* + Chức năng cho quản trị viên:
    - Danh mục thành phố, quận/huyện, phường/xã
    - Quản lý phương thức thanh toán
    - Quản lý khách hàng
    - Quản lý nhân viên và phân quyền người dùng
    - Quản lý sản phẩm(danh mục loại sản phẩm, thương hiệu, tồn kho)
    - Quản lý đơn hàng của khách hàng
    - Quản lý thanh toán
    - Quản lý nhập hàng
    - Quản lý chương trình khuyến mãi
    - Quản lý banner
    - Quản lý tìm kiếm sản phẩm(sử dụng elasticsearch)
    - Quản lý đánh giá sản phẩm
    - Quản lý thông báo
    - Chăm sóc khách hàng(Chat trực tiếp)
    - Thống kê doanh thu, thống kê kho hàng, thống kê đơn nhập.
* Chức năng cho nhân viên:
  + - Quản lý khách hàng
    - Quản lý sản phẩm(danh mục loại sản phẩm, thương hiệu, tồn kho)
    - Quản lý đơn hàng của khách hàng
    - Quản lý thanh toán
    - Quản lý nhập hàng
    - Quản lý đánh giá sản phẩm
    - Chăm sóc khách hàng(Chat trực tiếp).
* Chức năng của khách hàng
  + Đăng ký, đăng nhập hệ thống, quản lý tài khoản
  + Xem thông tin sản phẩm, tìm kiếm sản phẩm(tìm kiếm nâng cao)
  + Quản lý giỏ hàng, mua hàng, thanh toán đơn hàng(có thanh toán online)
  + Quản lý đơn hàng(theo dõi đơn hàng, hủy đơn hàng)
  + Quản lý thông báo
  + Quản lý thông tin nhận hàng
  + Đánh giá sản phẩm
  + Quản lý sản phẩm yêu thích
  + Quản lý voucher

Việc cần làm

* Quản lý các danh muc: hiển thị, thêm, sửa, xóa

Danh mục thành phố, quận/huyện, phường/xã

Quản lý khách hàng

Quản lý nhà cung cấp

Quản lý nhân viên và phân quyền người dùng

danh mục loại sản phẩm, thương hiệu

Quản lý sản phẩm

Quản lý banner

Done api các phần trên

Công việc tiếp theo:

Quản lý tìm kiếm sản phẩm(sử dụng elasticsearch)

Quản lý phương thức thanh toán

Quản lý chương trình khuyến mãi

Thống kê doanh thu, thống kê kho hàng, thống kê đơn nhập.

Thống nhất giao diện

Navbar:

* Danh mục địa chỉ(cha), tỉnh/thành phố(con), quận/huyện(con),xã (con)
* Quản lý người dùng(cha), khách hàng(con), nhà cung cấp(con), nhân viên(con)
* Quản lý sản phẩm(cha), Danh mục sản phẩm(con), Thương hiệu(con), danh sách sản phẩm(con)
* Cài đặt(cha), Banner(con)

quy định chung

Hiển thị

BaseEntity: Id, culture, createddate, createdby, updateddate, updatedby, isActive, isDeleted, displayorder

Hiển thị dạng bảng: các trường khác trong bảng và BaseEntity(createddate, createdby, updateddate, updatedby)

Thêm, sửa

Chỉ cập nhật các trường trong bảng trừ entity, nếu có khóa ngoại thì cho input là select option(chọn)

Chức năng:

Hiển thị: dạng table, có paging, có filter, search

Thêm sửa: Thêm error ở mỗi trường, hiện thỉ thông báo server thì bằng toast

Backend api

Input phía api

1. GetData

provinceId::4::eq

RequestFilterModel: skip(int), take(int), filters(string), orderby(string), dir(string), type(RequestType), paging(bool), restrictOrderBy(bool), keywords(string), propertyFilterModels([{

"field": "string",

"operator": "string",

"filterValue": "string"

}])

1. Post

Property có trong bảng trừ BaseEntity

Output phía api

+ Bình thường

{

"result": [

{

}

],

"isSuccess": true,

"message": "Thành công",

"errors": null

}

+ Grid

“result”:{

subset: [],

count,

…..

}

A screenshot of a computer program

Description automatically generated

Công việc tiếp theo:

Lưu ý phiếu giảm giấ

Điều kiện giảm giấ

**1. Điều kiện giảm giá cho từng DiscountType**

| **DiscountType** | **Điều kiện áp dụng phổ biến** |
| --- | --- |
| **Percentage** | ✅ Đơn hàng tối thiểu (**MinOrderValue**). ✅ Mức giảm tối đa (**MaxDiscountAmount**). |
| **FixedAmount** | ✅ Đơn hàng tối thiểu (**MinOrderValue**). ✅ Không được giảm quá giá trị đơn hàng (**CannotExceedTotal**). |
| **BuyXGetY** | ✅ Phải mua ít nhất X sản phẩm (**RequiredQuantity**). ✅ Sản phẩm tặng phải có trong kho (**StockAvailability**). |
| **FreeShipping** | ✅ Áp dụng cho đơn hàng từ mức tối thiểu (**MinOrderValue**). ✅ Giới hạn khu vực vận chuyển (**AllowedRegions**). |
| **TieredDiscount** | ✅ Giảm giá theo cấp bậc (**TierLevels**). |
| **BundleDiscount** | ✅ Chỉ áp dụng khi mua combo sản phẩm (**RequiredProducts**). |

