Прізвище: Йовбак
Ім'я: Андріанна
Група: КНСП-11
Варіант: 18

Кафедра: САПР
Дисципліна: Семантичні Веб- та Грід-мережі
Перевірив: Корпильов Д.В.

ЗВІТ
до лабораторних робіт № 2-3
на теми «Застосування технології ASP.NET при розробці Інтернет-Магазину» та «Забезпечення взаємодії Інтернет-Магазину з базою даних з застосуванням технології ASP.Net»

Мета роботи: розробка веб-сайту з використанням технології ASP .NET Core та забезпечення взаємодії Інтернет-Магазину з базою даних на прикладі SQL Server.

Короткі теоретичні відомості:

Entity Framework Core (EF Core) - це остання версія програми Entity Framework від Microsoft. Він був спроектований як легкий, здатний до розширення та підтримання крос-платформної розробки застосунок. Він також був розроблений для простішого використання та покращення продуктивності порівняно з попередніми версіями Entity Framework.

EF Core - це об'єктно-реляційний відображувач (ORM). Об'єктно-реляційне відображення - це методика, яка дозволяє розробникам працювати з даними об'єктно-орієнтованим способом, виконуючи роботу, необхідну для відображення між об'єктами, визначеними мовою програмування застосунку, та даними, що зберігаються у реляційних джерелах даних. EF Core - це вдосконалення до ADO.NET, що надає розробникам автоматизований механізм доступу та зберігання даних у базі даних. Він призначений для використання з застосунками .NET Core. Однак також може бути використаний у стандартних застосунках на основі .NET 4.5 і вище. EF Core підтримує застосування як на Windows, так і на Linux і на Mac OS.

База даних (БД) – це організована структура, яка призначена для зберігання, зміни та обробки об’єктів, що пов’язані між собою спільною ознакою або властивістю.

Головним завданням БД є збереження великих обсягів інформації та надання доступу до неї користувачеві, а також надання можливості керування цими даними через прикладну програму. Прикладну програму називають системою керування або управління базою даних. Таким чином база даних складається зі збереженої інформації та системи керування цією інформацією.

База даних має володіти такими властивостями:

1. Безпека. Забезпечує захист від несанкціонованого доступу та копіювання.
2. Цілісність. В кожний момент часу дані, що зберігаються повинні бути повними та несуперечливими. На етапі розробки бази даних встановлюються обмеження цілісності даних (діапазон допустимих значень), за рахунок чого досягається цілісність бази даних.
3. Ефективність. Мінімальний час виконання запиту користувача.

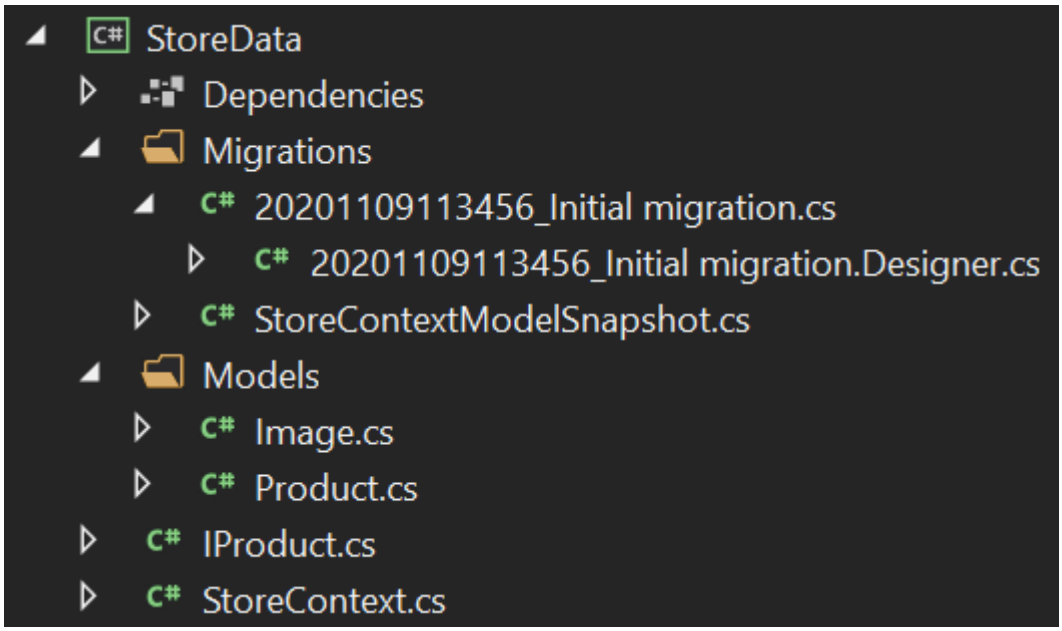
Бази даних розрізняють за структурою зберігання даних:

1. Ієрархічна. Ієрархічна структура може бути представлена у вигляді дерева, що складається з різних рівнів та має зв'язки типу "предок-нащадок". У такому випадку об'єкт може мати кілька нащадків або взагалі не мати нащадків, а кожен нащадок обов'язково має одного предка.
2. Мережева. Мережеві бази розширюють функціональність ієрархічних, тобто нащадки можуть тепер мати кілька предків, що дозволяє створювати складні відносини. Подаються у вигляді загального графа.
3. Реляційна. Реляційна база даних зберігає дані та зв'язки між даними за допомогою таблиць. Кожен рядок таблиці представляє окремий елемент даних.
4. Об'єктно-орієнтована. В об'єктно орієнтовній базі даних об'єкти зберігаються у вигляді моделей

Індивідуальне завдання: При розробленому макеті сайту, необхідно реалізувати різну функціональність, властиву Інтернет-Магазинам. Для того, щоб це зробити, одного тільки HTML уже не досить. Необхідно в рамках сайту реалізувати бізнес-логіку магазину. Підключення Веб-додатка на технології ASP.NET до бази даних, маніпуляція цими даними.

