C# Web

PRO - C# Web 1

DE HOGESCHOOL
MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Overzicht C# Web

MVC concept

- Model
  - Model Validation
- View
- Controller

Entity Tools

Entity Framework

Identity

- Users
- Roles

Built-in services

- services.AddControllersWithViews
- Services.AddDbContext

Middleware componenten

- `app.UseEndpoints`
- `App.UseRouting`

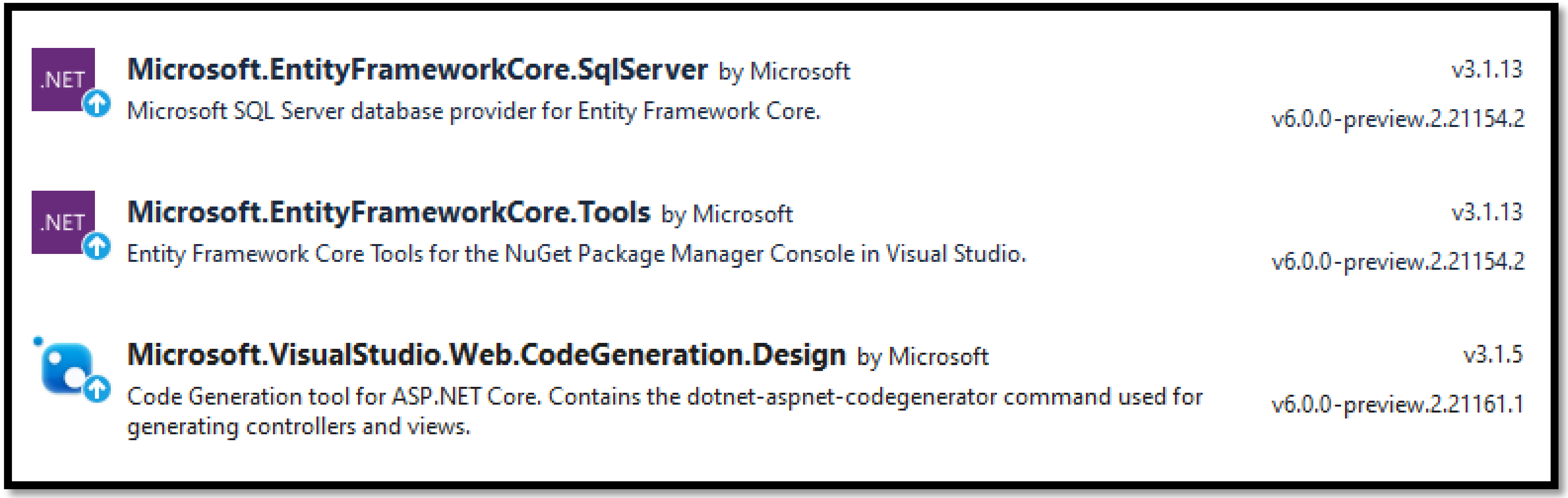
Doel

- MVC
  - Entity Framework
  - MVCSportStore

MVCSportStore

- New visual studio project
  - ASP.Net Core MVC
    - MVCSportStore

Nuget Package Manager



Microsoft.EntityFrameworkCore.SqlServer by Microsoft — v3.1.13

Microsoft SQL Server database provider for Entity Framework Core. — v6.0.0-preview.2.21154.2

Microsoft.EntityFrameworkCore.Tools by Microsoft — v3.1.13

Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio. — v6.0.0-preview.2.21154.2

Microsoft.VisualStudio.Web.CodeGeneration.Design by Microsoft — v3.1.5

Code Generation tool for ASP.NET Core. Contains the dotnet-aspnet-codegenerator command used for generating controllers and views. — v6.0.0-preview.2.21161.1

ASP.Net Core MVCSportStore

DATA

Data - DbContext

- **Create folder Data**
    - **Create class StoreDbContext**

```
using Microsoft.EntityFrameworkCore;

namespace MVCSportStore.Data
{
    public class StoreDbContext : DbContext
    {
         public StoreDbContext(DbContextOptions<StoreDbContext> options) : base(options)
         { }
}
```

