TRƯỜNG ĐẠI HỌC
SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH
HCMC University of Technology and Education

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

NGÀNH CÔNG NGHỆ THÔNG TIN

Môn học: Hệ quản trị cơ sở dữ liệu

BÁO CÁO ĐỒ ÁN CUỐI KÌ

WEB BÁN HÀNG

Nhóm sinh viên thực hiện:

Nguyễn Xuân Điền	**17110121**
Vũ Hoàng Hiệp	**17110138**
Tôn Long Hoàng Lãm	**17110166**

GVHD: cô Lê Thị Minh Châu

Tp. Hồ Chí Minh, tháng 12 – 2019

ĐIỂM SỐ

TIÊU CHÍ	NỘI DUNG	TRÌNH BÀY	TỔNG
ĐIỂM			

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Giáo viên hướng dẫn

(*ký và ghi họ tên*)

Lê Thị Minh Châu

LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, cô Lê Thị Minh Châu, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Cảm ơn thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn Khoa đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

TP.HCM, ngày 22 tháng 12 năm 2019

Nhóm sinh viên thực hiện

MỤC LỤC

Danh mục các hình *1*

Danh mục các bảng *2*

Chương 1: Đặc tả tổng quan chương trình *3*

1. Giới thiệu chung *3*

1.1. Giới thiệu Hệ quản trị cơ sở dữ liệu **3**

1.2. Sơ bộ về cách thiết kế Web bán hàng **3**

1.2.1. Yêu cầu đồ án **3**

1.2.3. Phương hướng thực hiện **3**

1.3. Một số khái niệm cần biết **4**

1.3.1. View **4**

1.3.2. Function **5**

1.3.3. Stored Procedure **5**

1.3.4. Trigger **6**

1.3.5. Angular **7**

2. Đặc tả Web bán hàng *7*

2.1. Mô tả về Web bán hàng **7**

2.2. Mục đích lựa chọn hình thức thiết kế quản trị bán hàng bằng web **7**

2.2.1. Tính năng chính **7**

2.2.2. Ứng dụng **8**

2.3. Yêu cầu kĩ thuật đối với trang web **8**

2.4. Công cụ và công nghệ sử dụng **8**

Chương 2: Kế hoạch thực hiện *9*

1. Kế hoạch *9*

2. Phân công công việc *9*

Chương 3: Thiết kế *10*

3. Thiết kế Model (Cơ sở dữ liệu) 10

3.1. Database Diagram 10

3.2. Bảng mô tả cách tạo các Table trong cơ sở dữ liệu 10

3.3. Bảng mô tả các Table trong cơ sở dữ liệu 12

4. Bảng mô tả các Properties của mỗi Table 15

4.1. Nhóm các sản phẩm 15

4.1.1. AbstractModel 15

4.1.2. Catalogs 15

4.1.3. Products 16

4.1.4. Customers 16

4.1.5. Orders 16

4.1.6. OrdersDetails 17

4.2. Nhóm các bài viết 17

4.2.1. Categories 17

4.2.2. Posts 17

4.3. Nhóm Users 18

5. Thiết kế View 18

6. Thiết Function 19

7. Thiết kế Store Procedure 20

7.1. Catalogs 20

7.2. Products 22

7.3. Category 24

7.4. Posts 26

7.5. Customer 27

7.6. Order 28

7.7. OrderDetail 29

7.8. CustomerOrders 30

7.9. Count (Dashboard) 31

8. Thiết kế Trigger 32

8.1. Product 32

8.2. Post 33

9. Seed Data 34

10. Thiết kế giao diện 35

Chương 4: Cài đặt và kiểm thử 37

1. Cài đặt 37

2. Kiểm thử 38

Chương 5: Kết luận và hướng phát triển 40

1. Kết luận 40

2. Hướng phát triển 41

Tài liệu tham khảo 42

Danh mục các hình

Hình 1 Mô tả View **4**

Hình 2 Cú pháp Trigger **6**

Hình 3 Lược đồ cơ sở dữ liệu **10**

Hình 4 SeedData **34**

Danh mục các bảng

Bảng 1 So sánh ưu và nhược điểm của View 5
Bảng 2 Ưu và nhược điểm của Stored Procedure 6
Bảng 3: Kế hoạch theo tuần 9
Bảng 4 Phân công công việc 9
Bảng 5 Tạo Table trong cơ sở dữ liệu 12
Bảng 6 Mô tả các Table trong cơ sở dữ liệu 15
Bảng 7 Mô tả properties của AbstractModel 15
Bảng 8 Mô tả properties của Catalogs 15
Bảng 9 Mô tả properties của Products 16
Bảng 10 Mô tả properties của Customers 16
Bảng 11 Mô tả properties của Orders 17
Bảng 12 Mô tả properties của OrdersDetails 17
Bảng 13 Mô tả properties của Categories 17
Bảng 14 Mô tả properties của Posts 18
Bảng 15 Mô tả properties của Users 18
Bảng 16 Bảng mô tả thiết kế View 19
Bảng 17 Bảng mô tả thiết kế Function 19
Bảng 18 Store Procedure của Catalogs 21
Bảng 19 Store Procedure của Products 24
Bảng 20 Store Procedure của Category 25
Bảng 21 Store Procedure của Posts 27
Bảng 22 Store Procedure của Customer 28
Bảng 23 Store Procedure của Order 29
Bảng 24 Store Procedure của OrderDetail 30
Bảng 25 Store Procedure của CustomerOrders 31
Bảng 26 Store Procedure của Count (Dashboard) 32
Bảng 27 Trigger của Product 33
Bảng 28 Trigger của Post 34
Bảng 29 Thiết kế giao diện 37
Bảng 30 Kiểm thử web 40

Chương 1: Đặc tả tổng quan chương trình

1. Giới thiệu chung

1.1. Giới thiệu Hệ quản trị cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu (Database Management System) có thể hiểu là hệ thống được thiết kế để quản lí một khối lượng dữ liệu nhất định một cách tự động và có trật tự. Các hành động quản lý này bao gồm chỉnh sửa, xóa, lưu thông tin và tìm kiếm (truy xuất thông tin) trong một nhóm dữ liệu nhất định.

Nói một cách dễ hiểu hơn, hệ quản trị cơ sở dữ liệu là hệ thống tự động giúp người dùng có thể kiểm soát các thông tin, tạo, cập nhật và duy trì các CSDL. Trong đó, hai thành phần chính trong một hệ quản trị cơ sở dữ liệu là: Bộ xử lí truy vấn (bộ xử lí yêu cầu) và bộ quản lí dữ liệu.

1.2. Sơ bộ về cách thiết kế Web bán hàng

1.2.1. Yêu cầu đồ án

- Sử dụng Triggers, constraint
- Views
- Stored Procedure và Function: Thực thi các chức năng của ứng dụng
- Tạo users, phân quyền và sử dụng các câu lệnh của DBMS

1.2.2. Phân tích đồ án

- Sử dụng Triggers, constraint, Views, Stored Procedure, Function.
- Tạo user, phân quyền
- Xây dựng các mối quan hệ giữa các đề mục cần quản lý để quản lý dễ dàng và hiệu quả hơn

1.2.3. Phương hướng thực hiện

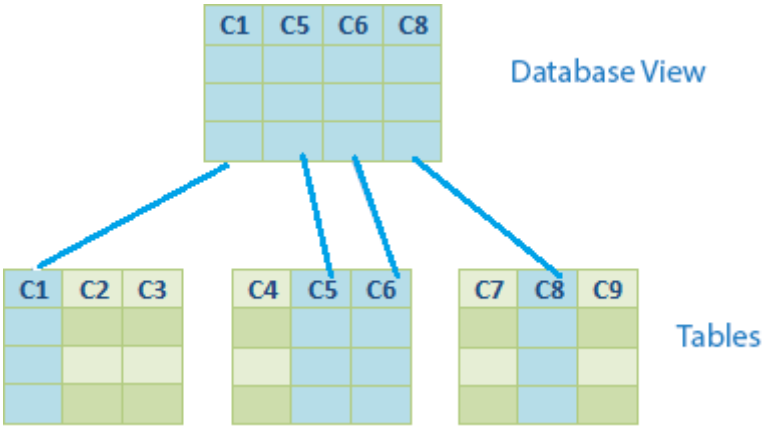
- Xây dựng web bán hàng sử dụng ADO.NET để truy vấn
- Xây dựng và thiết kế Views
- Xây dựng và thiết kế Function
- Xây dựng và thiết kế Stored Procedure
- Xây dựng và thiết kế Views
- Xây dựng và thiết kế Trigger

- Cài đặt trên giao diện và thực hiện data-seeding
- Thiết kế giao diện bằng HTML/CSS/ Angular (nhóm sử dụng Angular8)

1.3. Một số khái niệm cần biết

1.3.1. View

View được xem như là một bảng ảo trong CSDL có nội dung được định nghĩa thông qua một câu lệnh truy vấn(SELECT). Như vậy, View trông giống như một Table(bảng) với tên khung nhìn và tập hợp các dòng, cột. Điểm khác biệt giữa View và Table là: View không được xem là một cấu trúc dữ liệu tồn tại trong CSDL.



Hình 1 Mô tả View

Ưu điểm	Nhược điểm
Sử dụng khung nhìn để tập trung và đơn giản hóa dữ liệu: thông qua View ta có thể cung cấp cho người dùng những cấu trúc đơn giản, dễ hiểu về dữ liệu đồng thời giúp cho người dùng tập trung hơn trên những phần dữ liệu cần thiết. Sử dụng khung nhìn để tuỳ biến dữ liệu: cho phép tăng hoặc giảm các Fields tùy theo yêu cầu sử dụng bởi vì nó được tạo từ một câu truy vấn SELECT nên bạn có thể JOIN nhiều Table lại với nhau và lưu vào View	Khi truy vấn trong View có thể sẽ chậm hơn trong table Bị phụ thuộc vào Table gốc, nếu Table gốc thay đổi cấu trúc thì đòi hỏi View cũng phải thiết kế lại cho phù hợp

Sử dụng khung nhìn để xuất dữ liệu: Một View đóng vai trò như một đối tượng tập hợp dữ liệu từ nhiều bảng khác nhau vào trong một bảng -> Người sử dụng có thể thực hiện các yêu cầu truy vấn dữ liệu một cách đơn giản từ View thay vì phải đưa ra những câu truy vấn phức tạp. Sử dụng khung nhìn để bảo mật dữ liệu: View chỉ cho phép đọc (Read Only) nên việc hacker tấn công cập nhật dữ liệu là điều không thể.	

Bảng 1 So sánh ưu và nhược điểm của View

1.3.2. Function

Là một đối tượng trong cơ sở dữ liệu (CSDL) sử dụng trong các câu lệnh SQL, được biên dịch sẵn và lưu trong CSDL nhằm mục đích thực hiện xử lý nào đó như tính toán phức tạp và trả về kết quả là giá trị nào đó.

Đặc điểm của function:

- Luôn trả về giá trị
- Gồm 2 loại: Function hệ thống và Function do người dùng tự định nghĩa
- Function người dùng tự định nghĩa gồm 2 loại:
- Scalar-valued: Trả về giá trị vô hướng của các kiểu dữ liệu T-SQL
- Table-valued: Trả về bảng, là kết quả của một hoặc nhiều lệnh

1.3.3. Stored Procedure

Tương tự như khái niệm function trong các ngôn ngữ lập trình. Stored Procedure được tạo ra nhằm thực hiện các lệnh của sql theo một nhóm việc cụ thể thay vì thực hiện từng thao tác (insert,update,delete).