Код програми:

Створила додатковий проект StoreData, для опису моделей, необхідних для інтернет магазину:



Створила модель продукту інтернет-магазину:

```
namespace StoreData.Models
{
    12 references
    public class Product
    {
        4 references
        public int Id { get; set; }
        3 references
        public string Name { get; set; }
        5 references
        public IEnumerable<Image> Images { get; set; }
        3 references
        public string Color { get; set; }
        3 references
        public string Country { get; set; }
        1 reference
        public string Parameters { get; set; }
        1 reference
        public double Weight { get; set; }
        3 references
        public string Description { get; set; }
        1 reference
        public double Guarantee { get; set; }
        3 references
        public double Price { get; set; }
        2 references
        public string Type { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Text;

namespace StoreData.Models
{
    6 references
    public class Image
    {
        0 references
        public int Id { get; set; }
        0 references
        public Product Product { get; set; }
        5 references
        public string Url { get; set; }
        3 references
        public string AltText { get; set; }
    }
}
```

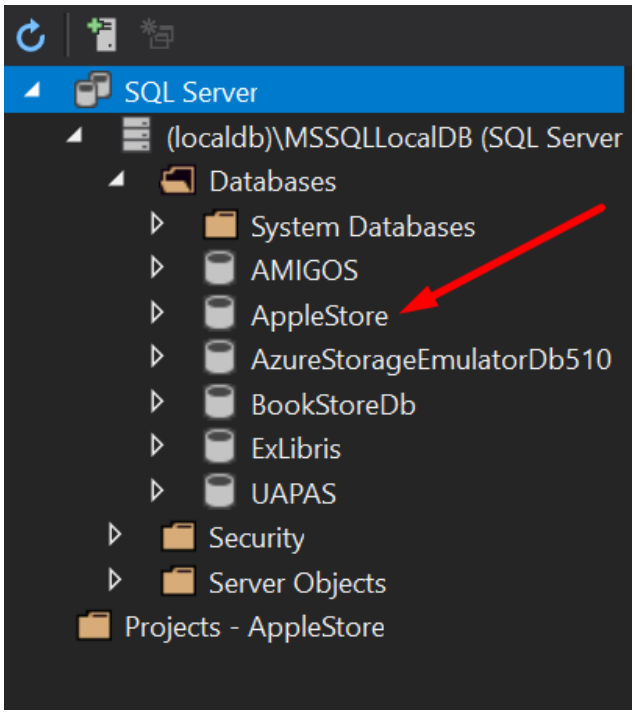
Додала «шлях підключення» до бази даних:

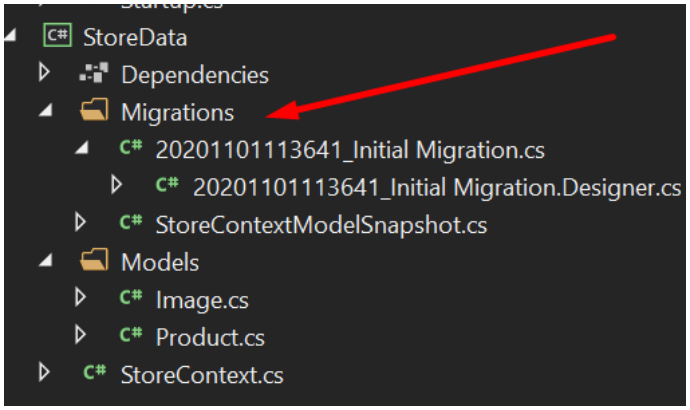
```
Schema: https://json.schemastore.org/appsettings
{
  "ConnectionStrings": {
    "StoreConnection": "Server=(localdb)\\MSSQLLocalDB;Database=AppleStore;Trusted_Connection=True;MultipleActiveResultSets=true;"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*"
}
```

Додала первинну міграцію та оновила базу даних:

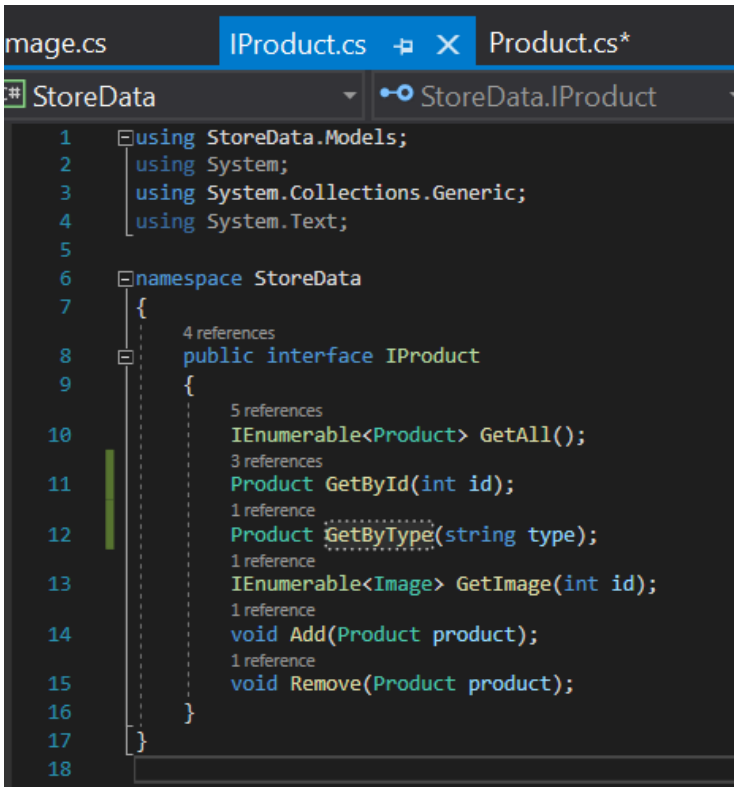
```
Package Manager Console
Package source: nuget.org    Default project: StoreData
PM> add-migration "Initial Migration"
Build started...
Build succeeded.
To undo this action, use Remove-Migration.
PM> update-database
Build started...
Build succeeded.
```

