WEB PROGRAMMING ASP.NET MVC CORE

PAGINATION VIEWMODEL

```csharp
public class PagingInfo {
        public const int DEFAULT_PAGE_SIZE = 5;
        public const int NUMBER_PAGES_SHOW_BEFORE_AFTER = 5;

        public int TotalItems { get; set; }

        public int PageSize { get; set; }

        public int CurrentPage { get; set; }

        public int TotalPages => (int)Math.Ceiling((double)TotalItems / PageSize);
}
```

PRODUCTSLIST VIEWMODEL

```csharp
public class ProductsListViewModel {
        public IEnumerable<Product> Products { get; set; }
        public PagingInfo Pagination { get; set; }
}
```

PAGINATION SUPPORT

```csharp
public IActionResult Index(int page = 1) {
    var pagination = new PagingInfo  {
        CurrentPage = page,
        PageSize = PagingInfo.DEFAULT_PAGE_SIZE,
        TotalItems = repository.Products.Count()
    };

    return View(
        new ProductsListViewModel {
            Products = repository.Products.OrderBy(p => p.Price)
                .Skip((page - 1) * pagination.PageSize).Take(pagination.PageSize),
            Pagination = pagination
        }
    );
}
```

VIEW

```
@model ProductsListViewModel

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

@foreach (var p in Model.Products) {
    @* ... *@
}

@{
    int initalPage = Model.Pagination.CurrentPage - PagingInfo.NUMBER_PAGES_SHOW_BEFORE_AFTER;
    if (initalPage < 1) initalPage = 1;
    int finalPage = Model.Pagination.CurrentPage + PagingInfo.NUMBER_PAGES_SHOW_BEFORE_AFTER;
    if (finalPage > Model.Pagination.TotalPages) finalPage = Model.Pagination.TotalPages;
}

@* ... *@
```

VIEW

```
@* ... *@

@for (int p = initalPage; p <= finalPage; p++) {
    if (p == Model.Pagination.CurrentPage) {
        <li class="page-item active"><a class="page-link" href="#">@p</a></li>
    } else {
        <li class="page-item"><a class="page-link" asp-action="Index"
            asp-route-page="@p">@p</a></li>
    }
}
```