Ưu điểm	Nhược điểm

• Stored Procedure làm tăng hiệu xuất sử lý giữ liệu • Làm giảm thời gian giao tiếp giữa ứng dụng với hệ quản trị cơ sở giữ liệu. thay vì gửi từng câu truy vấn thì nay sẽ chỉ gửi một Stored Procedure. • module hóa ứng dụng: do Stored Procedure có thể lưu lại và sử dụng lại nhiều lần. điều này sẽ làm cho việc maintenance trở lên dễ dàng hơn. • nâng cao tính bảo mật dữ liệu: việc thao tác với database thông qua stored procedure mà không cho thao tác trực tiếp với database sẽ làm giảm các cuộc tấn công thông qua sql injection	• Nếu bạn tạo ra quá nhiều Procedure thì hệ quản trị sẽ sử dụng bộ nhớ để lưu trữ các thủ tục này khá nhiều. Ngoài ra nếu bạn thực hiện quá nhiều xử lý trong mỗi thủ tục thì đồng nghĩa với việc CPU sẽ làm việc nặng hơn, điều này không tốt chút nào. • Nếu sử dụng thủ tục thì sẽ rất khó phát triển trong ứng dụng, gây khó khăn ở mức logic business. • Để phát triển ứng dụng thì bạn phải đòi hỏi có một kỹ năng nhất định mới có thể maintenance được.

Bảng 2 Ưu và nhược điểm của Stored Procedure

1.3.4. Trigger

Hiểu đơn giản thì Trigger là một stored procedure không có tham số. Trigger thực thi một cách tự động khi một trong ba câu lệnh Insert, Update, Delete làm thay đổi dữ liệu trên bảng có chứa trigger.

```
CREATE TRIGGER tên_trigger ON tên_bảng
FOR {DELETE, INSERT, UPDATE}
AS
  câu_lệnh_sql
```

Hình 2 Cú pháp Trigger

❖ **Vậy, Trigger dùng để làm gì?**

- Trigger thường được sử dụng để kiểm tra ràng buộc (check constraints) trên nhiều quan hệ (nhiều bảng/table) hoặc trên nhiều dòng (nhiều record) của bảng.

- Ngoài ra việc sử dụng Trigger để chương trình có những hàm chạy ngầm nhằm phục vụ nhưng trường hợp hữu hạn và thường không sử dụng cho mục đích kinh doanh hoặc giao dịch.

1.3.5. Angular

Angular là một Javascript framework để xây dựng những ứng dụng phía client sử dụng HTML, CSS và một ngôn ngữ lập trình như Javascript.

2. Đặc tả Web bán hàng

2.1. Mô tả về Web bán hàng

- Trong kỉ nguyên công nghệ 4.0 ngày nay, việc mua bán hàng trực tuyến đã và đang trở nên ngày càng phổ biến với mỗi con người chúng ta. Chỉ cần một thiết bị có kết nối internet, ta có thể mua một cái áo cách xa hàng chục km chỉ với một nút bấm. Đó chính là một trong những đóng góp tích cực của công nghệ vào trong cuộc sống. Với mong muốn tiếp cận công nghệ mua bán hiện nay, nhóm chúng em xin được tìm hiểu về đề tài

2.2. Mục đích lựa chọn hình thức thiết kế quản trị bán hàng bằng web

- Thiết kế quản trị bán hàng bằng Web cho phép người bán đưa thông tin sản phẩm mình mua đến với khách hàng dễ dàng hơn với internet. Qua đó, người bán có thể quảng cáo cho sản phẩm của mình, khách hàng muốn mua hàng cũng dễ dàng lựa chọn sản phẩm ưng ý từ xa, giảm thiểu các bất cập khi khách đến cửa hàng nhưng không chọn được món vừa ý khi đến cửa hàng. Bên cạnh đó, thiết kế bán hàng bằng web cũng mang đến nhiều ưu điểm hơn thiết kế bằng ứng dụng trước đây, người quản trị web bán hàng vẫn có đầy đủ các chức năng kiểm soát dữ liệu của cửa hàng mà còn có thể tương tác với người mua qua mạng xã hội, hoặc boxchat.

2.2.1. Tính năng chính

Web bán hàng của nhóm có các chức năng phù hợp với đối tượng sử dụng. Cụ thể hơn:

+ Người mua hàng (Customer): Mục đích của người mua khi truy cập trang Web bán hàng hiển nhiên là để mua hàng thôi, nên các chức năng mà khách hàng có thể sử dụng được đó là xem hàng, xem thông tin chi tiết một món hàng nào đó (Details), Thêm hàng vào giỏ, Thanh toán.

+ Nhân viên bán hàng (Salesperson): Nhiệm vụ của nhân viên là ghi hóa đơn, xuất phiếu thu và không cần quan tâm đến việc nhập hàng hay quản lý thu nhập của cửa hàng nên chức năng mà nhân viên bán hàng có thể thực hiện đó là thanh toán, thêm sửa xóa khách hàng khi khách hàng có nhu cầu đăng kí thành viên.

+ Chủ shop : chủ shop sẽ là người có quyền hạn cao nhất, có thể quản lý tất cả tài khoản của nhân viên, được quyền quản lýviệc xuất nhập hàng của shop (các tính năng thêm sửa xóa - CRUD).

2.2.2. Ứng dụng

Web bán hàng thích hợp với hầu hết các loại hình buôn bán lớn nhỏ hiện nay. Tuy nhiên, vì thời gian tìm hiểu và thực hiện đồ án còn hạn chế, nên web bán hàng của nhóm em chỉ có thể đáp ứng được nhu cầu của các hệ thống nhỏ, các cửa hàng nhỏ, hi vọng trong tương lai sẽ có cơ hội khắc phục và cải tiến trang web này tốt hơn.

2.3. Yêu cầu kĩ thuật đối với trang web

- Thực hiện được các chức năng thêm sửa xóa, các chức năng đặc trưng của các người dùng
- Áp dụng lập trình hướng đối tượng vào cơ sở dữ liệu, vào giao diện web.
- Thiết kế logic và thể hiện được mối quan hệ giữa các đối trượng trong cơ sở dữ liệu
- Giao diện web thân thiện với người dùng

2.4. Công cụ và công nghệ sử dụng

- Sử dụng Microsoft SQL Server Management Studio để thiết kế Views, Function, Stored Procedure, Trigger và đồng thời là Seed Data
- Thiết kế giao diện người dùng với HTML/CSS/Angular

Chương 2: Kế hoạch thực hiện

1. *Kế hoạch*

Bảng 3: Kế hoạch theo tuần

Tuần	Công việc
16	Tìm hiểu về ADO.NET
16	Tìm hiểu và thiết kế View
17	Tìm hiểu và thiết kế Function
18	Tìm hiểu và thiết kế Store Procedure
18	Tìm hiểu và thiết kế Trigger
19	Kiểm tra các chức năng trên giao diện và tạo data seeding
19	Bắt đầu viết báo cáo đồ án và ppt

2. *Phân công công việc*

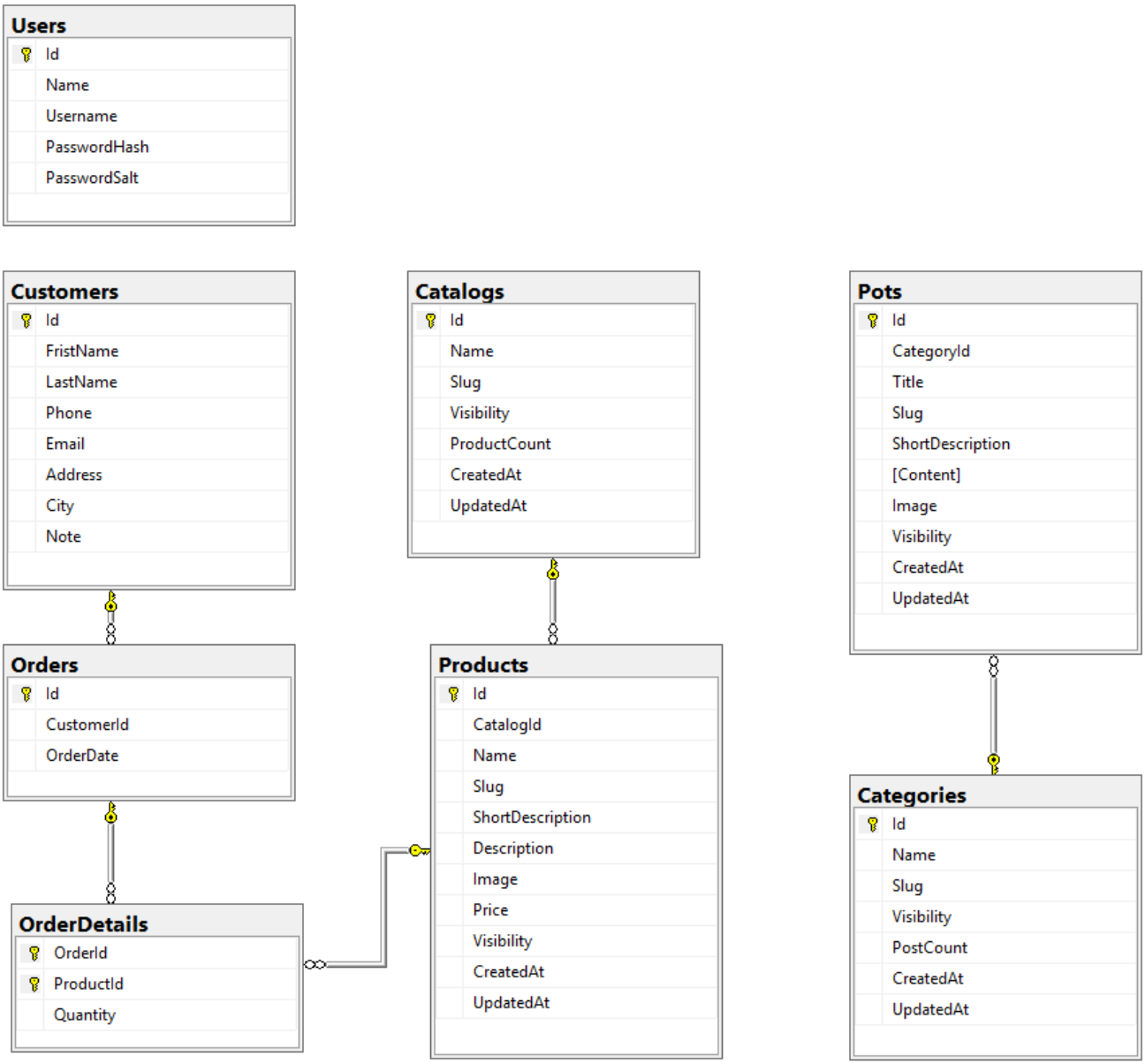
TT	Tên sinh viên	Miêu tả công việc	Đóng góp
1	Nguyễn Xuân Điền	- Thiết kế Trigger - Góp ý và chỉnh sửa Views - Góp ý chỉnh sửa Function - Góp ý chỉnh sửa Stored Procedure - Thiết kế giao diện web	40%
2	Vũ Hoàng Hiệp	- Thiết kế Views, Function - Hỗ trợ thiết kế Store Procedure - Góp ý chỉnh sửa Trigger	30%
3	Tôn Long Hoàng Lãm	- Thiết kế Stored Procedure - Seed Data - Góp ý chỉnh sửa Views, Function	30%