Після чого ми можемо побачити наші міграції та новостворену базу даних, разом з таблицями Products та Images:





Створила інтерфейс IProduct для роботи з продуктами:



Додала проект StoreService, в якому описала бізнес-логіку проекту та реалізувала клас ProductService:

```
ct.cs    ProductService.cs    StoreContext.cs    Im
eServices    StoreServices.ProductService
using Microsoft.EntityFrameworkCore;
using StoreData;
using StoreData.Models;
using System;
using System.Collections.Generic;
using System.Linq;

namespace StoreServices
{
    2 references
    public class ProductService : IProduct
    {
        private StoreContext _context;
        0 references
        public ProductService(StoreContext context)
        {
            _context = context;
        }
        1 reference
        public void Add(Product product)
        {
            _context.Add(product);
            _context.SaveChanges();
        }

        5 references
        public IEnumerable<Product> GetAll()
        {
            return _context.Products
                .Include(product => product.Images);
        }

        3 references
        public Product GetById(int id)
        {
            return GetAll()
                .FirstOrDefault(product => product.Id == id);
        }
```

```
StoreServices
    Dependencies
    ProductService.cs
```

```
        3 references
        public Product GetById(int id)
        {
            return GetAll()
                .FirstOrDefault(product => product.Id == id);
        }

        1 reference
        public Product GetByType(string type)
        {
            return GetAll()
                .FirstOrDefault(product => product.Type == type);
        }

        1 reference
        public IEnumerable<Image> GetImage(int id)
        {
            return GetById(id).Images;
        }

        1 reference
        public void Remove(Product product)
        {
            throw new NotImplementedException();
        }
    }
}
```

Додала моделі для відображення продуктів та списку продуктів каталогу:

```
Models
    Catalog
        ProductDetailModel.cs
        ProductIndexListingModel.cs
        ProductIndexModel.cs
```

```
namespace AppleStore.Models.Catalog
{
    3 references
    public class ProductIndexModel
    {
        3 references
        public IEnumerable<ProductIndexListingModel> Products { get; set; }
    }
}
```

```
namespace AppleStore.Models.Catalog
{
    4 references
    public class ProductDetailModel
    {
        1 reference
        public int Id { get; set; }
        2 references
        public string Name { get; set; }
        2 references
        public IEnumerable<Image> Images { get; set; }
        2 references
        public string Color { get; set; }
        2 references
        public string Country { get; set; }
        2 references
        public string Parameters { get; set; }
        2 references
        public double Weight { get; set; }
        2 references
        public string Description { get; set; }
        0 references
        public string Type { get; set; }
        2 references
        public double Guarantee { get; set; }
        2 references
        public double Price { get; set; }
    }
}
```

CatalogController:

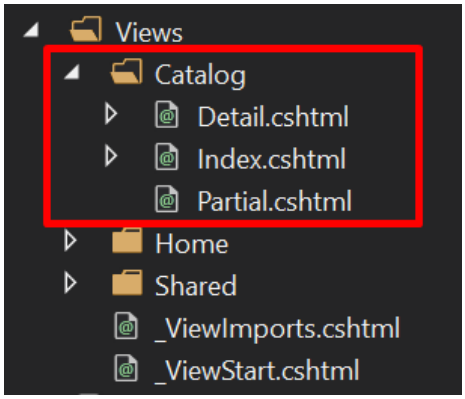
```
IProduct.cs    CatalogController.cs*    ProductService.cs
AppleStore    AppleStore.Controllers.Catalog
using AppleStore.Models.Catalog;
using Microsoft.AspNetCore.Mvc;
using StoreData;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace AppleStore.Controllers
{
    1 reference
    public class CatalogController: Controller
    {
        private IProduct _products;
        0 references
        public CatalogController(IProduct products)
        {
            _products = products;
        }
        0 references
        public IActionResult Index()
        {
            var productModels = _products.GetAll();
            var listingResult = productModels
                .Select(result => new ProductIndexListingModel
                {
                    Id = result.Id,
                    Image = result.Images.First(),
                    Name = result.Name,
                    Color = result.Color,
                    Country = result.Country,
                    Price = result.Price,
                    Description = result.Description
                });
            var model = new ProductIndexModel()
            {
                Products = listingResult
            };
            return View(model);
        }
```

```
0 references
public IActionResult Detail(int id)
{
    if (id == 0) id=1;
    var result = _products.GetById(id);

    var model = new ProductDetailModel()
    {
        Id = result.Id,
        Images = result.Images,
        Name = result.Name,
        Color = result.Color,
        Country = result.Country,
        Description = result.Description,
        Parameters = result.Parameters,
        Price = result.Price,
        Guarantee = result.Guarantee,
        Weight = result.Weight
    };

    return View(model);
}
```

```
public IActionResult Partial(string sort, string filter, string find)
{
    var productModels = _products.GetAll();

    var listingResult = productModels
        .Select(result => new ProductIndexListingModel
        {
            Id = result.Id,
            Image = result.Images.First(),
            Name = result.Name,
            Color = result.Color,
            Country = result.Country,
            Price = result.Price,
            Description = result.Description,
            Type = result.Type
        });

    if (sort != null)
    {
        if(sort=="name_up") listingResult = listingResult.OrderBy(el => el.Name);
        if(sort == "name_down") listingResult = listingResult.OrderByDescending(el => el.Name);
        if(sort == "price_up") listingResult = listingResult.OrderBy(el => el.Price);
        if(sort == "price_down") listingResult = listingResult.OrderByDescending(el => el.Price);
    }

    if (filter != null && filter != "all")
    {
        string[] filters = filter.Split('*');
        var filtered = listingResult.Where(e => filters.Contains(e.Type)).ToList();
        listingResult = filtered;
    }

    if(find != null) listingResult = listingResult.Where(el => el.Name.ToLower().Contains(find.ToLower()));

    var model = new ProductIndexModel()
    {
        Products = listingResult
    };
```

```
        return PartialView("Partial", model);
    }
}
}
```

