Microsoft.EntityFrameworkCore.Proxies

Microsoft.EntityFrameworkCore.Tools

Pomelo.EntityFrameworkCore.MySql

APP.XAML.CS

public static tradeContext context { get; } = new tradeContext();

APP.XAMl

StartupUri="Windows/Products.xaml">

ПАПКИ:

Classes

Class(Alldata):

-public static int ID;

public static int all;

-Class(Navigation):

public static Windows.Authorization auth;

public static string Atricle;

public static Windows.Products products

(CLASS, MODELS) USERS

public string FIO //дополнительное поле для записи ФИО в одном

{

get

{

return $"{UserSurname} { UserName} { UserPatronymic}";

}

}

public string dolgnost //дополнительное поле для записи должности

{

get

{

Roles role = App.context.Roles.ToList().Find(u => u.RoleId == UserRole);

return $"{role.RoleName}";

}

}

Pictures

UserControls

ProductCard

d:DesignHeight="250" d:DesignWidth="1300" FontSize="20" FontFamily="Comic Sans MS">

<Grid Height="250" Width="1300">

<Grid Background="Black" Height="240" Width="1290">

<Grid Background="White" Height="230" Width="1280">

<Label x:Name="SaleColor" Height="0" Background="#FFCC6600"/>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Orientation="Horizontal">

<Grid Width="320" Height="220" Background="Black">

<Grid Width="310" Height="210" Background="White">

<Image x:Name="Imgs" Width="400" Height="200" Source="/Pictures/picture.png"/>

</Grid>

</Grid>

<Label/>

<Grid Height="220" Width="700" Background="Black">

<Grid Height="210" Width="690" Background="White">

<StackPanel Width="680" Height="200">

<TextBlock x:Name="Title" FontWeight="Bold">Название</TextBlock>

<Label/>

<TextBlock x:Name="Description" Width="680" TextWrapping="Wrap" Height="90">Описание</TextBlock>

<Label/>

<TextBlock x:Name="Manufacturer">Производитель:</TextBlock>

<Label/>

<StackPanel Orientation="Horizontal">

<TextBlock>Цена:</TextBlock>

<TextBlock x:Name="Cost"/>

<Label/>

<TextBlock x:Name="CostDiscount" Visibility="Hidden"/>

</StackPanel>

</StackPanel>

</Grid>

</Grid>

<Label/>

<Grid Height="150" Width="200" Background="Black">

<Grid Height="140" Width="190" Background="White" x:Name="BackgroundSale">

<TextBlock x:Name="Stock" VerticalAlignment="Center" HorizontalAlignment="Right" TextWrapping="Wrap" Width="180"></TextBlock>

</Grid>

</Grid>

</StackPanel>

<TextBlock x:Name="Strike" TextDecorations="Strikethrough" Width="0"/>

</Grid>

</Grid>

</Grid>

</UserControl>

public partial class ProductCard : UserControl

{

string Article;

public ProductCard(string Atricle, string Title, string Description, string Manufacturer, string Cost, string Stock, decimal Sale, byte[] Img) //поля которые будут вызываться при использовании вызове

{

InitializeComponent();

this.Article = Atricle; //артикул

this.Title.Text = Title; //название

this.Description.Text = Description; //описание

Manufacturer manufacturers = App.context.Manufacturers.ToList().Find(m => m.ManufacturerId.ToString() == Manufacturer); //поиск производителя в бд

this.Manufacturer.Text += manufacturers.ManufacturerName; //производитель

this.Cost.Text += Cost.Split(',')[0] + " р."; //стоимость

this.Stock.Text = Stock + "%"; //скидка

if (Sale > 15)

BackgroundSale.Background = SaleColor.Background;

if (Sale > 0)

{

CostDiscount.Visibility = Visibility.Visible;

this.Cost.TextDecorations = Strike.TextDecorations;

CostDiscount.Text += (Convert.ToDecimal(Cost) - Convert.ToDecimal(Cost) \* Sale / 100).ToString().Split(',')[0] + " р."; //рассчёт стоимости с учётом скидки

}

try //сжатие и показ картинки

{

BitmapImage img = new BitmapImage();

MemoryStream MS = new MemoryStream(Img);

img.BeginInit();

img.StreamSource = MS;

img.EndInit();

this.Imgs.Source = img;

}

catch

{

}

}

}

}

WINDOWS

Products

Title="Продукты" MinHeight="820" MinWidth="1400" FontSize="20" FontFamily="Comic Sans MS" WindowStartupLocation="CenterScreen">

<Grid>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Height="784">

<Label x:Name="name" Content="" HorizontalAlignment="Right"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock VerticalAlignment="Center">Поиск:</TextBlock>

<Label/>

<TextBox x:Name="Search" Width="300" TextChanged="Search\_TextChanged"/>

<Label/>

<TextBlock VerticalAlignment="Center">Фильтрация:</TextBlock>

<Label/>

<ComboBox x:Name="Filtration" Width="200" DropDownClosed="Filtration\_DropDownClosed">

<ComboBoxItem>Все диапазоны</ComboBoxItem>

<ComboBoxItem>0-9.99%</ComboBoxItem>

<ComboBoxItem>10-14.99%</ComboBoxItem>

<ComboBoxItem>15% и больше</ComboBoxItem>

</ComboBox>

<Label/>

<TextBlock VerticalAlignment="Center">Сортировка:</TextBlock>

<Label/>

<StackPanel>

<RadioButton x:Name="Asc" Checked="Asc\_Checked">По возрастанию</RadioButton>

<RadioButton x:Name="Desc" Checked="Desc\_Checked">По убыванию</RadioButton>

</StackPanel>

</StackPanel>

<Label/>

<ScrollViewer Height="570" Width="1400">

<StackPanel x:Name="ProductView"/>

</ScrollViewer>

<Label/>

<TextBlock x:Name="All" Height="30" TextAlignment="Center"/>

<Button x:Name="exit" Background="#FFCC6600" FontFamily="Comic Sans MS" Click="exit\_Click">Выход</Button>

