C# Web – MVC

PRO/PRW - C# Web 1

DE HOGESCHOOL
MET HET NETWERK

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

Doel

- ASP.Net Core toepassing
  - MVC Web Application
    - TagHelper
      - SportStore
    - Partial View
      - Sportstore
    - ViewComponent
      - SportStore

ASP.NET CORE

MVC Web Application
MVCSportStore

- **TagHelper**

MVCSportStore

- Add folder
  - TagHelpers
    - Add TagHelper class
      - PageLinkTagHelper.cs

```
namespace MVCSportStore.TagHelpers
{
    [HtmlTargetElement("page-link")]
    public class PageLinkTagHelper : TagHelper
    {
        public PagingInfo PagingInfo { get; set; }
```

- Html
  - <page-link paging-info=“”></page-link>

- Views/Shared
  - _ViewImports.cshtml
    - addTagHelper *, MVCSportStore

MVCSportStore

PageLinkTagHelper.cs

```
namespace MVCSportStore.TagHelpers
{
  [HtmlTargetElement("page-link", Attributes="paging-info")]
  public class PageLinkTagHelper : TagHelper
  {
    public PagingInfo PagingInfo { get; set; }
    public override void Process(TagHelperContext context, TagHelperOutput output)
    {}
    private TagBuilder GetPaginationLinks(PagingInfo pagingInfo)
    {
      TagBuilder ul = new TagBuilder("ul");
      return ul;
    }
    private TagBuilder GetPaginationLink(int page, bool active)
    {
      string pageLinkActive = "btn border border-primary";
      string pagelink = "btn border border-secondary";
      TagBuilder li = new TagBuilder("li");
      return li;
    }
```

MVCSportStore

PageLinkTagHelper.cs - TagBuilder – GetPaginationLink()

```
private TagBuilder GetPaginationLink(int page, bool active)
{
    string pageLinkActive = "btn border border-primary";
    string pagelink = "btn border border-secondary";
    TagBuilder li = new TagBuilder("li");
    li.Attributes["class"] = "page-item";
    TagBuilder a = new TagBuilder("a");
    a.Attributes["class"] = (active) ? pageLinkActive : pagelink;
    a.Attributes["href"] = $"/Home/Index/{page}";
    a.Attributes["title"] = $"Click to go to page {page}";
    a.InnerHtml.Append($"{page}");
    li.InnerHtml.AppendHtml(a);
    return li;
}
```

```
<li class="page-item">
  <a class="btn border border-primary" href="/Home/Index/1" title="Click to go to page 1">1</a>
</li>
```

MVCSportStore

PageLinkTagHelper.cs - TagBuilder – GetPaginationLinks()

```
private TagBuilder GetPaginationLinks(PagingInfo pagingInfo)
{
    TagBuilder ul = new TagBuilder("ul");
    ul.Attributes["class"] = "pagination";
    for (int page = 1; page <= pagingInfo.TotalPages; page++)
    {
        ul.InnerHtml.AppendHtml(
       GetPaginationLink(page, page == pagingInfo.CurrentPage));
    }
    return ul;
}
```

```
<ul class="pagination">
 <li class="page-item">
   <a class="btn border border-primary" href="/Home/Index/1" title="Click to go to page 1">1</a>
 </li>
 …
</ul>
```