Створила Views для зовнішнього представлення:



CatalogController.cs* | Detail.cshtml* | ProductService.cs | StoreContext.cs

```
@using AppleStore.Models.Catalog
@model AppleStore.Models.Catalog.ProductDetailModel
@{
    ViewBag.Title = "Detail";
}
<div class="container">
    <div class="row py-3">
        <a asp-controller="Catalog" asp-action="index" class="black">< Back to catalog</a>
    </div>
    <div class="row align-items-center py-2">
        <div class="col-6">
            <div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
                <div class="carousel-inner">

                    @{ int i = 0;
                        foreach (var asset in @Model.Images)
                        {
                            if (i == 0)
                            {
                                <div class="carousel-item mw-100 active">
                                    <div class="c-item">
                                        <img class="d-block w-100 mh-100"
                                            data-src="@asset.Url"
                                            alt="@asset.AltText" src="@asset.Url"
                                            data-holder-rendered="true">
                                    </div>
                                </div>
                            }
                            else
                            {
                                <div class="carousel-item mw-100">
                                    <div class="c-item">
                                        <img class="d-block w-100 mh-100"
                                            data-src="@asset.Url" alt="@asset.AltText"
                                            src="@asset.Url" data-holder-rendered="true">
                                    </div>
                                </div>
                            }
                            i++;
                        }
                    }
```

```
                    </div>

                    <a class="carousel-control-prev" href="#carouselExampleControls"
                       role="button" data-slide="prev">
                        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                        <span class="sr-only">Previous</span>
                    </a>
                    <a class="carousel-control-next" href="#carouselExampleControls"
                       role="button" data-slide="next">
                        <span class="carousel-control-next-icon" aria-hidden="true"></span>
                        <span class="sr-only">Next</span>
                    </a>
                </div>
            </div>

            <div class="col-6 py-4">
                <h2>@Model.Name</h2>
                <hr />
                <p>Color: @Model.Color </p>
                <p>Country: @Model.Country </p>
                <p>Parameters: @Model.Parameters</p>
                <p>Weight: @Model.Weight g</p>
                <p>Guarantee: @Model.Guarantee months</p>
                <p>Description: @Model.Description</p>
                <hr />
                <p>Price:<span class="display-5"> @Model.Price $</span></p>
                <hr />
                <button class="btn btn-primary px-5">Buy</button>
            </div>
        </div>
    </div>
```

