



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
 - Models
 - Views
 - Controllers

ASP.NET CORE

MVC Web Application

MVCWebApp

ASP.NET Core Web App (Model-View-Controller)

C# Linux macOS Windows Cloud Service
Web

Project name

MVCWebApp

Location

C:\CSharpWeb01\22_23\VisualStudio

Solution name ⓘ

MVCWebApp

☐ Place solution and project in the same directory



ASP.NET Core Web App (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

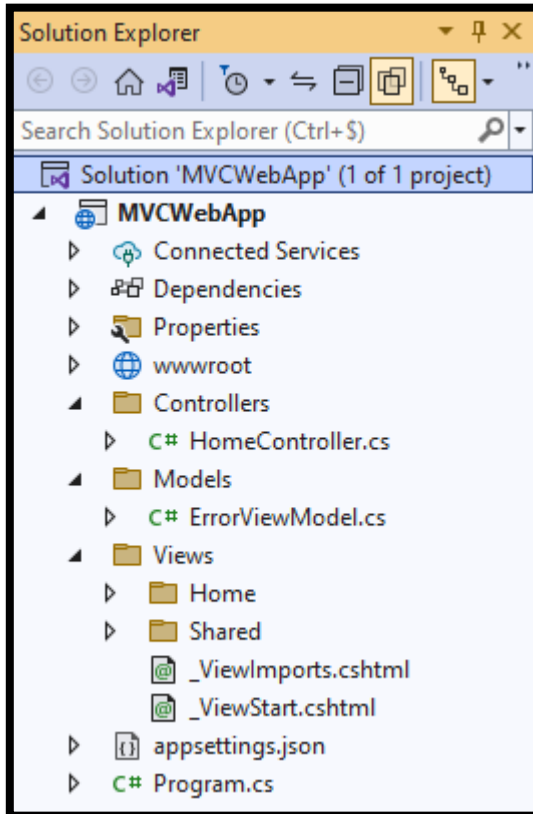
C# Linux macOS Windows Cloud Service Web

Solution Explorer

Search Solution Explorer (Ctrl+S)

Solution 'MVCWebApp' (1 of 1 project)

- ▲ MVCWebApp
 - ▶ Connected Services
 - ▶ Dependencies
 - ▶ Properties
 - ▶ wwwroot
 - ▲ Controllers
 - ▶ C# HomeController.cs
 - ▲ Models
 - ▶ C# ErrorViewModel.cs
 - ▲ Views
 - ▶ Home
 - ▶ Shared
 - ▶ _ViewImports.cshtml
 - ▶ _ViewStart.cshtml
 - ▶ appsettings.json
 - ▶ C# Program.cs



Program.cs

```
var builder = WebApplication.CreateBuilder(args);
// Add services to the container.
builder.Services.AddControllersWithViews();
var app = builder.Build();
// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}
app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseAuthorization();
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
app.Run();
```