MVCSportStore

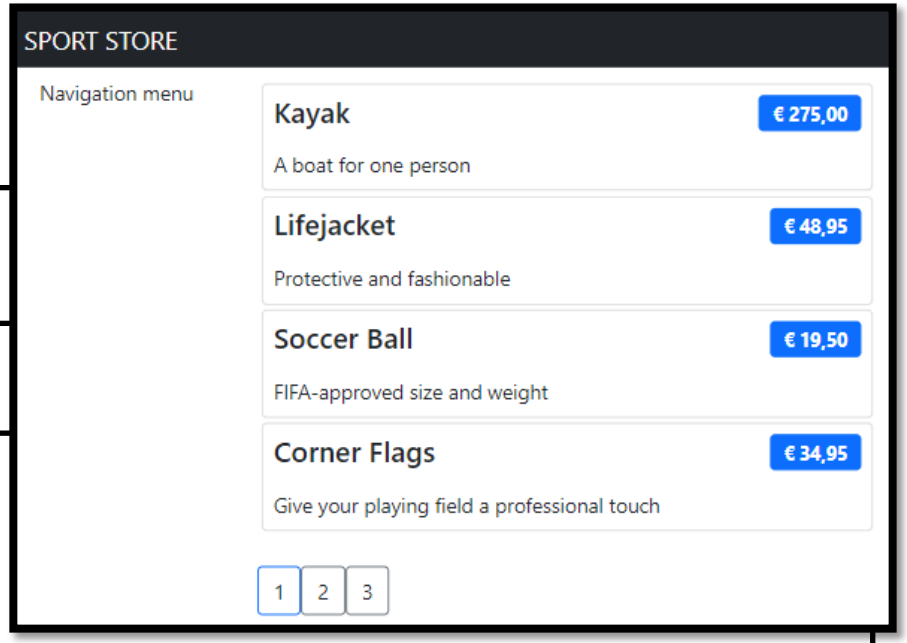
PageLinkTagHelper.cs - Process()

```
public override void Process(TagHelperContext context, TagHelperOutput output)
{
    output.TagName = "div";
    output.Content.AppendHtml(GetPaginationLinks(PagingInfo));
}
```

```
<div>
 <ul class="pagination">
  <li class="page-item">
   <a class="btn border border-primary" href="/Home/Index/1" title="Click to go to page 1">1</a>
  </li>
  <li class="page-item">
   <a class="btn border border-secondary" href="/Home/Index/2" title="Click to go to page 2">2</a>
  </li><li class="page-item">
   <a class="btn border border-secondary" href="/Home/Index/3" title="Click to go to page 3">3</a>
  </li>
 </ul>
</div>
```