logController.cs* | Index.cshtml* | Detail.cshtml* | StoreContext.cs

```
@using AppleStore.Models.Catalog
@model AppleStore.Models.Catalog.ProductIndexModel
@{
    ViewBag.Title = "Catalog";
}

<div class="container">
    <div class="row py-2">
        <div class="col-lg-7 col-12 filter d-flex align-items-center">
            <span class="d-none d-sm-block">Filter by:</span>
            <div class="pl-3 d-flex">
                <form>
                    <input id="all" type="checkbox" checked class="opacity-0 checkbox">
                    <label for="all">
                        <i class="fab display-5 fa-apple btn btn-secondary active"></i>
                    </label>
                    <input id="macbook" type="checkbox" class="opacity-0 checkbox">
                    <label for="macbook">
                        <i class="fas display-5 fa-laptop btn btn-secondary active"></i>
                    </label>
                    <input id="watch" type="checkbox" class="opacity-0 checkbox">
                    <label for="watch">
                        <i class="far display-5 fa-clock btn btn-secondary active"></i>
                    </label>
                    <input id="ipad" type="checkbox" class="opacity-0 checkbox">
                    <label for="ipad">
                        <i class="fas display-5 fa-tablet btn btn-secondary active"></i>
                    </label>
                    <input id="iphone" type="checkbox" class="opacity-0 checkbox">
                    <label for="iphone">
                        <i class="fas display-5 fa-mobile-alt btn btn-secondary active"></i>
                    </label>
                    <input id="earphones" type="checkbox" class="opacity-0 checkbox">
                    <label for="earphones">
                        <i class="fas display-5 fa-headphones-alt btn btn-secondary active"></i>
                    </label>
                </form>
            </div>
        </div>
```

```
        <div class="col-lg-3 col-12 offset-lg-2 pt-3 pt-lg-0">
            <div class="input-group mb-3 sort float-right">
                <div class="input-group-prepend btn btn-secondary py-0">
                    <label class="m-0" for="inputGroupSelect01" style="line-height: 35px;">Sort by</label>
                </div>
                <select class="custom-select" id="inputGroupSelect01">
                    <option selected>Choose...</option>
                    <option value="name_up">Name ↑</option>
                    <option value="name_down">Name ↓</option>
                    <option value="price_up">Price ↑</option>
                    <option value="price_down">Price ↓</option>
                </select>
            </div>
        </div>
    </div>
    <hr />
    <div class="row pt-2" id="products">

    </div>
</div>
```

```
61  <script src="~/lib/jquery/dist/jquery.js"></script>
62  <script>
63      var id = '#products';
64      let selected = '';
65      let search = '';
66      GetProducts();
67
68      $('select').on('change', function () {
69          selected = this.value;
70          GetProducts();
71      });
72
73      $('#search').on('keypress', function (e) {
74          if (e.charCode == 13) {
75              search = this.value;
76              GetProducts();
77              e.preventDefault();
78          }
79      });
80
81      $(".checkbox").change(function () {
82          if (this.id != 'all') $('#all').prop("checked", false);
83          if (this.id == 'all') {
84              $(".checkbox:checked").each(function () {
85                  $(this).prop("checked", false);
86              });
87              $('#all').prop("checked", true);
88          }
89
90          GetProducts();
91      });
92
93      function GetProducts() {
94          let filters_string = "";
95          $(".checkbox:checked").each(function (index) {
96              if (index != 0) filters_string += '*';
97              filters_string += this.id;
98          });
99
```

```
100         let filters = Object.assign({ sort: selected }, { filter: filters_string }, { find: search });
101
102         $(id).html('<img src="https://i.gifer.com/7pl0.gif" class="mx-auto" />');
103         $.ajax({
104             url: '/Catalog/Partial',
105             type: 'POST',
106             cache: false,
107             async: true,
108             dataType: "html",
109             beforeSend: function () {
110             },
111             data: filters
112         })
113             .done(function (result) {
114                 $(id).html(result);
115             }).fail(function (xhr) {
116                 console.log('error : ' + xhr.status + ' - ' + xhr.statusText + ' - ' + xhr.responseText);
117             });
118     }
119 </script>
```

```
CatalogController.cs*   Partial.cshtml   Index.cshtml*   Detail.cshtml*
1  @using AppleStore.Models.Catalog
2  @model AppleStore.Models.Catalog.ProductIndexModel
3
4  @foreach (var asset in @Model.Products)
5  {
6      <div class="col-lg-4 col-md-6 mb-4">
7          <a class="card h-100 p-1" asp-controller="Catalog" asp-action="Detail" asp-route-id="@asset.Id">
8              <span class="img-link">
9                  <img src="@asset.Image.Url" class="mw-100 mh-85" alt="@asset.Image.AltText" />
10             </span>
11             <div class="card-body pt-0">
12                 <h4 class="card-title display-6">
13                     @asset.Name
14                 </h4>
15                 <h5 class="display-6 weight-400">@asset.Price$</h5>
16                 <p class="card-text">@asset.Description</p>
17             </div>
18         </a>
19     </div>
20 }
```

Наповнила даними БД (таблиці зображення та продукти):

dbo.Images [Data] CatalogController.cs* Partial.cshtml Index.cshtml*

Max Rows: 1000

Id	ProductId	Url	AltText
2	1	/img/macbook...	MacBook Air G...
3	1	/img/macbook...	MacBook Air G...
4	1	/img/macbook...	MacBook Air G...
5	2	/img/macbook...	MacBook Air Gr...
6	2	/img/macbook...	MacBook Air Gr...
7	2	/img/macbook...	MacBook Air Gr...
8	3	/img/macbook...	MacBook Air Sil...
9	3	/img/macbook...	MacBook Air Sil...
10	3	/img/macbook...	MacBook Air Sil...
11	4	/img/macbook...	MacBook Air Gr...
12	4	/img/macbook...	MacBook Air Gr...
13	5	/img/macbook...	MacBook Air Sil...
14	5	/img/macbook...	MacBook Air Sil...
15	6	/img/watch/pin...	Watch Series 6 ...
16	6	/img/watch/pin...	Watch Series 6 ...
17	7	/img/watch/red...	Watch Series 6 ...
18	7	/img/watch/red...	Watch Series 6 ...
19	8	/img/watch/blu...	Watch Series 6 ...
20	8	/img/watch/blu...	Watch Series 6 ...
21	9	/img/ipad/gold...	iPad Gold
22	9	/img/ipad/gold...	iPad Gold
23	10	/img/ipad/silve...	iPad Silver
24	10	/img/ipad/silve...	iPad Silver