ASP.Net Core MVCSportStore

DATA MODELS

Data models - Product

Add Folder -> Data/Tables
Add Class -> Data/Tables/Product.cs

```
using System.ComponentModel.DataAnnotations.Schema;
namespace MVCSportStore.Data.Tables
{
    public class Product
    {
        public long ProductId { get; set; }
        public string Name { get; set; }
        public string Description { get; set; }
        [Column(TypeName = "decimal(8, 2)")]
        public decimal Price { get; set; }
        public string Category { get; set; }
    }

}
```

Data - DbContext

```
using Microsoft.EntityFrameworkCore;
using MVCSportStore.Data.Tables;

namespace MVCSportStore.Data
{
    public class StoreDbContext : DbContext
    {
         public StoreDbContext(DbContextOptions<StoreDbContext> options) : base(options)
         { }
        public DbSet<Product> Products { get; set; }
    }
}
```

Data – DbContext – Startup.cs

- **Startup.cs**
- **ConfigureServices**

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddControllersWithViews();
    services.AddDbContext<StoreDbContext>(options =>
        options.UseSqlServer(Configuration["ConnectionStrings:SportStoreConnection"]));
}
```

Data – DbContext – connection string

- **Connectionstring**
- **Appsettings.json**

Adding a Configuration Setting in the appsettings.json File in the project Folder

```
{
        "Logging": {
                "LogLevel": {
                "Default": "Information",
                "Microsoft": "Warning",
                "Microsoft.Hosting.Lifetime": "Information"
                }
        },
        "AllowedHosts": "*",
        "ConnectionStrings": {
                "SportStoreConnection":
                "Server=(localdb)\\MSSQLLocalDB; Database=MVCSportStore;
                        MultipleActiveResultSets=true"
        }}
```

ASP.Net Core MVCSportStore

NPM
MIGRATION

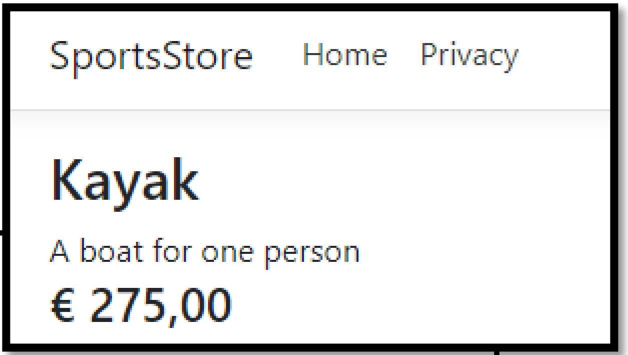
NPM – update database
Tools – Nuget Package Manage

- `Add-migration initial`
- `Update-Database`

ASP.Net Core MVCSportStore

DEFAULT DATA
SEEDDATA

Data - SeedData



SportsStore Home Privacy

Kayak

A boat for one person

€ 275,00

```
namespace MVCSportStore.Data
{
    public static class SeedData
    {
        public static void EnsurePopulated(IApplicationBuilder app)
        {
            StoreDbContext context = app.ApplicationServices.CreateScope()
                   .ServiceProvider
                   .GetRequiredService<StoreDbContext>();

            if (!context.Products.Any())
            {
                context.Products.AddRange(
                new Product
                {
                    Name = "Kayak",
                    Description = "A boat for one person",
                    Category = "Watersports",
                    Price = 275
                });
                //other products …
             }
         }
     }
}
```

Data – SeedData - Startup

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        app.UseHsts();
    }
    app.UseHttpsRedirection();
    app.UseStaticFiles();
    app.UseRouting();
    app.UseAuthorization();
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute(
            name: "default",
            pattern: "{controller=Home}/{action=Index}/{id?}");
    });
    SeedData.EnsurePopulated(app);
}
```

