<https://sweetalert.js.org/guides/#installation>

w klasie entity virtual ustawuala czy aktyweny na true albo przeladowywac save changes

https://select2.org/selections

Omowienie platmormy ASP NETMVC

APLIKACje asp .net core i wzorze projektowy mvc

Glowna strona projektu powinna być Index.cshtml, glowny layout projektu jest zdefiniowany w pages shared \_layout.cshtml -szablon strony tu wyświetlaj cala zawa

Layout to szablon stron zgodny z tym layoutem,

@RenderBody() w layoutcie oznaczacza ze w tym miejscu ma wyswietlac się cala zawartość strony zgodna z tym layoutem

Glowny plik ze stylami znajduje się w sekcji wwwroot\_>css->site.css

3 kolory na stronie\

Nasze projekty interntowe za pomocą ASP netcore Model View Controller

Będziemy używali wzorca projektowego Model View Controller, który dzieli aplikacje internetowa na trzy warsrtwy model zawiera klasy realizujące dostep do bazy danych oraz klasy logiki biznesowej (C#)

Warstwa View (HTML+C#) zawiera interfejsy użytkownika lub elementy tych interfejsow (partial view)

Wartswa Controller zawiera klasy posredniczace miedzy modelem a widokiem

Solucja to zbior projektów które sa ze sobą powiazane nasza solucja będzie skladac z 3 projektow 1 Firma.portalWWW to projekt dla klientów naszej firmy

Firma.Intranetprojekt dla pracowników firmy zarzadzania

Procedura tworzenia solucji

1. Utworz nowa solucje Blank solution
2. PPM na solution firma i add new projekt (C# web asp not net web app mvc none Firma.PortalWWW
3. PPM na solution firma I add new projekt C# mvc o nazwie Firma.Intranet invidual accont logowanie

Uwaga będzie jeszcze 2 projekty do bazy danych i logiki biznesowej

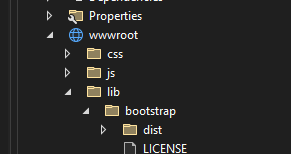
Firma.PortalWWW zrobimy za pomocą material designa 3 szablony css

Projekt Firma.Intranet zrobimy za pomocą bootsrapa

Procedura budowania layoutu Firma.PortalWWW przypomocy material designu <https://m3.material.io/>

<https://materializecss.com/>

<https://materializecss.com/getting-started.html> pobierz biblioteke

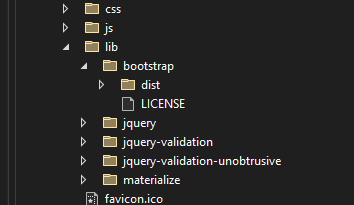


Ze strony <https://materializecss.com/getting-started.html> pobieramy biblioteke do materializa

Rozpakowujemy ta bibliotekę

3 przeciagami folder materialize do firma.portalWWW -> wwwroot-> lib (obok boot

Zobacz showcasy https://materializecss.com/showcase.html



wyszukaj bootsrapa i materialize dla tej firmy na ktorych sie bedzie wzorowac

templete do materialize znalesc

Procedura budowania portalu www

* + - 1. Do wwwroot-> lib -> dodaj folder materialize przeciagnij pobrany z internetu
      2. Na bazie wzoru (znalezionego z internetu utworz layot w views->shared->layout.cshtm <https://materializecss.com/templates/starter-template/preview.html> wyswietli zrodlo w kopiu do layout
      3. @RenderBody() wyswietlana zzawartosc strony zgodna z layoutem zanlesc ikony z materialize
      4. Edutyj view->home->index.cshtml i utworz kod strony zgodnej z layoutem
      5. Showcasy co to jest
      6. Edytuj view->shared layout i w sekcji nawigacyjne dodaj link do wybranych stron<li>@Html.ActionLink("nazwa wyswietlana","nazwa pliku","nazwa foldera w ktorym jest plik(controlera)")</li>
      7. Procedura dodaawania nowych stron (widokow) do projektow:

1. utworz nowy controler lub w istniejacym kontrolerze dodaj funckje (homecontroller

public IActionResult Kontakt()

{

return View();//widok bedzie nazywal sie tak samo jak funcja Kontakt

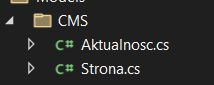
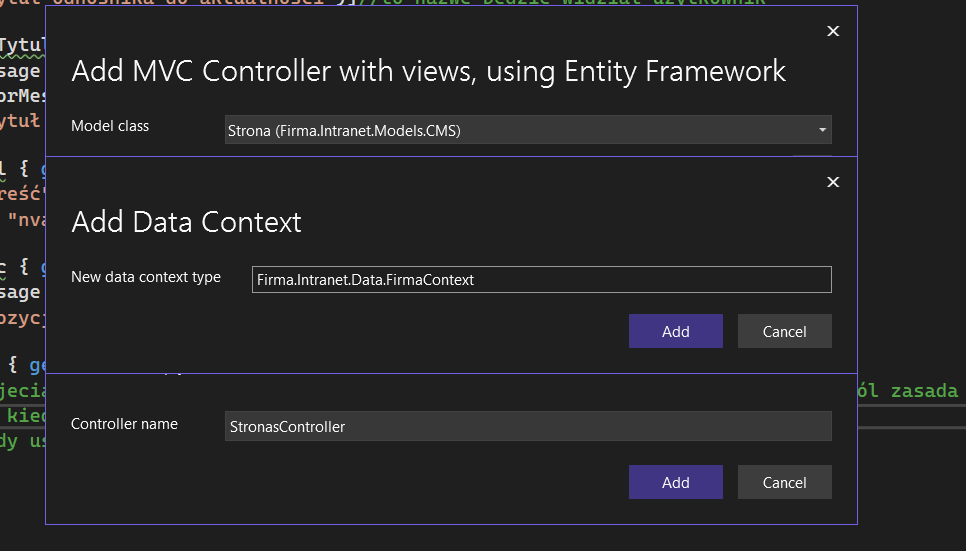
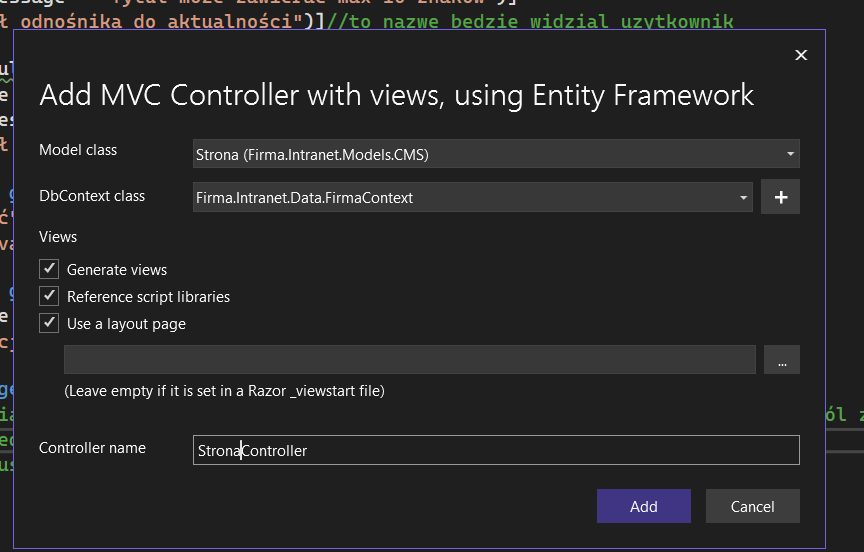
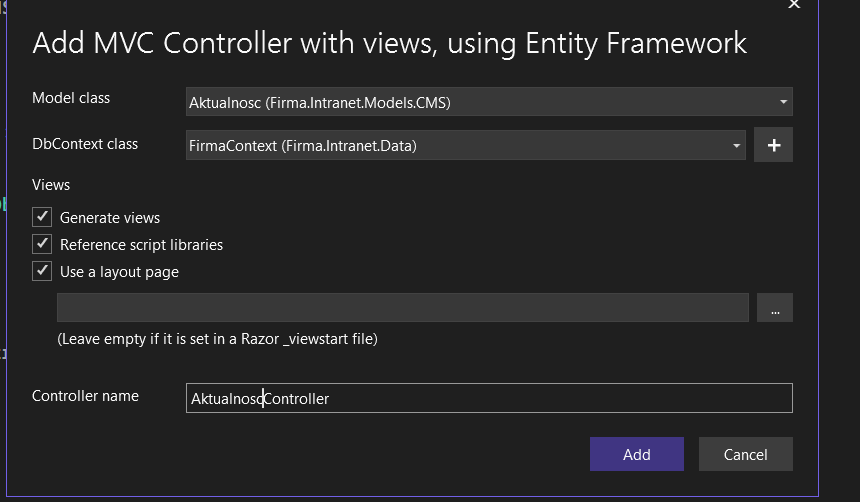
}

2. Naciskiem PPM na nazwe utorzonej fuknci Kontakt i add view razor view empty i nazywamy go tak jak funkcje Kontakt

3. Edytuj view->shared->\_layout.cshtml i dodaj link do utworzonego widoku na stronie f12

Layout w intranecie i w cliencie

Cel w intranecie zbudowac mechanizmy interfejsy do sterowania zawaroscia kazdej strony i aktualnosci procedura sterowania zawartoscia strony internetowej z bazy danych:

1. Przeczytaj dokumentacje https://learn.microsoft.com/en-us/ef/core/
2. Oznacz jako projekt startowy Firma.Intranet (wszystko nize w intranecie
3. PPM na models i add new folder o nazwie cms content managment system
4. PPM na cms add class o nazwie strona
5. Do operacji na bazei danych bedziemy uzywac entity framework w entity frameworku sa mozliwe nastepujace podejscia :
6. – database first (najpierw baze dany a potem entity framework tworzy klasy) w desktopowych
7. –code first (najpierw tworzymy klasy representujace tabele w bazie danych a entity framework z tego sam tworzy baze danych
8. PPM na models \_. Cms add class o nazwie aktualnosc 
9. Dokonaj kompilacji projektu
10. Nacisnij PPM na folder controllers add controler:
11. –mvc controler using entity fr anazwa strona data context nacisnij plus i name Aktualnosc
12. 
13. 
14. Jak blad to powtorz to samo,jak dalej blad to zle stworzona klasa strona
15. Analogicznie utworz aktualnosc controller PPM ba kontroler add entity framew 
16. Edytuj views ->shared ->\_layoutcsmtl i w sekcji navigacyjnej dodaj linki do wlasnie utworzonych stron index
17. Dokonaj kompilacji projektu
18. Uruchom menu tools menagera pakietu nuget i console menadzera
19. W konsoli ustaw projekt domyslny na firma intranet 
20. W konsoli wpisz

PM> Add-Migration InitialCreate -context FirmaContext

1. PM> Update-Database

Wykonac layput pro min 3 strony

2 layuty

intraner icony tabela, przycisku pro jak enova online strona indesx, dodawanie faktury online

Procedura wyodrebnienia bazy danych do osobnego projektu

PPm na cala solucje i add new projekt ->class library C# o nazwie Firma.Data

Ppm na Firma.Data(to dodaj jako classe an nie claaslibrary bo może być ballad) i add new folder o nazwie Data

Ppm na data folder i add folder CMS(content managment system)

W folderze cms utworz klase Aktualnosci i strona analogiczni jak w fiema intrae

Ppm na data i add new folder Sklep

Ppm na sklep i add class rodzaj oraz towar: na bazei tyc class relacja 1 do wielu 1 rodzaja towaru na wiele

Ppm na data folder i addclass firma context

W klasie firma context nacisni ppm na dbcontext wybirze zquick action istall package ms entity framework core

Aspi restowe (restapi mobilna i internetowa wystaw firma.data i do rest api wystaw te dane)