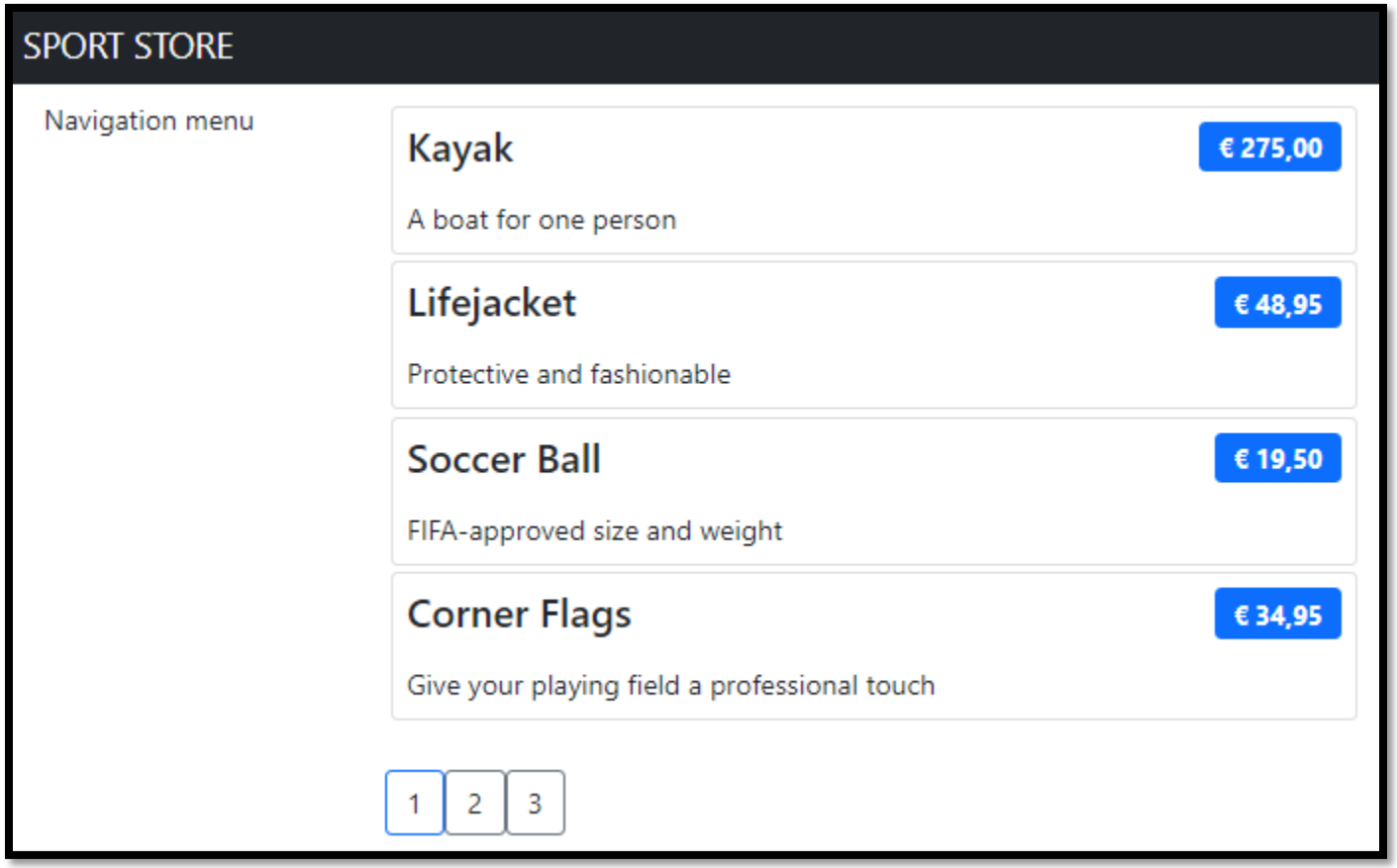
MVCSportStore

PageLinkTagHelper.cs
Views/Home/Index.cshtml

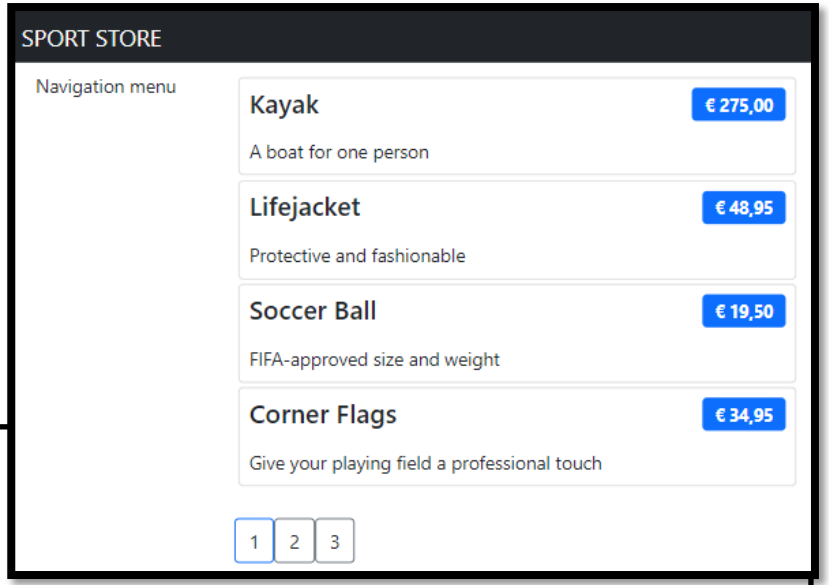



```
@model ProductModel
@foreach (var p in Model.Products)
{
    <div>
        <h3>@p.Name</h3>@p.Description
        <h4>@p.ProductPrice.ToString("c")</h4>
    </div>
}
<br>
<page-link paging-info="@Model.PagingInfo"></page-link>
```

MvcSportStore – styling



MVCSportStore – styling




```
@model ProductModel

@foreach (var p in Model.Products)
{
    <div class="card card-outline-primary m-1 p-1">
        <div class="bg-faded p-1">
            <h4>
                @p.Name
                <span class="badge bg-primary" style="float:right">
                    <small>@p.ProductPrice.ToString("c")</small>
                </span>
            </h4>
        </div>
        <div class="card-text p-1">@p.Description</div>
     </div>
}
```

ASP.NET CORE

MVC Web Application
MVCSportStore

- **Partial View**

Partial View - ProductSummary

Views/Shared
- Add empty razor view
  - ProductSummary.cshtml

Partial View - ProductSummary

_ViewImports.cshtml

```
@using MVCSportStore
@using MVCSportStore.Models
@using MVCSportStore.Models.Data
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@addTagHelper *, MVCSportStore
```