ASP.Net Core MVCSportStore

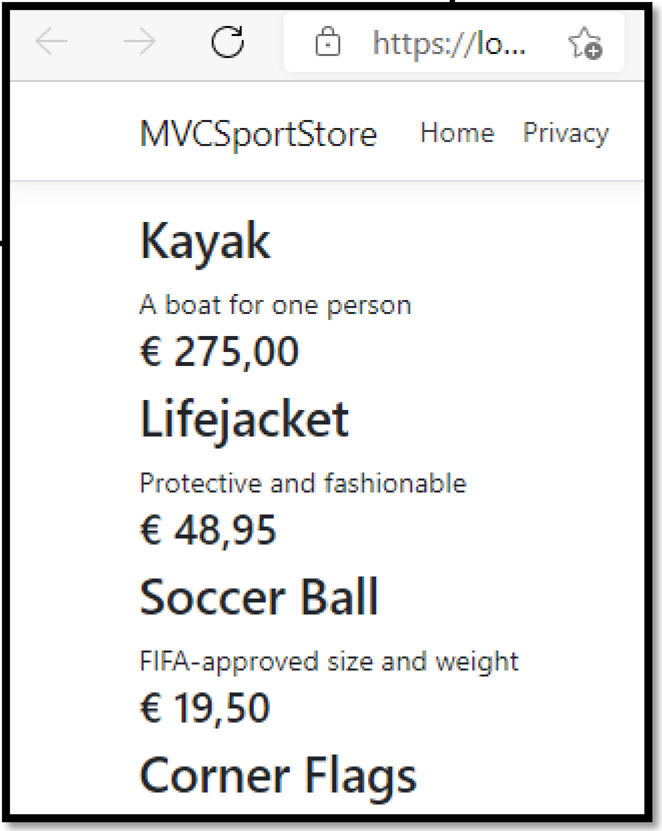
HOMECONTROLLER DEPENDENCY INJECTION

HomeController

```
namespace MVCSportStore.Controllers
{
    public class HomeController : Controller
    {
        private StoreDbContext _context;
        public HomeController(StoreDbContext context)
        {
            _context = context;
        }
        public IActionResult Index()
        {
            return View(_context.Products);
        }
    }
}
```

Views – Home – Index.cshtml

```
@model IEnumerable<Product>
@foreach (var p in Model)
{
    <div>
        <h3>@p.Name</h3>
        @p.Description
        <h4>@p.Price.ToString("c")</h4>
    </div>
}
```



Kayak
A boat for one person
€ 275,00
Lifejacket
Protective and fashionable
€ 48,95
Soccer Ball
FIFA-approved size and weight
€ 19,50
Corner Flags

ASP.Net Core MVCSportStore

MODELS
PAGING

Models – PagingInfo

```
namespace MVCSportStore.Models
{
    public class PagingInfo
    {
        public int TotalItems { get; set; }
        public int ItemsPerPage { get; set; }
        public int CurrentPage { get; set; }
        public int TotalPages =>
(int)Math.Ceiling((decimal)TotalItems / ItemsPerPage);
    }
}
```

Models – ProductsList

```
namespace MVCSportStore.Models
{
    public class ProductsList
    {
        public IEnumerable<Product> Products { get; set; }
        public PagingInfo PagingInfo { get; set; }
    }
}
```

Models – PagingSettings

```
namespace MVCSportStore.Models
{
    public static class PagingSettings
    {
        public static int ProductPagination = 4;
    }
}
```

Models – ProductRepository
Products – GetProducts()

```
public class ProductRepository
{
    StoreDbContext _context;
    public ProductRepository(StoreDbContext context)
    {
        _context = context;
    }
    private IEnumerable<Product> GetProducts(int productPage)
    {
        return _context.Products
                .OrderBy(p => p.ProductID)
                .Skip((productPage - 1) *
PagingSettings.ProductPagination)
                .Take(PagingSettings.ProductPagination);
    }
}
```

Models – ProductRepository PagingInfo – GetPagingInfo()

```
public class ProductRepository
{
    StoreDbContext _context;
    public ProductRepository(StoreDbContext context)
    {
        _context = context;
    }
    ...
    private PagingInfo GetPagingInfo(int productPage)
    {
        var pagingInfo = new PagingInfo();
        pagingInfo.CurrentPage = productPage;
        pagingInfo.ItemsPerPage = PagingSettings.ProductPagination;
        pagingInfo.TotalItems = _context.Products.Count();
        return pagingInfo;
    }
```

