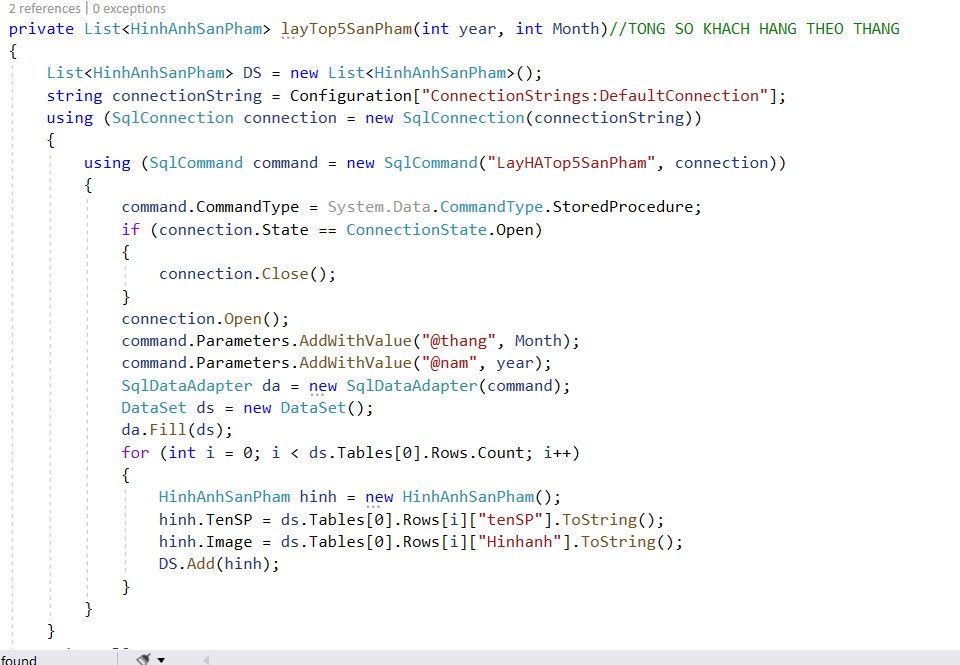
















# **4. CÁC LỖI BẢO MẬT – NGUYÊN NHÂN – GIẢI PHÁP NGĂN CHẶN CŨNG NHƯ CHỈNH SỬA CÁC LỖI XUẤT HIỆN TRONG PROJECT**

4.1. Access Control Flaw:

* Vấn đề:

Access control flaw có thể hiểu là những sai sót trong vấn đề ủy quyền. Nó có nghĩa là khi một hàm được gọi trên máy chủ, quá trình ủy quyền thích hợp đã không được thực hiện. Chúng ta dựa vào thực tế là phía máy chủ tạo ra giao diện người dùng và chúng ta nghĩ rằng khách hàng không thể truy cập các chức năng nếu không được cung cấp bởi máy chủ.

Tuy nhiên, kẻ tấn công luôn có thể yêu cầu các chức năng “ẩn” và sẽ không bị cản trở bởi việc giao diện người dùng không cho phép thực hiện các chức năng này. Hãy tưởng tượng trong giao diện người dùng chỉ có bảng điều khiển /admin và nút nếu người dùng thực sự là quản trị viên. Không có gì ngăn cản kẻ tấn công phát hiện ra những tính năng này và lạm dụng nó nếu không ủy quyền

* Giải pháp:
* Xây dựng tách biệt các chức năng trên các giao diện người dùng riêng biệt
* Kiểm tra quyền hạn người thực hiện trong phiên hoạt động hiện tại có quyền để thực hiện những thao tác trực tiếp với cơ sở dữ liệu
* Project

Do khi bắt đầu xây dựng project, Nhóm đã quy định rõ những chức năng nào mà người dùng và Admin của trang web được sử dụng. Khi đăng nhập với vai trò Admin thì giao diện sử dụng trang web sẽ khác với khi đăng nhập với vai trò là người dùng. Cho nên sau khi đăng nhập, giao diện và cách chức năng sẽ hiển thị tương ứng với quyền hạng của tài khoản đăng nhập, vậy nên hai giao diện với các chức năng sẽ tách biệt nhau hoàn toàn và không có các chức năng “ẩn”.

4.2. Client Side Filtering:

* Vấn đề:

Với nhu cầu nâng cao tốc độ truyền tải và yêu cầu giữa client và server, cùng sự ra đời của Ajax. Nhiều dữ liệu được load lên và ẩn đi ngay trên trang html với nhiều mục đích sử dụng khác nhau.

Bên cạnh đó do sự thiếu sót của người lập trình mà các dữ liệu có thể được load lên trang html một cách dư thừa hoặc load lên có chủ đích và ẩn đi với một số người dùng.

