

## WebShopApp - Част 7- Добавяне на статистическа информация

**Статистически модул** - Добавяне на статистическа информация за клиенти, поръчки и продукти.

Ще използваме менюто **Statistic Module**

WebShopApp	Home	Privacy	Products	My Orders	Statistic Modul ▾	Hello admin!
					Clients Orders Statistic	
Count Clients		Count Products	Count Orders		Total Sum Orders	
1		2	9		9040.00	

В `_Layout.cshtml` добавяме:

```
if ((this.User.Identity.IsAuthenticated) && (this.User.IsInRole("Administrator")))
{
    <li class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button" id="dropdownMenuButton1"
            data-bs-toggle="dropdown" aria-expanded="false">
            Statistic Modul
        </button>

        <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton1">
            <li><a class="dropdown-item" href="/Client/Index">Clients</a></li>
            <li><a class="dropdown-item" href="/Order/Index">Orders</a></li>
            <li><a class="dropdown-item" href="/Statistic/Index">Statistic</a></li>
        </ul>
    </li>
}
```

Ще направим статистика за:

- Брой регистрирани потребители на сайта
- Брой продукти
- Брой направени поръчки
- Обща сума на направените поръчки

Ще използваме отделен контролер за статистическата информация и ще имплементираме `IStatisticsService` за него. Първо трябва да създадем интерфейса `IStatisticsService` в папка **Contracts**.

WebShopApp

Home

Privacy

Products

My Orders

Statistic Modul

Hello admin!




Lc

All Orders

Clients

Orders

Statistic

OrderDate	User		Picture	Quantity	Price	Discount	TotalPrice
02-Nov-2023 11:49	toto	LG 123		1	1000.00	20.00	800.00
02-Nov-2023 03:02	admin	LG 123		2	1000.00	20.00	1600.00
02-Nov-2023 02:42	admin	FF 123		2	1230.00	0.00	2460.00

```
public interface IStatisticService
{
    int CountProducts();
    int CountClients();
    int CountOrders();
    decimal SumOrders();
}
```

В папка Services в клас StatisticsService имплементираме методите на интерфейса като преди това инжектираме ApplicationDbContext в конструктора.

```
public class StatisticService : IStatisticService
{
    private readonly ApplicationDbContext _context;

    public StatisticService(ApplicationDbContext context)
    {
        _context = context;
    }

    public int CountClients()
    {
        return _context.Users.Count() - 1;
    }

    public int CountOrders()
    {
        return _context.Orders.Count();
    }

    public int CountProducts()
    {
        return _context.Products.Count();
    }

    public decimal SumOrders()
    {
        var suma = _context.Orders.Sum(x => x.TotalPrice);
        return suma;
    }
}
```

В Program.cs регистрираме създадените сървиси.

```
builder.Services.AddTransient<ICategoryService, CategoryService>();
builder.Services.AddTransient<IBrandService, BrandService>();
builder.Services.AddTransient<IProductService, ProductService>();
builder.Services.AddTransient<IOrderService, OrderService>();
builder.Services.AddTransient<IStatisticService, IStatisticService>();
```

Създаваме вю модел StatisticVM в папка Models/Statistic

```
public class StatisticVM
{
    [Display(Name = "Count Clients")]
    public int CountClients { get; set; }
    [Display(Name = "Count Products")]
    public int CountProducts { get; set; }
    [Display(Name = "Count Orders")]
    public int CountOrders { get; set; }
    [Display(Name = "Total Sum Orders")]
    public decimal SumOrders { get; set; }
}
```

Създаваме StatisticsController като инжектираме IStatisticsService и имплементираме екшън Index.

```
public class StatisticController : Controller
{
    private readonly IStatisticService statisticsService;

    public StatisticController(IStatisticService statisticsService)
    {
        this.statisticsService = statisticsService;
    }

    public IActionResult Index()
    {
        StatisticVM statistics = new StatisticVM();

        statistics.CountClients = statisticsService.CountClients();
        statistics.CountProducts = statisticsService.CountProducts();
        statistics.CountOrders = statisticsService.CountOrders();
        statistics.SumOrders = statisticsService.SumOrders();

        return View(statistics);
    }
}
```

Остана да създадем вю на екшъна Index на StatisticsController

```
@using WebShopApp.Models.Statistic
@model StatisticVM

@{
    ViewData["Title"] = "Statistics";
}

<h4 class="text-center">Statistic</h4>
<div class="text-center">
    <div class="col-xl-8">
        <div class="card-group pb-0">
            <div class="card-columns text-center">
            </div>
        </div>
        <table class="table table-striped " style="width:150%">
            <thead>
                <tr>
                    <th>
                        @Html.DisplayNameFor(model => model.CountClients)
                    </th>
                    <th>
                        @Html.DisplayNameFor(model => model.CountProducts)
                    </th>
                    <th>
                        @Html.DisplayNameFor(model => model.CountOrders)
                    </th>
                    <th>
                        @Html.DisplayNameFor(model => model.SumOrders)
                    </th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>
                        @Html.DisplayFor(model => model.CountClients)
                    </td>
                    <td>
                        @Html.DisplayFor(model => model.CountProducts)
                    </td>
                    <td>
                        @Html.DisplayFor(model => model.CountOrders)
                    </td>
                    <td>
                        @Html.DisplayFor(model => model.SumOrders)
                    </td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
```

Тествайте приложението с администраторски профил.

WebShopApp   Home   Privacy   Products   My Orders   **Statistic Modul**   Hello admin!   Log

Statistic			
Count Clients	Count Products	Count Orders	Total Sum Orders
1	2	9	9040.00

Разгледайте работата на приложението и направете промени по дизайна, например редактирайте имената на страниците и табове, уеднаквете хипервръзките за нов запис, детайли, изтриване и редактиране /всички да са като бутони или само като хипервръзки/.

Оформете началната страница и добавете страница за контакти.

Ако решите да имате страница за поверителност, напишете информация в нея. Ако не – премахнете я от менюто, премахнете съответния изглед и екшън от контролера.

Проверете дали дизайнът е адаптивен и направете поправки, ако е необходимо.

**Идеи за допълнителни функционалности към приложението:**

- Изтриване и редактиране на поръчка от администратора
- Следене на статуса на поръчките
- Възможност за поръчка на повече от един продукт – количка за пазаруване
- Предоставяне на допълнителни ваучери/отстъпки за пазаруване на конкретни потребители
- Виртуален портфейл при пазаруване – предоставяне на възможност на потребителите да използват свой собствен виртуален портфейл с опции за зареждане и следене на баланса по него
- Потребителски мнения и коментари
- Ревюта на продукти
- Бюлетин със специални оферти
- други