Views/Shared

- Add empty razor view
  - ProductSummary.cshtml

```
@model Product
<div class="card card-outline-primary m-1 p-1">
    <div class="bg-faded p-1">
        <h4>
            @Model.Name
            <span class="badge bg-primary" style="float:right">
                <small>@Model.ProductPrice.ToString("c")</small>
            </span>
        </h4>
    </div>
    <div class="card-text p-1">@Model.Description</div>
</div>
```

Partial View - ProductSummary

Views/Home/Index.cshtml

```
@model ProductModel

@foreach (var p in Model.Products)
{
    <partial name="ProductSummary" model="p" />
}
<br />
<page-link paging-info="@Model.PagingInfo"></page-link>
```

ASP.NET CORE

MVC Web Application
MVCSportStore

- **ViewComponent**

Navigation menu - ViewComponent

Navigation Menu

- Project - Add folder: **Components**
  - Folder Components – Add Class: **NavigationMenuViewComponent**

- Views/Shared
  - Nieuwe folder: Components/NavigationMenu
    (Views/Shared/Components/NavigationMenu)
    - Add empty razor view
      - Default.cshtml

Navigation menu - ViewComponent

```
namespace MVCSportStore.Components
{
    public class NavigationMenuViewComponent : ViewComponent
    {
        private StoreDbContext _context;
        public NavigationMenuViewComponent(StoreDbContext context)
        {
            _context = context;
        }
        public IViewComponentResult Invoke()
        {
            return View(_context.Products
                .Select(x => x.Category)
                .Distinct()
                .OrderBy(x => x));
        }
    }
}
```

Navigation menu - ViewComponent

Navigation Menu - Views/Shared

- Views/Shared/Components/NavigationMenu/Default.cshtml

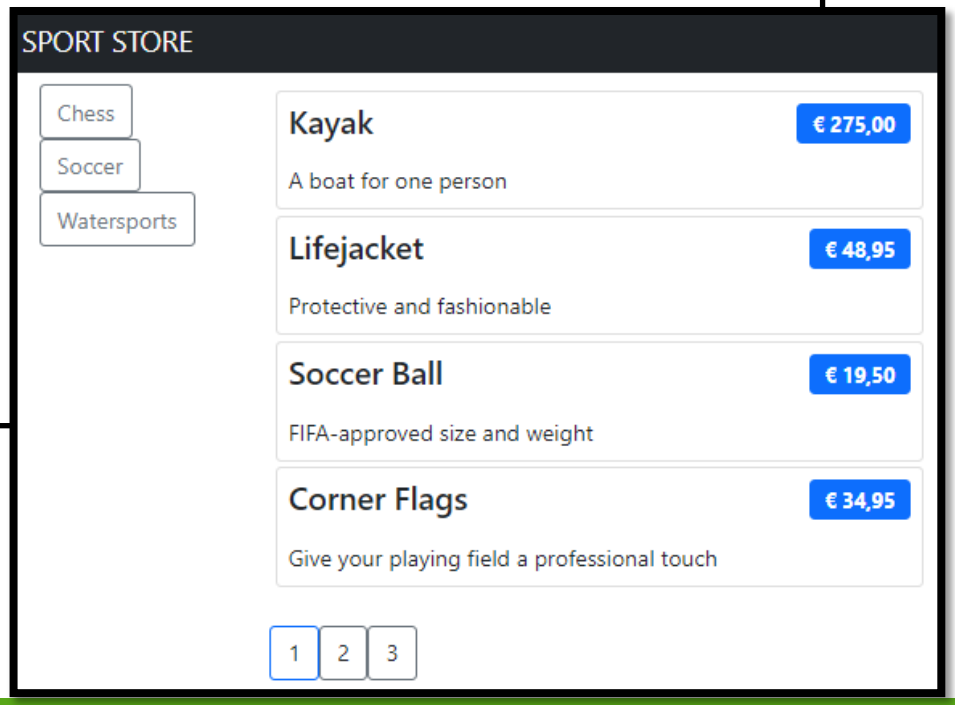
```
@model IEnumerable<string>
@foreach (string category in Model)
{
    <a class="btn btn-outline-secondary btn-block m-2"
       asp-action="Index"
       asp-controller="Home"
       asp-route-id="1">
            @category
    </a>
    <br>
}
```