Những trường hợp trên đã vô tình tạo ra cơ hội cho kẻ tấn công, chỉ bằng cách đơn giản là kiểm tra mã nguồn của trang web thì kẻ tấn công đã có thể dễ dàng có được thông tin. Trong trường hợp các chức năng bị ẩn với một số người dùng cũng được load lên thì kẻ tấn công có thể chỉnh sửa mã nguồn trang html và sử dụng các chức năng này

* Giải pháp:

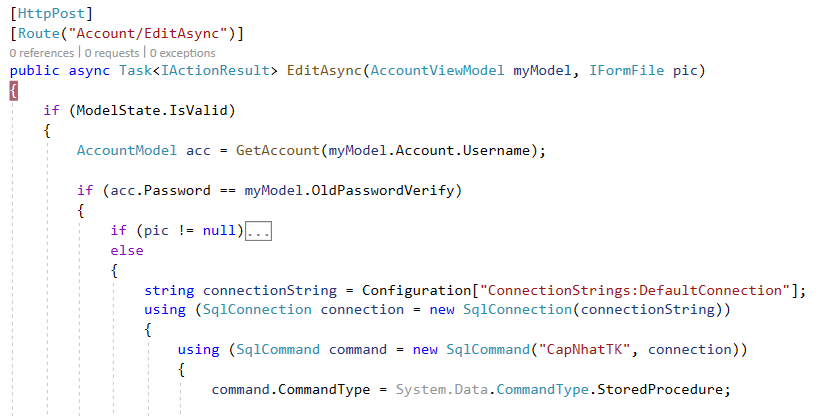
Khi load thông tin lên trang HTML cần lưu ý:

* Chỉ load đầy đủ thông tin cần thiết.
* Load những thông tin đúng với quyền hạn mà người dùng được phép
* Project:

Trên project của nhóm đã xãy ra việc load dư thừa thông tin ở các phần đổi mật khẩu tài khoản. Mật khẩu của tài khoản sau khi đăng nhập được lưu vào 1 thẻ input kèm thuộc tính hidden để ẩn đi.

Mục đích đặt mật khẩu người dùng trên trang HTML là để kiểm tra việc nhập mật khẩu mới có trùng với mật khẩu cũ và xác nhận lại mật khẩu trước khi đổi mật khẩu.



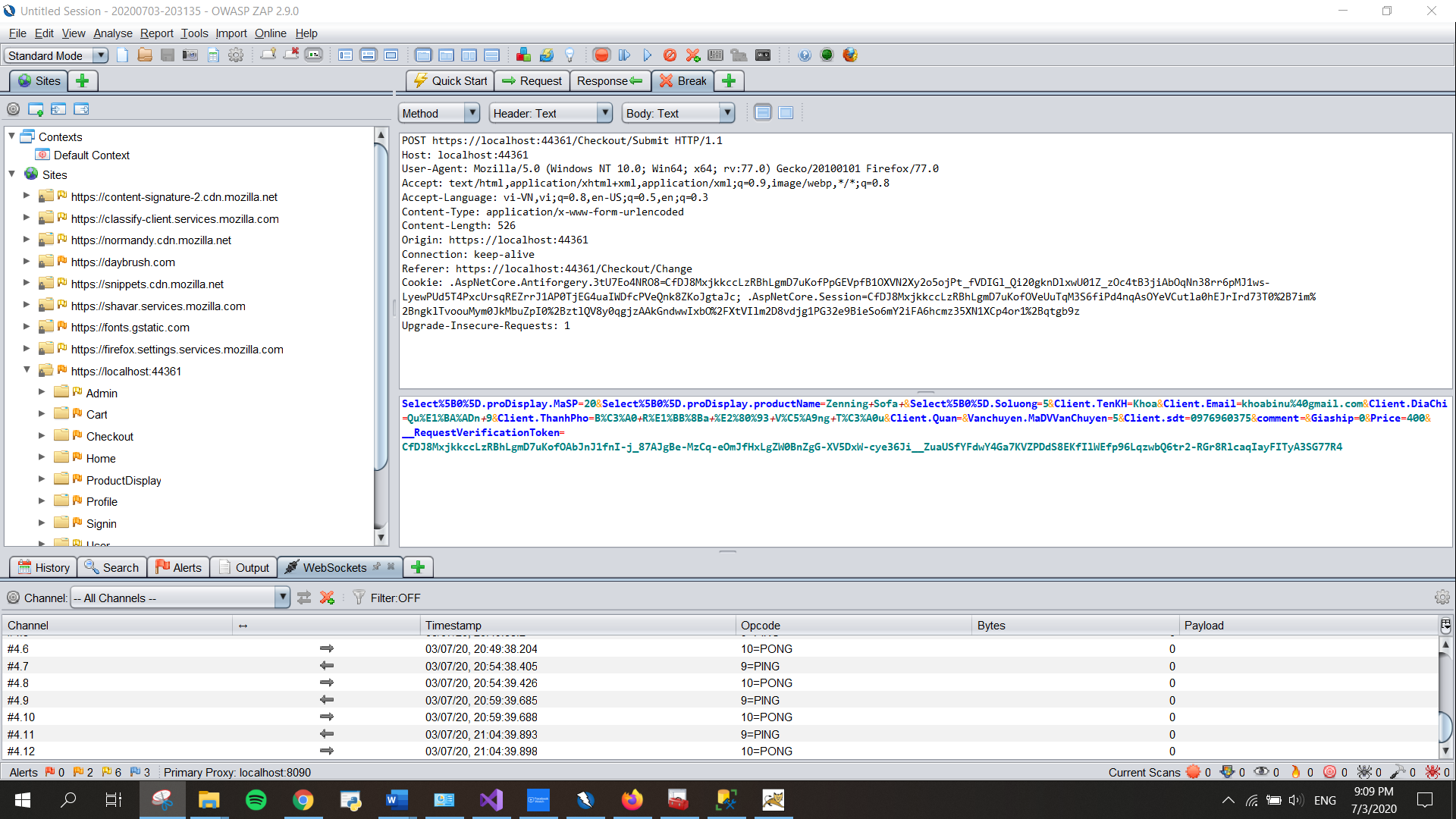
* Khắc phục: Chỉnh sửa lại trang HTML, xóa đi những dữ liệu dư thừa, chuyển việc xử lý so sánh mật khẩu cũ và mới sang phía server

