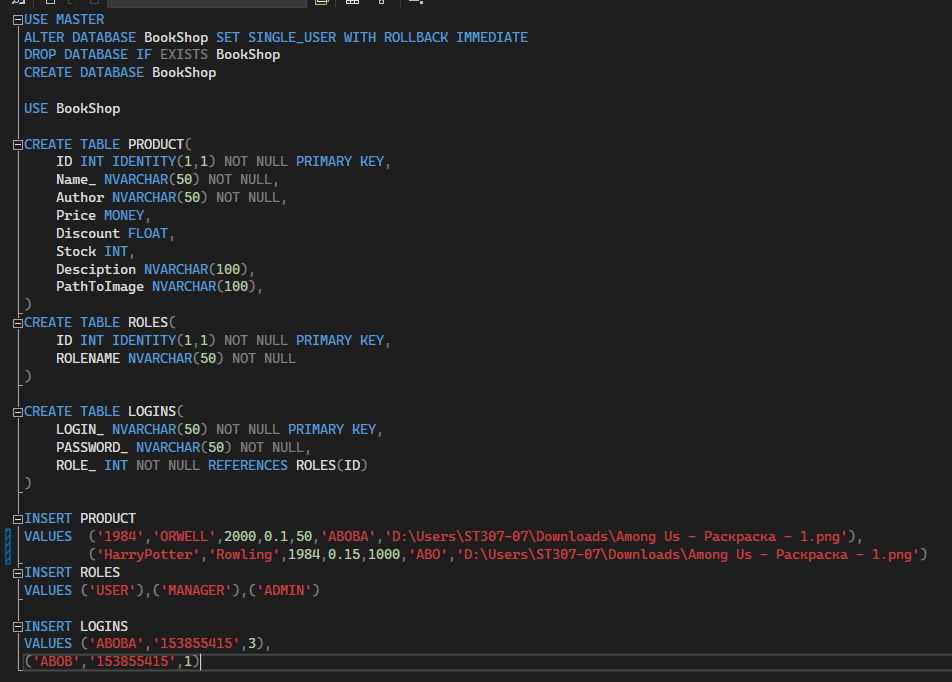
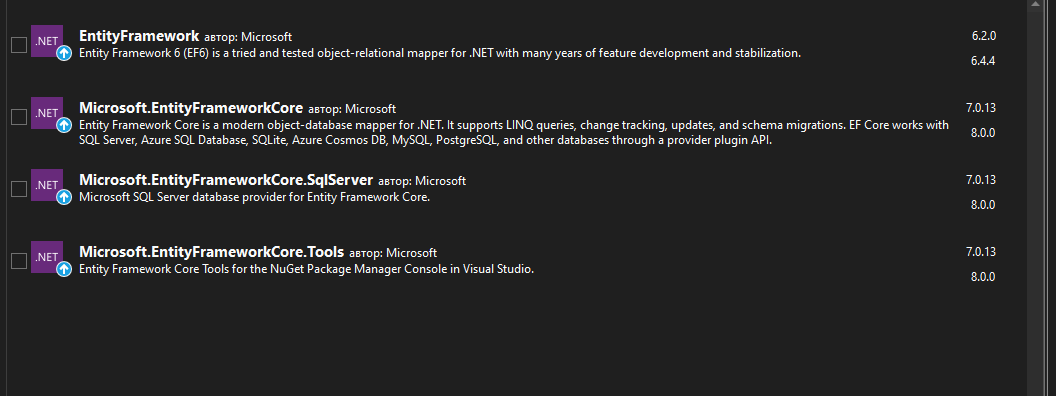
Создать бД запросом:



После этого установить через Nuget следующие пакеты



После этого в консоль диспетчера пакетов ввести команду

Scaffold-DbContext "Server=ИМЯСЕРВЕРА;Database=ИМЯБАЗЫДАННЫХ;Trusted\_Connection=True;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer

После создания контекста базы данных (файл ИМЯБАЗЫДАННЫХContext.cs) создать три окна – основное,окно авторизации и окно корзины, а так же 2 UserControl – один для товаров в каталоге, другой для товаров в корзине

UserControl для продукта в каталоге создается со следующим содержанием Grid(Xaml)

<Grid x:Name="MainGrid">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="1\*"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<Image x:Name="ProductImage" Grid.Column="0" Margin="5"/>

<Label x:Name="ProductNameLabel" Grid.Column="1" Margin="5"/>

<Label x:Name="ProductPriceLabel" Grid.Column="2" Margin="5"/>

<Button x:Name="ProductDeleteButton" Content="Удалить" Click="ProductDeleteButton\_Click" Grid.Column="4" Margin="5" IsEnabled="False"/>

<Button x:Name="ProductAddButton" Content="Добавить в корзину" Click="ProductAddButton\_Click" Grid.Column="3" Margin="5"/>

</Grid>

И следующим содержанием класса(.CS)

private Product ThisProduct;

private MainWindow EvokingWindow;

public ProductVisual(MainWindow MainWindow, int productId,bool admined)

{

InitializeComponent();

BookShopContext dbCont = new BookShopContext();

ThisProduct = dbCont.Products.Find(productId);//поиск в конткесте БД текущей записи товара по ID переданному в конструкторе