Navigation menu - ViewComponent

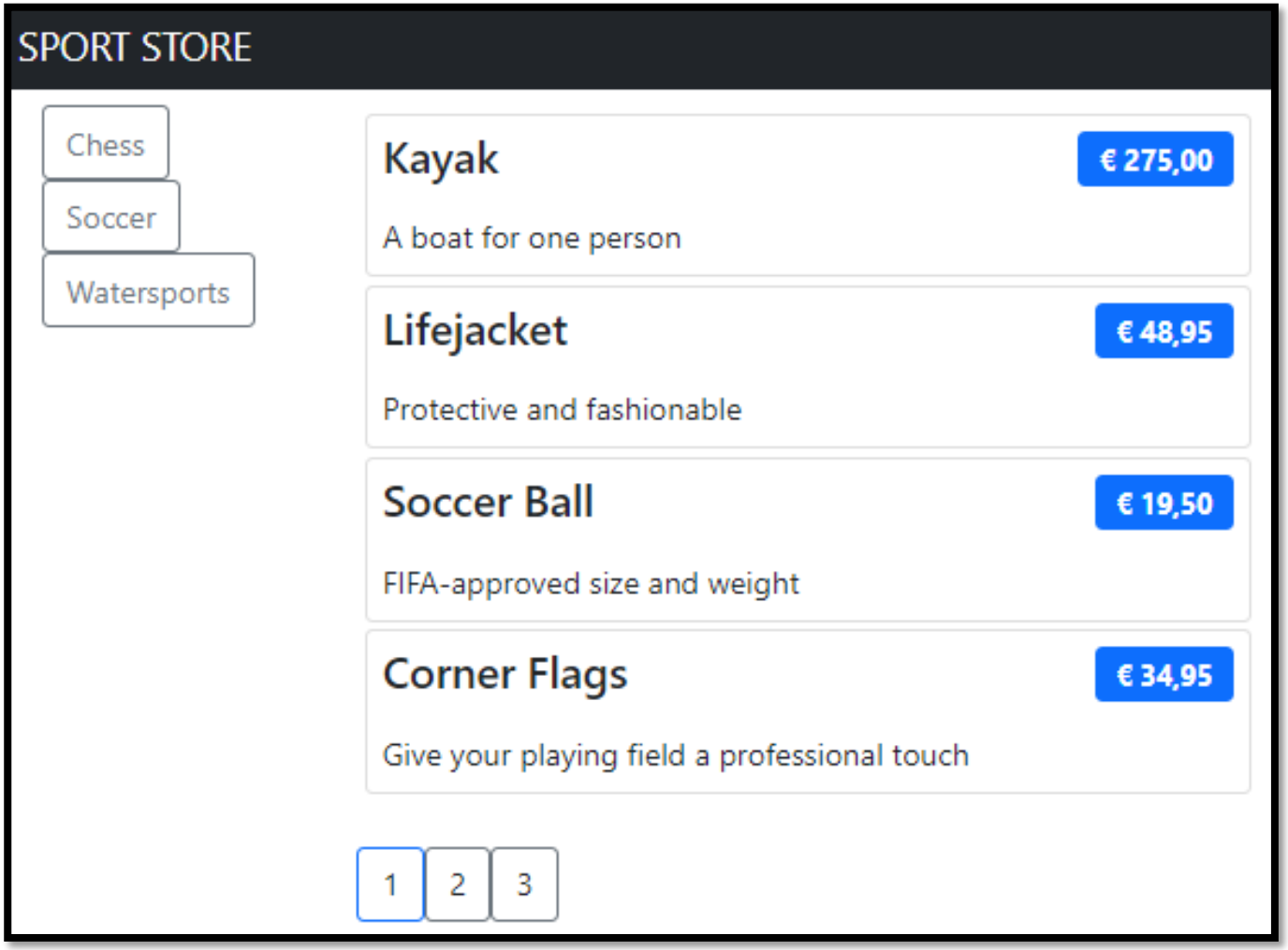
Navigation Menu - Views/Shared

- Views/Shared/_Layout.cshtml

```
<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">
            <vc:navigation-menu />
        </div>
        <div class="col-9">
            @RenderBody()
        </div>
    </div>
</body>
```