4.3. Json Injection:

* Vấn đề:

JSON Injection là kĩ thuật khai thác dựa trên việc can thiệp trả lời từ máy chủ và tiêm nhiễm vào hệ thống một giá trị nào đó tùy ý.

* Project:
* Trong project của chúng em, qua quá trình tìm hiểu thì em phát hiện rằng hệ thống của mình có thể tấn công dạng Json Injection từ phía client.
* Khi tiến hàng thanh toán đơn hàng. Form checkout sẽ được gửi về hệ thống dưới dạng json chứa các thông tin về đơn hàng bao gồm cả tổng giá tiền của đơn hàng.
* Khi tiêm nhiễm vào file json gửi về, kẻ tấn công có thể thay đổi tổng giá tiền của đơn hàng.



Nguyên nhân: Hệ thống nhận giá trị được tính toán từ javascript và gửi thẳng xuống mà không tính toán lại dẫn đến việc kẻ tấn công có thể thay đổi tổng giá thành sản phẩm mà không phải gặp vấn đề gì.

* Khắc phục: Thay vì tính toán giá trị đơn hàng ở phía người dùng. Bọn em chỉ hiển thị giá trị đó lên nhưng không nhận về. Mà chỉ nhận về các giá trị từ giỏ hàng và thông tin hãng vận chuyển. Sau đó tính toán lại và nhận vào cơ sở dữ liệu.

4.3. Authentication Flaws:

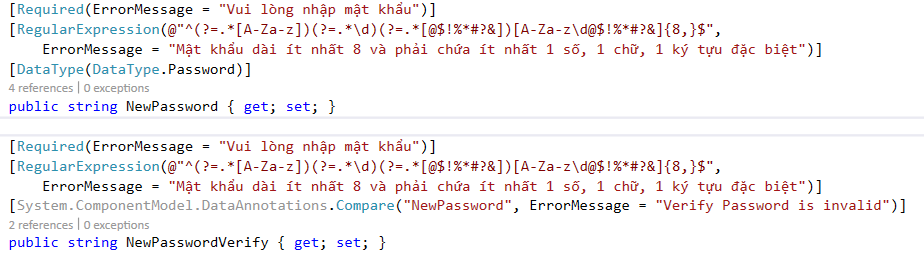
* Vấn đề:

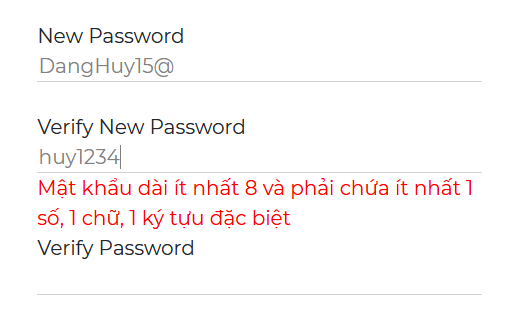
Authentication là một vấn đề vô cùng quan trọng trong lập trình web. Một ví dụ cụ thể cho việc xác thực chính là việc đăng nhập, xác thực ở chức năng đăng nhập chỉ đơn giản là việc nhập tên tài khoản và mật khẩu. Các trang web hiện nay đều khuyến khích hoặc bắt buộc người dùng luôn luôn đặt mật khẩu phức tạp cho tài khoản để đảm bảo mật khẩu khó có thể đoán được, bên cạnh đó người lập trình còn sử dụng những thuật toán mã hóa để mã hóa và gải mã mật khẩu khi lưu trữ và truy vấn mật khẩu từ CSDL.

* Giải pháp:
* Mật khẩu phải được quy định rõ độ dài (tối thiểu, tối đa), các ký tự bắt buộc (chữ hoa, chữ thường, các ký tự đặc biệt, …)
* Có thể sử dụng các thuật toán mã hóa mật khẩu
* Project:

Mật khẩu trong chức năng tạo tài khoản trên project của nhóm được đặt một cách tùy ý người dùng, không có yêu cầu nào là bắt buộc.