Bảng 4 Phân công công việc

Chương 3: Thiết kế

3. *Thiết kế Model (Cơ sở dữ liệu)*

3.1. Database Diagram



Hình 3 Lược đồ cơ sở dữ liệu

3.2. Bảng mô tả cách tạo các Table trong cơ sở dữ liệu

TT	Class	Cách tạo	Sinh viên phụ trách

1	Catalogs	```CREATE TABLE [dbo].[Catalogs]([Id][bigint] IDENTITY(1,1) PRIMARY KEY, [Name][varchar](255) NOT NULL, [Slug][varchar](255) NOT NULL, [Visibility][bit] NOT NULL, [ProductCount][int] NOT NULL, [CreatedAt][datetime] NOT NULL, [UpdatedAt][datetime] NOT NULL,) GO``` 1	Vũ Hoàng Hiệp
2	Products	```CREATE TABLE [dbo].[Products]([Id][bigint] IDENTITY(1,1) PRIMARY KEY, [CatalogId][bigint] NOT NULL, [Name][varchar](255) NOT NULL, [Slug][varchar](255) NOT NULL, [ShortDescription][varchar](255) NOT NULL, [Description][text] NOT NULL, [Image][varchar](255) NOT NULL, [Price][decimal](18,4) NOT NULL, [Visibility][bit] NOT NULL, [CreatedAt][datetime] NOT NULL, [UpdatedAt][datetime] NOT NULL, CONSTRAINT FK_CatalogProduct FOREIGN KEY (CatalogId) REFERENCES Catalogs(Id)) GO``` 1	Vũ Hoàng Hiệp
3	Customer	```CREATE TABLE [dbo].[Customers] ([Id][bigint] IDENTITY(1,1) PRIMARY KEY, [FristName][varchar](50) NOT NULL, [LastName][varchar](50) NOT NULL, [Phone][varchar](50) NOT NULL, [Email][varchar](255) NOT NULL, [Address][varchar](255) NOT NULL, [City][varchar](100) NOT NULL, [Note][varchar](255) NOT NULL) GO``` 1	Nguyễn Xuân Điền
4	Orders	```CREATE TABLE [dbo].[Orders] ([Id][bigint] IDENTITY(1,1) PRIMARY KEY, [CustomerId][bigint] NOT NULL, [OrderDate][datetime] NOT NULL, CONSTRAINT FK_CusromerOrder FOREIGN KEY (CustomerId) REFERENCES Customers(Id)) GO``` 1	Nguyễn Xuân Điền

5	OdersDetails	```CREATE TABLE [dbo].[OrderDetails] ([OrderId][bigint] NOT NULL, [ProductId][bigint] NOT NULL, [Quantity][int] NOT NULL, CONSTRAINT PK_OrderDetail PRIMARY KEY (OrderId,ProductId), FOREIGN KEY (OrderId) REFERENCES Orders(Id), FOREIGN KEY (ProductId) REFERENCES Products(Id)) GO```	Vũ Hoàng Hiệp
6	Categories	```CREATE TABLE [dbo].[Categories]([Id][bigint] IDENTITY(1,1) PRIMARY KEY, [Name][varchar](255) NOT NULL, [Slug][varchar](255) NOT NULL, [Visibility][bit] NOT NULL, [PostCount][int] NOT NULL, [CreatedAt][datetime] NOT NULL, [UpdatedAt][datetime] NOT NULL,) GO```	Tôn Long Hoàng Lãm
7	Posts	```CREATE TABLE [dbo].[Pots]([Id][bigint] IDENTITY(1,1) PRIMARY KEY, [CategoryId][bigint] NOT NULL, [Title][varchar](255) NOT NULL, [Slug][varchar](255) NOT NULL, [ShortDescription][varchar](255) NOT NULL, [Content][text] NOT NULL, [Image][varchar](255) NOT NULL, [Visibility][bit] NOT NULL, [CreatedAt][datetime] NOT NULL, [UpdatedAt][datetime] NOT NULL, CONSTRAINT FK_CategoryPost FOREIGN KEY (CategoryId) REFERENCES Categories(Id)) GO```	Tôn Long Hoàng Lãm
8	Users	```CREATE TABLE [dbo].[Users] ([Id][bigint] IDENTITY(1,1) PRIMARY KEY, [Name][varchar](255) NOT NULL, [Username][varchar](255) NOT NULL, [PasswordHash][varbinary](MAX) NOT NULL, [PasswordSalt][varbinary](MAX) NOT NULL) GO```	Nguyễn Xuân Điền

Bảng 5 Tạo Table trong cơ sở dữ liệu

3.3. Bảng mô tả các Table trong cơ sở dữ liệu

TT	Class	Kế thừa	Mô tả	Sinh viên phụ trách

1	Catalogs		Catalogs (Danh mục sản phẩm): 1. Gồm nhiều Products (Sản phẩm) 2. 2. PrimaryKey: Id	Vũ Hoàng Hiệp
2	Products	Catalogs	Products (Sản phẩm): 1. Mỗi Products nằm trong 1 Catalogs (Danh mục sản phẩm). 2. PrimaryKey: Id 3. ForeignKey: CatalogsId (tham chiếu đến Id của Catalogs) 4. 4. Không thể xóa Catalogs chứa 1 sản phẩm nào đó	Vũ Hoàng Hiệp
3	Customers		Customers (khách hàng): 1. Là người đến mua hàng. Việc quản lý thêm sửa xóa khách hàng nhằm phục vụ cho mục đích tìm lại thông tin mua bán hàng trước đó, đối chiếu khi cần thiết. 2. PrimaryKey: Id	Vũ Hoàng Hiệp
4	Oders	Customers	Orders: 1. Oders sinh ra khi khách hàng đặt hàng, mỗi Customers có thể có nhiều Orders. 2. PrimaryKey: Id	Vũ Hoàng Hiệp

			3. ForeignKey: CustomersId (tham chiếu đến Id của Customers) 4. Không thể xóa 1 Customers đang có Oders	
5	OdersDetails	Products Oders	OdersDetails (Chi tiết đơn đặt hàng): 1. Chứa 2 khóa ngoại: 1 tham chiếu đến Products và 1 tham chiếu đến Orders 2. PrimaryKey: Id 3. ForeignKey: ProductsId (tham chiếu đến Id của Products) 4. ForeignKey: Oders ID(tham chiếu đến Id của Oders) 5. Không thể xóa Products hoặc Oders có OrdersDetails	Vũ Hoàng Hiệp – Nguyễn Xuân Điền (cùng góp ý xây dựng và thực hiện)
6	Categories		Categories (Danh mục bài viết): 1. Bao gồm các bài viết, hình ảnh mô tả về sản phẩm sẽ bán 2. PrimaryKey: Id	Tôn Long Hoàng Lãm
7	Posts	Categories	Posts (Bài viết): 1. Bài viết được viết bởi người quản trị 2. PrimaryKey: Id 3. ForeignKey: CategoriesId (tham chiếu đến Id của Categories)	Tôn Long Hoàng Lãm

8	Users		1. Users chứa các tài khoản có thể thực hiện thao tác trên web 2. Properties của Users còn bao gồm: 3. PasswordHash (Password được mã hóa vì lí do bảo mật) 4. PasswordSalt (Key giải mã mật khẩu dùng để đăng nhập)	Nguyễn Xuân Điền

Bảng 6 Mô tả các Table trong cơ sở dữ liệu

4. *Bảng mô tả các Properties của mỗi Table*

4.1. Nhóm các sản phẩm

4.1.1. AbstractModel

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Id`	`long`	Id cho các bảng kế thừa
2	`CreatedAt`	`DateTime`	Lưu ngày tháng khởi tạo một bảng
3	`UpdateAt`	`DateTime`	Lưu ngày tháng update một bảng

Bảng 7 Mô tả properties của AbstractModel

4.1.2. Catalogs

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Name`	`string`	Lưu tên
2	`Slug`	`string`	FORMAT trường Name làm SEO URL
3	`ShortDescription`	`string`	Lưu những mô tả ngắn và danh mục
4	`Description`	`string`	Lưu mổ tả chi tiết về danh mục
5	`Visibility`	`bool`	Thuộc tính quyết định danh mục này có hiển thị với người dùng không

Bảng 8 Mô tả properties của Catalogs

4.1.3. Products

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Name`	`string`	Lưu tên
2	`Slug`	`string`	FORMAT trường Name làm SEO URL
3	`ShortDescription`	`string`	Lưu những mô tả ngắn và danh mục
4	`Description`	`string`	Lưu mổ tả chi tiết về danh mục
5	`Visibility`	`bool`	Thuộc tính quyết định danh mục này có hiển thị với người dùng không

Bảng 9 Mô tả properties của Products

4.1.4. Customers

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Id(FK)`	`Long`	Lưu id của khách hàng
2	`FirstName`	`string`	Lưu tên khách hàng
3	`LastName`	`string`	Lưu tên họ khách hàng
4	`Phone`	`string`	Lưu số điện thoại khách hàng
5	`Email`	`string`	Lưu email khách hàng
6	`Address`	`string`	Lưu địa chỉ của khách hàng
7	`City`	`string`	Lưu thành phố nơi khách hàng sống
8	`Note`	`String`	Lưu các ghi chú về khách hàng

Bảng 10 Mô tả properties của Customers

4.1.5. Orders

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Id(PK)`	`Long`	Id của lần đặt hàng
2	`OrderDate`	`DateTime`	Ngày đặt hàng

3	`CustomerId(FK_Oders_Customers_Id)`	`long`	Lưu id của khách hàng mua

Bảng 11 Mô tả properties của Orders

4.1.6. OrdersDetails

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`OrderId(FK_OrdersDetails_Orders_Id)`	`long`	Tham chiếu đến khóa bảng Orders, lưu thông tin Orders
2	`ProductId(FK_OrdersDetails_Products_id)`	`long`	Tham chiếu đến bảng Products, lưu thông tin các sản phẩm khách hàng mua
3	`Quantity`	`int`	Lưu số lượng sản phẩm được đặt hàng

Bảng 12 Mô tả properties của OrdersDetails

4.2. Nhóm các bài viết

4.2.1. Categories

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Name`	`string`	Lưu tên Categories
2	`ShortDescription`	`string`	Lưu mô tả ngắn gọn về danh mục
3	`Description`	`string`	Lưu mô tả về danh mục
4	`Visibility`	`bool`	Hiển thị hay không hiển thị Category đó

Bảng 13 Mô tả properties của Categories

4.2.2. Posts

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Name`	`string`	Lưu tên Categories
2	`ShortDescription`	`string`	Lưu mô tả ngắn gọn về danh mục
3	`Content`	`string`	Nội dung bài viết

4	`Image`	`string`	Lưu đường dẫn tới hình ảnh
5	`CategoryId(FK_Posts_Categories_id)`	`long`	Khóa ngoại tham chiếu đến Categories

Bảng 14 Mô tả properties của Posts

4.3. Nhóm Users

TT	Tên thuộc tính (Properties)	Kiểu dữ liệu	Mục đích
1	`Id(PK)`	`long`	Id của User
2	`Name`	`string`	Lưu tên User
3	`Username`	`string`	Lưu tên người đăng kí User
4	`PasswordHash`	`byte[]`	Lưu passwords dạng hash (đã mã hóa)
5	`PasswordSalt`	`byte[]`	Lưu key để giải mã passwordHash

Bảng 15 Mô tả properties của Users

5. *Thiết kế View*

TT	Tên View	Tạo View	Mục đích

1	view_CustomerOrders	`CREATE VIEW [dbo].[view_CustomerOrders] AS`
		`SELECT [o].[Id] as OrderId, [OrderDate],`
		`[c].[Id] as CustomerId, [c].[FristName], [c].[LastName], [c].[Phone],`
		`[c].[Email], [c].[Address], [c].[City], [c].[Note]`
		`FROM Customers as c`
		`INNER JOIN Orders as o`
		`ON c.Id = o.CustomerId`
		`GO`
			Lấy thông tin đơn hàng cùng với thông tin khách hàng

2	view_CustomerOrderDetails	`CREATE VIEW [dbo].[view_CustomerOrderDetails] AS`
		`SELECT [o].[Id] as OrderId, [OrderDate],`
		`[c].[Id] as CustomerId, [c].[FristName], [c].[LastName], [c].[Phone],`
		`[c].[Email], [c].[Address], [c].[City], [c].[Note],`
		`[od].[ProductId] as ProductId, [od].[Quantity],`
		`[p].[Name] as ProductName, [p].[Price]`
		`FROM [dbo].[Orders] as o`
		`INNER JOIN [dbo].[Customers] as c`
		`ON [c].[Id] = [o].[CustomerId]`
		`INNER JOIN [dbo].[OrderDetails] as od`
		`ON [od].[OrderId] = [o].[Id]`
		`INNER JOIN [dbo].[Products] as p`
		`ON [od].[ProductId] = [p].[Id]`
		`GO`
			Lấy thông tin đơn hàng cùng với thông tin khách hàng cùng với số

			lượng sản phẩm và thông tin sản phẩm

Bảng 16 Bảng mô tả thiết kế View

6. Thiết Function

TT	Tên Function	Tạo Function	Mục đích

1	Func_Calculate PriceQuantity Product	`CREATE FUNCTION [dbo].[func_CalculatePriceQuantityProduct]`
		`(`
		`  @quantity int,`
		`  @price decimal`
		`)`
		`RETURNS decimal`
		`AS`
		`BEGIN`
		`  DECLARE @TotalPrice decimal`
		`  set @TotalPrice = @quantity * @price`
		`  return @TotalPrice`
		`END`
		`GO`
			Tính tổng giá tiền phải trả cho số lượng sản phẩm của 1 sản phẩm