Ppm na firma.data i wybierz build

W projekcie firma.intranet wykasu zawarotsc foldera data, migration, oraz models

Firma.intranet ppm na dependencies i add project reference a nastepnie solution i zaznacz firma.data (te operacje wykonujemy po to zeby projekt firma intranet widzial projekt firma.data i mogl z niego korzystac)

Wejdz do fidokow firma.intranet i zmien naglowki tych widokow na @model Firma.Data.Data.CMS.Aktualnosc

Wejdz do wszystkich kontrolerow aktualnosci i stron i popraw kod aby kontrolery uzywaly firmy data using Firma.Data.Data;

wProgram.cs usun using Firma.Intranet.Data;

ppm na firma.intranet i wybieramy opjce set as sturtup projekt

dokonaj kompilacji projektu

upewnij sie ze na pewno wszystkie projektu uzywaja ten samej wersji entity framework core w dependance package

wejdz do tools nuget package manager -> package manager console i wybierz tam default prije firma.intranet oraz wpisz: dotnet restore

w firma intranet wejdz do app seting json i ustaw nazwe bazy danych na firma context 2023 "FirmaContext": "Server=KACPER;Database=FirmaContext2023;Trusted\_Connection=True;MultipleActiveResultSets=true"

dokonaj kompilacji lub bildowania wszystkich projektow

upewnij sie ze firma intranet jest projektem startowym

w package manager console ustaw defaultowy projekt na firma.data(gdzie baza danych ) (a startowy tam gdzie connection string( oraz wpisz add migration m1 wpisz: add-migration m1

firma data -> migration m1 usunac cala sekcje using cala w tym pliku migration m1 nacisna ppm na migration przy dziedziczeniu quick acton i zainslatowa entity framework core w tej samej wersji co w intranecie i forma.data (default firma data a startowy firma.intranet) TU ZOBACZ

podobne operacje wykoran w pliku frima.context.snapshot qick action istal i use lokal version

dokonaj bulidowana projektu firma.data

urucvhom package manager cosolen default projekt firma.data i wpisz update-database (jaki startup mialem intranet ?)

uruchom projekt i wpisz dane Strona1 tresc strony pierwszej

zajecia 16.04.2023 (summary zamiast tworzyc w shared partial view moge utworzyc je w folderze i np partial Odnosniki przeklejam z layout

Partial View to technologia przy pomocy ktorej w jeden widok mozemy wstawiac inny widok tzw partialny

Procedura wyswietlania zawartosci bazy danych w projekcie firma.PortalWWW

PPM na Firma.PortalWWW opjac set as starup projekt. Wszystwkie ponizsze operacje w PortalWWW

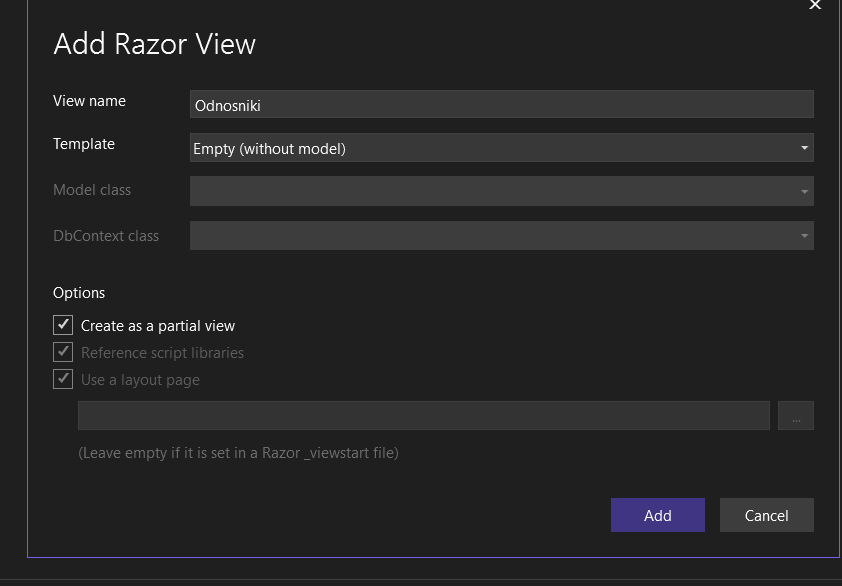
Edytuj plik appsetings.json i dodaj sekcje connection string wzorujac sie na firma intranet

PPM na dependencie i add project refrerence a nastepnie w solution wybrac firma.data

Edytuj plik Program.cs i dodaj elementy zwiazane z firma kontext podobnie jak firma intranet.

1. using Firma.Data.Data;
2. using Microsoft.AspNetCore.Identity;
3. using Microsoft.EntityFrameworkCore;
4. using Microsoft.Extensions.DependencyInjection;
5. var builder = WebApplication.CreateBuilder(args);
6. builder.Services.AddDbContext<FirmaContext>(options =>
7. options.UseSqlServer(builder.Configuration.GetConnectionString("FirmaContext") ?? throw new InvalidOperationException("Connection string 'FirmaContext' not found.")));

5. PPM na folder Views->Shared i add view (razor view) o nazwie Odnosniki jako partial view (partial view odnosniki) wyswietlam linki do podstron do innych stron



Z laypput wytnij wszystko do odnoscnikow.cshmt <div class="nav-wrapper container">

<a id="logo-container" href="#" class="brand-logo">Logo</a>

<ul class="right hide-on-med-and-down">

<li>@Html.ActionLink("Strona glowna","Index","Home")</li>

<li>@Html.ActionLink("O Firmie","OFirmie","Home")</li>

<li>@Html.ActionLink("Kontakt","Kontakt","Home")</li>

<li></li>

</ul>

<ul id="nav-mobile" class="sidenav">

<li><a href="#">Navbar Link</a></li>

</ul>

<a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-icons">menu</i></a>

</div>

Efekt

<nav class="light-blue lighten-1" role="navigation">

@await Html.PartialAsync("Odnosniki")

@\* <div class="nav-wrapper container">

<a id="logo-container" href="#" class="brand-logo">Logo</a>

<ul class="right hide-on-med-and-down">

<li>@Html.ActionLink("Strona glowna","Index","Home")</li>

<li>@Html.ActionLink("O Firmie","OFirmie","Home")</li>

<li>@Html.ActionLink("Kontakt","Kontakt","Home")</li>

<li></li>

</ul>

<ul id="nav-mobile" class="sidenav">

<li><a href="#">Navbar Link</a></li>

</ul>

<a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-icons">menu</i></a>

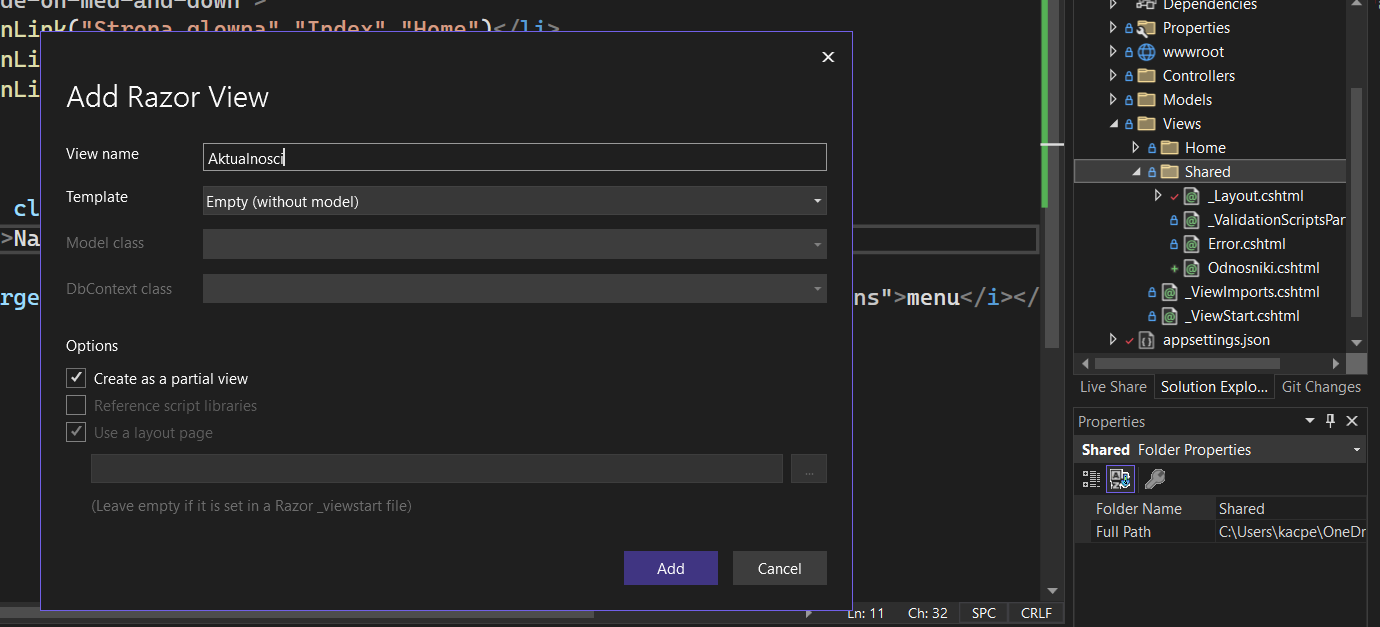
</div>\*@

</nav>

Tak asynchronicznie wywolujemy widok partialny @await Html.PartialAsync("Odnosniki")

6. Uzypelnij kod widoku odnosnikki jak powyzej

7. PPM na shared w views i add view razor view o nazwie aktualnosci jako partial view



Wytnij z layoout section

<!-- Icon Section -->

@await Html.PartialAsync("Aktualnosci")

@\* <div class="row">

<div class="col s12 m4">

<div class="icon-block">

<h2 class="center light-blue-text"><i class="material-icons">flash\_on</i></h2>

<h5 class="center">O firmie</h5>

<p class="light">Tu teks o firmie</p>

</div>

</div>

<div class="col s12 m4">

<div class="icon-block">

<h2 class="center light-blue-text"><i class="material-icons">group</i></h2>

<h5 class="center">O towarach</h5>

<p class="light">Tu bedzie tekst o towarach.</p>

</div>

</div>

<div class="col s12 m4">

<div class="icon-block">

<h2 class="center light-blue-text"><i class="material-icons">settings</i></h2>

<h5 class="center">o sklepie</h5>

<p class="light">Tu bedzie tekst o sklepie.</p>

</div>

</div>

</div>\*@

I wklej to do aktualnosci.cshtml

Jezeli widok bedzie uzywal kolekcji np stron to piszemy w Odnosniki.cshtml @model IEnumerable<Firma.Data.Data.CMS.Strona>

Jezeli wodko bedzie uzywal tylko 1 strony to piszemy to samo tylko bez Ienumerable<>

6 Edytuj HomeController i uzupelnij jego kod tak zeby pobieral strony i aktualnosci z bazy danych

7 Edytuj Layout.cshtml i do partial view przekaz ViewBag @await Html.PartialAsync("Odnosniki", (IEnumerable<Firma.Data.Data.CMS.Strona>) ViewBag.ModelStrony))

8. Edytuj Views->Home->index i pobieraj dane do wyswietlana z bazy danych

@model Firma.Data.Data.CMS.Strona

@\*tu nie mam ienume po wyswitlam tylko 1 sostrone\*@

@{

ViewData["Title"] = Model.LinkTytul;//z bazy danych pobierz link tutyl

}

<h1 class="header center orange-text">@Model.Tytul </h1>

<div class="row center">

<h5 class="header col s12 light">@Model.Tresc</h5>

</div>

Edytuj Views shared layout.cshtml i w sekcji title pobieraj dane z bazy danych <title>@ViewData["Title"] - KA</title>

Edytuj kontroller->Homecontroller i dodaj pobierranie aktualnosci do

ViewBag.ModelAktualnosci =

(

from aktualnosc in \_context.Aktualnosc

orderby aktualnosc.Pozycja

select aktualnosc

).ToList();