* Giải pháp: Yêu cầu mật khẩu phải được đặt sao cho phù hợp với một số quy định.

Mã nguồn xử lý:

Kết quả:

4.4. SQL Injection:

* Vấn đề:

**SQL Injection** là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp.

SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác, delete, insert, update, ... trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy.

**SQL Injection** thường được biết đến như là một vật trung gian tấn công trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase...

Các cuộc tấn công SQL Injection được thực hiện bằng cách gửi lệnh SQL độc hại đến các máy chủ cơ sở dữ liệu thông qua các yêu cầu của người dùng mà website cho phép. Bất kỳ kênh input nào cũng có thể được sử dụng để gửi các lệnh độc hại, bao gồm các thẻ<input>, chuỗi truy vấn (query strings), cookie và tệp tin.

* Giải pháp:
* Kiểm tra các kênh đầu vào
* Sử dụng các câu lệnh tham số.
* Sử dụng Strored procedure để hạn chế tấn công SQL injection
* Project:

Trên project tại những phương thức truy vấn cơ sở dữ liệu có sử dụng các Sqlcommand đều có khả năng xảy ra SQL injection. Project của nhóm có thể bị tấn công Sql Injection ở dạng tấn công vượt qua kiểm tra lúc đăng nhập

Với dạng tấn công này, kẻ tấn công có thể dễ dàng vượt qua các trang đăng nhập nhờ vào lỗi khi dùng các câu lệnh SQL thao tác trên cơ sở dữ liệu của ứng dụng web. Thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu.

Khi nhập “ = 0’ or 1=1------- “ vào ô username trên trang đăng nhập với mật khẩu tùy ý thì, kẻ tấn công đã có thể vào được trang admin.

Giải thích: Trong phương thức đăng nhập trên Project chúng em có sử dụng câu lệnh SQL

select TaiKhoan,MatKhau,QuyenTruyCap from TaiKhoan where TaiKhoan = '" + model.UserName + "' and MatKhau = '" + model.PassWord + "'"

với model.Username và model.PassWord do người dùng nhập vào khi người dùng nhập vào username là “= 0’ or 1=1-------" và password là chuỗi bất kỳ ví dụ “abc” câu lệnh trở thành

select TaiKhoan,MatKhau,QuyenTruyCap from TaiKhoan where TaiKhoan = '0' or 1=1 -------- and MatKhau = 'abc'

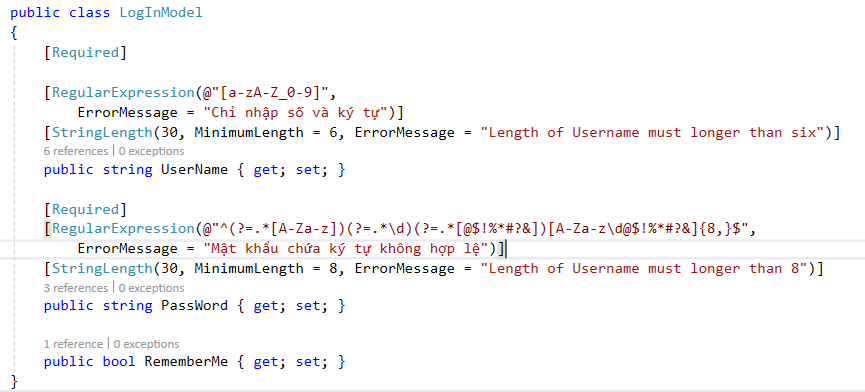
Câu lệnh sql như trên với điêù kiện sau Where TaiKhoan = '0' or 1=1 luôn đúng.

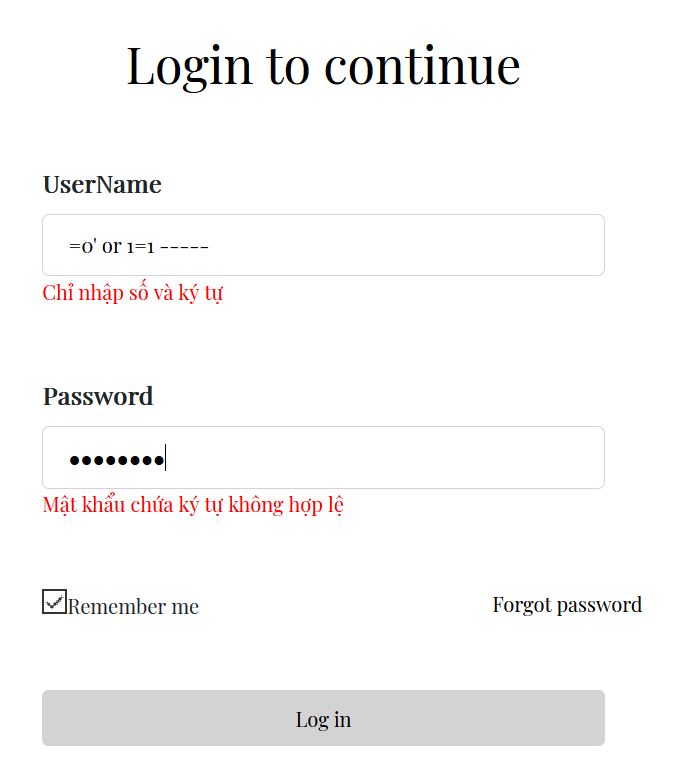
Các dấu -------- có chức năng là không thực hiện những lệnh phía sau nó cho nên với mật khẩu nhập bất kỳ nhập vào thì điều kiện vẫn đúng. Lúc này chúng ta đã có thể lấy ra hàng đầu tiên trong bảng TaiKhoan tương ứng với một tài khoản nào đó với quyền là Admin. Và kẻ tấn công đã có thể vào được trang quản trị.