2	Func_Calculate TotalPriceOne Order	`CREATE FUNCTION [dbo].[func_CalculateTotalPriceOneOrder]()`
		`RETURNS @TotalBillPrice table`
		`(`
		`  OrderId bigint,`
		`  TotalProduct int,`
		`  TotalPrice decimal`
		`)`
		`AS`
		`BEGIN`
		`  INSERT INTO @TotalBillPrice`
		`  select OrderId, SUM(Quantity) as TotalProduct ,`
		`  SUM([dbo].[func_CalculatePriceQuantityProduct]([c].[Quantity], [c].[Price]))`
		`  AS TotalPrice`
		`  FROM [dbo].[view_CustomerOrderDetails] as c`
		`  GROUP BY OrderId`
		`  return`
		`END`
		`GO`
			Tính tổng số tiền khách hàng phải trả cho lần Order

3	Func_TotalIncome	`CREATE FUNCTION [dbo].[func_TotalIncome]()`
		`RETURNS @TotalPrice table`
		`(`
		`  TotalIncome decimal`
		`)`
		`AS`
		`BEGIN`
		`  INSERT INTO @TotalPrice`
		`  SELECT ISNULL(SUM(t.TotalPrice), 0) as TotalIncome`
		`  FROM (SELECT * FROM [dbo].[func_CalculateTotalPriceOneOrder]()) as t`
		`  GROUP BY t.OrderId, t.TotalProduct`
		`  return`
		`END`
		`GO`
			Tính tổng thu nhập

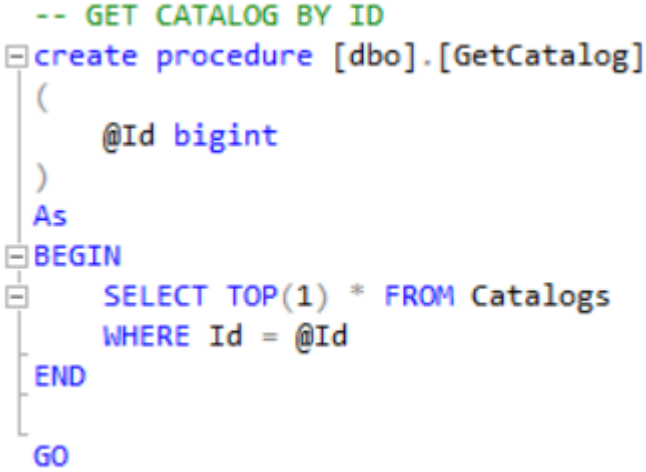
Bảng 17 Bảng mô tả thiết kế Function

7. *Thiết kế Store Procedure*

7.1. Catalogs

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	GetCatalogs	
		`create procedure [dbo].[GetCatalogs]`
		`(`
		`@SearchValue varchar(255) = null,`
		`@SortOrderName varchar(50),`
		`@SortOrder varchar(4),`
		`@CurrentPage int,`
		`@PageSize int`
		`)`
		`As`
		`BEGIN`
		`IF(@SearchValue IS NULL)`
		`BEGIN`
		`SELECT * FROM Catalogs as c`
		`ORDER BY`
		`CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [c].[Id] END DESC,`
		`CASE WHEN @SortOrderName = 'Id' THEN [c].[Id] END,`
		`CASE WHEN @SortOrderName = 'Name' AND @SortOrder = 'DESC' THEN [c].[Name] END DESC,`
		`CASE WHEN @SortOrderName = 'Name' THEN [c].[Name] END,`
		`CASE WHEN @SortOrderName = 'Slug' AND @SortOrder = 'DESC' THEN [c].[Slug] END DESC,`
		`CASE WHEN @SortOrderName = 'Slug' THEN [c].[Slug] END,`
		`CASE WHEN @SortOrderName = 'Visibility' AND @SortOrder = 'DESC' THEN [c].[Visibility] END DESC,`
		`CASE WHEN @SortOrderName = 'Visibility' THEN [c].[Visibility] END,`
		`CASE WHEN @SortOrderName = 'CreatedAt' AND @SortOrder = 'DESC' THEN [c].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'CreatedAt' THEN [c].[CreatedAt] END,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' AND @SortOrder = 'DESC' THEN [c].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' THEN [c].[UpdatedAt] END`
		`OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY`
		`END`
		`ELSE`
		`BEGIN`
		`SELECT * FROM Catalogs as c`
		`WHERE CHARINDEX(@SearchValue, [c].[Name]) > 0`
		`ORDER BY`
		`CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [c].[Id] END DESC,`
		`CASE WHEN @SortOrderName = 'Id' THEN [c].[Id] END,`
		`CASE WHEN @SortOrderName = 'Name' AND @SortOrder = 'DESC' THEN [c].[Name] END DESC,`
		`CASE WHEN @SortOrderName = 'Name' THEN [c].[Name] END,`
		`CASE WHEN @SortOrderName = 'Slug' AND @SortOrder = 'DESC' THEN [c].[Slug] END DESC,`
		`CASE WHEN @SortOrderName = 'Slug' THEN [c].[Slug] END,`
		`CASE WHEN @SortOrderName = 'Visibility' AND @SortOrder = 'DESC' THEN [c].[Visibility] END DESC,`
		`CASE WHEN @SortOrderName = 'Visibility' THEN [c].[Visibility] END,`
		`CASE WHEN @SortOrderName = 'CreatedAt' AND @SortOrder = 'DESC' THEN [c].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'CreatedAt' THEN [c].[CreatedAt] END,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' AND @SortOrder = 'DESC' THEN [c].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' THEN [c].[UpdatedAt] END`
		`OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY`
		`END`
		`END`
			Thực hiện lấy tất cả các danh mục sản phẩm (Catalogs) nếu giá trị tìm kiếm (SearchValue) bằng null (chưa nhập hoặc rỗng) thì lấy tất cả đồng thời phân trang và sắp xếp tăng dần giảm dần theo tên attribute

2	GetCatalog	
		`-- GET CATALOG BY ID`
		`create procedure [dbo].[GetCatalog]`
		`(`
		`@Id bigint`
		`)`
		`As`
		`BEGIN`
		`SELECT TOP(1) * FROM Catalogs`
		`WHERE Id = @Id`
		`END`
		`GO`
			Lấy thông tin Catalog thông qua Id truyền vào, mục đích là khi click vào 1 Catalog thì nhận thông tin của Catalogs đó

3	InsertCatalog	```
		-- INSERT CATALOG
		create procedure [dbo].[InsertCatalog]
		(
		@Name varchar(255),
		@Slug varchar(255),
		@Visibility bit,
		@ProductCount int,
		@CreatedAt datetime,
		@UpdatedAt datetime
		)
		As
		BEGIN
		INSERT INTO Catalogs(Name, Slug, Visibility, ProductCount, CreatedAt, UpdatedAt)
		VALUES(@Name, @Slug, @Visibility, @ProductCount, @CreatedAt, @UpdatedAt)
		SELECT SCOPE_IDENTITY()
		END
		GO
		``
			Thêm một Catalog mới

4	UpdateCatalog	```
		-- UPDATE CATALOG
		create procedure [dbo].[UpdateCatalog]
		(
		@Id bigint,
		@Name varchar(100),
		@Slug text,
		@Visibility bit,
		@ProductCount int,
		@CreatedAt datetime,
		@UpdatedAt datetime
		)
		As
		BEGIN
		UPDATE Catalogs
		SET Name = @Name, Slug = @Slug, Visibility = @Visibility, ProductCount=@ProductCount,
		CreatedAt = @CreatedAt, UpdatedAt = @UpdatedAt
		WHERE Id = @Id;
		END
		GO
		``
			Sửa thông tin một Catalog

5	DeleteCatalog	```
		-- DELETE CATALOG
		create procedure [dbo].[DeleteCatalog]
		(
		@Id bigint
		)
		As
		BEGIN
		DELETE FROM Catalogs WHERE Id = @Id
		END
		``
			Xóa một Catalog được chọn, khi click vào 1 Catalog hàm GetCatalog sẽ lấy địa chỉ Id của Catalog đó, nếu muốn xóa Catalog Id này sẽ được truyền vào hàm DeleteCatalog

Bảng 18 Store Procedure của Catalogs

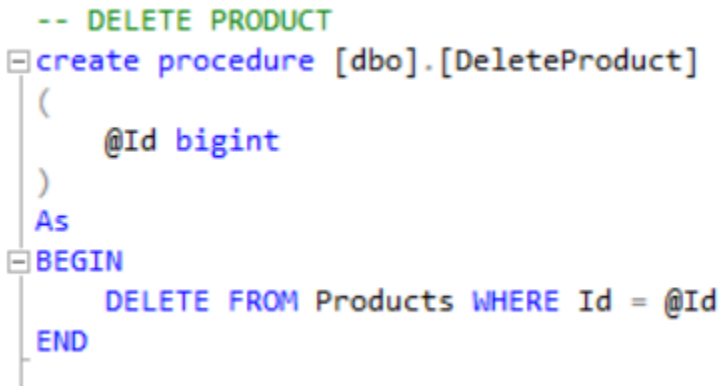
7.2. Products

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	GetProducts	
		`create procedure [dbo].[GetProducts]`
		`(`
		`@SearchValue varchar(255) = null,`
		`@SortOrderName varchar(50),`
		`@SortOrder varchar(4),`
		`@CurrentPage int,`
		`@PageSize int`
		`)`
		`As`
		`BEGIN`
		`IF(@SearchValue IS NULL)`
		`BEGIN`
		`SELECT * FROM Products as p`
		`INNER JOIN Catalogs as c`
		`ON [p].[CatalogId] = c.Id`
		`ORDER BY`
		`CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [p].[Id] END DESC,`
		`CASE WHEN @SortOrderName = 'Id' THEN [p].[Id] END,`
		`CASE WHEN @SortOrderName = 'Name' AND @SortOrder = 'DESC' THEN [p].[Name] END DESC,`
		`CASE WHEN @SortOrderName = 'Name' THEN [p].[Name] END,`
		`CASE WHEN @SortOrderName = 'Slug' AND @SortOrder = 'DESC' THEN [p].[Slug] END DESC,`
		`CASE WHEN @SortOrderName = 'Slug' THEN [p].[Slug] END,`
		`CASE WHEN @SortOrderName = 'Price' AND @SortOrder = 'DESC' THEN [p].[Price] END DESC,`
		`CASE WHEN @SortOrderName = 'Price' THEN [p].[Price] END,`
		`CASE WHEN @SortOrderName = 'Visibility' AND @SortOrder = 'DESC' THEN [p].[Visibility] END DESC,`
		`CASE WHEN @SortOrderName = 'Visibility' THEN [p].[Visibility] END,`
		`CASE WHEN @SortOrderName = 'CreatedAt' AND @SortOrder = 'DESC' THEN [p].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'CreatedAt' THEN [p].[CreatedAt] END,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' AND @SortOrder = 'DESC' THEN [p].[UpdatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' THEN [p].[UpdatedAt] END`
		`OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY`
		`END`
		`ELSE`
		`BEGIN`
		`SELECT * FROM Products as p`
		`INNER JOIN Catalogs as c`
		`ON [p].[CatalogId] = c.Id`
		`WHERE CHARINDEX(@SearchValue, [p].[Name]) > 0`
		`ORDER BY`
		`CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [p].[Id] END DESC,`
		`CASE WHEN @SortOrderName = 'Id' THEN [p].[Id] END,`
		`CASE WHEN @SortOrderName = 'Name' AND @SortOrder = 'DESC' THEN [p].[Name] END DESC,`
		`CASE WHEN @SortOrderName = 'Name' THEN [p].[Name] END,`
		`CASE WHEN @SortOrderName = 'Slug' AND @SortOrder = 'DESC' THEN [p].[Slug] END DESC,`
		`CASE WHEN @SortOrderName = 'Slug' THEN [p].[Slug] END,`
		`CASE WHEN @SortOrderName = 'Price' AND @SortOrder = 'DESC' THEN [p].[Price] END DESC,`
		`CASE WHEN @SortOrderName = 'Price' THEN [p].[Price] END,`
		`CASE WHEN @SortOrderName = 'Visibility' AND @SortOrder = 'DESC' THEN [p].[Visibility] END DESC,`
		`CASE WHEN @SortOrderName = 'Visibility' THEN [p].[Visibility] END,`
		`CASE WHEN @SortOrderName = 'CreatedAt' AND @SortOrder = 'DESC' THEN [p].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'CreatedAt' THEN [p].[CreatedAt] END,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' AND @SortOrder = 'DESC' THEN [p].[UpdatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' THEN [p].[UpdatedAt] END`
		`OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY`
		`END`
		`END`
			Thực hiện lấy tất cả các sản phẩm (Products) nếu giá trị tìm kiếm (SearchValue) bằng null (chưa nhập hoặc rỗng) thì lấy tất cả đồng thời phân trang và sắp xếp tăng dần giảm dần theo tên attribute