W projektcie strona (zamiast aktualnosci ciekawe oferty w aktulanosci moge dodac ikone nazwaikony np. flash\_on)

4 Edytuj views-.shares->layout,cshtml i przekaz do partial view viewbag z aktulanosciami @await Html.PartialAsync("Aktualnosci",(IEnumerable<Firma.Data.Data.CMS.Aktualnosc>) ViewBag.ModelAktualnosci )

Edytuj views->shared->aktualnosci.cshtml i zmien ten widok zeby uzywal bazy danych

@model IEnumerable<Firma.Data.Data.CMS.Aktualnosc>

<div class="row">

@foreach (var item in Model)

{

<div class="col s12 m4">

<div class="icon-block">

<h2 class="center light-blue-text"><i class="material-icons">flash\_on</i></h2>

<h5 class="center">@item.Tytul</h5>

<p class="light">@item.Tresc</p>

</div>

</div>

}

</div>

<h5 class="center"> @Html.ActionLink(item.Tytul, "Index", "Aktualnosc", new { id = item.IdAktualnosci },null)</h5>

PPM na Controllers i add controller empty o nazwie AktualnoscController

public class AktualnoscController : Controller

{

private readonly FirmaContext \_context;//tu jest baza danych

//bazowy kontroller i z baza danych i dziedziczy z niego wszystkie kontrolery

public AktualnoscController(FirmaContext context)

{

\_context = context;

}

//tez index dziedziczyc moge, mozna to asynchronicznie zrobic

public async Task<IActionResult> Index(int id)//to jest id kliknietej aktualnosci id nie moge byc nullem

{

//viewbag to taki listonosz ktory przenosi dane miedzy kontrolerem a widokiem modelstrony to nazwa zmiennej

ViewBag.ModelStrony =

(

from strona in \_context.Strona

orderby strona.Pozycja

select strona

).ToList();

ViewBag.ModelAktualnosci =

(

from aktualnosc in \_context.Aktualnosc

orderby aktualnosc.Pozycja

select aktualnosc

).ToList();

//odnadujemy aktualnosc o danym kliknietym id i przekazujemy do widoku

var item = await \_context.Aktualnosc.FindAsync(id);

return View(item);

}

}

}

W AktualnoscController PPM na nazwe funckji Index i dodaj widok razorview empty o nazwie Index.cshtml

@model Firma.Data.Data.CMS.Aktualnosc

@\*tu nie mam ienume po wyswitlam tylko 1 sostrone\*@

@{

ViewData["Title"] = Model.LinkTytul;//z bazy danych pobierz link tutyl

}

<h1 class="header center orange-text">Aktualnosc: @Model.Tytul </h1>

<div class="row center">

<h5 class="header col s12 light">@Model.Tresc</h5>

</div>

Zobacz <https://www.mapeditor.org/2023/04/04/tiled-1-10-1-released.html> ?

**editor tile html https://www.tiny.cloud/**

Zadanie 1

Wzorujac si na aktualnosciach i stronach steruj za pomoca bazy danych sekcja dodatkowych informacji np RODO

Wzorujac sie na aktualnosciach i stronach steruj za pomoca bazy danych sekcja Connect i zmien ja na jakas swoja sekcje

Parametr id 1 WSB-nlu

Parametr id 2 Projekt realizazowany

Parametr id 3 made by wsb-nlu

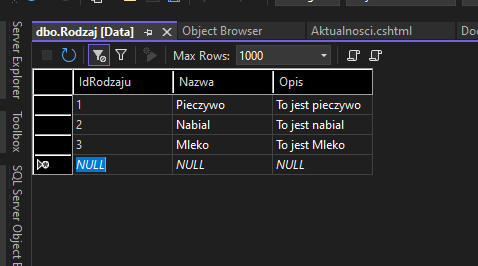
Utworzyc tabele parametry o polach Id, wartosc, i opis i do tej tabeli parametry (jako rekordy) wpisac teksty wyswietlane z lewej strony paska pomaranczowego. Zamiast tekstu wpisanych w html na sztywno wyswietlac teksty z lewej strony paska pomaranczowego z bazy danych

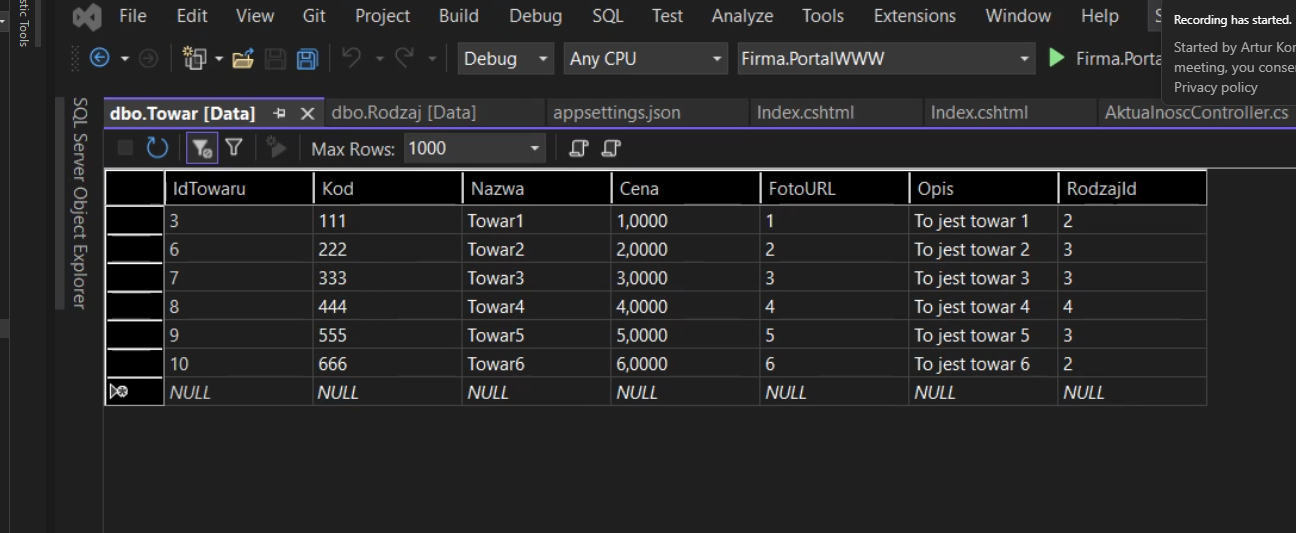
do projektu dodaj obsluge w czesci portalu minimum 3 tabel. wszystkimi tekstami steruj z bazy danych.

W layout footer (lub parametry moge dodac np @await Html.PartialAsync("MediaSpolecznosciowe", (IEnumerable<Firma.Data.Data.CMS.MediaSpolecznosciowe>) ViewBag.ModelMediowSpolecznosciowych)

30.04







Procedura tworzenia sklepu internetowego punkt 1 PPM na fr4ma portal www i wybierz set as sturtup project .Jeden kontroler może ofdpowiadac wielu widokom ale dla każdego widoku powinna być strorzona osobna 1 funckcja w kontrolerze , (w folderze kontrolerssa foldery i w nich sa kontrolery- pozamykać w folderach)

2. Wszystkie poniższe elementu wykonywane w portalu www dany projekt może mieć wiele layoutów

3. PPPm na views-> shared i add view razor view empty o nazwie \_SklepLayout (moze byc zupelnie inny – ttu skopiowalem z \_Layout)

4. dokanaj kompilacji lubctrl+B

5. PPm na kontrollers i add kontroler (MVC kontroler empty) o nazwie SklepController

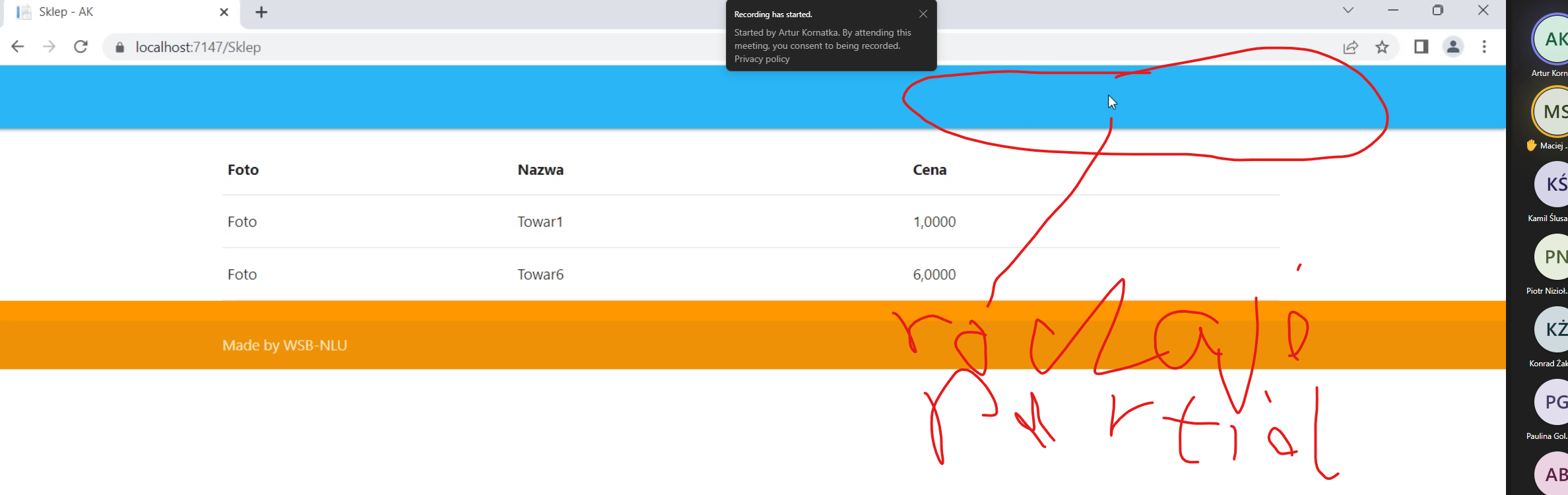
6. W sklep controller PPM na nazwe funkcji index i add view razor view empty o nazwie Index

7. Edytuj views/shared/Odnosniki.cshtml i dodaj link do sklepu

<li>

@Html.ActionLink("Sklep internetowy", "Index", "Sklep")

</li>



8 PPm na views/shared i add view (razor view empty create as a partial view) o nazwie rodzaje Menu

@model IEnumerable<Firma.Data.Data.Sklep.Rodzaj>

<div class="nav-wrapper container">

<a id="logo-container" href="#" class="brand-logo">Logo</a>

<ul class="right hide-on-med-and-down">

<li>

@Html.ActionLink("Strona glowna", "Index", "Home")