* Với các lỗ hổng Sql injection này nhóm đã chuyển hết các câu lệnh command với các tham số truyền trực tiếp như trường hợp đăng nhập được nêu ở trên, sang Stored Procedure hỗ trợ bởi SQL server. Stored Procedure được sử dụng trong lập trình web với mục đích nhằm giảm sự phức tạp trong ứng dụng và tránh sự tấn công trong kỹ thuật SQL Injection.

Nhưng Stored Procedure cũng chỉ có thể hạn chế một phần tấn công SQL injection vào hệ thống, chứ không tránh được hoàn toàn.

Để đảm bảo hơn các input đầu vào đều phải được kiểm tra, giới hạn các các ký tự, kiểu dữ liệu được cho phép.

Kiểm tra đầu vào để tránh Sql injection:

Kết quả:

Sử dụng stored procedure thay vì Sqlcommand:

4.6. Cross site Scripting

* Vấn đề:

Cross-Site Scripting được gọi tắt là XSS là một kỹ thuật tấn công bằng cách chèn vào các website động một đoạn mã HTML hay những đoạn mã script (chủ yếu là các mã Client-Site Script như JavaScript, JScript, DHTML) và có thể gây nguy hiểm cho người sử dụng (mất cookies, mật khẩu, session). XSS gồm có 3 loại:

* Stored XSS: Là lỗi XSS mà các đoạn mã được lưu trữ trên Server (các hacker có thể submit lên server bằng cách nào đó) như là trong database dưới dạng comment trong blog, messages trong form. Vì các đoạn mã này được lưu trữ trên server cho nên khi một người dùng nào đó truy cập vào trang web có chứa đoạn mã này thì sẽ gây nguy hại cho người dùng(có thể chuyển người dùng đến 1 trang web giả mạo, có thể yêu cầu người dùng cung cấp thông tin …).
* Reflected XSS: Theo một thống kê thì có tới 75% tấn công XSS dựa trên kỹ thuật reflected XSS này. Gọi là reflected(phản xạ) bởi vì theo kịch bản khai thác lỗi này thì hacker bằng cách nào đó phải gửi cho nạn nhân một URL mà trong đó có chưa các mã script độc.  
  Một ví dụ trong thực tiễn như sau: Khi người dùng truy cập vào website và được gán session => bằng cách nào đó hacker gửi cho user một URL chứa mã script mà trong đoạn mã đó đọc thông tin document.cookie và request đến site mà hacker mong muốn => nếu người dùng truy cập vào URL chứa mã độc này thì nó đoạn mã trên sẽ đọc thông tin trong document.cookie và gửi đến trang web của hacker => Từ đó hacker có thể đọc thông tin cần thiết trong document.cookie này.
* DOM(Document Object Model) based: Là kỹ thuật khai thác lỗ hổng XSS dựa trên cơ chế thay đổi cấu trúc DOM của tài liệu, cụ thể ở là HTML.  
  Xét ví dụ sau: Một website có URL truy cập đến trang login và bao gồm 1 param mess trên url.=> Hacker có thể kết hợp với Reflected XSS để thay URL trên với param bằng cách add thêm các element => website sẽ thực thi các element này và hiển thị trên giao diện(có thể là các tag HTML cung cấp thông tin thêm, hoặc có thể là post data đến 1 website)=> Hacker có thể đánh cắp thông tin của người dùng.
* Giải pháp:

Ngăn chặn XSS trong .NET:

Trong khi phát triển web bằng XSS thì mặc định ko cho người dùng nhập vào mã script hoặc trên URL (nếu cố tình nhập vào khi post lên server sẽ ra lỗi server error).

Trường hợp nếu muốn cho người sử dụng nhập thì thêm validateRequest = ”false” vào .aspx(chỉ cho trang chứa nó) hoặc add vào web.config(cho toàn bộ ứng dụng).

* Validate All Input: Validate tất cả các input của ứng dụng Web
* Encoding HTML: Encode trước khi save DB hoặc show data
* AntiXSS Library: đã được đưa vào trong .NET 4.5, đối với các phiên bản .NET cũ hơn thì add thêm thư viện.
* ASP.NET Request Validation: sử dụng validateRequest = ”false” để ngăn chặng input script nếu ứng dụng cho phép.