2	GetCatalogProducts	
		`-- GET CATALOG PRODUCTS`
		`create procedure [dbo].[GetCatalogProducts]`
		`(`
		`@SearchValue varchar(255) = null,`
		`@SortOrderName varchar(50),`
		`@SortOrder varchar(4),`
		`@CurrentPage int,`
		`@PageSize int`
		`)`
		`As`
		`BEGIN`
		`SELECT * FROM Products as p`
		`INNER JOIN Catalogs as c`
		`ON [p].[CatalogId] = c.Id`
		`WHERE [p].[CatalogId] = @SearchValue`
		`ORDER BY`
		`CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [p].[Id] END DESC,`
		`CASE WHEN @SortOrderName = 'Id' THEN [p].[Id] END,`
		`CASE WHEN @SortOrderName = 'Name' AND @SortOrder = 'DESC' THEN [p].[Name] END DESC,`
		`CASE WHEN @SortOrderName = 'Name' THEN [p].[Name] END,`
		`CASE WHEN @SortOrderName = 'Slug' AND @SortOrder = 'DESC' THEN [p].[Slug] END DESC,`
		`CASE WHEN @SortOrderName = 'Slug' THEN [p].[Slug] END,`
		`CASE WHEN @SortOrderName = 'Price' AND @SortOrder = 'DESC' THEN [p].[Price] END DESC,`
		`CASE WHEN @SortOrderName = 'Price' THEN [p].[Price] END,`
		`CASE WHEN @SortOrderName = 'Visibility' AND @SortOrder = 'DESC' THEN [p].[Visibility] END DESC,`
		`CASE WHEN @SortOrderName = 'Visibility' THEN [p].[Visibility] END,`
		`CASE WHEN @SortOrderName = 'CreatedAt' AND @SortOrder = 'DESC' THEN [p].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'CreatedAt' THEN [p].[CreatedAt] END,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' AND @SortOrder = 'DESC' THEN [p].[UpdatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' THEN [p].[UpdatedAt] END`
		`OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY`
		`END`
		`GO`
			Lấy những sản phẩm thuộc nhóm Catalog nhất định show ra

3	GetProduct	 -- GET PRODUCT BY ID create procedure [dbo].[GetProduct] (@Id bigint) As BEGIN SELECT TOP(1) * FROM Products as p INNER JOIN Catalogs as c ON [p].[CatalogId] = c.Id WHERE [p].[Id] = @Id END GO	Lấy thông tin của Product khi click vào, nhận Id của product để truy xuất thông tin
4	InsertProduct	 -- INSERT PRODUCT create procedure [dbo].[InsertProduct] (@Name varchar(255), @Slug varchar(255), @ShortDescription varchar(255), @Description text, @Price decimal (18,4), @Image varchar(255), @Visibility bit, @CatalogId bigint, @CreatedAt datetime, @UpdatedAt datetime) As BEGIN INSERT INTO Products(Name, Slug,ShortDescription, Description, Price, Image, Visibility, CatalogId, CreatedAt, UpdatedAt) VALUES(@Name, @Slug, @ShortDescription, @Description, @Price, @Image, @Visibility, @CatalogId,@CreatedAt,@UpdatedAt) SELECT SCOPE_IDENTITY() END GO	Thêm một sản phẩm
5	UpdateProduct	 -- UPDATE PRODUCT create procedure [dbo].[UpdateProduct] (@Id bigint, @Name varchar(255), @Slug varchar(255), @ShortDescription varchar(255), @Description text, @Price decimal (18,4), @Image varchar(255), @Visibility bit, @CatalogId bigint, @CreatedAt datetime, @UpdatedAt datetime) As BEGIN UPDATE Products SET Name = @Name, Slug = @Slug, ShortDescription = @ShortDescription, Description = @Description, Price = @Price, Image = @Image, Visibility = @Visibility, CatalogId = @CatalogId, CreatedAt = @CreatedAt, UpdatedAt = @UpdatedAt WHERE Id = @Id; END GO	Sửa thông tin một sản phẩm

6	DeleteProduct		Xóa một sản phẩm có Id được truyền vào

Bảng 19 Store Procedure của Products

7.3. Category

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích
1	GetCategories		Thực hiện lấy tất cả các danh mục bài viết (Categories) nếu giá trị tìm kiếm (SearchValue) bằng null (chưa nhập hoặc rỗng) thì lấy tất cả Categories đồng thời phân trang và sắp xếp tăng dần giảm dần theo tên attribute

2	GetCategory	`-- GET CATEGORY BY ID`
		`create procedure [dbo].[GetCategory]`
		`(`
		`@Id bigint`
		`)`
		`As`
		`BEGIN`
		`SELECT TOP(1) * FROM [dbo].[Categories]`
		`WHERE Id = @Id`
		`END`
		`GO`
			Lấy thông tin của Category khi click vào, nhận Id của product để truy xuất thông tin

3	InsertCategory	`-- INSERT CATEGORY`
		`create procedure [dbo].[InsertCategory]`
		`(`
		`@Name varchar(255),`
		`@Slug varchar(255),`
		`@Visibility bit,`
		`@PostCount int,`
		`@CreatedAt datetime,`
		`@UpdatedAt datetime`
		`)`
		`As`
		`BEGIN`
		`INSERT INTO Categories(Name, Slug, Visibility, PostCount, CreatedAt, UpdatedAt)`
		`VALUES(@Name, @Slug, @Visibility, @PostCount, @CreatedAt, @UpdatedAt)`
		`END`
		`GO`
			Thêm một danh mục bài viết (Category)

4	UpdateCategory	`-- UPDATE CATEGORY`
		`create procedure [dbo].[UpdateCategory]`
		`(`
		`@Id bigint,`
		`@Name varchar(255),`
		`@Slug varchar(255),`
		`@Visibility bit,`
		`@PostCount int,`
		`@CreatedAt datetime,`
		`@UpdatedAt datetime`
		`)`
		`As`
		`BEGIN`
		`UPDATE Categories`
		`SET Name = @Name, Slug = @Slug, Visibility = @Visibility, PostCount=@PostCount,`
		`CreatedAt = @CreatedAt, UpdatedAt = @UpdatedAt`
		`WHERE Id = @Id;`
		`END`
		`GO`
			Sửa thông tin một danh mục bài viết (Category)

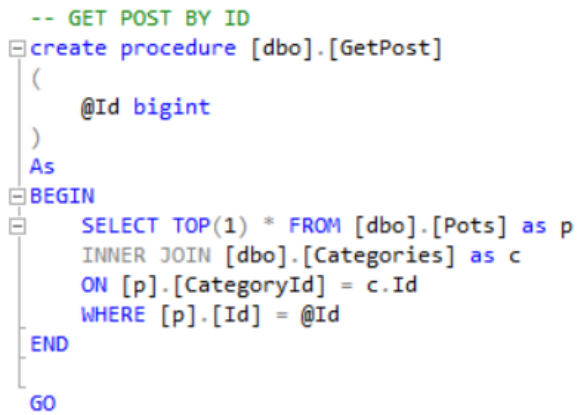
5	DeleteCategory	`-- DELETE CATEGORY`
		`create procedure [dbo].[DeleteCategory]`
		`(`
		`@Id bigint`
		`)`
		`As`
		`BEGIN`
		`DELETE FROM Categories WHERE Id = @Id`
		`END`
		`GO`
			Xóa một danh mục bài viết (Category) có Id được truyền vào

Bảng 20 Store Procedure của Category

7.4. Posts

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	GetPosts	
		`create procedure [dbo].[GetPosts]`
		`(`
		`@SearchValue varchar(255) = null,`
		`@SortOrderName varchar(50),`
		`@SortOrder varchar(4),`
		`@CurrentPage int,`
		`@PageSize int`
		`)`
		`As`
		`BEGIN`
		`IF(@SearchValue IS NULL)`
		`BEGIN`
		`SELECT * FROM [dbo].[Pots] as p`
		`INNER JOIN [dbo].[Categories] as c`
		`ON [p].[CategoryId] = c.Id`
		`ORDER BY`
		`CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [p].[Id] END DESC,`
		`CASE WHEN @SortOrderName = 'Id' THEN [p].[Id] END,`
		`CASE WHEN @SortOrderName = 'Title' AND @SortOrder = 'DESC' THEN [p].[Title] END DESC,`
		`CASE WHEN @SortOrderName = 'Title' THEN [p].[Title] END,`
		`CASE WHEN @SortOrderName = 'Slug' AND @SortOrder = 'DESC' THEN [p].[Slug] END DESC,`
		`CASE WHEN @SortOrderName = 'Slug' THEN [p].[Slug] END,`
		`CASE WHEN @SortOrderName = 'Visibility' AND @SortOrder = 'DESC' THEN [p].[Visibility] END DESC,`
		`CASE WHEN @SortOrderName = 'Visibility' THEN [p].[Visibility] END,`
		`CASE WHEN @SortOrderName = 'CreatedAt' AND @SortOrder = 'DESC' THEN [p].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'CreatedAt' THEN [p].[CreatedAt] END,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' AND @SortOrder = 'DESC' THEN [p].[UpdatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' THEN [p].[UpdatedAt] END`
		`OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY`
		`END`
		`ELSE`
		`BEGIN`
		`SELECT * FROM [dbo].[Pots] as p`
		`INNER JOIN [dbo].[Categories] as c`
		`ON [p].[CategoryId] = c.Id`
		`WHERE CHARINDEX(@SearchValue, [p].[Title]) > 0`
		`ORDER BY`
		`CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [p].[Id] END DESC,`
		`CASE WHEN @SortOrderName = 'Id' THEN [p].[Id] END,`
		`CASE WHEN @SortOrderName = 'Title' AND @SortOrder = 'DESC' THEN [p].[Title] END DESC,`
		`CASE WHEN @SortOrderName = 'Title' THEN [p].[Title] END,`
		`CASE WHEN @SortOrderName = 'Slug' AND @SortOrder = 'DESC' THEN [p].[Slug] END DESC,`
		`CASE WHEN @SortOrderName = 'Slug' THEN [p].[Slug] END,`
		`CASE WHEN @SortOrderName = 'Visibility' AND @SortOrder = 'DESC' THEN [p].[Visibility] END DESC,`
		`CASE WHEN @SortOrderName = 'Visibility' THEN [p].[Visibility] END,`
		`CASE WHEN @SortOrderName = 'CreatedAt' AND @SortOrder = 'DESC' THEN [p].[CreatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'CreatedAt' THEN [p].[CreatedAt] END,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' AND @SortOrder = 'DESC' THEN [p].[UpdatedAt] END DESC,`
		`CASE WHEN @SortOrderName = 'UpdatedAt' THEN [p].[UpdatedAt] END`
		`OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY`
		`END`
		`END`
			Thực hiện lấy tất cả các bài viết (Posts) nếu giá trị tìm kiếm (SearchValue) bằng null (chưa nhập hoặc rỗng) thì lấy tất cả Posts đồng thời phân trang và sắp xếp tăng dần giảm dần theo tên attribute

