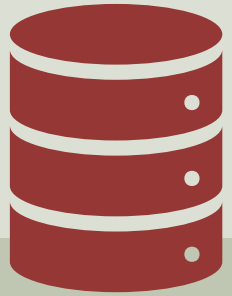
https://www.halvorsen.blog

ASP.NET Core

ASP.NET Core with Razor and SQL Server



Hans-Petter Halvorsen

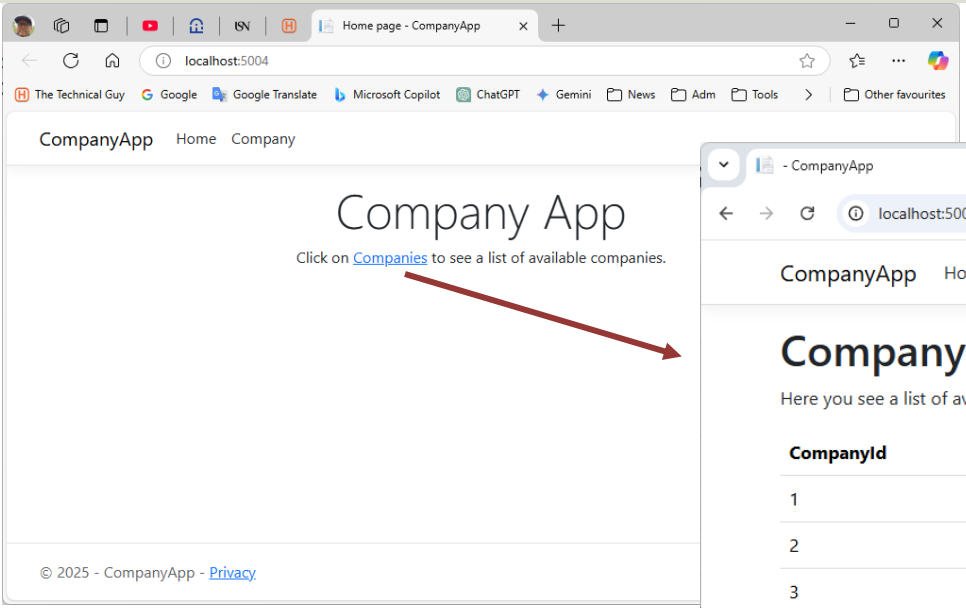
Contents

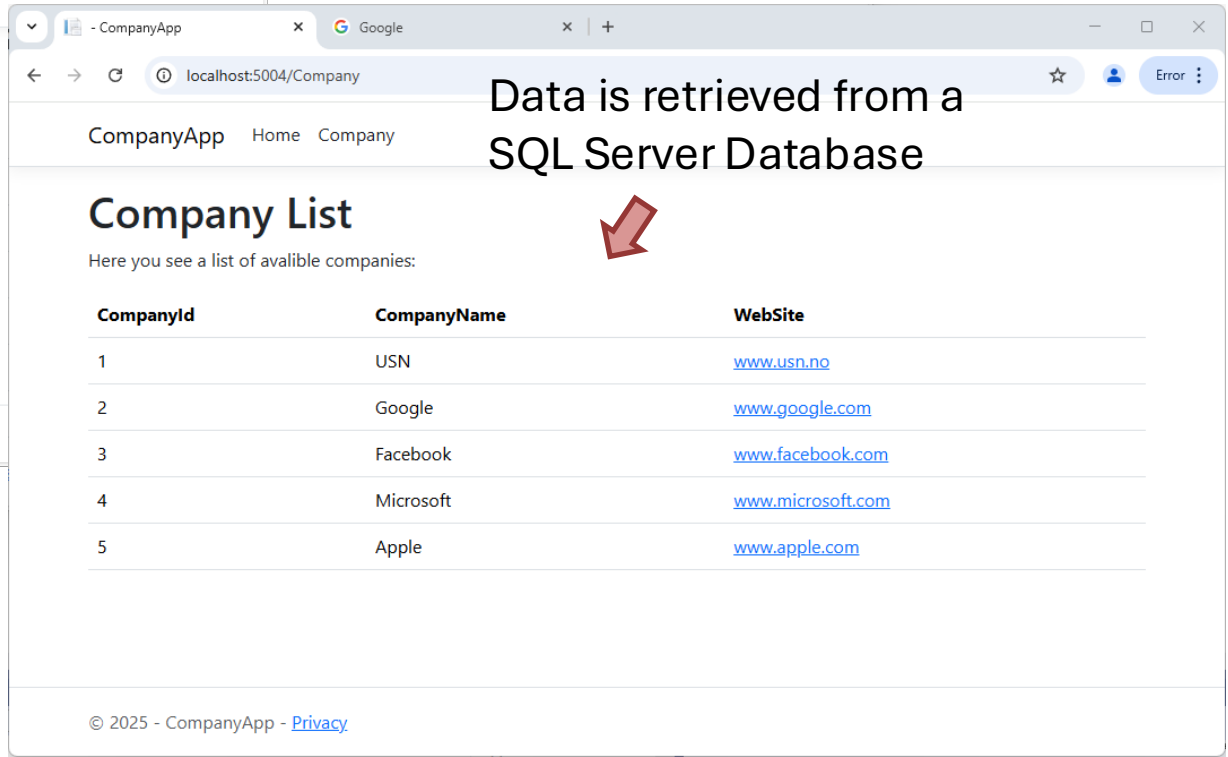
Introduction

Hans-Petter Halvorsen

Table of Contents

ASP.NET Core Razor Web App





ASP.NET Core

- ASP.NET Core is a framework for web development.
- ASP.NET Core is based on .NET and C#.
- What is the difference between ASP.NET Core and .NET?
  - ASP.NET Core is specifically designed for web development, while the .NET covers a broader range of application types, including Windows desktop, mobile, and web applications.
- In ASP.NET Core Razor code and layout are separated into 2 files; The layout file has the extension ". cshtml", and the code-behind file has the extension ". cshtml.cs" (where "cs" is short for C#).
- The layout files ". cshtml" use something called Razor syntax and are mixed with HTML.
- ASP, ASP.NET and ASP.NET Core is made by Microsoft.
- Homepage: https://dotnet.microsoft.com/en-us/apps/aspnet