Trong Asp.net core 2.1, HTML được mã hóa bằng Razor

Công cụ Razor được sử dụng trong việc tự động mã hóa tất cả đầu ra có nguồn gốc từ các biến, trừ khi chúng ta ngăn chặn cdoong cụ này. Nó sử dụng các quy tắc mã hóa thuộc tính HTML bất cứ khi nào bạn sử dụng lệnh @.

Ví dụ:

HTML Encode mã hóa các ký tự đặc biệt trong chuỗi được sử dụng trong tài liệu HTML để tránh nhầm lẫn với các yếu tố HTML như

"<hello>world</hello>"

Trở thành:

"&lt;hello&gt;world&lt;/hello&gt;"

URL Encoding hiện một tương tự điều cho các giá trị chuỗi trong URL như thay đổi

"hello+world = hello world"

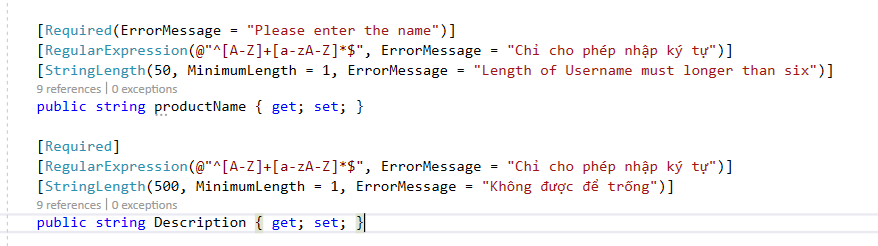
Trở thành

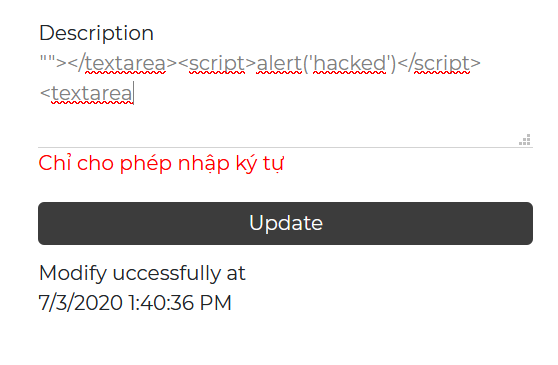
"hello%2Bworld+%3D+hello+world"

Project:

Project chúng em được xây dựng bằng ASP.net core 2.1, nên các trang cshtml đã được từ động encode, mặc dù các thẻ input đầu vào không được xác thực kiểm tra nhưng các đoạn script dù có lưu vào được CSDL cũng không thể gây ra lỗi trên trang web.

Nhưng để làm rõ vấn đề XSS chúng em đã sử dụng lệnh @html.raw() ở một số trang html trong project. Lệnh này có tác dụng là cho phép html thuần tránh bị encode bởi razor. Khi chúng em đưa vào các input là các lệnh độc hại, mặc nhiên trang web đã bị tấn công.

* Khắc phục: Quy định chỉ cho phép ký tự hoặc chữ số tùy thuộc vào kiểu dữ liệu của giá trị đầu. Sử dụng [RegularExpression] của C#

Khi nhập cái ký tự không hợp lệ hệ thống sẽ lập tức thông báo và ngăn cản Submit để gửi request.

4.5. CSRF:

* Vấn đề:

CSRF ( Cross Site Request Forgery**)** là kĩ thuật tấn công bằng cách sử dụng quyền chứng thực của người sử dụng đối với 1 website khác. Các ứng dụng web hoạt động theo cơ chế nhận các câu lệnh HTTPtừ người sử dụng, sau đó thực thi các câu lệnh này.

Kẻ tấn công sử dụng phương pháp CSRF để lừa trình duyệt của người dùng gửi đi các câu lệnh http đến các ứng dụng web. Trong trường hợp phiên làm việc của người dùng chưa hết hiệu lực thì các câu lệnh trên sẽ dc thực hiện với quyền chứng thực của người sử dụng.

* Giải pháp:

Dựa trên nguyên tắc của CSRF "lừa trình duyệt của người dùng (hoặc người dùng) gửi các câu lệnh HTTP", các kĩ thuật phòng tránh sẽ tập trung vào việc tìm cách phân biệt và hạn chế các câu lệnh giả mạo.