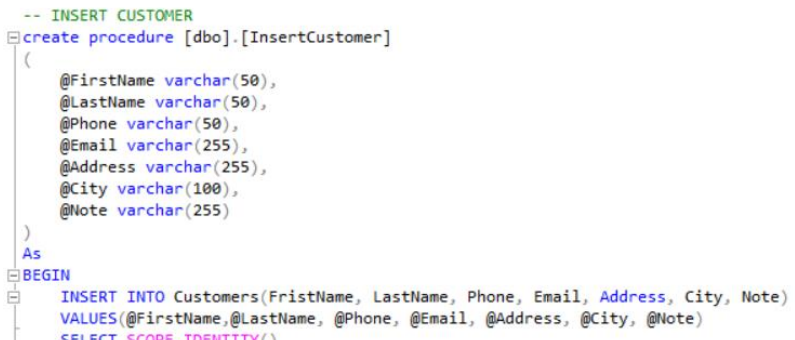
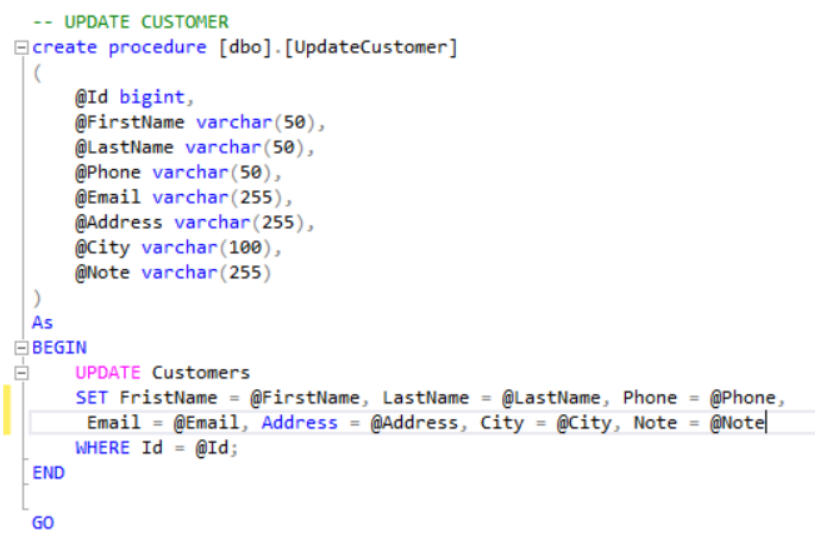
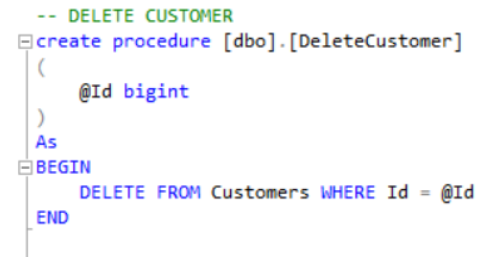
2	GetPost	
		`-- GET POST BY ID`
		`create procedure [dbo].[GetPost]`
		`(`
		`@Id bigint`
		`)`
		`As`
		`BEGIN`
		`SELECT TOP(1) * FROM [dbo].[Pots] as p`
		`INNER JOIN [dbo].[Categories] as c`
		`ON [p].[CategoryId] = c.Id`
		`WHERE [p].[Id] = @Id`
		`END`
		`GO`
			Lấy thông tin của Post khi click vào, nhận Id của post để truy xuất thông tin

3	InsertPost	-- INSERT POST create procedure [dbo].[InsertPost] (@Title varchar(255), @Slug varchar(255), @ShortDescription varchar(255), @Content text, @Image varchar(255), @Visibility bit, @CategoryId bigint, @CreatedAt datetime, @UpdatedAt datetime) As BEGIN INSERT INTO [dbo].[Pots] (Title, Slug, ShortDescription, Content, Image, Visibility, CategoryId, CreatedAt, UpdatedAt) VALUES(@Title, @Slug, @ShortDescription, @Content, @Image, @Visibility, @CategoryId, @CreatedAt, @UpdatedAt) END GO	Thêm một bài viết (Post)
4	UpdatePost	-- UPDATE POST create procedure [dbo].[UpdatePost] (@Id bigint, @Title varchar(255), @Slug varchar(255), @ShortDescription varchar(255), @Content text, @Image varchar(255), @Visibility bit, @CategoryId bigint, @CreatedAt datetime, @UpdatedAt datetime) As BEGIN UPDATE [dbo].[Pots] SET Title = @Title, Slug = @Slug, ShortDescription = @ShortDescription, Content = @Content, Image = @Image, Visibility = @Visibility, CategoryId = @CategoryId, CreatedAt = @CreatedAt, UpdatedAt = @UpdatedAt WHERE Id = @Id; END GO	Sửa thông tin một bài viết(Post)
5	DeletePost	-- DELETE POST create procedure [dbo].[DeletePost] (@Id bigint) As BEGIN DELETE FROM [dbo].[Pots] WHERE Id = @Id END	Xóa một bài viết (Post) có Id được truyền vào

Bảng 21 Store Procedure của Posts

7.5. Customer

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	InsertCustomer		Thêm một khách hàng
2	UpdateCustomer		Sửa thông tin một khách hàng
3	DeleteCustomer		Xóa một khách hàng có Id được truyền vào

Bảng 22 Store Procedure của Customer

7.6. Order

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	InsertOrder	`-- INSERT ORDER`
		`create procedure [dbo].[InsertOrder]`
		`(`
		`@OrderDate datetime,`
		`@CustomerId bigint`
		`)`
		`As`
		`BEGIN`
		`INSERT INTO Orders(OrderDate, CustomerId)`
		`VALUES(@OrderDate,@CustomerId)`
		`SELECT SCOPE_IDENTITY()`
		`END`
		`GO`
			Thêm một đơn đặt hàng

2	UpdateOrder	`-- UPDATE ORDER`
		`create procedure [dbo].[UpdateOrder]`
		`(`
		`@Id bigint,`
		`@OrderDate datetime,`
		`@CustomerId bigint`
		`)`
		`As`
		`BEGIN`
		`UPDATE Orders`
		`SET OrderDate = @OrderDate, CustomerId = @CustomerId`
		`WHERE Id = @Id;`
		`END`
		`GO`
			Sửa thông tin một đơn đặt hàng

3	DeleteOrder	`-- DELETE ORDER`
		`create procedure [dbo].[DeleteOrder]`
		`(`
		`@Id bigint`
		`)`
		`As`
		`BEGIN`
		`DELETE FROM Orders WHERE Id = @Id`
		`END`
			Xóa một đơn đặt hàng có Id được truyền vào

Bảng 23 Store Procedure của Order

7.7. OrderDetail

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	InsertOrderDetail	```
		-- INSERT ORDER DETAIL
		create procedure [dbo].[InsertOrderDetail]
		(
		@OrderId bigint,
		@ProductId bigint,
		@Quantity int
		)
		As
		BEGIN
		
		INSERT INTO OrderDetails(OrderId, ProductId,Quantity)
		VALUES(@OrderId, @ProductId,@Quantity)
		
		END
		``
			Thêm một đơn đặt hàng chi tiết

2	GetOrderDetail	```
		create procedure [dbo].[GetOrderDetail]
		(
		@OrderId bigint
		)
		As
		BEGIN
		select *, [dbo].[func_CalculatePriceQuantityProduct]([c].[Quantity],
		[c].[Price]) as TotalPriceProduct
		FROM [dbo].[view_CustomerOrderDetails] as c
		WHERE [c].[OrderId] = @OrderId
		END
		
		GO
		``
			Lấy thông tin chi tiết của đơn đặt hàng

Bảng 24 Store Procedure của OrderDetail

7.8. CustomerOrders

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	GetCustomerOrders	```
		create procedure [dbo].[GetCustomerOrders]
		(
		@SearchValue varchar(255) = null,
		@SortOrderName varchar(50),
		@SortOrder varchar(4),
		@CurrentPage int,
		@PageSize int
		)
		As
		BEGIN
		IF(@SearchValue IS NULL)
		BEGIN
		SELECT * FROM [dbo].[view_CustomerOrders] as c
		ORDER BY
		CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [c].[OrderId] END DESC,
		CASE WHEN @SortOrderName = 'Id' THEN [c].[OrderId] END,
		CASE WHEN @SortOrderName = 'FristName' AND @SortOrder = 'DESC' THEN [c].[FristName] END DESC,
		CASE WHEN @SortOrderName = 'FristName' THEN [c].[FristName] END,
		CASE WHEN @SortOrderName = 'LastName' AND @SortOrder = 'DESC' THEN [c].[LastName] END DESC,
		CASE WHEN @SortOrderName = 'LastName' THEN [c].[LastName] END,
		CASE WHEN @SortOrderName = 'Email' AND @SortOrder = 'DESC' THEN [c].[Email] END DESC,
		CASE WHEN @SortOrderName = 'Email' THEN [c].[Email] END,
		CASE WHEN @SortOrderName = 'City' AND @SortOrder = 'DESC' THEN [c].[City] END DESC,
		CASE WHEN @SortOrderName = 'City' THEN [c].[City] END,
		CASE WHEN @SortOrderName = 'OrderDate' AND @SortOrder = 'DESC' THEN [c].[OrderDate] END DESC,
		CASE WHEN @SortOrderName = 'OrderDate' THEN [c].[OrderDate] END
		OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY
		END
		ELSE
		BEGIN
		SELECT * FROM [dbo].[view_CustomerOrders] as c
		WHERE CHARINDEX(@SearchValue, [c].[FristName]) > 0
		ORDER BY
		CASE WHEN @SortOrderName = 'Id' AND @SortOrder = 'DESC' THEN [c].[OrderId] END DESC,
		CASE WHEN @SortOrderName = 'Id' THEN [c].[OrderId] END,
		CASE WHEN @SortOrderName = 'FristName' AND @SortOrder = 'DESC' THEN [c].[FristName] END DESC,
		CASE WHEN @SortOrderName = 'FristName' THEN [c].[FristName] END,
		CASE WHEN @SortOrderName = 'LastName' AND @SortOrder = 'DESC' THEN [c].[LastName] END DESC,
		CASE WHEN @SortOrderName = 'LastName' THEN [c].[LastName] END,
		CASE WHEN @SortOrderName = 'Email' AND @SortOrder = 'DESC' THEN [c].[Email] END DESC,
		CASE WHEN @SortOrderName = 'Email' THEN [c].[Email] END,
		CASE WHEN @SortOrderName = 'City' AND @SortOrder = 'DESC' THEN [c].[City] END DESC,
		CASE WHEN @SortOrderName = 'City' THEN [c].[City] END,
		CASE WHEN @SortOrderName = 'OrderDate' AND @SortOrder = 'DESC' THEN [c].[OrderDate] END DESC,
		CASE WHEN @SortOrderName = 'OrderDate' THEN [c].[OrderDate] END
		OFFSET ((@CurrentPage-1) * @PageSize) ROWS FETCH NEXT @PageSize ROWS ONLY
		END
		END
		``
			Lấy danh sách đơn hàng kèm thông tin khách hàng (có thể search được tên khách hàng) đồng thời phân trang và sắp xếp tăng dần giảm dần theo tên attribute

Bảng 25 Store Procedure của CustomerOrders

7.9. Count (Dashboard)

Dù phần này cũng thuộc các bảng trên nhưng sẽ được viết riêng ra vì chức năng của chúng để tính toán tổng các sản phẩm, hiển thị ra Dashboard.