EvokingWindow = MainWindow;//окно которое вызвало UserControl

ProductImage.Source = new BitmapImage(new Uri(ThisProduct.PathToImage, UriKind.RelativeOrAbsolute));

ProductNameLabel.Content = ThisProduct.Name;

ProductPriceLabel.Content = ThisProduct.Price;

if (admined) {//создание дополнительной колонки для отображения кнопки удаления

ColumnDefinition col = new ColumnDefinition();

col.Width = new GridLength(0.5, GridUnitType.Star);

MainGrid.ColumnDefinitions.Add(col);

ProductDeleteButton.IsEnabled = true;

}

if (ThisProduct.Stock < 1)//если колво товара меньше 1 то товар нельзя добавить в корзину и он окраивается в серый

{

ProductAddButton.IsEnabled = false;

this.Background = Brushes.Gray;

}

}

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

{//кнопка добавления товара в корзину

if (EvokingWindow.Basket.ContainsKey(ThisProduct.Id))

EvokingWindow.Basket[ThisProduct.Id] += 1;

else EvokingWindow.Basket.Add(ThisProduct.Id, 1);

BasketWindow.UpdateBasket();//метод обновления корзины

EvokingWindow.TurnOnBasketButton();//включить кнопку корзины в основном окне

MessageBox.Show("Товар успешно добавлен!");

}

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

{//кнопка удлания для адмнистратора

BookShopContext db = new BookShopContext();

db.Products.Remove(ThisProduct);//запрос в БД на удаление текущей записи

db.SaveChanges();

EvokingWindow.ProductPanel.Children.Remove(this);

}

UserControl для товара в корзине создается со следующим содержанием разметки

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1.5\*"/><!--картинка-->

<ColumnDefinition Width="1.5\*"/><!--название-->

<ColumnDefinition Width="1\*"/><!--колво-->

<ColumnDefinition Width="1\*"/><!--стоимость( + цена за штуку)-->

<ColumnDefinition Width="1\*"/><!--кнопка удаления-->

</Grid.ColumnDefinitions>

<Image x:Name="BasketProductImage" Grid.Column="0" Margin="5 5 5 5"/>

<Label x:Name="BasketProductNameLabel" Grid.Column="1" Margin="5 5 5 5"/>

<Grid Grid.Column="2" Margin="20" Height="50">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.2\*"/>

<ColumnDefinition Width="0.4\*"/>

<ColumnDefinition Width="0.2\*"/>

</Grid.ColumnDefinitions>

<Button x:Name="ProductCountUpButton" Content="+" Grid.Column="2" Margin="1 1 1 1" Click="ProductCountUpButton\_Click"/>

<Button x:Name="ProductCountDownButton" Content="-" Grid.Column="0" Margin="1 1 1 1" Click="ProductCountDownButton\_Click"/>

<TextBox x:Name="ProductCountTextBox" Grid.Column="1" Margin="1 1 1 1" IsEnabled="False" />

</Grid>

<Grid Grid.Column="3" Margin="5">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Label Grid.Row="0" x:Name="BasketProductSumLabel" Grid.Column="3"/>

<Label Grid.Row="1" x:Name="BasketProductPriceLabel" Grid.Column="3"/>

<Label Grid.Row="2" x:Name="BasketProductDiscountLabel" Grid.Column="3"/>

</Grid>

<Button x:Name="DeleteProductButton" Click="DeleteProductButton\_Click" Margin="5" Grid.Column="4" Content="Delete"/>

</Grid>

И следующим содержанием класса(.CS)

private BasketWindow EvokingWindow;

private int Quantity;

Product ThisProduct;

public BasketProductVisual(BasketWindow BasketWindow, int productId, int quantity)

{

InitializeComponent();

BookShopContext dbCont = new BookShopContext();

ThisProduct = dbCont.Products.Find(productId);

EvokingWindow = BasketWindow;

if (ThisProduct.PathToImage != null) {

BasketProductImage.Source = new BitmapImage(new Uri(ThisProduct.PathToImage, UriKind.RelativeOrAbsolute));

}

BasketProductNameLabel.Content = ThisProduct.Name;

BasketProductSumLabel.Content = new string($"{ThisProduct.Price \* quantity \* (1-(Decimal)ThisProduct.Discount) }");

BasketProductPriceLabel.Content = new string($"{ThisProduct.Price} за ед.");

BasketProductDiscountLabel.Content = new string($"Скидка {ThisProduct.Discount \* 100}%");

ProductCountTextBox.Text = quantity.ToString();

Quantity = quantity;

}

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

{

Quantity++;

EvokingWindow.Basket[ThisProduct.Id]++;

EvokingWindow.RefreshBasket();

UpdatePrice();

}

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

{

Quantity--;

EvokingWindow.Basket[ThisProduct.Id]--;

EvokingWindow.RefreshBasket();

UpdatePrice();

}

private void UpdatePrice() {

BasketProductPriceLabel.Content = ThisProduct.Price \* Quantity;

ProductCountTextBox.Text = Quantity.ToString();

}

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

{

EvokingWindow.Basket.Remove(ThisProduct.Id);

UpdatePrice();

EvokingWindow.RefreshBasket();

}

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

{

Quantity = Convert.ToInt32(ProductCountTextBox.Text);

UpdatePrice();

EvokingWindow.RefreshBasket();

}