</StackPanel>

</Grid>

</Window>

public partial class Products : Window

{

public Products()

{

InitializeComponent();

List<Models.Product> list = App.context.Products.ToList();

Classes.Alldata.all = list.Count;

LoadData();

}

private void Desc\_Checked(object sender, RoutedEventArgs e)

{

LoadData();

}

private void Asc\_Checked(object sender, RoutedEventArgs e)

{

LoadData();

}

private void Filtration\_DropDownClosed(object sender, EventArgs e)

{

LoadData();

}

private void Search\_TextChanged(object sender, TextChangedEventArgs e)

{

LoadData();

}

public void LoadData() //обновление окна при каких-то изменениях

{

ProductView.Children.Clear();

List<Models.Product> list = App.context.Products.ToList();

if (Filtration.SelectedItem != null)

{

switch (Filtration.SelectedIndex)

{

case 0:

list = App.context.Products.ToList();

break;

case 1:

list = list.Where(p => p.ProductDiscountAmount > 0 && p.ProductDiscountAmount < Convert.ToDecimal(9.99)).ToList();

break;

case 2:

list = list.Where(p => p.ProductDiscountAmount > 10 && p.ProductDiscountAmount < Convert.ToDecimal(14.99)).ToList();

break;

case 3:

list = list.Where(p => p.ProductDiscountAmount >= 15 && p.ProductDiscountAmount < 100).ToList();

break;

}

}

if (Search.Text.Length != 0)

list = list.Where(p => p.ProductName.Contains(Search.Text)).ToList();

if (Asc.IsChecked == true)

list = list.OrderBy(p => p.ProductCost).ToList();

else if (Desc.IsChecked == true)

list = list.OrderByDescending(p => p.ProductCost).ToList();

foreach (var item in list)

ProductView.Children.Add(new UserControls.ProductCard(item.ProductArticleNumber, item.ProductName, item.ProductDescription, item.ManufacturerId.ToString(), item.ProductCost.ToString(), item.ProductDiscountAmount.ToString(), Convert.ToDecimal(item.ProductDiscountAmount), item.ProductPhoto));

All.Text = list.Count.ToString() + " из " + Classes.Alldata.all.ToString();

}

private void exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

172.17.4.11

<http://172.17.4.11:3000/>

user\_20is\_11

4231

Scaffold-DbContext "Server=localhost;User=root;Password=1234;Database=trade" "Pomelo.EntityFrameworkCore.MySql" -OutputDir "Models"

create database user\_20is\_11;

use user\_20is\_11;

create table Roles

(

RoleID int primary key auto\_increment,

RoleName varchar(100) not null

);

create table Users

(

UserID int primary key auto\_increment,

UserSurname varchar(100) not null,

UserName varchar(100) not null,

UserPatronymic varchar(100) not null,

UserLogin text not null,

UserPassword text not null,

UserRole int not null,

foreign key (UserRole) references Roles(RoleID)

);

create table Category

(

CategoryID int primary key auto\_increment,

CategoryName text not null

);

create table Manufacturers

(

ManufacturerID int primary key auto\_increment,

ManufacturerName text not null

);

create table Units

(

UnitID int primary key auto\_increment,

UnitName text not null

);

create table Suppliers

(

SupplierID int primary key auto\_increment,

SupplierName text not null

);

create table PickUpPoints

(

PickUpPointID int primary key auto\_increment,

PickUpPointIndex text not null,

PickUpPointName text not null

);

create table Products

(

ProductArticleNumber nvarchar(100) primary key,

ProductName text not null,

UnitID int not null,

ProductCost decimal(19,2) not null,

MaxDiscountAmount decimal(19,2) not null,

ManufacturerID int not null,

SupplierID int not null,

CategoryID int not null,

ProductDiscountAmount decimal(19,2) null,

ProductQuantityInStock int not null,

ProductDescription text not null,

foreign key (CategoryID) references Category(CategoryID),

ProductPhoto longblob,

foreign key (ManufacturerID) references Manufacturers(ManufacturerID),

foreign key (SupplierID) references Suppliers(SupplierID),

foreign key (UnitID) references Units(UnitID)

);

# Название проекта \* ООО\_ФЛОРИСТ\_1\*

Магазин по продаже цветов и материалов для флористов

## Запуск приложения

Для запуска приложения переходим в каталог C:\Users\shara\source\repos\ООО\_ФЛОРИСТ\_1\ООО\_ФЛОРИСТ\_1\bin\Debug\netcoreapp3.0 и запускаем файл \*ООО\_ФЛОРИСТ\_1\*.ехе

## Средства разработки

- MySQL Workbench

- Visual Studio 2022

- C#

## Автор

\*ФАМИЛИЯ ИМЯ\*, 20ИС-1

# Название проекта \*ООО\_\*

Магазин по продаже цветов и материалов для флористов

## Запуск приложения

Для запуска приложения переходим в каталог \*Название проекта\*\bin\Debug и запускаем файл \*название проекта\*.ехе

## Средства разработки

- MySQL Workbench

- Visual Studio 2022

- C#

## Автор

\*ФАМИЛИЯ ИМЯ\* 20ИС-1