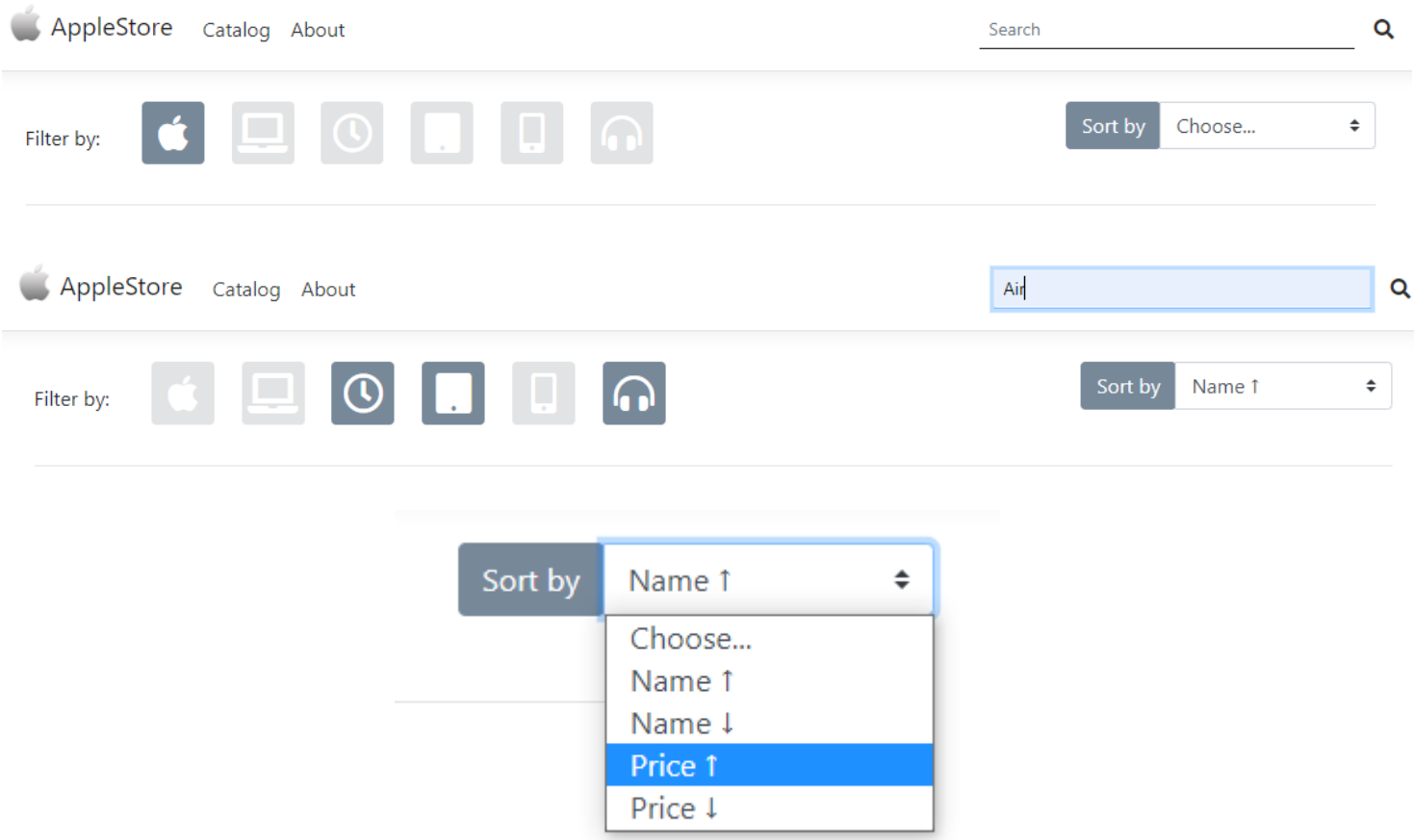
dbo.Products [Data] | dbo.Images [Data] | CatalogController.cs* | Partial.cshtml

Max Rows: 1000

Id	Name	Color	Count...	Parameters	Weig...	Description	G...	Price	Type
1	MacBook Air 1...	Gold	China	30.41 x 21.24 x ...	1,29	Screen 13.3 "IP...	12	1299	macbook
2	MacBook Air 1...	Gray	China	30.41 x 21.24 x ...	1,29	Screen 13.3 "IP...	12	1299	macbook
3	MacBook Air 1...	Silver	China	30.41 x 21.24 x ...	1,29	Screen 13.3 "IP...	12	1299	macbook
4	MacBook Pro 1...	Gray	USA	304.1 x 212.4 x ...	1,37	Screen 13.3 "IP...	24	1699	macbook
5	MacBook Pro 1...	Silver	USA	304.1 x 212.4 x ...	1,37	Screen 13.3 "IP...	24	1799	macbook
6	Watch Series 6 ...	Pink	USA	44 x 38 x 10.4	0,365	Wireless charge...	12	399	watch
7	Watch Series 6 ...	Red	USA	44 x 38 x 10.4	0,365	Wireless charge...	12	499	watch
8	Watch Series 6 ...	Blue	USA	44 x 38 x 10.4	0,365	Wireless charge...	12	299	watch
9	iPad 10.2" 2020...	Gold	USA	174.1 x 250.6 x ...	0,49	A12 Bionic proc...	12	899	ipad
10	iPad 10.2" 2020...	Silver	USA	174.1 x 250.6 x ...	0,49	A12 Bionic proc...	12	999	ipad
11	iPad 10.2" 2020...	Gray	USA	174.1 x 250.6 x ...	0,49	A12 Bionic proc...	12	799	ipad
12	iPhone 12 128...	Black	USA	71.5 x 146.7 x 7...	0,13	Communicatio...	12	1299	iphone
13	iPhone 12 128...	Blue	USA	71.5 x 146.7 x 7...	0,13	Communicatio...	12	1299	iphone
14	iPhone 11 64G...	Green	USA	71.5 x 146.7 x 7...	0,13	Communicatio...	12	999	iphone
15	iPhone 11 64G...	Purple	USA	71.5 x 146.7 x 7...	0,13	Communicatio...	12	999	iphone
16	iPhone 11 64G...	Yellow	USA	71.5 x 146.7 x 7...	0,13	Communicatio...	12	999	iphone
17	iPhone SE 64GB...	Red	USA	71.5 x 146.7 x 7...	0,13	Communicatio...	12	799	iphone
18	iPhone SE 64GB...	White	USA	71.5 x 146.7 x 7...	0,13	Communicatio...	12	799	iphone
19	AirPods with Ch...	White	USA	44.3 ? 21.3 ? 53...	0,048	AirPods sensor...	6	199	earphones