Models – ProductRepository PagingInfo – GetProductsList()

```
public class ProductRepository
{
    StoreDbContext _context;
    public ProductRepository(StoreDbContext context)
    {
        _context = context;
    }
    ...
    public ProductsList GetProductsList(int productPage)
    {
        var pl = new ProductsList();
        pl.Products = GetProducts(productPage);
        pl.PagingInfo = GetPagingInfo(productPage);
        return pl;
    }
}
```

ASP.Net Core MVCSportStore

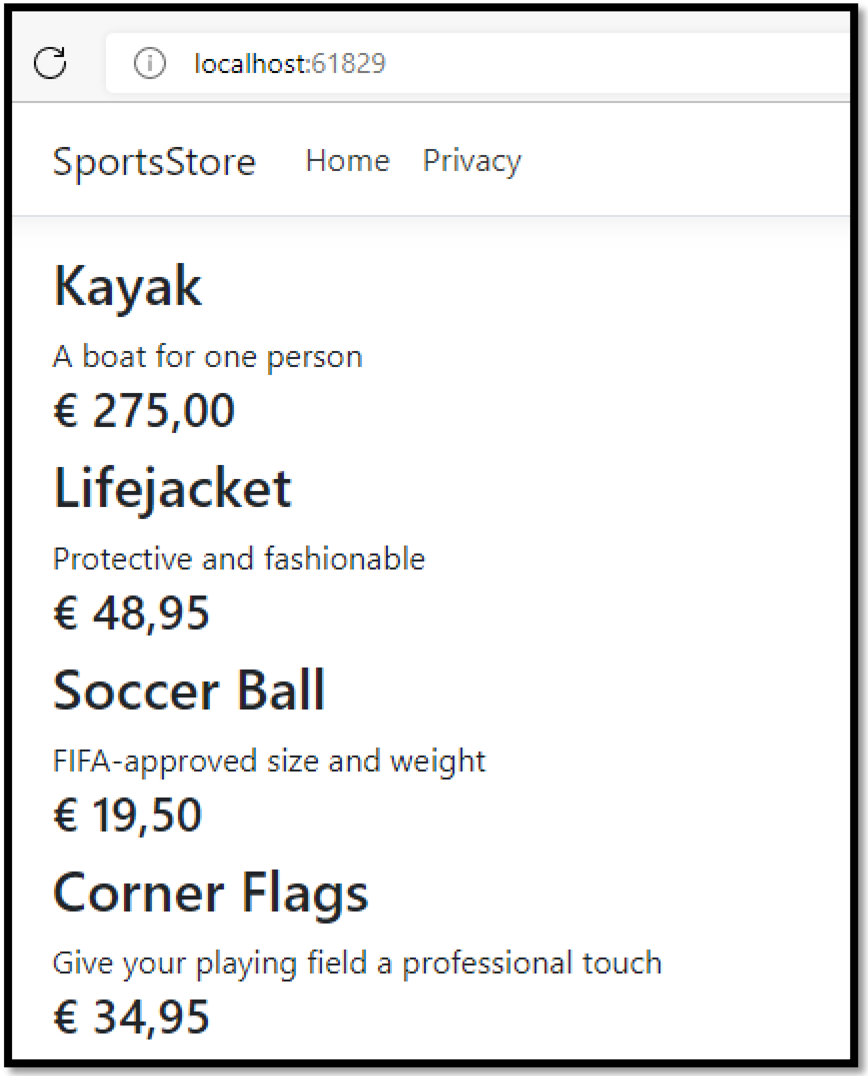
PRODUCTPAGINATION HOMECONTROLLER

HomeController – Index

```
public class HomeController : Controller
{
    private StoreDbContext _context;
    private ProductRepository _repo;
    public HomeController(StoreDbContext context)
    {
        _context = context;
        _repo = new ProductRepository(_context);
    }
    public IActionResult Index(int id = 1)
    {
        var productsList = _repo.GetProductsList(id);
        return View(productsList);
        //return View(_context.Products);
    }
```

Views – Home - Index

```
@model ProductsList
@foreach (var p in Model.Products)
{
    <div>
        <h3>@p.Name</h3>@p.Description
        <h4>@p.Price.ToString("c")</h4>
    </div>
}
```