Navigation menu - ViewComponent



Navigation menu - ViewComponent

Navigation Menu - Views/Shared

- Views/Shared/Components/NavigationMenu/Default.cshtml

```
@model IEnumerable<string>
@{
    string style = "btn btn-outline-secondary btn-block m-2";
    string styleActive = "btn btn-outline-primary btn-block m-2";
}
@foreach(string category in Model)
{
    <div class="row">
        <a class="@((category==ViewBag.SelectedCategory)
                        ? styleActive : style)"
            asp-controller="Home"
            asp-action="Index"
            asp-route-id="1">
                @category
        </a>
    </div>
}
```

Navigation menu - ViewComponent

Navigation Menu - Views/Shared

- Views/Shared/Components/NavigationMenu/Default.cshtml

```
<div class="row">
    <a class="@(category==ViewBag.SelectedCategory
                    ? styleActive : style)"
       asp-controller="Home"
       asp-action="Index"
       asp-route-id="1“
       asp-route-category="@category">
        @category
    </a>
</div>
```

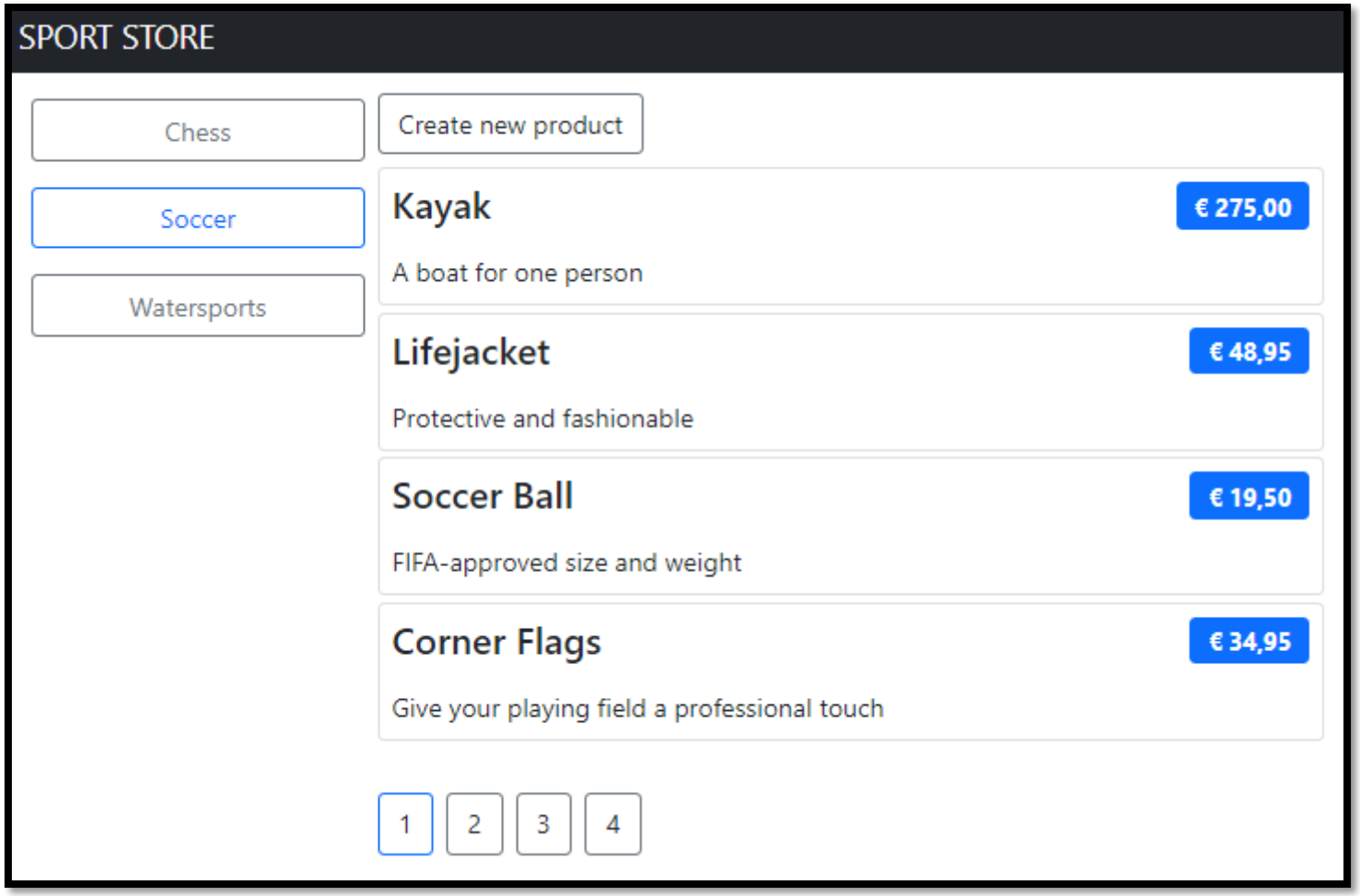
HomeController- category (Route)

```
public IActionResult Index(int id = 1, string category=null)
{
    if (category != null)
        AddRouteKey("category", category);
    var productModel = _repo.GetProductModel(id);
    return View(productModel);
}
private void AddRouteKey(string key, string keyValue)
{
    if (base.RouteData.Values.ContainsKey(key))
        base.RouteData.Values[key] = keyValue;
    else
        base.RouteData.Values.Add(key, keyValue);
}
```