Результат виконання програми:

Панель з навігацією, пошуком, фільтрами та кнопкою сортування:



MacBook Air 13" 2020 Gold

1299$

Screen 13.3" IPS (2560x1600), glossy / Intel Core i5 (1.1 - 3.2 GHz) / RAM 8 GB / SSD 256 GB / Intel Iris Plus Graphics / Wi-Fi / Bluetooth / macOS Catalina / 1.29 kg / gold

MacBook Air 13" 2020 Gray

1299$

Screen 13.3" IPS (2560x1600), glossy / Intel Core i5 (1.1 - 3.2 GHz) / RAM 8 GB / SSD 256 GB / Intel Iris Plus Graphics / Wi-Fi / Bluetooth / macOS Catalina / 1.29 kg / gray

MacBook Air 13" 2020 Silver

1299$

Screen 13.3" IPS (2560x1600), glossy / Intel Core i5 (1.1 - 3.2 GHz) / RAM 8 GB / SSD 256 GB / Intel Iris Plus Graphics / Wi-Fi / Bluetooth / macOS Catalina / 1.29 kg / silver

MacBook Pro 13" Retina Space Gray

1699$

Screen 13.3" IPS (2560x1600), glossy / Intel Core i5 (2.4 GHz (2.4 - 4.1 GHz) / RAM 8 GB / SSD 512 GB / Intel Iris Plus Graphics 655 / Wi-Fi / Bluetooth / macOS Mojave / 1.37 kg / gray

MacBook Pro 13" Retina Silver

1799$

Screen 13.3" IPS (2560x1600), glossy / Intel Core i5 (2.4 GHz (2.4 - 4.1 GHz) / RAM 8 GB / SSD 512 GB / Intel Iris Plus Graphics 655 / Wi-Fi / Bluetooth / macOS Mojave / 1.37 kg / silver

Watch Series 6 GPS 44mm Pink

399$

Wireless charge. Alarm clock. Built-in microphone. Protection against water dust. Control of daily activity. With a compass. Stopwatch. Messages. Sleep tracking. Camera control. Music player control

Watch Series 6 GPS 44mm Red

499$

Wireless charge. Alarm clock. Built-in microphone. Protection against water dust. Control of daily activity. With a compass. Stopwatch. Messages. Sleep tracking. Camera control. Music player control

Watch Series 6 GPS 44mm Blue

299$

Wireless charge. Alarm clock. Built-in microphone. Protection against water dust. Control of daily activity. With a compass. Stopwatch. Messages. Sleep tracking. Camera control. Music player control

iPad 10.2" 2020 Gold

899$

A12 Bionic processor with 64-bit architecture. Neural Engine System. Touch ID. Fingerprint-resistant coating. Apple Pencil (1st generation) support. Wi-Fi 802.11a / b / g / n / ac support for two bands (2.4 GHz and 5 GHz) with MIMO technology

iPad 10.2" 2020 Silver

999$

A12 Bionic processor with 64-bit architecture. Neural Engine System. Touch ID. Fingerprint-resistant coating. Apple Pencil (1st generation) support. Wi-Fi 802.11a / b / g / n / ac support for two bands (2.4 GHz and 5 GHz) with MIMO technology

iPad 10.2" 2020 Gray

799$

A12 Bionic processor with 64-bit architecture. Neural Engine System. Touch ID. Fingerprint-resistant coating. Apple Pencil (1st generation) support. Wi-Fi 802.11a / b / g / n / ac support for two bands (2.4 GHz and 5 GHz) with MIMO technology

iPhone 12 128GB Black

1299$

Communication standard: 5G (sub6) / 5G(E) 5G (mmWave) / LAN(E) / WiMax 4G (LTE) 5G. Screen diagonal - 6.1. The number of megapixels of the main camera is - 12 Mp + 12 Mp. The number of megapixels of the front camera is 12 megapixels. Display resolution - 2532x1170. Built-in memory: 128 GB

iPhone 12 128GB Blue

1299$

Communication standard: 5G (sub6) / 5G(E) 5G (mmWave) / LAN(E) / WiMax 4G (LTE) 5G. Screen diagonal - 6.1. The number of megapixels of the main camera is - 12 Mp + 12 Mp. The number of megapixels of the front camera is 12 megapixels. Display resolution - 2532x1170. Built-in memory: 128 GB

iPhone 11 64GB Green

999$

Communication standard: 5G (sub6) / 5G(E) 5G (mmWave) / LAN(E) / WiMax 4G (LTE) 5G. Screen diagonal - 6.1. The number of megapixels of the main camera is - 12 Mp + 12 Mp. The number of megapixels of the front camera is 12 megapixels. Display resolution - 2532x1170. Built-in memory: 128 GB

iPhone 11 64GB Purple

999$

Communication standard: 5G (sub6) / 5G(E) 5G (mmWave) / LAN(E) / WiMax 4G (LTE) 5G. Screen diagonal - 6.1. The number of megapixels of the main camera is - 12 Mp + 12 Mp. The number of megapixels of the front camera is 12 megapixels. Display resolution - 2532x1170. Built-in memory: 128 GB

iPhone 11 64GB Yellow

999$

Communication standard: 5G (sub6) / 5G(E) 5G (mmWave) / LAN(E) / WiMax 4G (LTE) 5G. Screen diagonal - 6.1. The number of megapixels of the main camera is - 12 Mp + 12 Mp. The number of megapixels of the front camera is 12 megapixels. Display resolution - 2532x1170. Built-in memory: 128 GB

iPhone SE 64GB Red

799$

Communication standard: 5G (sub6) / 5G(E) 5G (mmWave) / LAN(E) / WiMax 4G (LTE) 5G. Screen diagonal - 6.1. The number of megapixels of the main camera is - 12 Mp + 12 Mp. The number of megapixels of the front camera is 12 megapixels. Display resolution - 2532x1170. Built-in memory: 128 GB

iPhone SE 64GB White

799$

Communication standard: 5G (sub6) / 5G(E) 5G (mmWave) / LAN(E) / WiMax 4G (LTE) 5G. Screen diagonal - 6.1. The number of megapixels of the main camera is - 12 Mp + 12 Mp. The number of megapixels of the front camera is 12 megapixels. Display resolution - 2532x1170. Built-in memory: 128 GB