</li>

@foreach (var item in Model)//item to rodzaj

{

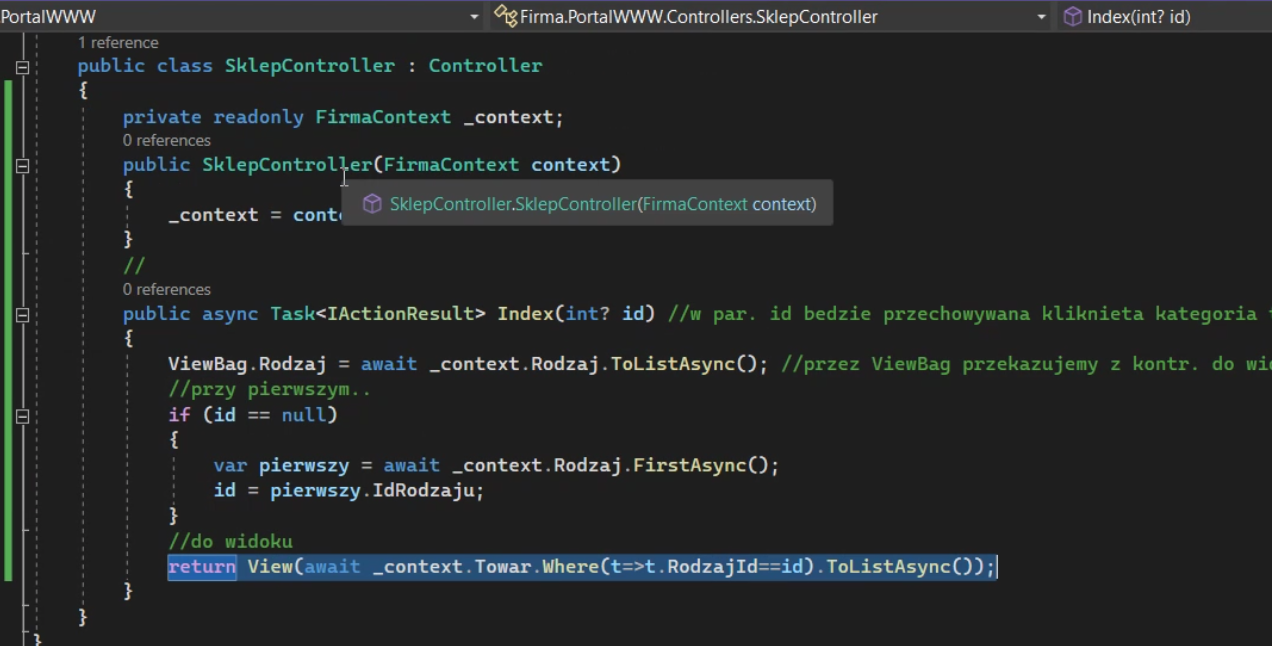
<li>

@Html.ActionLink(item.Nazwa, "Index", "Sklep", new { id = item.IdRodzaju },null)//odnosninik do trony glownej w sklepie z parametrem

</li>

}

</ul>

//towary z kliknietego rodzaju strona glowna 

**9.Edutuj view shared \_skleplayout** i sekcji nawigacyjnej dodaj wywolanie rodzaje menu

<nav class="light-blue lighten-1" role="navigation">

@await Html.PartialAsync("RodzajeMenu", (IEnumerable<Firma.Data.Data.Sklep.Rodzaj>) ViewBag.Rodzaj))

</nav>

KOntroller który wystawi 1 towar, widok wyswiertli 1 towar i wirko

10. Wejdz do sklepController i dodaj funckje szczegoly

public class SklepController : Controller

{

private readonly FirmaContext \_context;

public SklepController(FirmaContext context)

{

\_context = context;

}

//funckja index wystawia wszytskie towary danego rodzaju widokowi index ze sklepu

public async Task<IActionResult> Index(int? id)//w parametrze id bedzie przechowywana kliknieta kategoria towaru, katororia towaru bedzie nullem przy pierwszym wejsciu na strone

{

// var rodzaje = await \_context.Rodzaj.ToListAsync();//wszystkie rodzaje, zeby nie wywolywac bazy

ViewBag.Rodzaj = await \_context.Rodzaj.ToListAsync();//przez viewbag przekazujem z kontrolera do widoku wszystkie rodzaje

//przy pierwszym wejsciu do sklepu id kategori jest ppuste i podstawimy pod to piersza kategorie, tak zeby przy pierwszym wejsciu do sklepu wyswietlaly sie towary pierszej kategori (potem beda promowane)

if(id ==null)

{

var pierwszy = await \_context.Rodzaj.FirstAsync();//1 rodzaj ktory jest w bazie danych, jak firstordefault i wyswietlic erro czy nie wjest nullem

id = pierwszy.IdRodzaju;

}

//do widoku przekazujemy wszystkie towary klliknietego rodzaju lub w przypadku pierwszego wejscia do sklepu wszystkie towary pierwszej kategorii

return View(await \_context.Towar.Where(t=>t.RodzajId == id).ToListAsync());

}

public async Task<IActionResult> Szczegoly(int? id)//w param id bvedzie umieszczone id kliknietego towaru ktorego szczegoly mamy wyswietlic

{

ViewBag.Rodzaj = await \_context.Rodzaj.ToListAsync();

// return View(await \_context.Towar.FirstOrDefaultAsync(t => t.IdTowaru == id));//zwracamy widok szczegoly i towary ktorego id jest rowne id z parametru

//do widoku przekjazujemy towar o danym id ktory kliknieto (uzuj find)

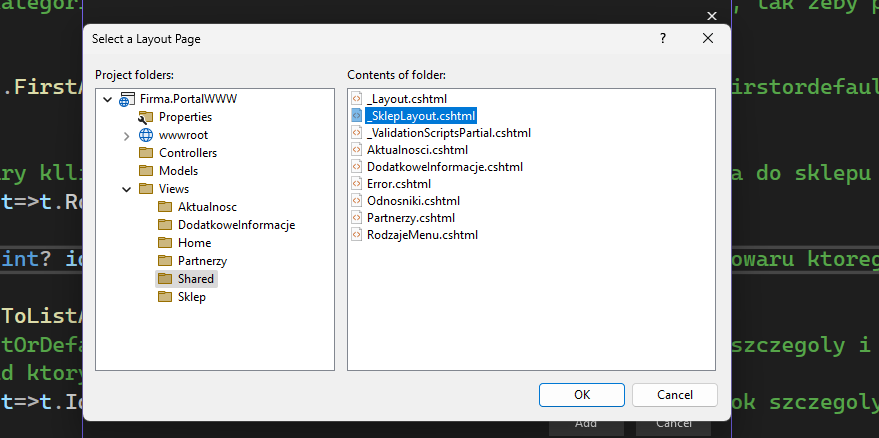
return View(await \_context.Towar.Where(t=>t.IdTowaru ==id).FirstOrDefaultAsync());//zwracamy widok szczegoly i towary ktorego id jest rowne id z parametru

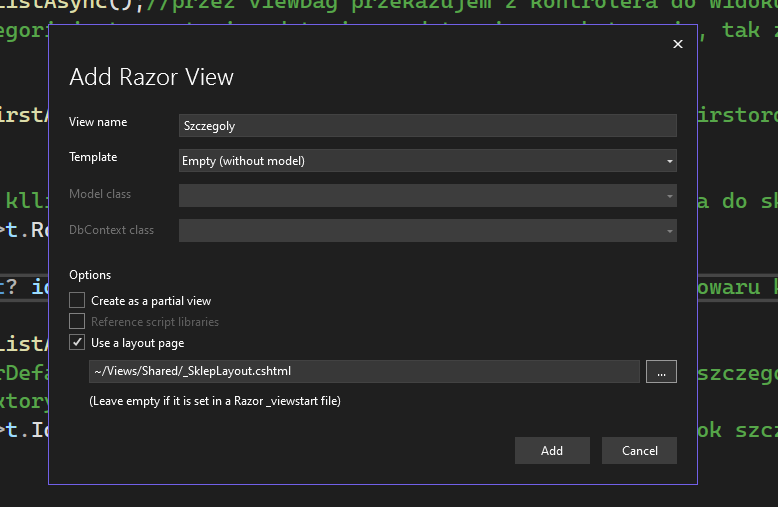
}

}

}

PPM na szczegoly w kontrolerze sklepController i add view razor view o nazwie Szczegoly empty zgodny z layoutemSklep



Szczegoly.cshtml

@model Firma.Data.Data.Sklep.Towar

@{

ViewData["Title"] = "Szczegoly";

Layout = "~/Views/Shared/\_SklepLayout.cshtml";

}

<div>

Foto

</div>

<div>

@Model.Nazwa

</div>

<div>

@Model.Kod

</div>

<div>

@Model.Cena

</div>

<div>

@Model.Opis

</div>

Edytuj views/sklep/index.cshtml i dodaj link do szczegolow towaru

@model IEnumerable<Firma.Data.Data.Sklep.Towar>

@{

ViewData["Title"] = "Sklep";

Layout = "~/Views/Shared/\_SklepLayout.cshtml";

}

<div>

<table>

<thead>

<tr>

<th>Foto</th>

<th>Nazwa</th>

<th>Cena</th>

</tr>

</thead>

<tbody>

@foreach (var towar in Model)

{

<tr>

<td>

Foto

</td>

<td>

@Html.ActionLink(towar.Nazwa,"Szczegoly", "Sklep", new { id = towar.IdTowaru }, null)

</td>

<td>

@towar.Cena

</td>

</tr>

}

</tbody>

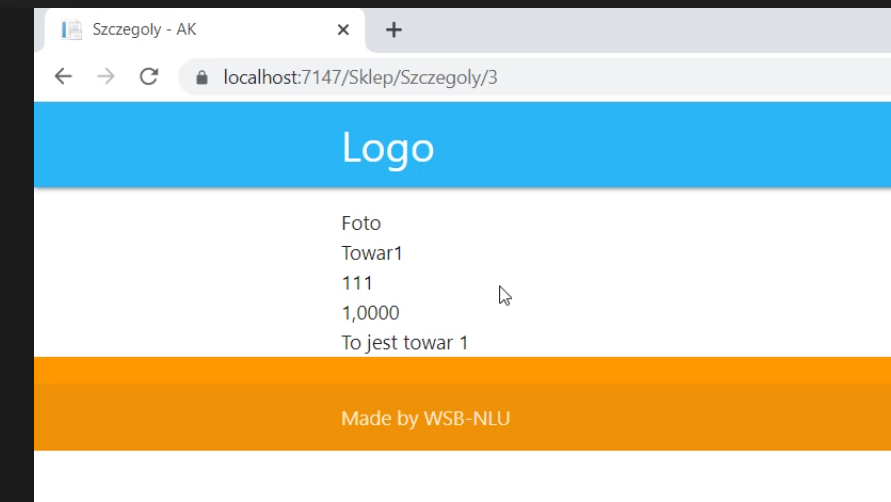
</table>

</div>

Zadanie 1 materialize-cvomponents ->cards wyświetlanie towarów zamiast tabelki

Przebudować widok wyświetlający towary tak żeby wyglądał bardzo profesjonalnie (np korzystając z kart materialize). Ogłaszamy konkurs na najpiękniejszy widok towarów

Zad 2.

Zrobić profesjonalny widok szczegółów towarów. 

Zad 3.