MVC Project – Folders & Files

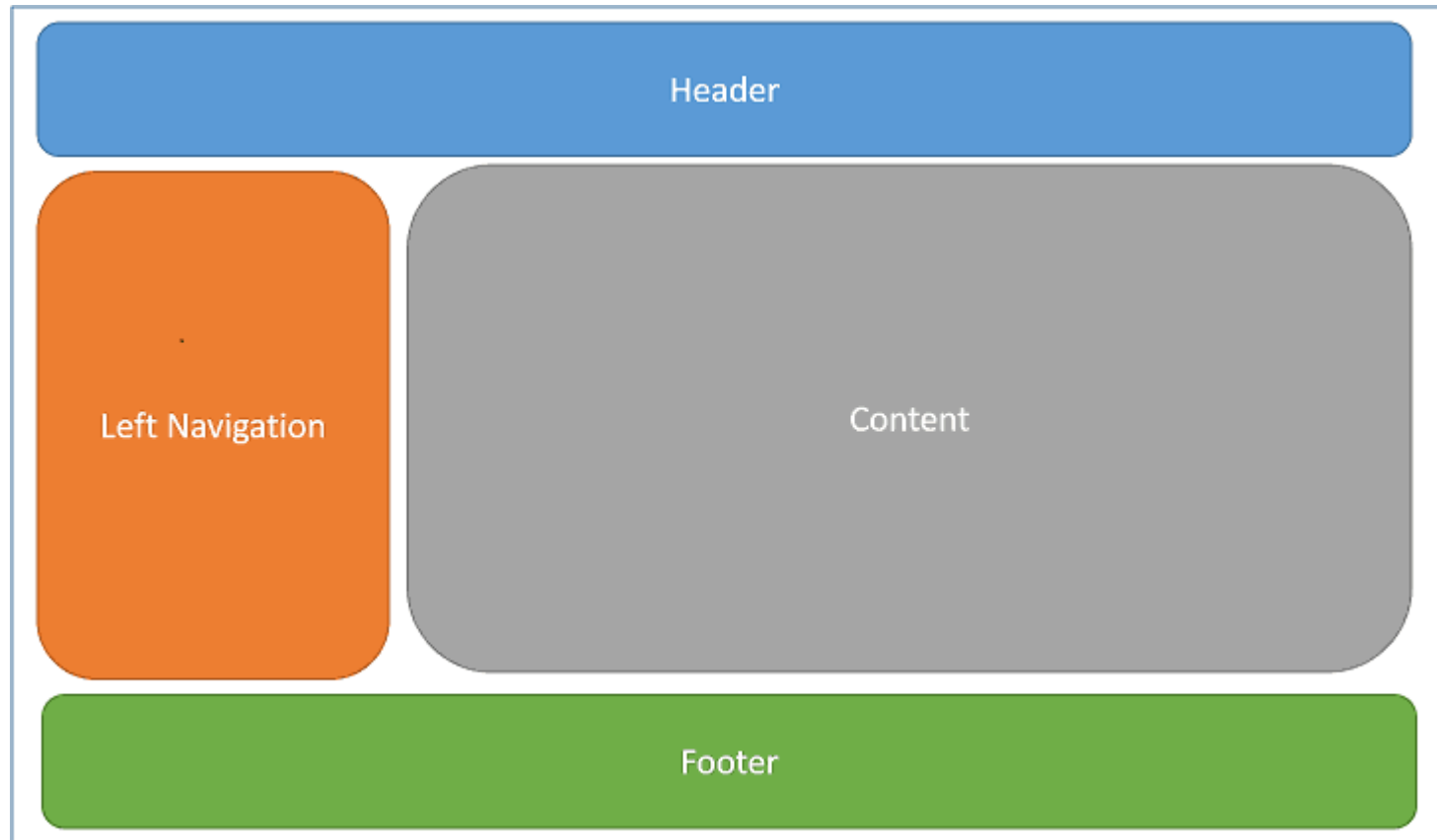
- Models
- Views
 - Home
 - Index.cshtml
 - Shared
 - _Layout.cshtml
- Controllers
 - HomeController
 - Index() Action (HttpGet)
- wwwroot
- Program.cs
- Appsettings.json