Navigation menu - ViewComponent

```
namespace MVCSportStore.Components
{
    public class NavigationMenuViewComponent : ViewComponent
    {
        private StoreDbContext _context;
        public NavigationMenuViewComponent(StoreDbContext context)
        {
            _context = context;
        }
        public IViewComponentResult Invoke()
        {
            ViewBag.SelectedCategory = RouteData?.Values["category"];
            return View(_context.Products
                .Select(x => x.Category)
                .Distinct()
                .OrderBy(x => x));
        }
    }
}
```

Navigation menu - ViewComponent



Navigation menu - ViewComponent

HomeController – Index actie

```
public IActionResult Index(int id = 1, string category=null)
{
    if(category != null)
       AddRouteKey("category", category);
    //category moeten we nog toevoegen
    var productModel = _repo.GetProductModel(id);
    return View(productModel);}
```

Navigation menu - ViewComponent

```
ProductRepository – GetProducts()
```

```
private IEnumerable<Product> GetProducts(int productPage, string category)
{
    return _context.Products
            .Where(p => category == null || p.Category == category)
            .OrderBy(p => p.ProductId)
            .Skip((productPage - 1) * PagingSettings.ProductPagination)
            .Take(PagingSettings.ProductPagination);
}
```

Navigation menu - ViewComponent

ProductRepository – GetPagingInfo()

```
private PagingInfo GetPagingInfo(int productPage, string category)
{
    var pagingInfo = new PagingInfo();
    var totalItems = (category == null)
        ? _context.Products.Count()
        : _context.Products.Where(p => p.Category == category).Count();
    pagingInfo.CurrentPage = productPage;
    pagingInfo.PageItems = PagingSettings.ProductPagination;
    pagingInfo.TotalItems = totalItems;
    return pagingInfo;
}
```

Navigation menu - ViewComponent

ProductRepository

```
public ProductModel GetProductModel(int productPage, string category)
{
    var productModel = new ProductModel();
    productModel.Products = GetProducts(productPage, category);
    productModel.PagingInfo = GetPagingInfo(productPage, category);
    return productModel;
}
```