TT	Tên Store Procedure	Tạo Store Procedure	Mục đích

1	GetCatalogCount	```
		-- ** Catalog
		create procedure [dbo].[GetCatalogCount]
		As
		BEGIN
		SELECT COUNT(*) as Total FROM [dbo].[Catalogs]
		END
		
		GO
		``
			Đếm số lượng Catalog

2	GetProductCount	`-- ** Product`
		`create procedure [dbo].[GetProductCount]`
		`As`
		`BEGIN`
		`SELECT COUNT(*) as Total FROM [dbo].[Products]`
		`END`
		`GO`
			Đếm số lượng sản phẩm hiện có

3	GetCategoryCount	`-- ** Category`
		`create procedure [dbo].[GetCategoryCount]`
		`As`
		`BEGIN`
		`SELECT COUNT(*) as Total FROM [dbo].[Categories]`
		`END`
		`GO`
			Đếm số lượng danh mục sản phẩm

4	GetPostCount	`-- ** Post`
		`create procedure [dbo].[GetPostCount]`
		`As`
		`BEGIN`
		`SELECT COUNT(*) as Total FROM [dbo].[Pots]`
		`END`
		`GO`
			Đếm số lượng bài viết hiện có

5	GetOrderCount	`-- ** Order`
		`create procedure [dbo].[GetOrderCount]`
		`As`
		`BEGIN`
		`SELECT COUNT(*) as Total FROM [dbo].[Orders]`
		`END`
		`GO`
			Đếm số lượng Order đang có

6	GetTotalIncome	`-- ** Income`
		`create procedure [dbo].[GetTotalIncome]`
		`As`
		`BEGIN`
		`SELECT TotalIncome FROM [dbo].[func_TotalIncome]()`
		`END`
			Tính tổng thu nhập

Bảng 26 Store Procedure của Count (Dashboard)

8. *Thiết kế Trigger*

8.1. Product

TT	Tên Trigger	Tạo Trigger	Mục đích

1	Trig_CreateProduct	`-- TRIG CREATE PRODUCT`
		`CREATE TRIGGER trig_CreateProduct`
		`ON [dbo].[Products] AFTER INSERT AS`
		`DECLARE @CatalogId AS BIGINT`
		`BEGIN`
		`  SET @CatalogId = (SELECT CatalogId FROM inserted)`
		
		`  UPDATE [dbo].[Catalogs]`
		`  SET [Catalogs].[ProductCount] = [Catalogs].[ProductCount] + 1`
		`  WHERE [Catalogs].[Id] = @CatalogId`
		`END`
			Khi thêm mới 1 sản phẩm (product) => Cập nhật Catalogs (cộng 1 vào Product Count)

2	Trig_UpdateProduct	`-- TRIG UPDATE PRODUCT`
		`CREATE TRIGGER trig_UpdateProduct`
		`ON [dbo].[Products] AFTER UPDATE AS`
		`DECLARE @Price AS DECIMAL`
		`SET @Price = (SELECT TOP 1 Price FROM [dbo].[Products] WHERE [Products].[Price] <= 0)`
		`IF @Price < 0`
		`BEGIN`
		`  RAISERROR('Cannot update price < 0', 16, 10);`
		`  ROLLBACK`
		`END`
		
		`-- TEST TRIG UPDATE PRODUCT`
		`--UPDATE Products`
		`--SET Price = -2.3`
		`--WHERE Id = 2`
		
		`GO`
			Validate khi sản phẩm có giá dưới <=0 thì báo lỗi không thể cập nhật giá

3	Trig_DeleteProduct	`-- TRIG DELETE PRODUCT`
		`CREATE TRIGGER [dbo].[trig_DeleteProduct]`
		`ON [dbo].[Products] AFTER DELETE AS`
		`DECLARE @CatalogId AS BIGINT`
		`BEGIN`
		`  SET @CatalogId = (SELECT CatalogId FROM deleted)`
		
		`  UPDATE [dbo].[Catalogs]`
		`  SET [Catalogs].[ProductCount] = [Catalogs].[ProductCount] - 1`
		`  WHERE [Catalogs].[Id] = @CatalogId AND [Catalogs].[ProductCount] > 0`
		`END`
			Khi xóa 1 sản phẩm (product) => Cập nhật Catalogs (trừ 1 vào Product Count)

Bảng 27 Trigger của Product

8.2. Post

TT	Tên Trigger	Tạo Trigger	Mục đích

1	Trig_CreatePost	`-- TRIG CREATE POST`
		`CREATE TRIGGER [dbo].[trig_CreatePost]`
		`ON [dbo].[Pots] AFTER INSERT AS`
		`DECLARE @CategoryId AS BIGINT`
		`BEGIN`
		`  SET @CategoryId = (SELECT CategoryId FROM inserted)`
		
		`  UPDATE [dbo].[Categories]`
		`  SET [Categories].[PostCount] = [Categories].[PostCount] + 1`
		`  WHERE [Categories].[Id] = @CategoryId`
		`END`
		
		`GO`
			Khi thêm mới 1 bài viết (post) => Cập nhật Categories (cộng 1 vào PostCount)

2	Trig_DeletePost	`-- TRIG DELETE POST`
		`CREATE TRIGGER [dbo].[trig_DeletePost]`
		`ON [dbo].[Pots] AFTER DELETE AS`
		`DECLARE @CategoryId AS BIGINT`
		`BEGIN`
		`  SET @CategoryId = (SELECT CategoryId FROM deleted)`
		
		`  UPDATE [dbo].[Categories]`
		`  SET [Categories].[PostCount] = [Categories].[PostCount] - 1`
		`  WHERE [Categories].[Id] = @CategoryId AND [Categories].[PostCount] > 0`
		`END`
			Khi thêm mới 1 bài viết (post) => Cập nhật Categories (cộng 1 vào PostCount)

Bảng 28 Trigger của Post

9. *Seed Data*

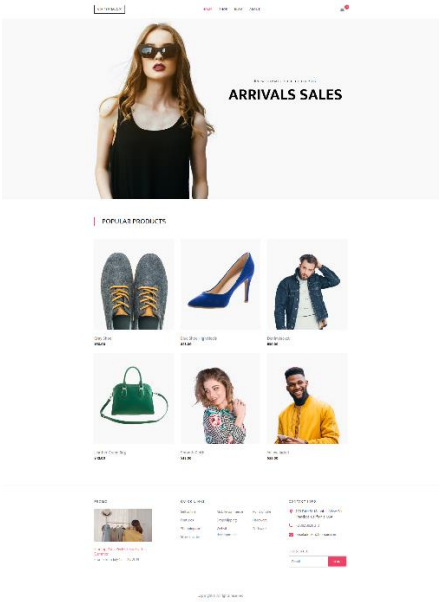
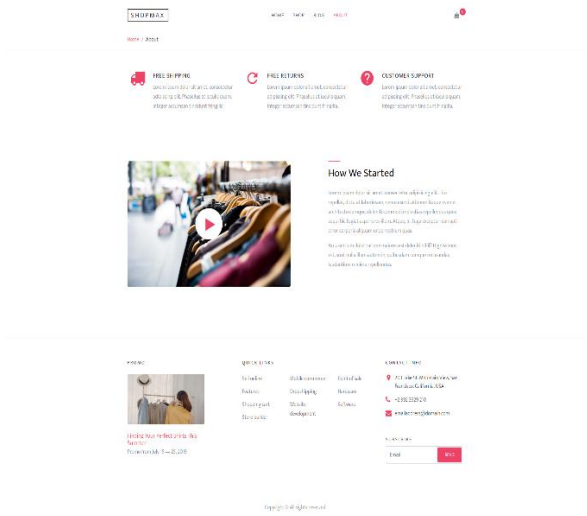
Mục đích của Seed data là khi run web sẽ có các dữ liệu sẵn trong database, chủ yếu phục vụ cho việc demo hiển thị được đầy đủ chức năng nhất của web.

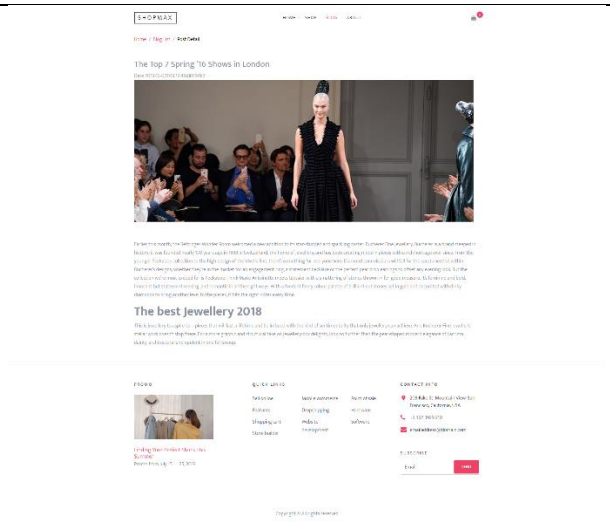
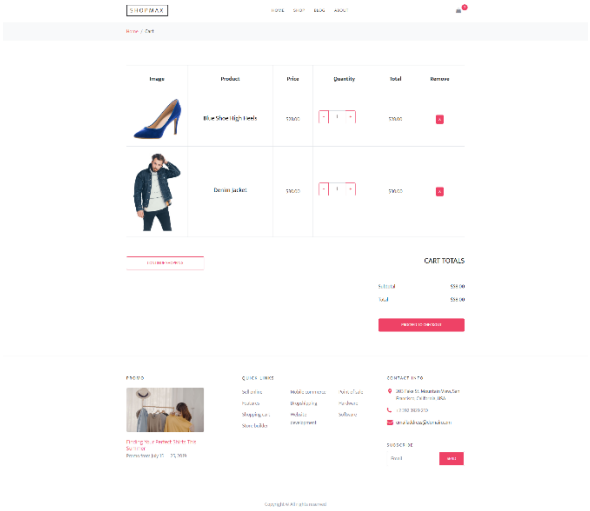
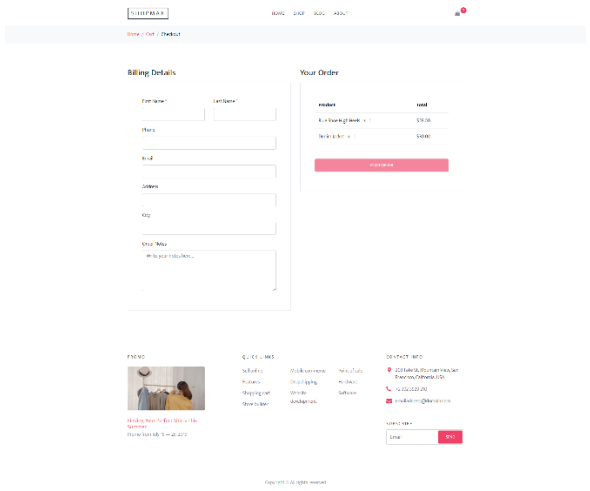
```
-- INSERT CATALOG
EXEC [dbo].[InsertCatalog] @Name="Men", @Slug="men", @Visibility=true,
 @ProductCount=0, @CreatedAt="2019-12-19T22:22:40", @UpdatedAt="2019-12-19T22:22:40"
EXEC [dbo].[InsertCatalog] @Name="Women", @Slug="wonmen", @Visibility=true,
 @ProductCount=0, @CreatedAt="2019-12-19T22:23:40", @UpdatedAt="2019-12-19T22:23:40"
EXEC [dbo].[InsertCatalog] @Name="Children", @Slug="children", @Visibility=true,
 @ProductCount=0, @CreatedAt="2019-12-19T22:24:40", @UpdatedAt="2019-12-19T22:24:40"

-- INSERT PRODUCT
EXEC [dbo].[InsertProduct] @Name="Product 01", @Slug="product-01",
 @ShortDescription="Lorem ipsum dolor sit amette, velit aperiam quis.",
  @Description="Lorem ipsum dolor sit amette, velit aperiam quis.", @Price=24,
   @Image="product-01.png", @Visibility=true, @CatalogId=1, @CreatedAt="2019-12-19T22:22:40",
    @UpdatedAt="2019-12-19T22:22:40"


EXEC [dbo].[InsertProduct] @Name="Product 02", @Slug="product-02",
 @ShortDescription="Lorem ipsum dolor sit amette, velit aperiam quis.",
  @Description="Lorem ipsum dolor sit amette, velit aperiam quis.", @Price=32,
   @Image="product-01.png", @Visibility=true, @CatalogId=2, @CreatedAt="2019-12-19T22:23:40",
    @UpdatedAt="2019-12-19T22:23:40"

EXEC [dbo].[InsertProduct] @Name="Product 03", @Slug="product-03",
 @ShortDescription="Lorem ipsum dolor sit amette, velit aperiam quis.",
  @Description="Lorem ipsum dolor sit amette, velit aperiam quis.", @Price=58,
  @Image="product-01.png", @Visibility=true, @CatalogId=3, @CreatedAt="2019-12-19T22:24:40",
   @UpdatedAt="2019-12-19T22:24:40"
```