AirPods with Charging Case

199$

AirPods come in one each earbuds. Dual directional microphones. Dual optical sensors. Motion recognition accelerometer. Voice activity detection accelerometer

AppleStore Catalog About

Search

Filter by:

Sort by Price ↑

AirPods with Charging Case

199$

AirPods sensors (in each earbud), Dual directional microphones, Dual optical sensors, Motion recognition accelerometer, Voice activity detection accelerometer

Watch Series 6 GPS 44mm Blue

299$

Wireless charger, Alarm clock, Built-in microphone, Protection against water, dust, Control of daily activity, With a compass Stopwatch, Messages, Sleep tracking, Camera control, Music player control

Watch Series 6 GPS 44mm Pink

399$

Wireless charger, Alarm clock, Built-in microphone, Protection against water, dust, Control of daily activity, With a compass Stopwatch, Messages, Sleep tracking, Camera control, Music player control

Watch Series 6 GPS 44mm Red

499$

Wireless charger, Alarm clock, Built-in microphone, Protection against water, dust, Control of daily activity, With a compass Stopwatch, Messages, Sleep tracking, Camera control, Music player control

© 2020 - AppleStore - Privacy

AppleStore Catalog About

Search

« Back to catalog

MacBook Air 13" 2020 Gray

Color: Gray

Country: China

Parameters: 30.41 x 21.24 x 1.61

Weight: 1.29 g

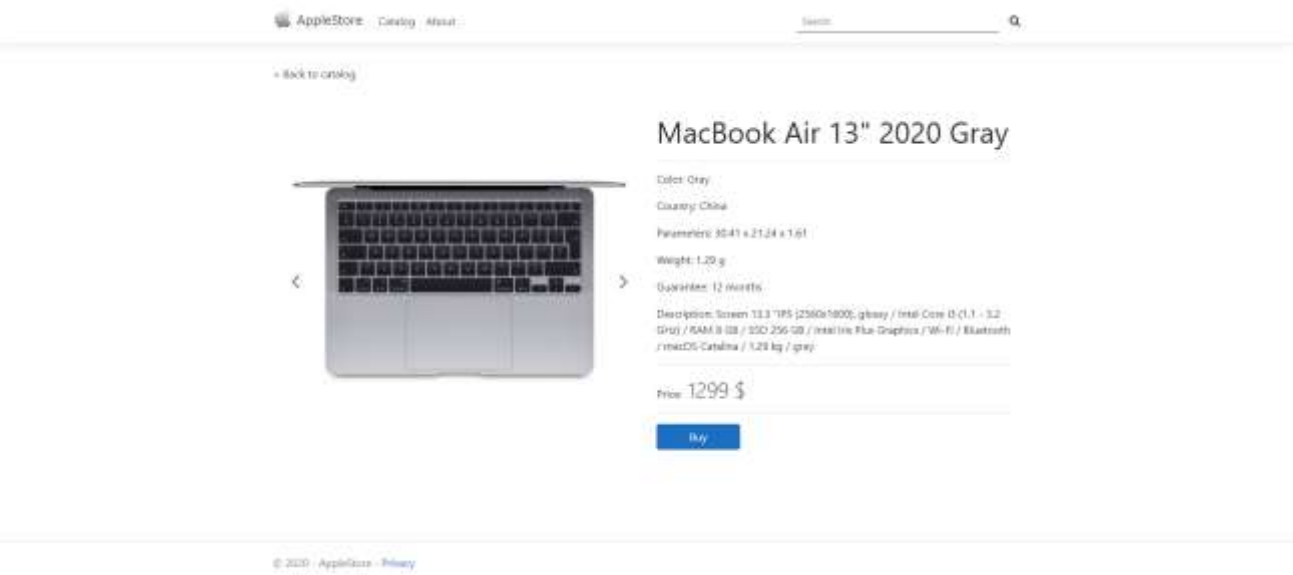
Guarantee: 12 months

Description: Screen 13.3 "IPS (2560x1600), glossy / Intel Core i5 (1.1 - 3.2 GHz) / RAM 8 GB / SSD 256 GB / Intel Iris Plus Graphics / Wi-Fi / Bluetooth / macOS Catalina / 1.29 kg / gray

Price: 1299 $

Buy

© 2020 - AppleStore - Privacy



Аналіз результатів, висновки: Під час виконання лабораторної роботи було реалізовано різну функціональність, властиву Інтернет-Магазинам (відображення каталогу продуктів, зображення їх детальної інформації, можливість пошуку, сортування та фільтрування даних) при розробленому макеті сайту. В рамках сайту було реалізовано бізнес-логіку магазину, підключено Веб-додаток на технології ASP.NET Core до бази даних. За допомогою бібліотек jQuery та Bootstrap 4 було розроблено юай частину інтернет магазину.