Zapisywać ścieżkę do zdjęcia w bazie i wyświetlać zdjęcia z bazy (img, albo ścieżka do zdjęcia)

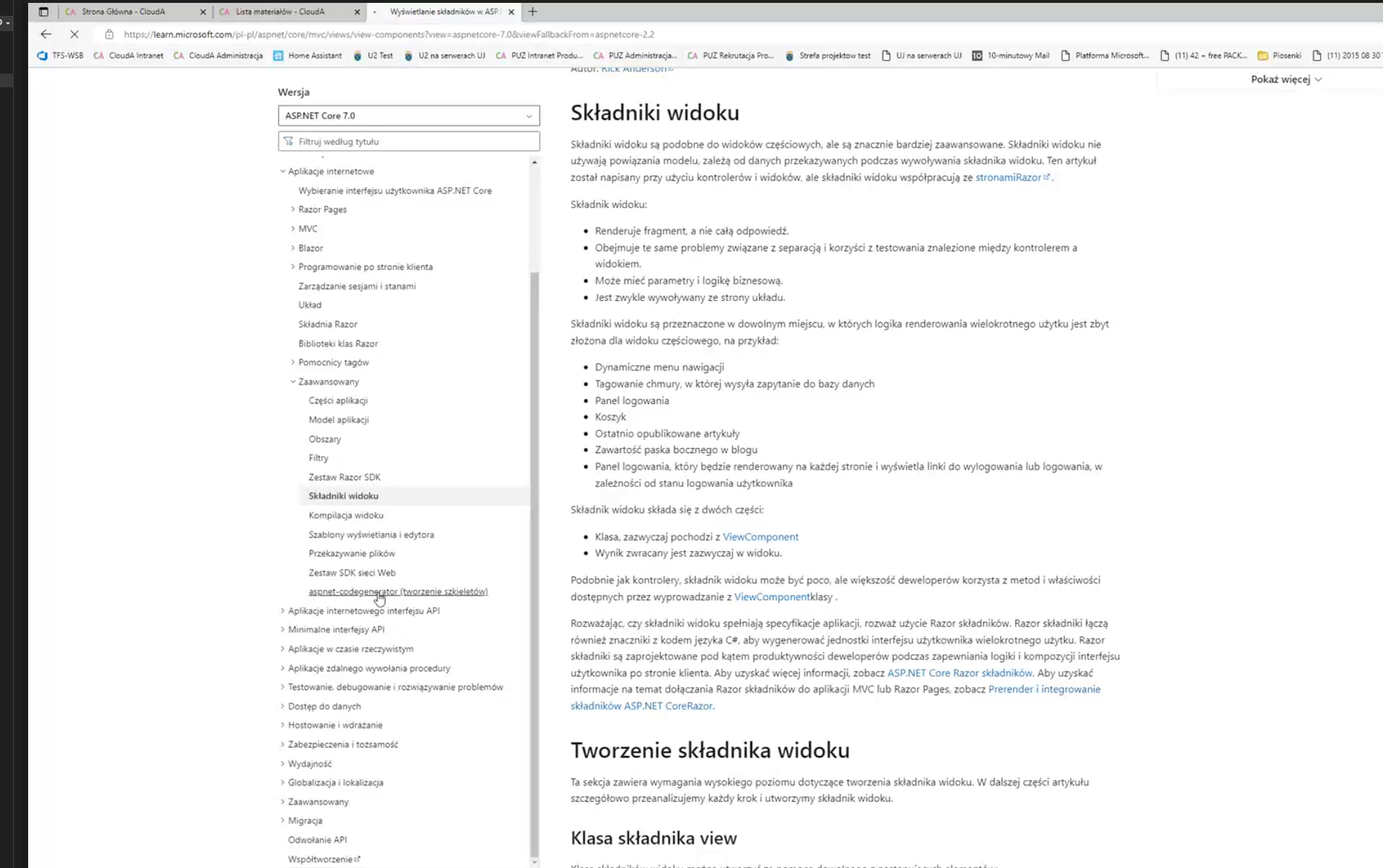
Jako Image jak zapisywać

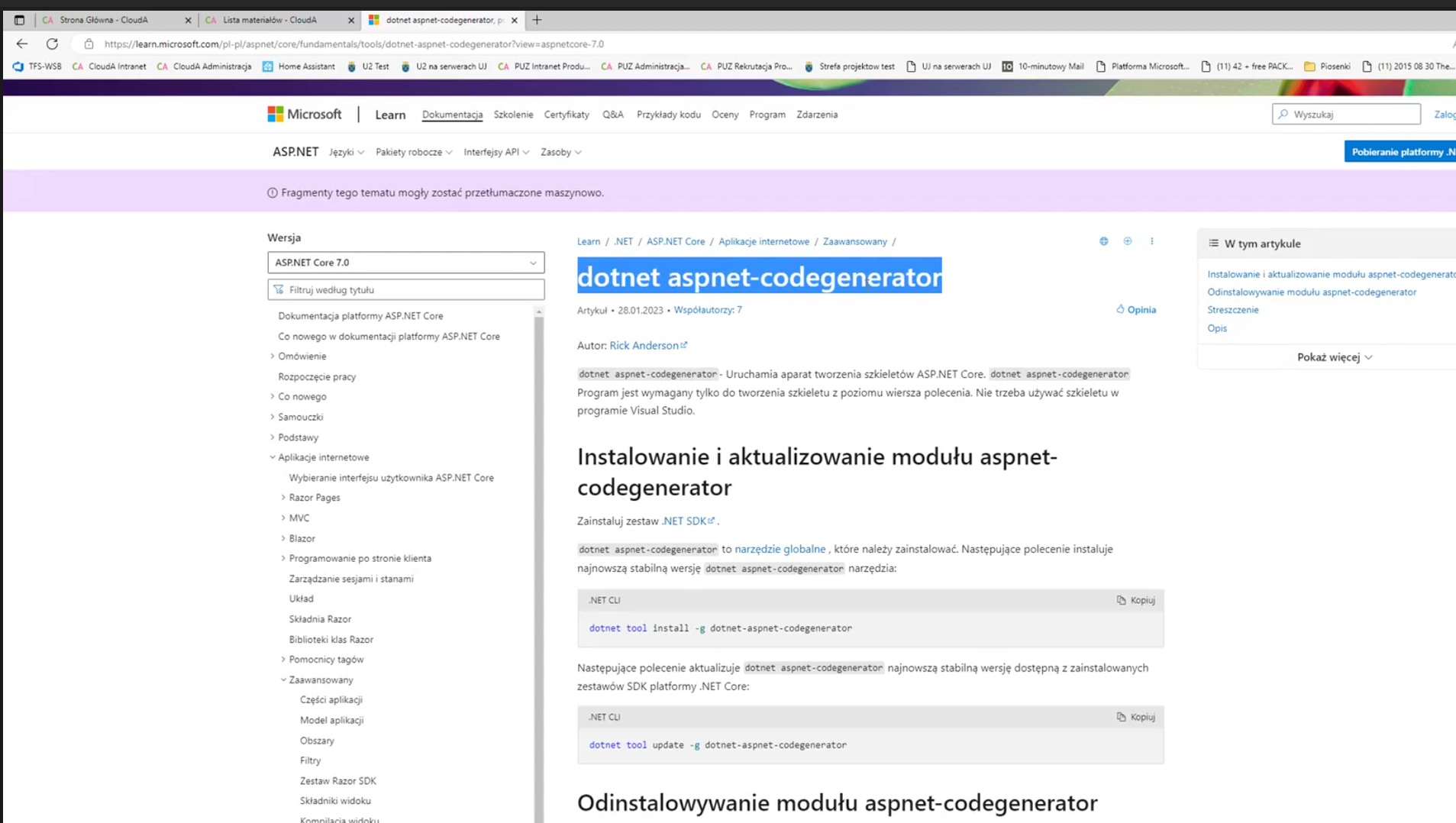
Zad 4.

Zrobić sterowanie towarami w intranecie razem ze zdjęciami

Alternatywa do view baga i partial view to componenty

7.05 Componenty dziedziczenie controllerow





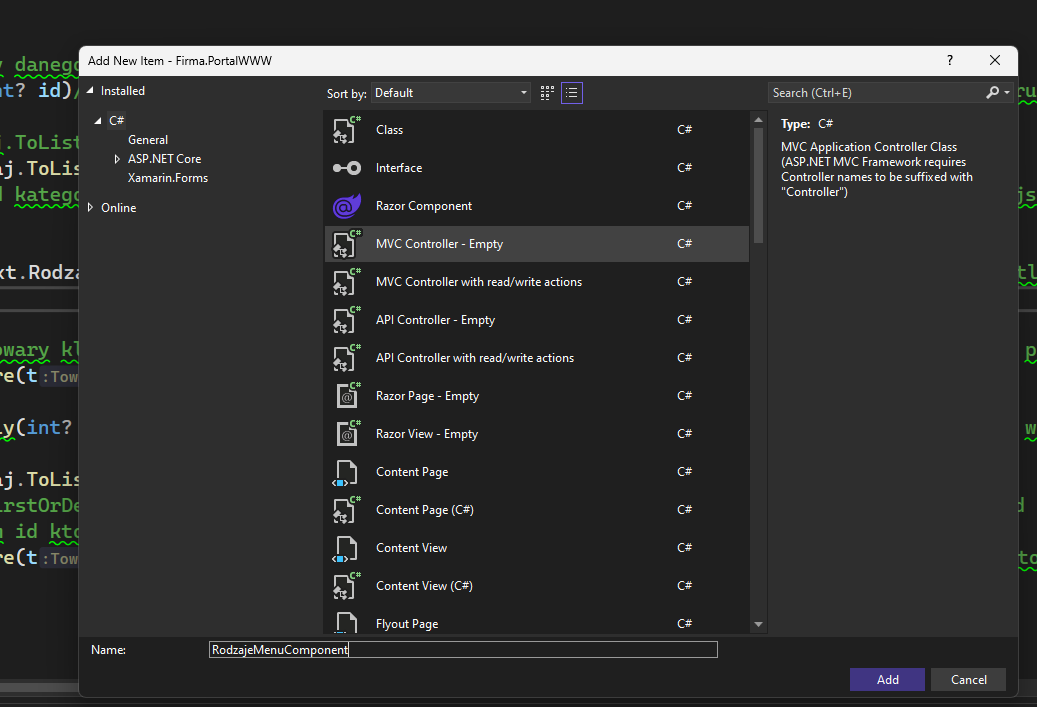
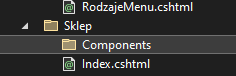
<https://learn.microsoft.com/pl-pl/aspnet/core/fundamentals/tools/dotnet-aspnet-codegenerator?view=aspnetcore-7.0>

zabezpieczenie strona z errorem jak w baize danych nie ma strony

do firma data using Microsoft.EntityFrameworkCore.Design;  
using Microsoft.EntityFrameworkCore;

namespace Firma.Data.Data  
{  
    public class plicationDbContextFactory: IDesignTimeDbContextFactory<FirmaContext>  
    {  
        public FirmaContext CreateDbContext(string[] args)  
        {  
            var optionsBuilder = new DbContextOptionsBuilder<FirmaContext>();  
            optionsBuilder.UseSqlServer("Server=(localdb)\\mssqllocaldb;Database=FirmaContext2023;Trusted\_Connection=True;MultipleActiveResultSets=true");  
            return new FirmaContext(optionsBuilder.Options);  
        }  
    }  
}

Procedura tworzenia componenmtu(iinny sposob wysweitlania rodzajow)

* + - 1. Zapoznaj się z dokumentacja <https://docs.microsoft.com/pl-pl/aspnet/core/mvc/views/view-components?view=aspnetcore-2.2>
      2. Ustaw jako projekt startowy firma.Portal WWW
      3. PPM na kontrollers i add mvc cotroller empty RodzajeMenuComponent
      4. 
      5. PPm na views->sklep i nowy folder on nazwie Components
      6. 
      7. PPm na komponents I add folder RodzajeMenuComponent
      8. PPM na RodzajeMenuComponent i add razor view empty o nazwie RodzajeMenuComponent
      9. Utworz kod tego componentu wzorując się na view->shared->rodzajeMenu

@model IEnumerable<Firma.Data.Data.Sklep.Rodzaj>

<div class="nav-wrapper container">

<a id="logo-container" href="#" class="brand-logo">Logo</a>

<ul class="right hide-on-med-and-down">

<li>

@Html.ActionLink("Strona glowna", "Index", "Home")

</li>

@foreach (var item in Model)

{

<li>

@Html.ActionLink(item.Nazwa, "Index", "Sklep", new { id = item.IdRodzaju },null)

</li>

}

</ul>

<ul id="nav-mobile" class="sidenav">

<li><a href="#">Navbar Link</a></li>

</ul>

<a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-icons">menu</i></a>

* + - 1. </div>
      2. Edytuj plik views->shared->\_SklepLayout i w sekcji nawigacyjnej dodaj wywowanie komponentu

@\* @await Html.PartialAsync("RodzajeMenu", (IEnumerable<Firma.Data.Data.Sklep.Rodzaj>) ViewBag.Rodzaj))

\*@

* + - 1. @await Component.InvokeAsync("RodzajeMenuComponent")

W coitrollers i sklepcontroller i zakomentoj // ViewBag.Rodzaj = await \_context.Rodzaj.ToListAsync();//zakomentowany viewbag bo uzuwam componetow

* + - 1. Na początku przy wejscou do witryny trzymaj poprane dane w sesji np. jedno polaczenie z baza danych

ZADANIA:

Zadanie  1

Przerobić wyświetlanie stron na components

zadanie 2

Przerobić aktualności na component

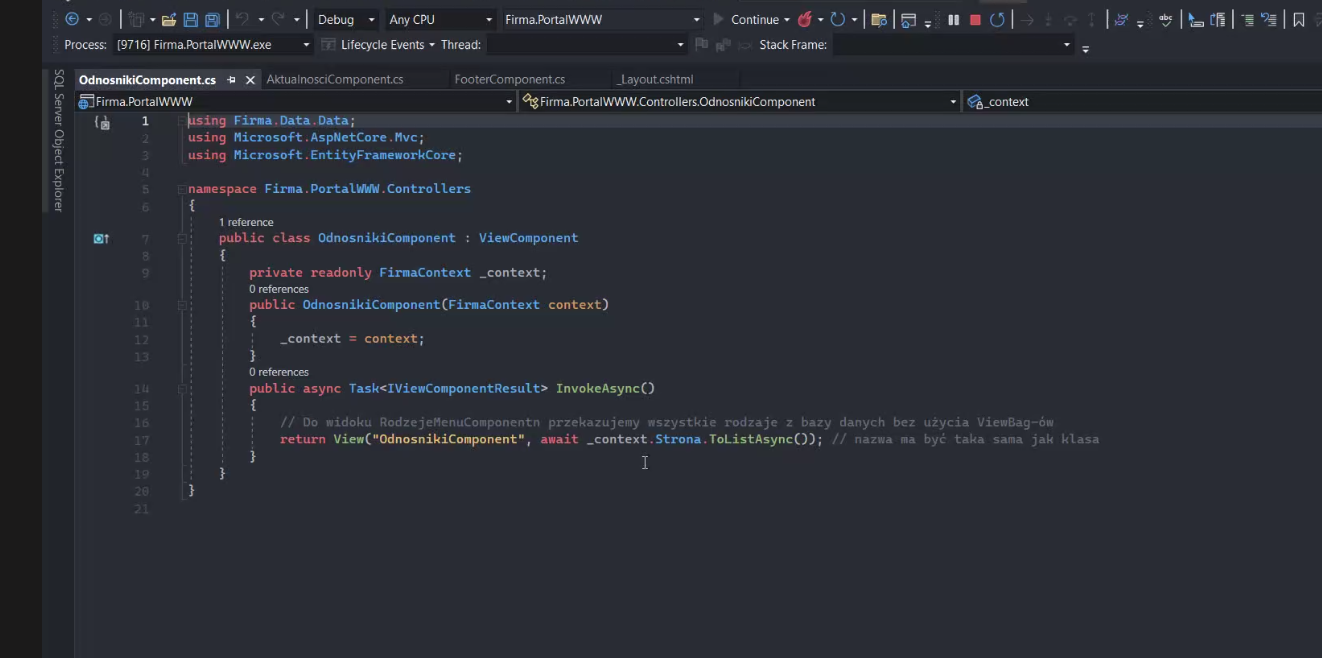
zadanie 3

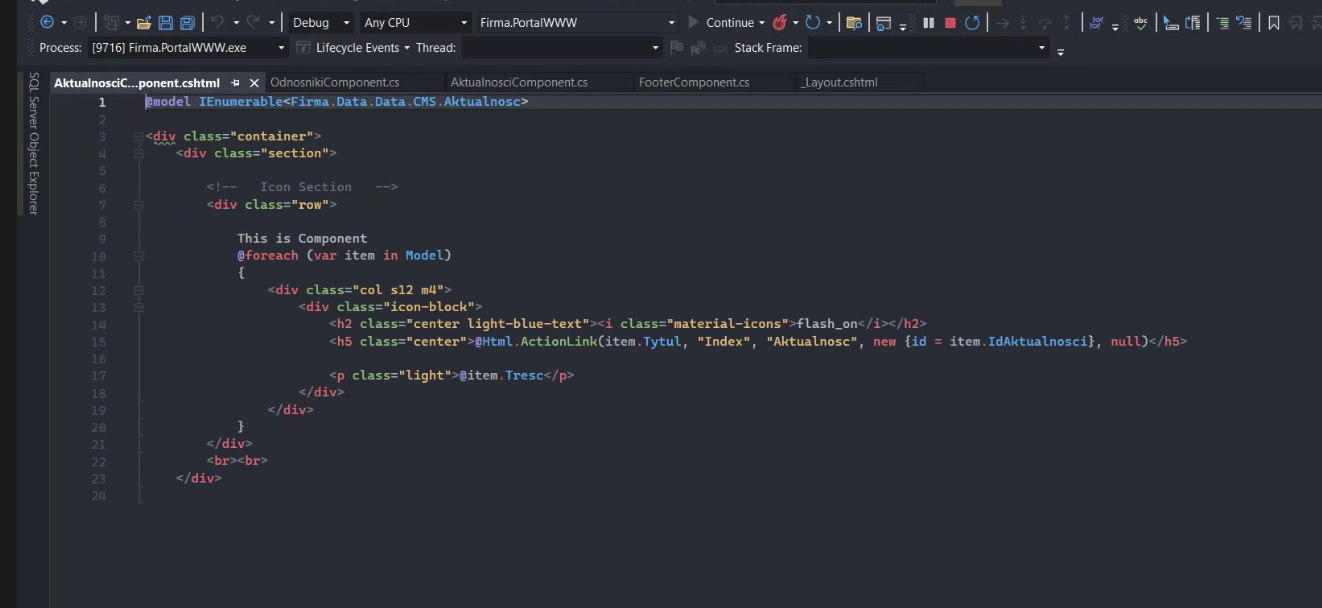
Poprawić wygląd menu w sklepie

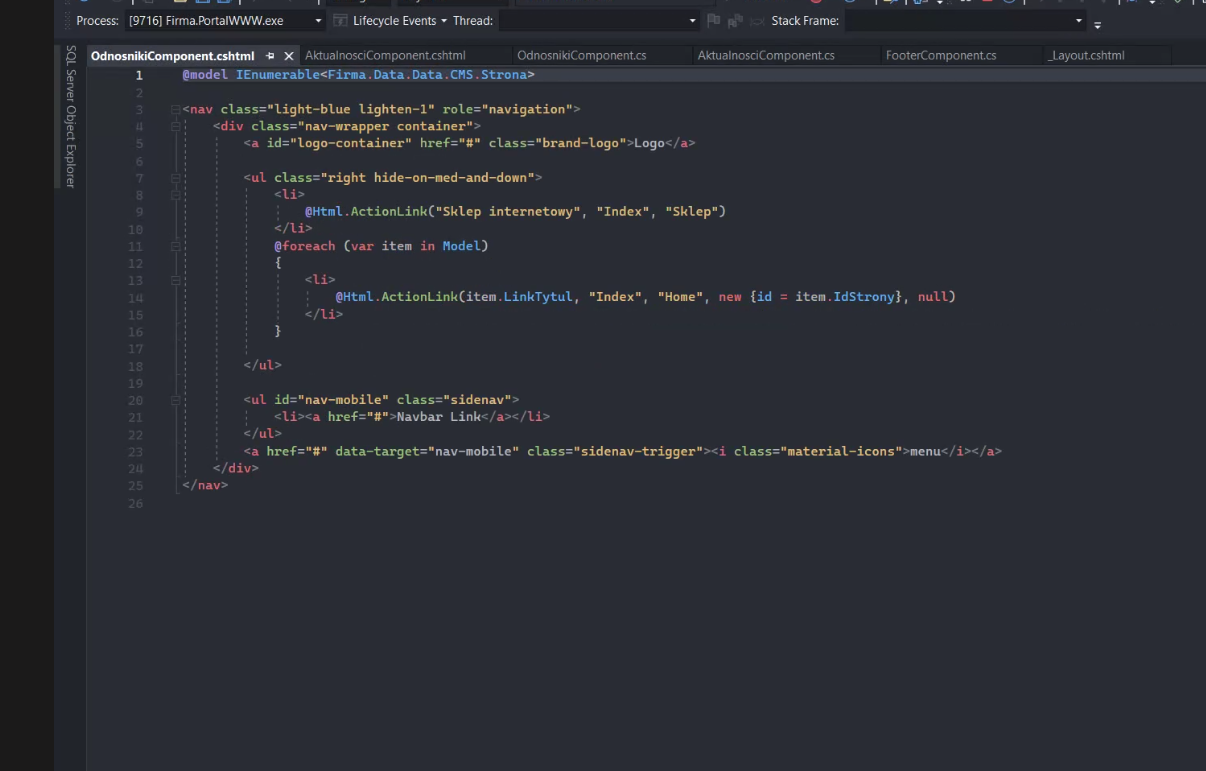
zadanie 4

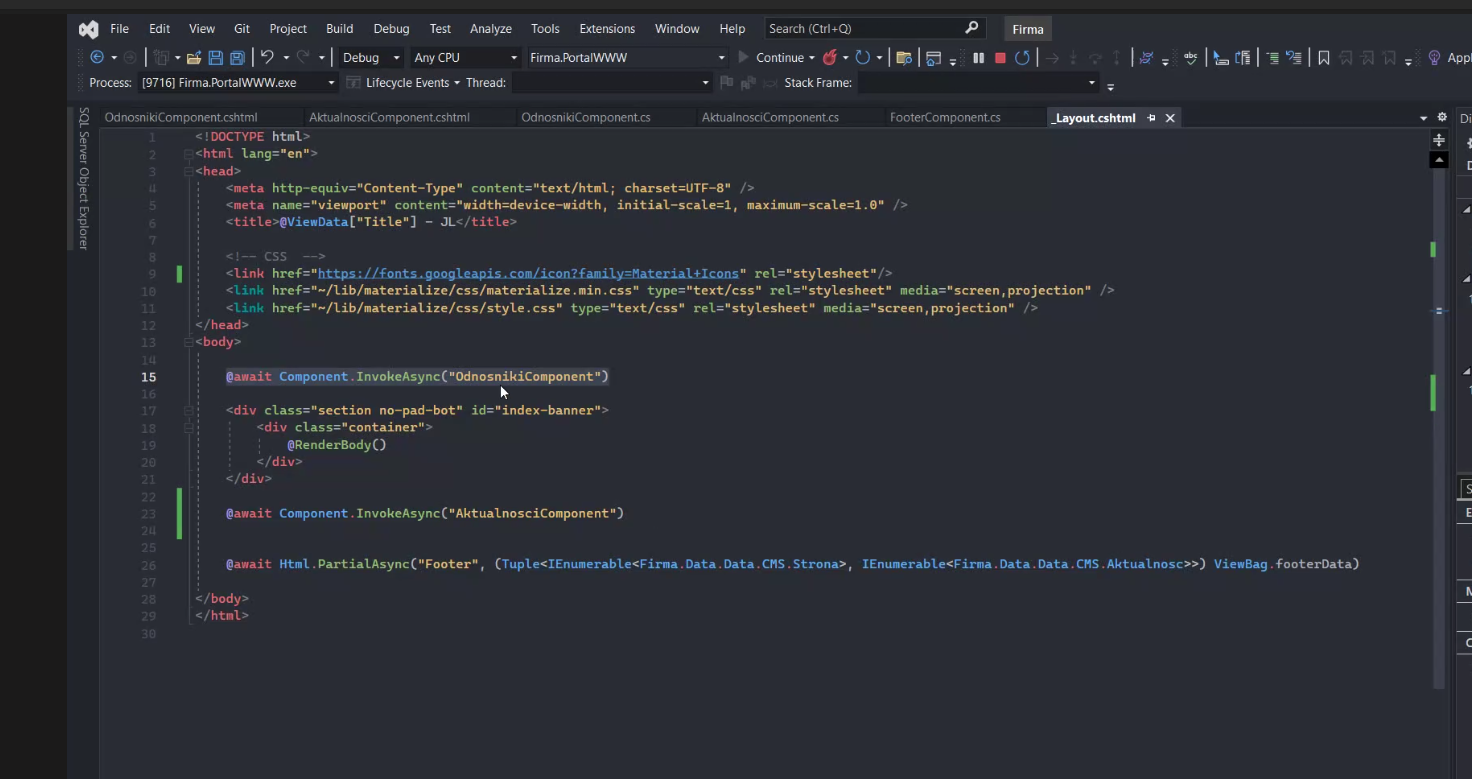
poprawić wygląd aktualności

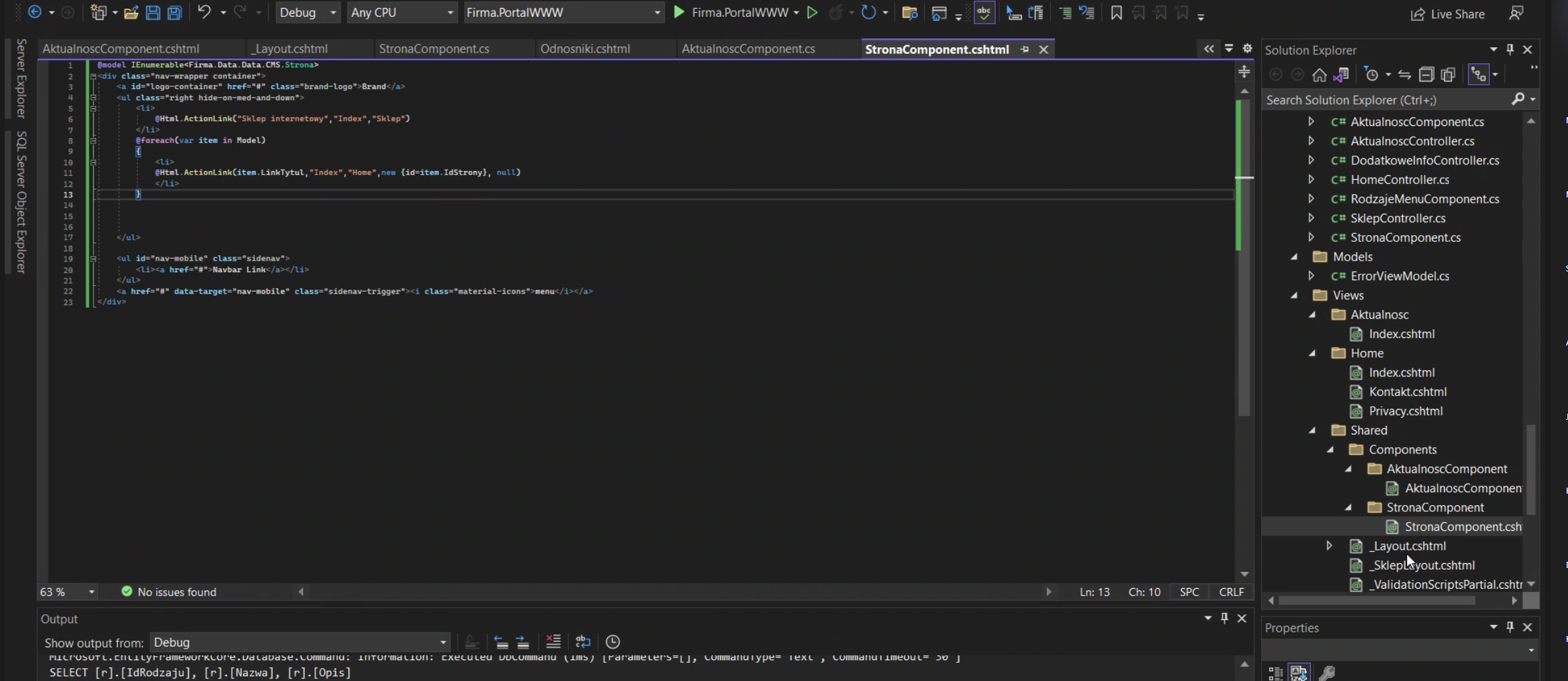




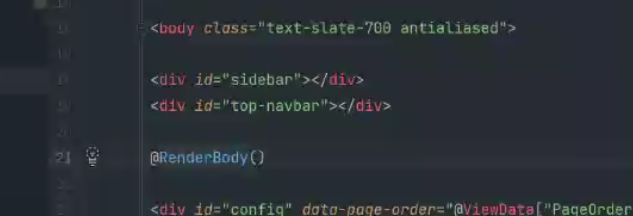


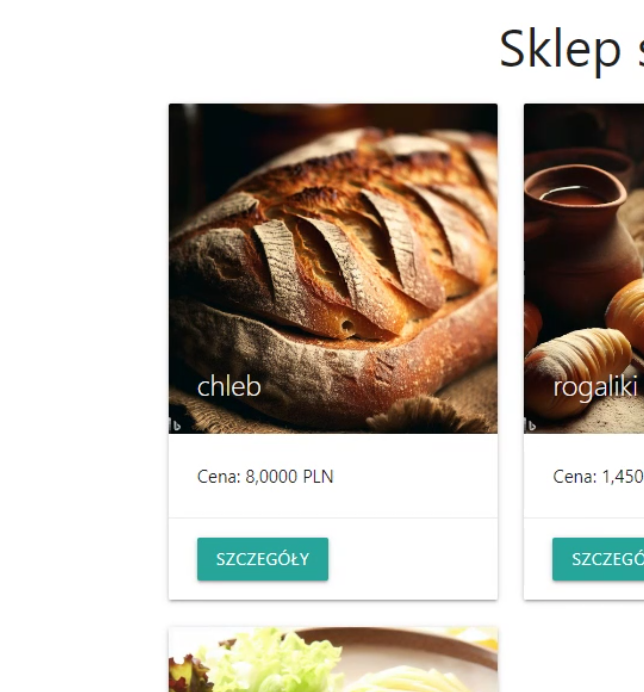


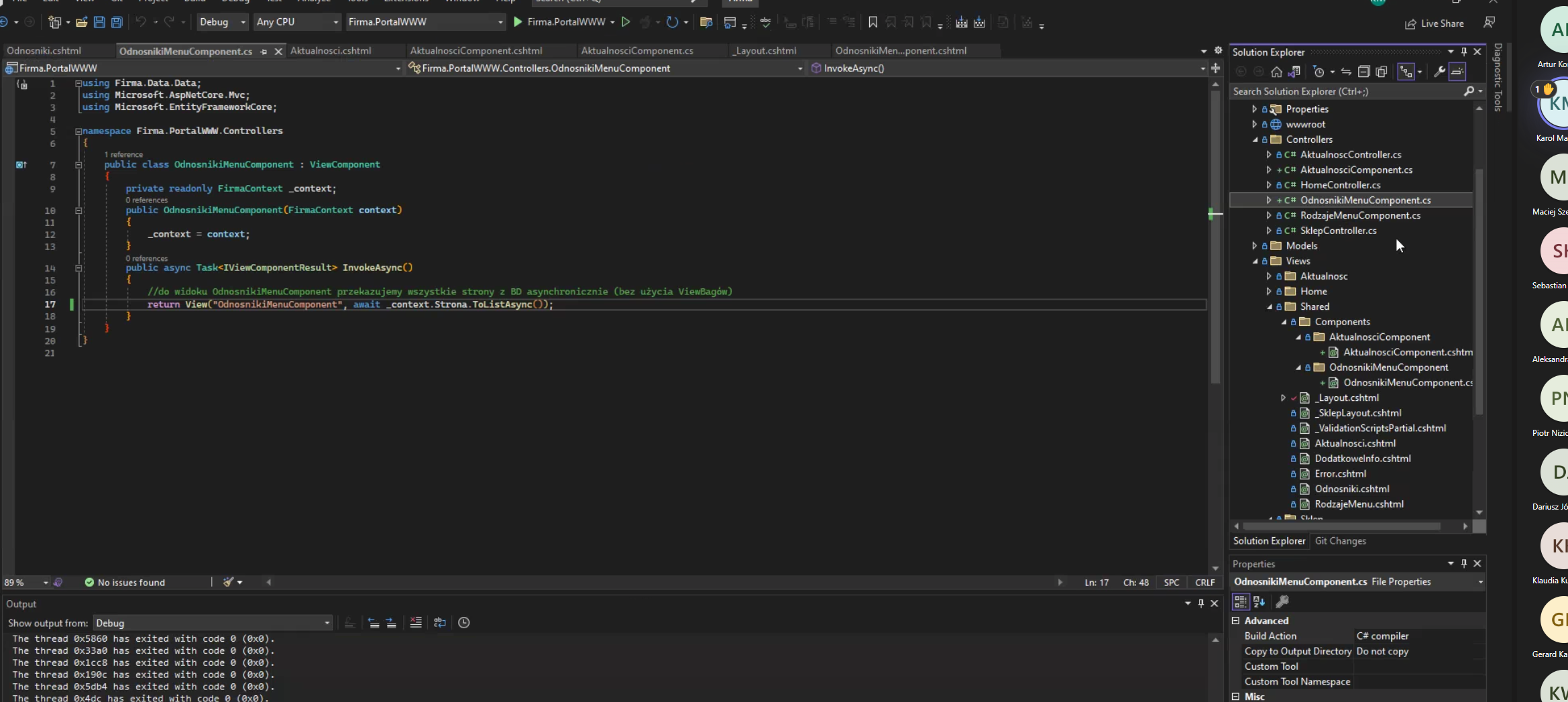




Viewjs

ViegData =thru //przekazuje zebywyswietlac np. czy strona ma tabs

04.06.2023



14.05

Foreign key – wypelniamy w desktopach okna modalne, combobox

W create utorz okno modalne do obslugi klucza obcego w indekcie wysweitlic

Rodzaj zrob samemu, dodawanie dodaru z kluczem obcym

Procedura tworzenia kontrolerów i widokow realizujacych operacje elementarne z dziedziczeniem wraz z obsluga kluczy opcymi ustaw jako projekt startowy firma intranet i wszystkie operacje wykonuj w intranecie

PPM na controlers i add kontroller empty o nazwie towarkontroller

PPM na views i add folder o nazwie towar

PPM na views->towar i add view o nazwie Index aldbo edytuj towar kontoler i PPM na Nazwe funckji index i dodaj widoku razor view ermpty

Edytuj views->shared->\_layout.cshtml i w sekcji navigacyjnej dodaj link do utworzonej strony index w kontrolerze towar