* Phía user

Để phòng tránh trở thành nạn nhân của các cuộc tấn công CSRF, người dùng internet nên thực hiện một số lưu ý sau:

* Nên thoát khỏi các website quan trọng:
* Không nên click vào các đường dẫn mà bạn nhận được qua email, qua facebook
* Không lưu các thông tin về mật khẩu tại trình duyệt của mình (không nên chọn các phương thức "đăng nhập lần sau", "lưu mật khẩu" …
* Trong quá trình thực hiện giao dịch hay vào các website quan trọng không nên vào các website khác, có thể chứa các mã khai thác của kẻ tấn công.
* Phía Server:
* Tương tự như XSS, để phòng tránh CSRF chúng ta cần kiểm tra đầu vào
* Sử dụng GET và POST đúng cách.
* Dùng GET nếu thao tác là truy vấn dữ liệu.
* Dùng POST nếu các thao tác tạo ra sự thay đổi hệ thống. Nên cân nhắc các trường hợp sử dụng

#### Sử dụng token: Tạo ra một token tương ứng với mỗi form, token này sẽ là duy nhất đối với mỗi form và thường thì hàm tạo ra token này sẽ nhận đối số là"SESSION" hoặc được lưu thông tin trong SESSION. Khi nhận lệnh HTTP POST về, hệ thống sẽ thực hiên so khớp giá trị token này để quyết định có thực hiện hay không

* Project

Trên project nhóm chúng em có khả năng bị tiêm nhiễm CSRF từ trang người dùng khi người dùng có thể chỉnh sửa các thông tin cá nhân của họ mà các đầu vào không được kiểm soát, người sử dụng trang web có thể dễ dàng lưu xuống các lệnh độc hại khi người quản trị tìm kiếm và xem danh sách người dùng thì các lệnh độc hại đó sẽ được thực thi gây ảnh hướng đến hệ thống.

* Mặc dù Razor đã giải quyết các vấn đề mã hóa HTML trên projet, nhưng chúng em sẽ kiểm tra các đầu vào. Với các phương thức quan trọng thao tác trực tiếp với CSDL trên project, đặc biệt là delete Nhóm sẽ chỉnh sửa cho hợp lý hơn(cần truyền về mật khẩu hiện tại để so sánh xác thực,….).



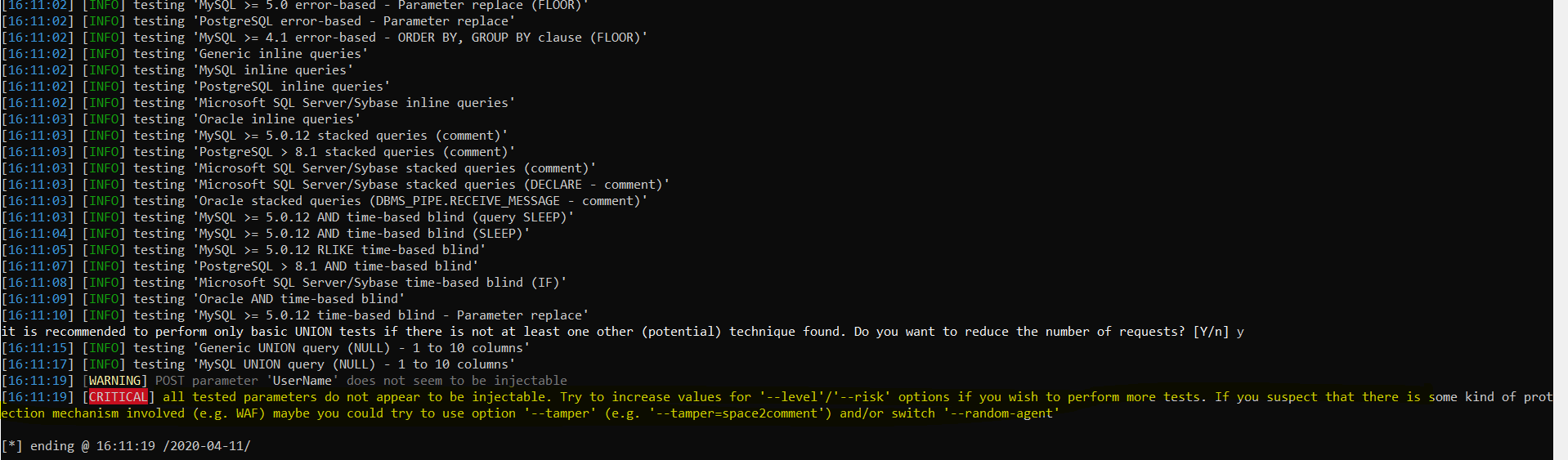
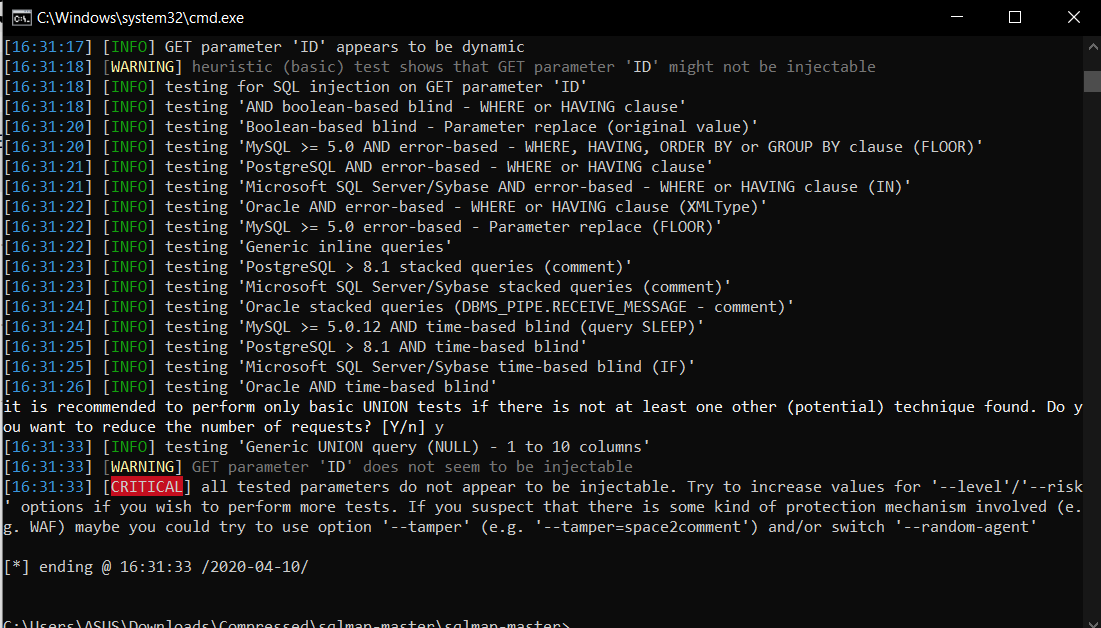
Route mapping đến Action Delete: [Route("Profile/Delete")]