Views/Home

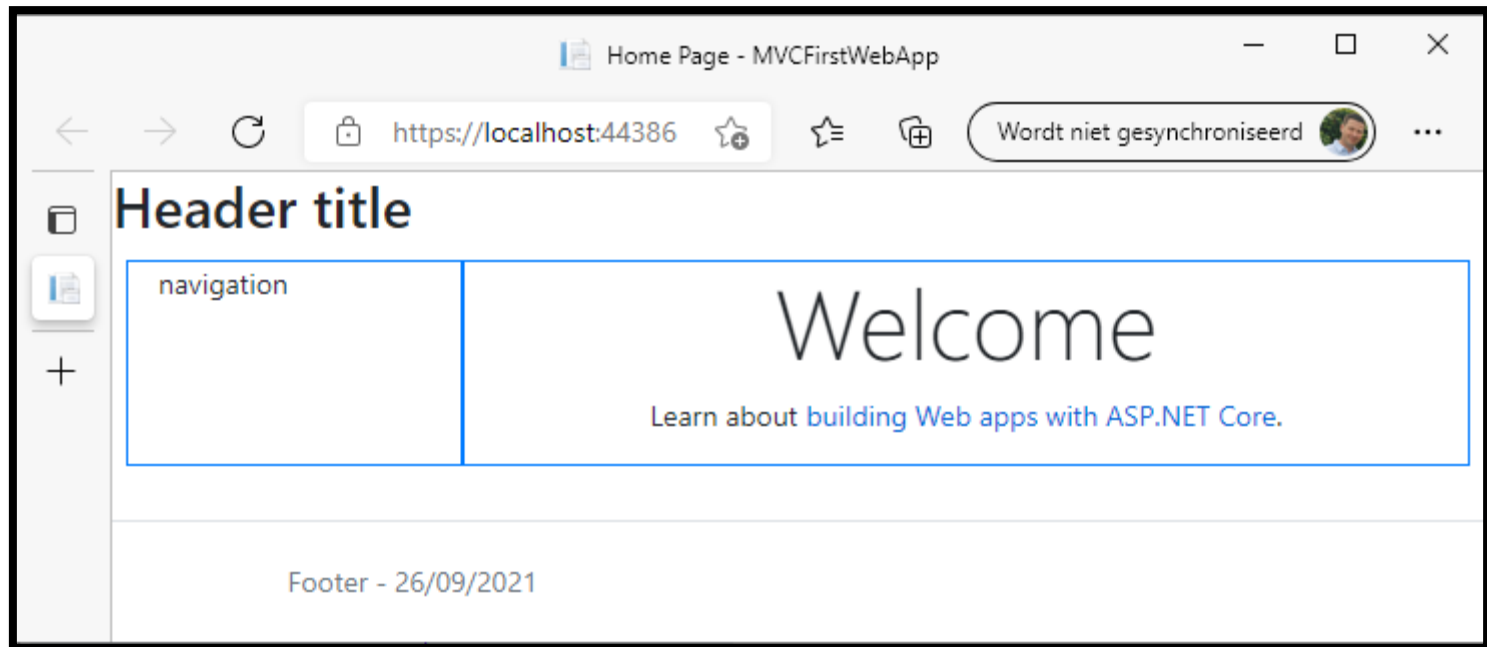
- Home/Index.cshtml

Views/Shared

- _Layout.cshtml



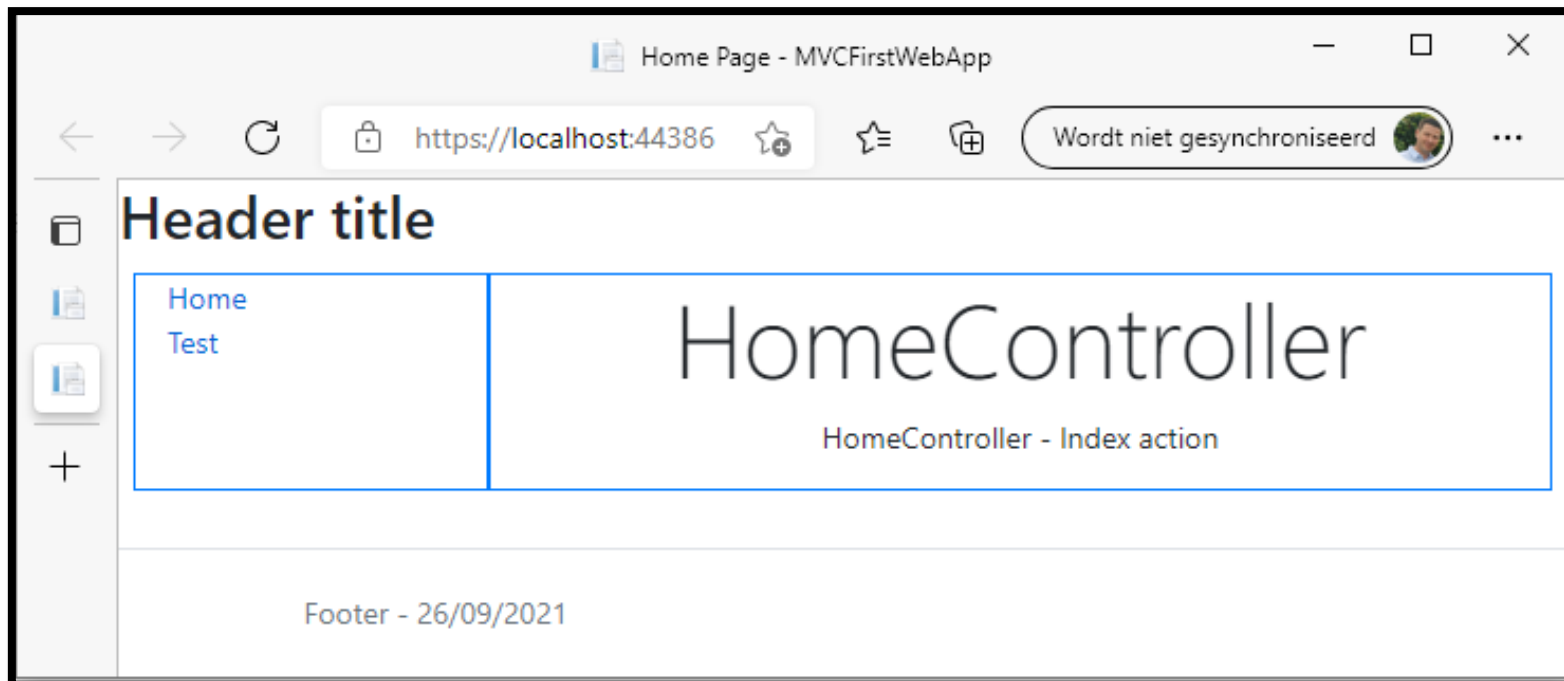
Styling the content



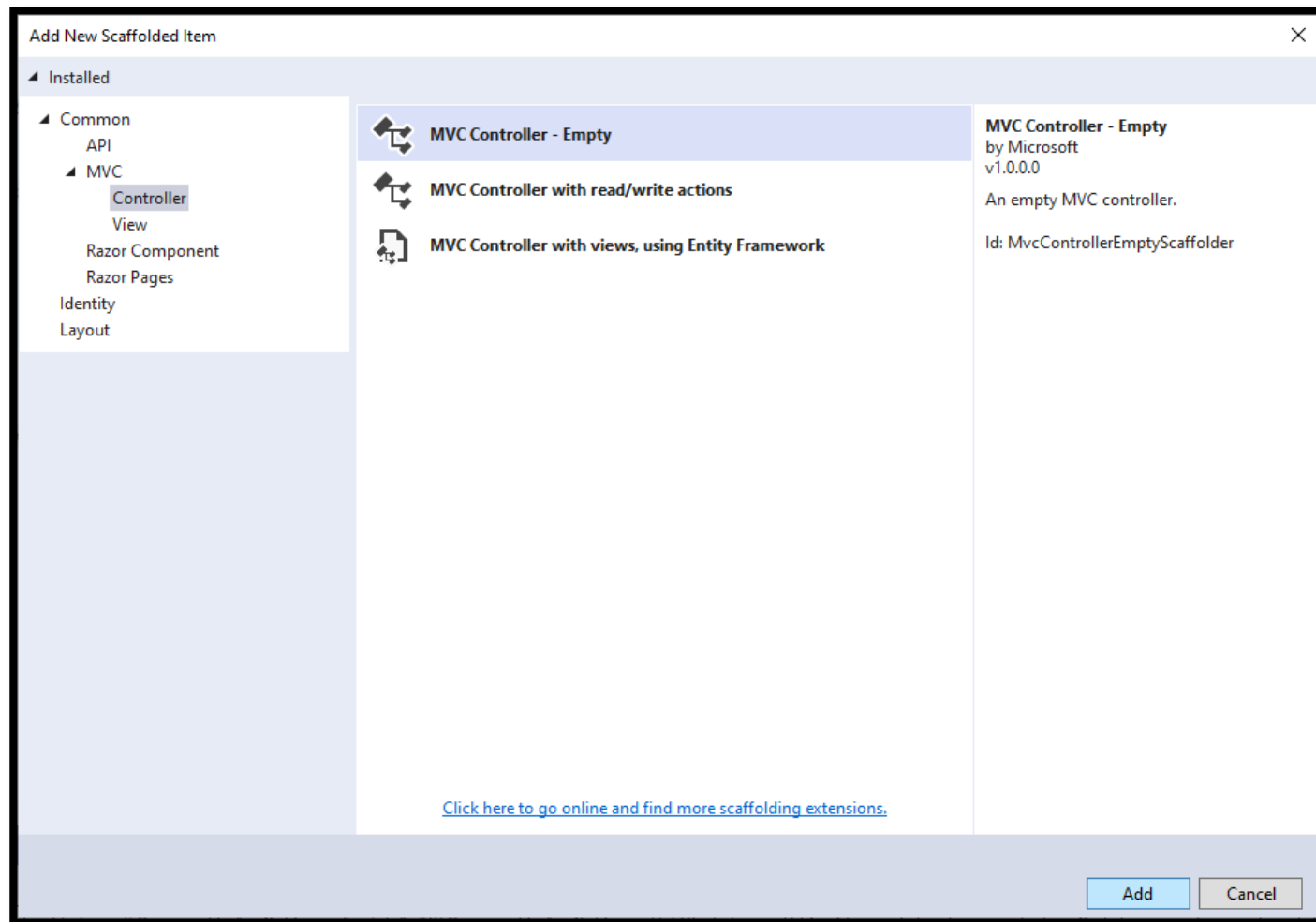
Styling the content – Views/Shared/_Layout.cshtml

```
<body>
  <header >
    <div>
      <h2>Header title</h2>
    </div>
  </header>
  <div class="row m-1 p-1">
    <div id="menu" class="col-3 border border-primary">
      navigation
    </div>
    <div id="main" class="col-9 border border-primary">
      @RenderBody()
    </div>
  </div>
  <footer class="border-top footer text-muted">
    <div class="container">
      Footer - @DateTime.Now.ToString("dd/MM/yyyy")
    </div>
  </footer>
  ...
</body>
```