W projekcie firma.data ppm na data i add class Tentity

Zmien lub dodaj dziedziczenie we wczystkich classach z folderu CMS i sklep w firmie data na : Tentity

PPM na kontrollers i add class basecontroller

Edytuj towar kontroller i zmien w nim dzieczyczenie na baseconrtoller. Podobnie zmien dziecdziczenie we wszystkich klasach crud strona aktualnosc i rodzaj. Ponnadto zmodyfikuj kod tego kontrolera tak zeby nie powtarzal kodu

Edytuj views->towar->index i na pocztaku widoku pzez tabela dodaj akcje create

Edutyj kontrollers -> towarkontroller i dodaj tam funkcje create w dwoch odslonach (dwoch wersjach)

PPM na views->towar i add view o nazwie create

Przenies funckje create z towarcontroller do basecontroller

Dodaj w towarcontroller funckje ktora relizuje tzn POST do dodawania towary funkcja create z httppost

28.05

Przenieść z towarcontroller do baseController//index wyswietla wszystkie towary

public async Task<IActionResult> Index()

{

// z bazy danynych pobieray asynchronicznie liste wszystkich towarow

// var list = await \_context.Towar.Where(t=>t.IsActive ==true).ToListAsync();//where isActive

var list = await \_context.Towar.Include(t=>t.Rodzaj).ToListAsync();//where isActive

//tu linq ktore pobiera kulumny ktore sa potrzebne

//maper kolumn z bazy danych na widok ?