**🔹 2. Lưu điều kiện vào Database**

Dùng bảng VoucherConditions để lưu từng điều kiện.

sql

Sao chépChỉnh sửa

CREATE TABLE VoucherConditions (

Id INT PRIMARY KEY AUTO\_INCREMENT,

VoucherId INT NOT NULL,

ConditionType ENUM(

'MinOrderValue',

'MaxDiscountAmount',

'RequiredQuantity',

'AllowedRegions',

'TierLevels',

'AllowedPaymentMethods',

'MinPointsRequired',

'RequiredProducts'

),

ConditionValue VARCHAR(255) NOT NULL,

FOREIGN KEY (VoucherId) REFERENCES Vouchers(Id) ON DELETE CASCADE

);

**🔹 3. Ví dụ chi tiết từng loại DiscountType**

Dưới đây là **cách lưu điều kiện giảm giá** cho từng loại vào database.

**🔸 1. Percentage (Giảm theo phần trăm)**

sql

Sao chépChỉnh sửa

INSERT INTO VoucherConditions (VoucherId, ConditionType, ConditionValue)

VALUES (1, 'MinOrderValue', '500000'), -- Đơn hàng tối thiểu 500.000 VNĐ

(1, 'MaxDiscountAmount', '100000'); -- Giảm tối đa 100.000 VNĐ

**🔸 2. FixedAmount (Giảm số tiền cố định)**

sql

Sao chépChỉnh sửa

INSERT INTO VoucherConditions (VoucherId, ConditionType, ConditionValue)

VALUES (2, 'MinOrderValue', '200000'); -- Đơn hàng tối thiểu 200.000 VNĐ

**🔸 3. BuyXGetY (Mua X tặng Y)**

sql

Sao chépChỉnh sửa

INSERT INTO VoucherConditions (VoucherId, ConditionType, ConditionValue)

VALUES (3, 'RequiredQuantity', '2'); -- Mua ít nhất 2 sản phẩm

**🔸 4. FreeShipping (Miễn phí vận chuyển)**

sql

Sao chépChỉnh sửa

INSERT INTO VoucherConditions (VoucherId, ConditionType, ConditionValue)

VALUES (4, 'MinOrderValue', '300000'), -- Đơn hàng tối thiểu 300.000 VNĐ

(4, 'AllowedRegions', 'Hanoi,HoChiMinh'); -- Chỉ áp dụng cho Hà Nội & HCM

**🔸 5. TieredDiscount (Giảm giá theo cấp bậc)**

sql

Sao chépChỉnh sửa

INSERT INTO VoucherConditions (VoucherId, ConditionType, ConditionValue)

VALUES (5, 'TierLevels', '2:5%,3:10%'); -- Mua 2 sản phẩm giảm 5%, mua 3 giảm 10%

**🔸 8. BundleDiscount (Giảm giá khi mua combo)**

sql

Sao chépChỉnh sửa

INSERT INTO VoucherConditions (VoucherId, ConditionType, ConditionValue)

VALUES (8, 'RequiredProducts', '101,102,103'); -- Chỉ áp dụng khi mua sản phẩm ID 101, 102, 103

**🔹 4. Kiểm tra điều kiện trong C#**

Sau khi lưu điều kiện vào database, bạn có thể kiểm tra chúng như sau:

csharp

Sao chépChỉnh sửa

public bool ValidateVoucher(Order order, Voucher voucher)

{

var conditions = \_dbContext.VoucherConditions

.Where(vc => vc.VoucherId == voucher.Id)

.ToList();

foreach (var condition in conditions)

{

switch (condition.ConditionType)

{

case "MinOrderValue":

if (order.TotalPrice < Convert.ToDecimal(condition.ConditionValue))

return false;

break;

case "MaxDiscountAmount":

if (voucher.DiscountType == DiscountType.Percentage)

{

decimal maxDiscount = Convert.ToDecimal(condition.ConditionValue);

decimal discountAmount = (order.TotalPrice \* voucher.Value) / 100;

if (discountAmount > maxDiscount) return false;

}

break;

case "RequiredQuantity":

if (order.TotalQuantity < Convert.ToInt32(condition.ConditionValue))

return false;

break;

case "AllowedRegions":

var allowedRegions = condition.ConditionValue.Split(',');

if (!allowedRegions.Contains(order.ShippingRegion))

return false;

break;

case "MinPointsRequired":

if (order.Customer.Points < Convert.ToInt32(condition.ConditionValue))

return false;

break;

case "RequiredProducts":

var requiredProducts = condition.ConditionValue.Split(',').Select(int.Parse);

if (!order.Items.Any(item => requiredProducts.Contains(item.ProductId)))

return false;

break;

}

}

return true;

}