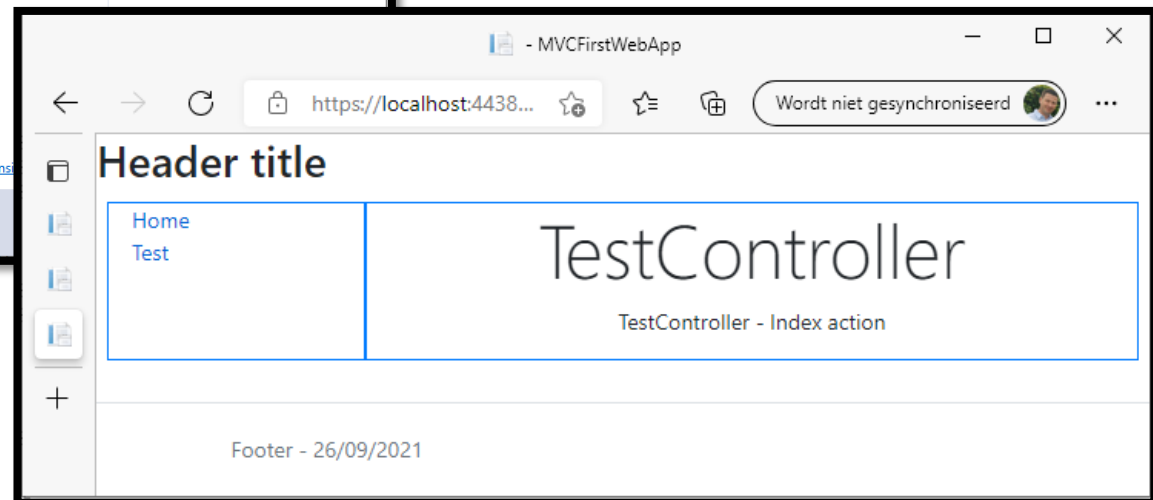
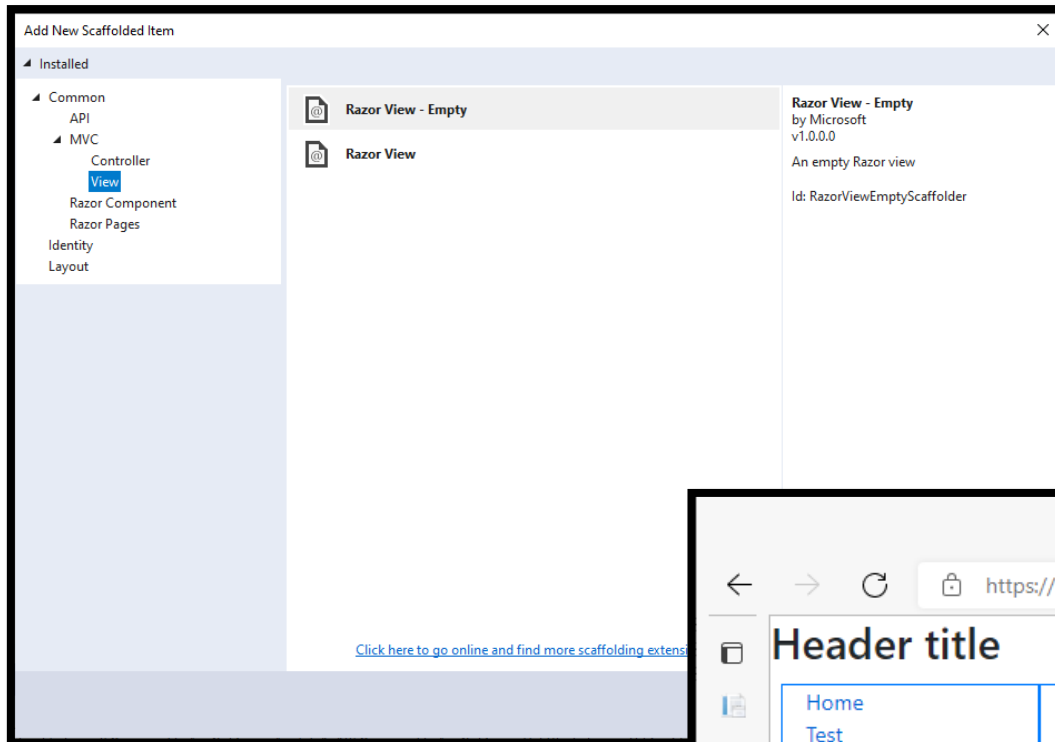
- Voeg een controller toe! (Empty MVC Controller)
 - Controllers\TestController
- Voeg de bijhorende View Folder toe!
 - Views\Test
 - Voeg de Index View toe (Empty razor View)
 - Views\Test\Index.cshtml
- Voeg een hyperlink toe naar de nieuwe TestController – Index view!
 - Pas de inhoud aan naar TestController – Index action.
- Voeg een hyperlink toe naar de bestaande HomeController – Index view!
 - Pas de inhoud aan naar HomeController – Index action.