//i przekazujemy ja do widoku,! view model (tutaj view model pobrac do funkcji index) ktore pola pobierac potem mapper ktory mapuje co pobierac

//.Include(t=>t.Rodzaj). include zaladowanie klucza opbcego

return View(list);

}

Procedura:

Procedura towarzenia kontrolerów i widoków realizujących operacje elementarne z dziedziczeniem wraz z obsługą kluczy obcych.  
1. Ustaw jako projekt startowy Firma.Intranet i wszystkie poniższe operacje wykonuj w intrancecie.  
2. PPM na Controllers i Add Controller (Empty) o nazwie TowarController.  
3. PPM na Views i Add Folder o nazwie Towar.  
4. PPM na Views -> Towar i Add View (Empty) o nazwie Index - lub Edytuj Towar Controller i naciśnij PPM na nazwe funkcji Index i dodaj widok.  
5. Edytuj Views -> Shared -> \_Layout.cshtml i w sekcji nawigacyjnej dodaj link do utworzonej strony Index w kontrolerze Towar.  
6. W projkecie Firma.Data PPM na Data i Add Class TEntity.  
7. Zmień lub dodaj dziedziczenie we wszystkich klasach w forlderach CMS i Sklep (w Firma.Data) na TEntity.  
8. PPM na Controllers i Add Class o nazwie BaseController.   
9. Edytuj TowarController i zmień w nim dziedziczenia na BaseController. Podobnie zmień dziedziczenie we wszystkich klasach CRUD kontrolerowych, czyli Strona, Aktualnosc i Rodzaj.  
Ponadto zmodyfikuj kod tego kontrolera, tak żeby nie powtarzał kodu.  
10. Edytuj View -> Towar -> Index i na początku widoku przed tabelą dodaj akcje create.  
11. Edytuj Controllers -> TowarController i dodaj tam funkcje Create w dwóch odsłonach (dwóch wersjach).  
12. PPM na Views -> Towar i Add View o nazwie Create.  
13. Przenieś funckje Create z TowarController do BaseController.  
14. Dodaj w TowarController funkcje, która realizuje tzw POST do dodawania towaru, czyli funkcje create z [HTTPPost].

Tutaj:

//to jest kontroler bazowy z niego beda dziedziczyc wszystkie kontrolery realizujace operacje elementarne CRUD

//nad base controllerem jest IController ktory zawiera CRUDY interfejs

//klasa baseController uzywa typu generycznego TEntity pod ktory w przypadku towarow zostanie wyswietlony towar

public abstract class BaseController<TEntity> : Controller where TEntity : class

{

protected readonly FirmaContext \_context;//lepiej zrob propertisa zamiast protecked

public BaseController(FirmaContext context)

{

\_context = context;

}

//funckja jest abstrakcyjna jezeli nie ma bloku(nie wiemy jak ja napisac dopiero wiemy jak ja napisac w klasach dziedziczacych jezeli klasa ma co najm 1 funckje abstarkcyja to klassa musi byc abstrakcyjna)

public abstract Task<List<TEntity>> GetEntityList();//dla towarow bedzie to lista towarow

//index wyswietla wszystkie towary

public async Task<IActionResult> Index()

{

// z bazy danynych pobieray asynchronicznie liste wszystkich towarow

// var list = await \_context.Towar.Where(t=>t.IsActive ==true).ToListAsync();//where isActive

// var list = await \_context.Towar.Include(t => t.Rodzaj).ToListAsync();//where isActive//zamiast tego zrobimy funkcje abstrakcyjna

//tu linq ktore pobiera kulumny ktore sa potrzebne

//maper kolumn z bazy danych na widok ?

//i przekazujemy ja do widoku,! view model (tutaj view model pobrac do funkcji index) ktore pola pobierac potem mapper ktory mapuje co pobierac

//.Include(t=>t.Rodzaj). include zaladowanie klucza opbcego

var list = await GetEntityList();

return View(list);

}

//to funcja uruchamia sie przy wejsciu na strone dodawania nowego towaru

public IActionResult Create()

{

return View();//f create wywoluje,generuje widok o tej samej nazwie czyli create.cshtml

}

}

Wytnij z towarcontroller do base controller

//to jest funckja ktora wywoluje sie po kliknieciu w przycisk create, ale jest jeszcze z dziedziczenia

//[Bind("IdTowaru,Kod,Nazwa,Cena,FotoUrl,Opis,RodzajId zamiast tego tworz viewmodele i wtedy wszystko będę z viewmodelu !!

[HttpPost]

public async Task<IActionResult> Create([Bind("IdTowaru,Kod,Nazwa,Cena,FotoUrl,Opis,RodzajId")] Towar towar)

{

\_context.Add(towar);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

Basecontroller //to jest kontroler bazowy z niego beda dziedziczyc wszystkie kontrolery realizujace operacje elementarne CRUD

//nad base controllerem jest IController ktory zawiera CRUDY interfejs

//klasa baseController uzywa typu generycznego TEntity pod ktory w przypadku towarow zostanie wyswietlony towar

public abstract class BaseController<TEntity> : Controller where TEntity : class

{

protected readonly FirmaContext \_context;//lepiej zrob propertisa zamiast protecked

public BaseController(FirmaContext context)

{

\_context = context;

}

//funckja jest abstrakcyjna jezeli nie ma bloku(nie wiemy jak ja napisac dopiero wiemy jak ja napisac w klasach dziedziczacych jezeli klasa ma co najm 1 funckje abstarkcyja to klassa musi byc abstrakcyjna)

public abstract Task<List<TEntity>> GetEntityList();//dla towarow bedzie to lista towarow

//index wyswietla wszystkie towary

public async Task<IActionResult> Index()

{

// z bazy danynych pobieray asynchronicznie liste wszystkich towarow

// var list = await \_context.Towar.Where(t=>t.IsActive ==true).ToListAsync();//where isActive

// var list = await \_context.Towar.Include(t => t.Rodzaj).ToListAsync();//where isActive//zamiast tego zrobimy funkcje abstrakcyjna

//tu linq ktore pobiera kulumny ktore sa potrzebne

//maper kolumn z bazy danych na widok ?

//i przekazujemy ja do widoku,! view model (tutaj view model pobrac do funkcji index) ktore pola pobierac potem mapper ktory mapuje co pobierac

//.Include(t=>t.Rodzaj). include zaladowanie klucza opbcego

var list = await GetEntityList();//do widoku tyo przekazauje raz towary , raz pracownikow

return View(list);

}

//to funcja uruchamia sie przy wejsciu na strone dodawania nowego towaru

public IActionResult Create()

{

return View();//f create wywoluje,generuje widok o tej samej nazwie czyli create.cshtml

}

//to jest funckja ktora wywoluje sie po kliknieciu w przycisk create, ale jest jeszcze z dziedziczenia

[HttpPost]

public async Task<IActionResult> Create(TEntity entity)

{

await \_context.AddAsync(entity);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

}

}

public class TowarController : BaseController<Towar> //pod <TEntity> podstawiamy <Towar> z BaseController.cs dzieczic bo basze kontroler a pod tentity podstaw towear

{

//klasa towarcontroller dziedziczy po basecontroller podstawiajac za TEntity towar

public TowarController(FirmaContext context)

: base(context)

{

}

//w towarze napisz funkcje getentitylist

public override async Task<List<Towar>> GetEntityList()//towar zamiast <TEntity>//jak pracownicy kotroller to tylko to napisze

{

return await \_context.Towar.Include(t => t.Rodzaj).ToListAsync();//lub okno modalne wybor rodzajow

}

}

15. Przeniesc wszystkie elementy z towar kontroler do base controller tak żeby nie powtarzac kodu

16. Do klasy BaseCotroller dodaj funckje setSelectList

//funcjka setselectlist inicjalizuje dane do wyboru w combobosie jezeli tabela (encja) bedzie miala klucze obce to wtedy nadpiszemy te funckje w klasach dziedziczacych

// /jezeli tabela nie bedzie miala kluczy obcych to nie besdzuemy te funckje nadpisywac zwrocimy nulla

//jak ma klucze obce to nadpiszemy jak nie ma to zwrocimy nulla

public virtual Task SetSelectList()//tak funkcja ustawia dane do wyboru z comboboxa - selecta

{

return null;//gdy wszzystkie nasze tabele nie mialy klucze obce to wtedy abstacyjne, jak nie ma kluczy obcych to null

}

//to funcja uruchamia sie przy wejsciu na strone dodawania nowego towaru

public async Task<IActionResult> CreateAsync()

{

await SetSelectList();//przed wywolanie widoku inicjalizujemy comboboxy funkcja setselectlist

return View();//f create wywoluje,generuje widok o tej samej nazwie czyli create.cshtml

}

17 Edytuj klase towar controller i dodaj tam funckcje nadpisana setselectlist

public override async Task SetSelectList()

{

//bobieramy wszystkie rodzaje z bazy danych

var rodzaje = await \_context.Rodzaj.ToListAsync();//\_context.Rodzaj.Where(r => r.IsActive == true).ToListAsync();

//do select listy przekjazujemy te rodzaje kazemy wybierac id rodzaju a wyswietlamy nazwe

ViewBag.Rodzaje = new SelectList(rodzaje, "IdRodzaju", "Nazwa");

}

Edytuj view->Towar->create i w sekcji dotyczącej rodzaju dodac selecta

<div class="form-group">

<label asp-for="RodzajId" class="control-label"></label>

@\*<input asp-for="RodzajId" class="form-control" />\*@

<select asp-for="RodzajId" class="form-control" asp-items="ViewBag.Rodzaje"></select>

<span asp-validation-for="RodzajId" class="text-danger"></span>

</div>

Edytuj firma.data->sklep->towar.cs i w sekcji dotyczącej rodzaj id dodaj czytelny dla użytkownika display

//jeden do wielu realizacja klucza obcego

[Display(Name = "Wybierz rodzaj towaru")]

public int RodzajId { get; set; }

public Rodzaj Rodzaj { get; set; }

Procedura tworzenia koszyka towarów:

1 Otworz projekt z poprzednikj zajec

2 PPM na firma data->data->sklep i dodaj nowa klase o nazwie element koszyka(co jest w koszyku)

3 Edytuj firma.data->data->firmacontext i dodaj db seta element koszyka

4 PPM na models(firma portalwww-> models) i dodaj nowy folder o nazwie BussinessLogic

5 PPM na businessLogic i add class KoszykB (koszyk do bisnezowych)

Do projektu dodaj BaseController oraz obsluge minimum jednej klasy entities z kluczem obcym przypomocy BaseController(odpowiednika towarów) który powinien wystawiać kontrolerowidziedziczacemt wszystkich funkcji do realizacji funkcji elementarnych (to co dla towarów tutaj)

Edytuj plik program.cs i dodaj do tego pliku obsluge sessji

PPM na models i dodaj nowy folder sklep a w nim klase dane do koszyka

PPm na kontroller i add kontroller empty o nazwie koszyk kontroller (model logika biznesowa kontroller potem widok)DOKONAJ kompilacji projektu

PPM na views ->shared i dodajemy widok o nazwie (razor view empty) o nazwie \_KoszykLayout (wzorujac sie na skleplayput

Edytuj plik koszykkontroller i nacisnij ppm na nazwe funckji index dodajac widok index wyswietlajacy koszyk

Edytuj views->sklep\_>szczeguly i dodaj tam przycisk link do dodawania nowego towaru do koszyka

Ustaw jako project startowy firma.intranet

Dokonaj buildowania wszytskich projektow

Wywolaj konsole tools nudgetpacka console w konsoli ustaw jako default projekt firma.data upewnij sie ze projektem starowym jest firma.intranet a defaultowy firma.data

Sprawdz jaki masz ostatni numer migracji firma.data->migration

W konsoli wpisz add-migration m2

W konsoli wpisz update-database

Edytuj views->sklep\_>componets->rodzajemenucomponents i dodak link do kosyzka

(usuwanie z koszyka) okna modalne