Kayak

A boat for one person

€ 275,00

Lifejacket

Protective and fashionable

€ 48,95

Soccer Ball

FIFA-approved size and weight

€ 19,50

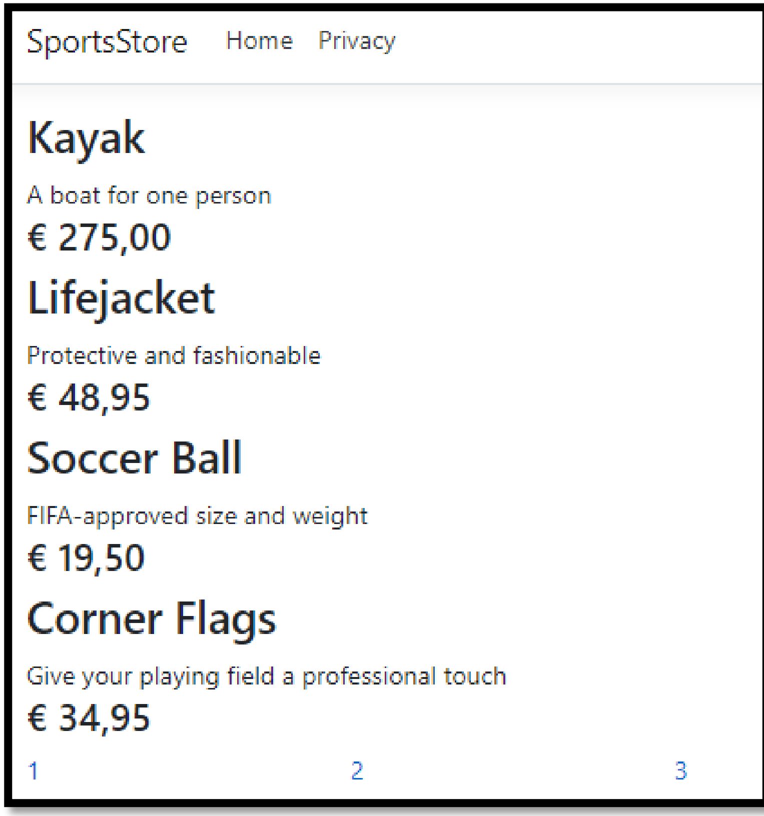
Corner Flags

Give your playing field a professional touch

€ 34,95

Views – Home - Index

```
@model ProductsList
@foreach (var p in Model.Products)
{
    <div>
        <h3>@p.Name</h3>@p.Description
        <h4>@p.Price.ToString("c")</h4>
    </div>
}
<div class="row">
    @for(int i=1;i<=Model.PagingInfo.TotalPages;i++)
    {
        <div class="col">
            <a asp-controller="Home" asp-
action="Index" asp-route-id="@i">@i</a>
        </div>
    }
</div>
```



Kayak

A boat for one person

€ 275,00

Lifejacket

Protective and fashionable

€ 48,95

Soccer Ball

FIFA-approved size and weight

€ 19,50

Corner Flags

Give your playing field a professional touch

€ 34,95

1 2 3

HomeController – Index

```
public class HomeController : Controller
{
    private StoreDbContext _context;
    private ProductRepository _repo;
    public HomeController(StoreDbContext context)
    {
        _context = context;
        _repo = new ProductRepository(_context);
    }
    public IActionResult Index(int id = 1)
    {
       var productsList = _repo.GetProductsList(id);
       return View(productsList);
       //return View(_context.Products);
    }
```

ASP.Net Core MVCSportStore

STYLING THE CONTENT _LAYOUT.CSHTML

_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Sports Store</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
</head>
<body>
    <div class="bg-dark text-white p-2">
        <span class="navbar-brand ml-2">SPORTS STORE</span>
    </div>
    <div class="row m-1 p-1">
        <div id="categories" class="col-3">
            Navigation menu
        </div>
        <div class="col-9">
            @RenderBody()
        </div>
    </div>
</body>
</html>
```

Pluralsight

https://app.pluralsight.com/library/courses/building-aspdotnet-core-mvc-web-applications

https://app.pluralsight.com/library/courses/csharp-collections-beginning

ASP.Net Core MVCSportStore

OEFENING 09
PROJECT