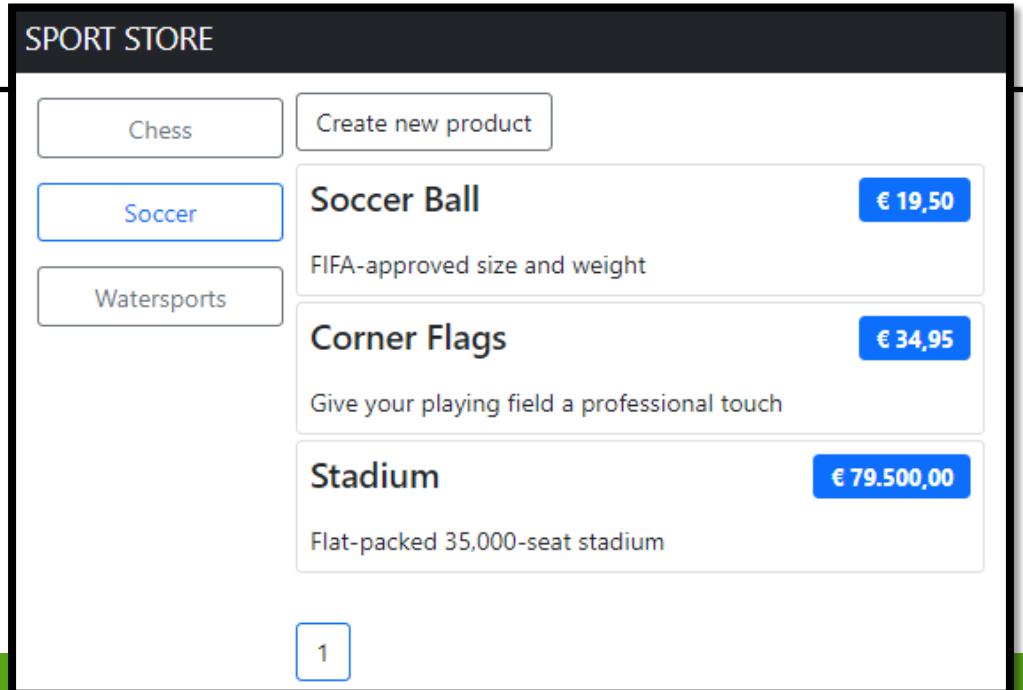
Navigation menu - ViewComponent

HomeController – Index action

```
public IActionResult Index(int id = 1, string category=null)
{
    if (category != null)
        AddRouteKey("category", category);

    var productModel = _repo.GetProductModel(id, category);
    return View(productModel);

}
```



ViewModels - PagingInfo

```
public class PagingInfo
{
    public int TotalItems { get; set; }
    public int PageItems { get; set; }
    public int CurrentPage { get; set; }
    public string Category { get; set; }
    public int TotalPages =>
(int)Math.Ceiling((decimal)TotalItems / ItemsPerPage);
}
```

TagHelper - PageLink

```
private TagBuilder GetPaginationLink(int page, string category, bool active)
{
    string pageLinkActive = "btn border border-primary m-1";
    string pageLink = "btn border border-secondary m-1";
    var li = new TagBuilder("li");
    li.Attributes["class"] = "page-item";
    TagBuilder a = new TagBuilder("a");
    if(category == null)
    {
       a.Attributes["href"] = $"/Home/Index/{page}";
    }
    else
    {
       a.Attributes["href"] = $"/Home/Index/{page}?category={category}";
    }
    a.Attributes["title"] = $"Click to go to page {page}";
    a.Attributes["class"] = (active) ? pageLinkActive : pageLink;
    a.InnerHtml.Append($"{page}");
    li.InnerHtml.AppendHtml(a);
    return li;
}
```

TagHelper - PageLink

```
private TagBuilder GetPaginationLinks(PagingInfo pagingInfo)
{
    var ul = new TagBuilder("ul");
    ul.Attributes["class"] = "pagination";
    bool active = false;
    for (int page = 1; page <= pagingInfo.TotalPages; page++)
    {
        active = (page == pagingInfo.CurrentPage);
        ul.InnerHtml.AppendHtml(
            GetPaginationLink(page, pagingInfo.Category, active));
    }
    return ul;
}
```

TagHelper - PageLink

```
private PagingInfo GetPagingInfo(int productPage, string category)
{
    var pagingInfo = new PagingInfo();
    var totalItems = (category == null)
        ? _context.Products.Count()
        : _context.Products.Where(p => p.Category == category).Count();
    pagingInfo.CurrentPage = productPage;
    pagingInfo.Category = category;
    pagingInfo.ItemsPerPage = PagingSettings.ProductPagination;
    pagingInfo.TotalItems = totalItems;
    return pagingInfo;
}
```

ASP.NET CORE

MVC Web Application
Oefening 14

- **ViewComponent**