- Voeg een controller toe!
 - Controllers\TestController



- Voeg de bijhorende View Folder toe!
 - Views\Test
 - Voeg de Index View toe
 - Views\Test\Index.cshtml



TagHelper MVC hyperlink

- Views/Shared
 - _Layout.cshtml

```
<div id="menu" class="col-3 border border-primary">  
  <a asp-controller="Home" asp-action="Index">Home</a>  
  <br />  
  <a asp-controller="Test" asp-action="Index">Test</a>  
</div>
```



Data doorgeven tussen Controller en View

Voeg een View toe in de Views/Test map -> TestViewbag

Plaats de volgende tekst op het scherm

Dit is test 1 uit de viewbag!

Dit is test 2 uit de viewbag!

TestController

```
public IActionResult TestViewbag()
{
    ViewBag.Teller = 3;
    ViewData["Teller"] = 3;
    return View();
}
```

Views/Test/TestViewBag

```
@{
    Layout = "_Layout";
    int teller = (int)ViewBag.Teller;
    teller = (int)ViewData["Teller"];
}
<div>
    <ul>
        @for (int i = 1; i < teller; i++)
        {
            <li>
                Dit is test @i uit de viewbag!
            </li>
        }
    </ul>
</div>
```

TagHelper MVC hyperlink

- Views/Shared
 - _Layout.cshtml

```
<div id="menu" class="col-3 border border-primary">
  <a asp-controller="Home" asp-action="Index">Home</a>
  <br />
  <a asp-controller="Test" asp-action="Index">Test</a>
  <br />
  <a asp-controller="Test" asp-action="TestViewBag">TestViewBag</a>
</div>
```

Voeg een formulier toe!

- Views/Test
 - Formulier.cshtml
- Controllers/TestController
 - Formulier action

```
public IActionResult Formulier()  
{  
    return View();  
    //Deze action verwijst naar onze  
    Views/Test/Formulier.cshtml  
}
```

```
<div>  
    <form asp-action="FormulierPost">  
        <div class="form-group">  
            Invoer:  
            <input name="TestInvoer" />  
            <input type="submit" value="submit"/>  
        </div>  
    </form>  
</div>
```


Voeg een view toe!

- Views/Test
 - FormulierPost.cshtml
- Controllers/TestController
 - FormulierPost action

Models

- Add Model – TestModel.cs

```
public class TestModel
{
    public string TestInvoer{get;set;}
}
```

Views/Test/FormulierPost.cshtml

@model TestModel

<div>

De invoer die u in het formulier hebt ingevuld is:

@Model.TestInvoer

</div>

```
public IActionResult FormulierPost()
{
    string invoer = Request.Form["TestInvoer"];
    TestModel model = new TestModel();
    model.TestInvoer = invoer;
    return View(model);
    //Deze action verwijst naar onze
    Views/Test/FormulierPost.cshtml
}
```

TagHelper MVC hyperlink

- Views/Shared
 - _Layout.cshtml

```
<div id="menu" class="col-3 border border-primary">  
  <a asp-controller="Home" asp-action="Index">Home</a>  
  <br />  
  <a asp-controller="Test" asp-action="Index">Test</a>  
  <br />  
  <a asp-controller="Test" asp-action="TestViewBag">  
    Test - Viewbag</a>  
  <br />  
  <a asp-controller="Test" asp-action="Formulier">  
    Test - formulier</a>  
</div>
```

ASP.NET CORE

MVC Web Application – Oefening04