Với kiểu route mapping như thế này khi người dùng đã đăng nhập và chưa đăng xuất. Nếu người dùng truy cập vào 1 trang chứa các mã độc tạo ra request để server theo URL: <https://localhost:44361/Profile/Delete>. Thì hệ thống sẽ lấy ra username của người dùng đang đăng nhập được lưu trong session và thực hiện xóa tài khoản.

Lỗ hổng được sinh ra bởi thiếu sót trong quá trình xây dựng project, để khắc phục trường hợp chức năng delete này. Nhóm sẽ chỉnh sửa mã nguồn yêu cầu truyền giá trị password hoặc username để so sánh các giá trị đó có khớp với username, password đang lưu trong session hay không.



4.7. Sử dụng công cụ sqlmap chưa tìm kiếm lỗ hổng sql injection trên đồ án

Nhóm đã thực hiện cài đặt Sqlmap và quét tìm lỗi Sql injection trên đồ án của nhóm. Hiện tại sqlmap chưa t được các lỗ hổng Sql Injection trên đồ án 

4.8. Sử dụng ZAP quét các lỗ hổng của Project

# **CHƯƠNG 4: KẾT LUẬN**

1. **Về Chương trình**

## **Kết quả đạt được:**

* Giới thiệu sản phẩm đến khách hàng. Đối với khách hàng: cho phép thực hiện các chức năng xem hàng hóa.
* Sau khi mua sản phẩm, đặt mua sản phẩm.
* Khách hàng có thể thay đổi mật khẩu và mật khẩu thanh toán trong thông tin cá nhân. Khách hàng có thể xem thông tin đơn hàng của mình.
* Hoàn thành các chức năng cơ bản của một trang web bán hàng.

## **Các hạn chế:**

Về dữ liệu:

* Dữ liệu chưa được đầy đủ.
* Dữ liệu trên trang web hiện tại không đúng với thực tế, sử dụng nguồn dữ liệu ảo.

Về giao diện:

* Giao diện chưa sinh động
* Giao diện cứng, không thể thay đổi template cho giao diện.

## **Hướng phát triển:**

Với những thiếu sót như phân tích trên, để phần mềm ngày càng hoàn thiện, phục vụ tốt cho nhu cầu của người dùng thì nên có những cập nhật sau đây:

Thiết kế trang web đẹp hơn, tiện dụng hơn, có nhiều chức năng và sinh động hơn. Xây dựng thêm một diễn đàn trên trang web để khách hàng có thể trao đổi bình luận về sản phẩm. Thiết kế thêm vị trí để cập nhật giá cả thị trường. Nâng cao khả năng bảo mật cho chương trình. Ngăn ngừa tối đa các trường hợp đột nhập dữ liệu hệ thống. Chỉnh sữa dữ liệu cho chính xác với thực tế và sẽ nâng cấp dần website để đáp ứng được những nhu cầu của khách hàng trong thời đại mới.

1. **Về bảo mật và khắc phục các lỗ hổng bảo mật:**
   1. **Kết quả:**

Dựa trên cơ sở kiến thức đã học được ở môn bảo mật web, kiến thức thực hành webGoat, cũng nhưng các bài thực hành liên quan. Nhóm đã nghiên cứu và thực hành bằng tay, sử dụng một số công cụ hỗ trợ như ZAP, Sqlmap để tìm ra những lỗ hổng bảo mật tồn tại trên project của nhóm và đề ra được các cách để khắc phục các lỗ hổng này.

Tuy nhiên với kiến thức còn non yếu và chưa có nhiều kinh nghiệm, nên quá trình tìm hiểu của nhóm còn có thể có những sai sót.