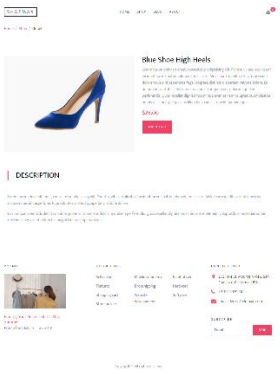
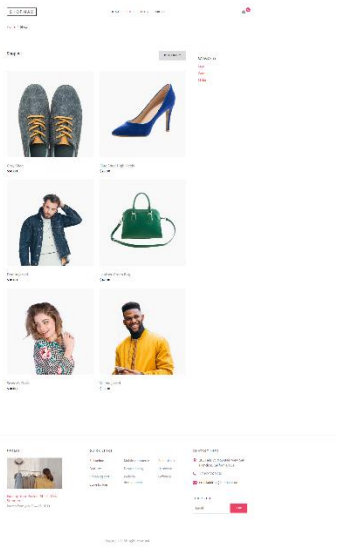
Hình 4 SeedData

Cách thức SeedData tương đối đơn giản, chỉ cần EXEC insert các phần muốn seed data như hình 2.

10. Thiết kế giao diện

TT	Màn hình	Mục đích	Giải thích
1	Home 	Giao diện tổng quan sản phẩm	Thiết kế: Nguyễn Xuân Điền Người dùng xem các sản phẩm nổi bật, hoặc các danh mục cần tìm
2	About 	Hiện thêm thông tin về cửa hàng	Thiết kế: Nguyễn Xuân Điền Người dùng có thể xem chi tiết các thông tin về cửa hàng
3	Blog	Mô tả sản phẩm	-Thiết kế: Nguyễn Xuân Điền Người dùng có thể truy cập để xem các thông tin về sản

			phẩm, xu hướng thời trang và các sản phẩm nổi bật của cửa hàng
4	Cart 	Giỏ hàng khi khách hàng mua hàng	-Thiết kế: Nguyễn Xuân Điền Khi khách hàng nhấn vào mua hàng, hàng sẽ được chuyển vào giỏ, trong giỏ sẽ hiển thị các thông tin như trên
5	Check out 	Hiển thị form nhập thông tin người mua	-Thiết kế: Nguyễn Xuân Điền Khách hàng điền vào thông tin để mua hàng, sau này sẽ phát triển sử dụng các ưu đãi khách hàng

6	Details 	Hiển thị thông tin chi tiết của sản phẩm	-Thiết kế: Nguyễn Xuân Điền Khách hàng xem thông tin chi tiết của sản phẩm để quyết định có mua hay không, nếu mua thì click "Add to cart"
7	Shop all 	Hiển thị toàn bộ các sản phẩm của shop	-Thiết kế: Nguyễn Xuân Điền Khách hàng xem các sản phẩm của cửa hàng, xem thích sản phẩm nào để quyết định chọn mua

Bảng 29 Thiết kế giao diện

Chương 4: Cài đặt và kiểm thử

1. Cài đặt

INSTALL & SETUP

Step 01: Download & Install nodejs

⇨ https://nodejs.org/dist/v12.13.1/node-v12.13.1-x64.msi

Step 02: (ASP.NET CORE)

- Open command line(cmd)
- Write: cd backend => enter

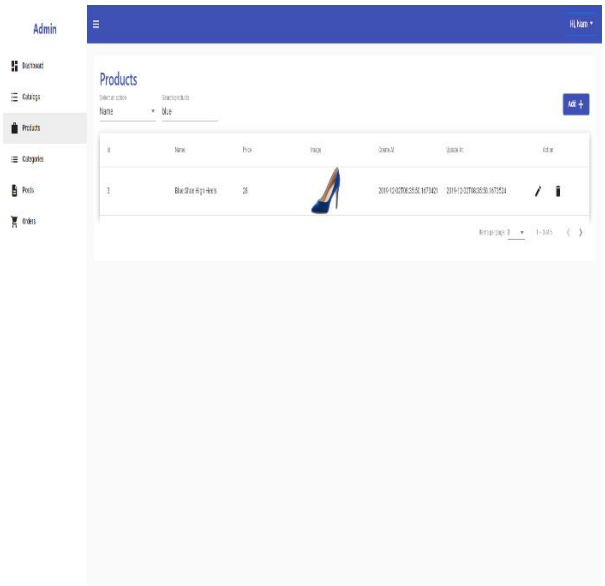
- Write: dotnet watch run => enter

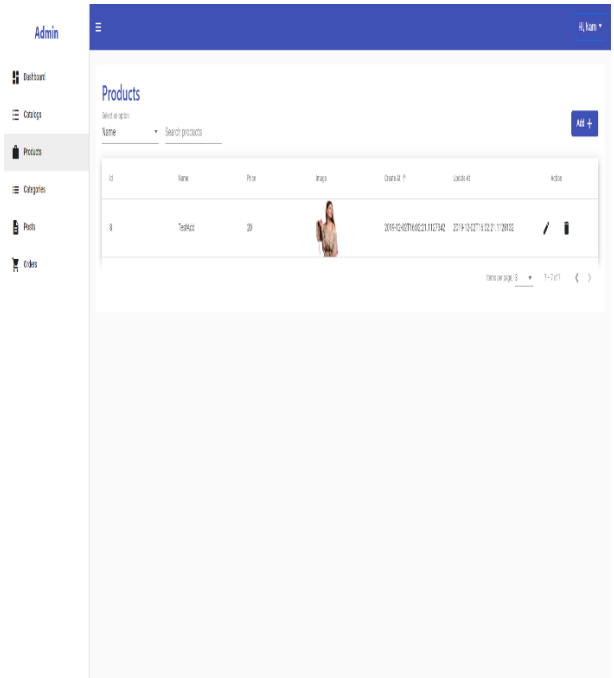
Step 03: (Angular):

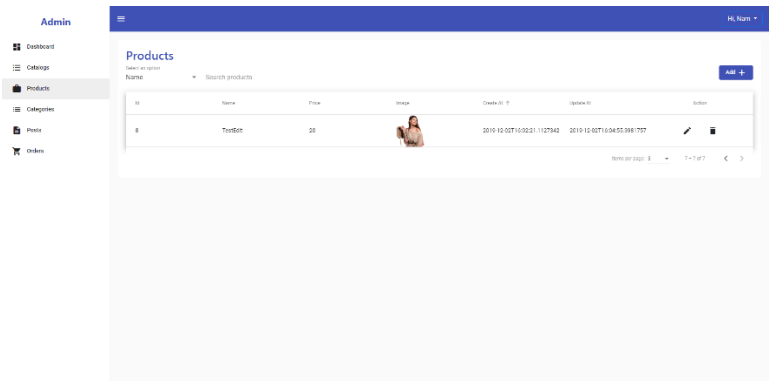
- Open command line(cmd)
- Write: cd frontend => enter
- Write: npm install => enter
- Write: npm start => enter

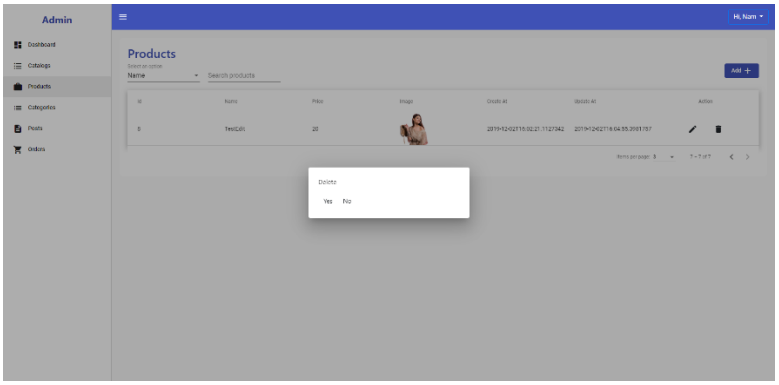
⇨ **Cuối cùng Truy cập => http://localhost:4200/**

2. Kiểm thử

TT	Tình huống	Mục đích	Giải thích	Thực tế
1	Input: “blue”	Kiểm tra chức năng tìm kiếm sản phẩm	Tìm kiếm sản phẩm có chứa “blue” trong tên	

2	Input:
	{
	Name: TestAdd;
	ShortDescription: Test;
	Price: 20;
	Image: (hình)
	Select an catalogs: Women;
	Description: test
	}
		Kiểm tra chức năng thêm	Kiểm tra xem thêm sản phẩm có được hay không	

3	Input (như test case 2) { Name: TestEdit;	Kiểm tra chức năng sửa	Kiểm tra xem khi sửa một sản phẩm có bị lỗi hay không	

	}			
4		Kiểm tra chức năng xóa sản phẩm	Kiểm tra xem có thể xóa một sản phẩm đã chọn hay không	

Bảng 30 Kiểm thử web

Chương 5: Kết luận và hướng phát triển

1. Kết luận

Về cơ bản, nhóm tự nhận xét phần mềm của nhóm đã giải quyết được được hơn 90% yêu cầu mà đồ án đặt đặt ra. Sau đây là ưu điểm cũng như tồn tại của phần mềm

- Ưu điểm:
  - Đã áp dụng các kĩ thuật trong yêu cầu đặt ra
  - Giao diện gọn gàng, dễ tiếp cận, dễ làm quen.
  - Ứng dụng single page application, tối ưu tốc độ load trang
  - Chương trình chạy ổn định cho tất cả các loại trình duyệt, cho ra kết quả chính xác, không bị exception trong quá trình thực thi yêu cầu người dùng.
- Nhược điểm:
  - Database còn đơn giản, chưa giải quyết được các yêu cầu trong thực tế.
  - Chưa ứng dụng unittest cho chương trình.

2. *Hướng phát triển*

- Cải thiện giao diện đẹp và bắt mắt hơn, thân thiện với người dùng hơn.
- Thêm các chức năng quản lý khuyến mãi, mã giảm giá, khách thân thiết,…
- Phân quyền chi tiết các đối tượng sử dụng.
- Thêm thống kê doanh thu, biểu đồ thu nhập năm, so sánh với thu nhập những năm gần đây.
- Cải thiện và nâng cấp database.

Tài liệu tham khảo

[1]. Tìm hiểu về Database View. – Tác giả: Lavender https://viblo.asia/p/tim-hieu-ve-database-view-RQqKLx1pK7z (Ngày 22/12/2019)

[2]. Function người dùng tự định nghĩa trong SQL – Tác giả: ThaoDTD https://viblo.asia/p/function-nguoi-dung-tu-dinh-nghia-trong-sql-GrLZDva35k0 (Ngày 22/12/2019)

[3]. TÌM HIỂU VỀ STORED PROCEDURE TRONG MYSQL – Tác giả: Nguyen Manh Ky

https://viblo.asia/p/tim-hieu-ve-stored-procedure-trong-mysql-157G5npBvAje (Ngày 22/12/2019)

[4]. Sử dụng trigger trong SQL qua ví dụ cơ bản. – Tác giả: Phạm Anh Tuấn

https://viblo.asia/p/su-dung-trigger-trong-sql-qua-vi-du-co-ban-aWj538APK6m (Ngày 22/12/2019)

[5]. Angular – tutorial series – Tác giả: Đội ngũ Google https://angular.io/docs (Ngày 22/12/2019)