ASP.NET Core

ASP.NET Core is a framework for building Web Applications and Services with .NET and C#. ASP.NET Core supports different types, here are some examples:

- ASP.NET Core with Razor Pages — The focus in this Tutorial!
- ASP.NET Core MVC
- ASP.NET Core Blazor Web Apps
- ASP.NET Core Web API with controllers
- ASP.NET Core Minimal Web APIs

SQL Server

Setting up the SQL Server Database

Table of Contents

Hans-Petter Halvorsen

SQL Server

- SQL Server Express.
  - Free version of SQL Server that has all we need for the examples in this Tutorial.
- SQL Server Express consist of 2 parts (separate installation packages):
  - SQL Server Express.
  - SQL Server Management Studio (SSMS) – This software can be used to create Databases, create Tables, Insert/Retrieve or Modify Data, etc.
- Download: https://www.microsoft.com/sql-server/sql-server-downloads
- SQL Server Express Installation YouTube: https://youtu.be/hhhggAlUYo8

Database Table

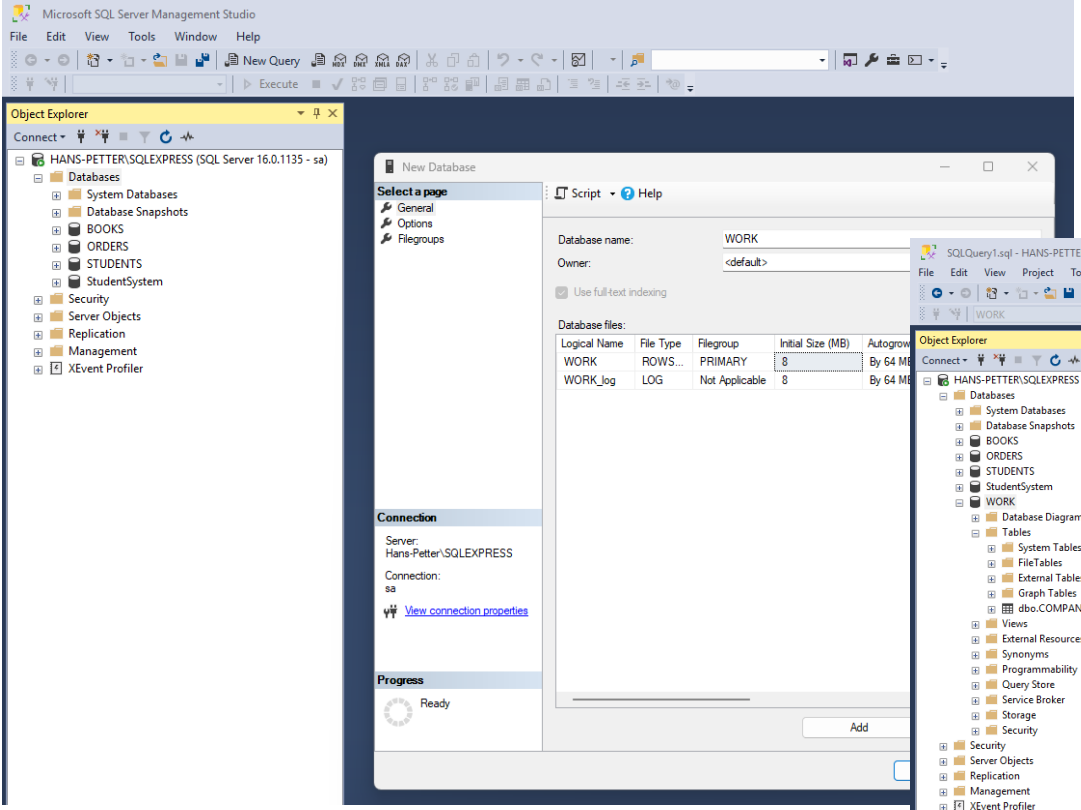
Let's create a basic Table like this:

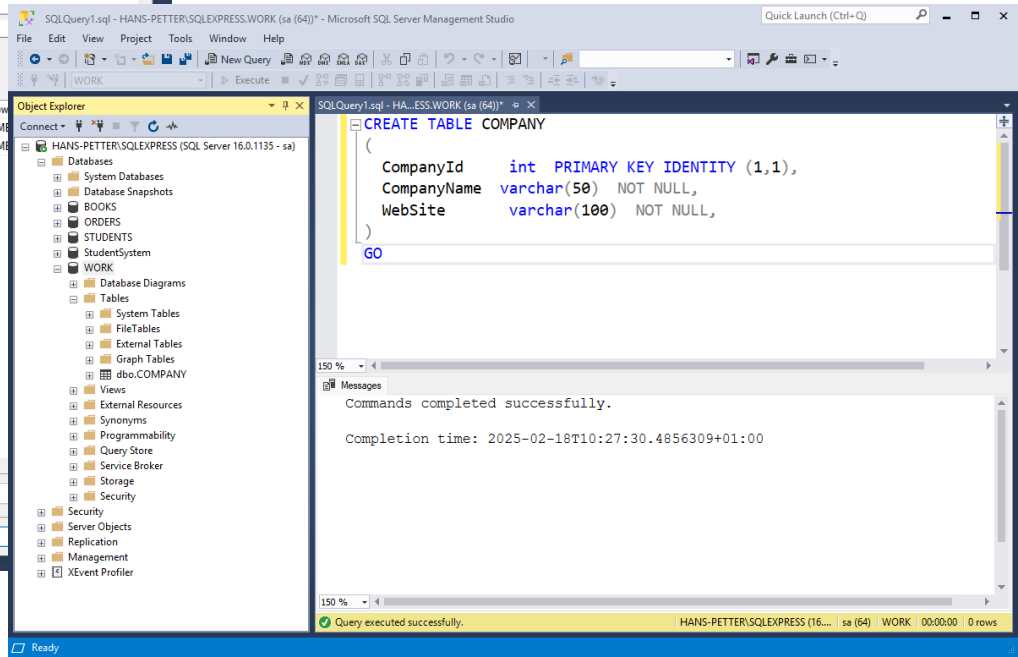
```
CREATE TABLE COMPANY
(
   CompanyId      int   PRIMARY KEY IDENTITY (1,1),
   CompanyName    varchar(50)   NOT NULL,
   WebSite        varchar(100)   NOT NULL,
)
GO
```

This is just a basic example, here you can add more columns like Address, Phone, etc.

Create Database and Table

We use SQL Server Management Studio to create the Database and the Table





Create Data

We also use SQL Server Management Studio to insert some dummy data into the COMPANY table:

```
insert into COMPANY (CompanyName, WebSite) values ('USN', 'www.usn.no')
GO

insert into COMPANY (CompanyName, WebSite) values ('Google', 'www.google.com')
GO

insert into COMPANY (CompanyName, WebSite) values ('Facebook', 'www.facebook.com')
GO

insert into COMPANY (CompanyName, WebSite) values ('Microsoft', 'www.microsoft.com')
GO

insert into COMPANY (CompanyName, WebSite) values ('Apple', 'www.apple.com')
GO
```

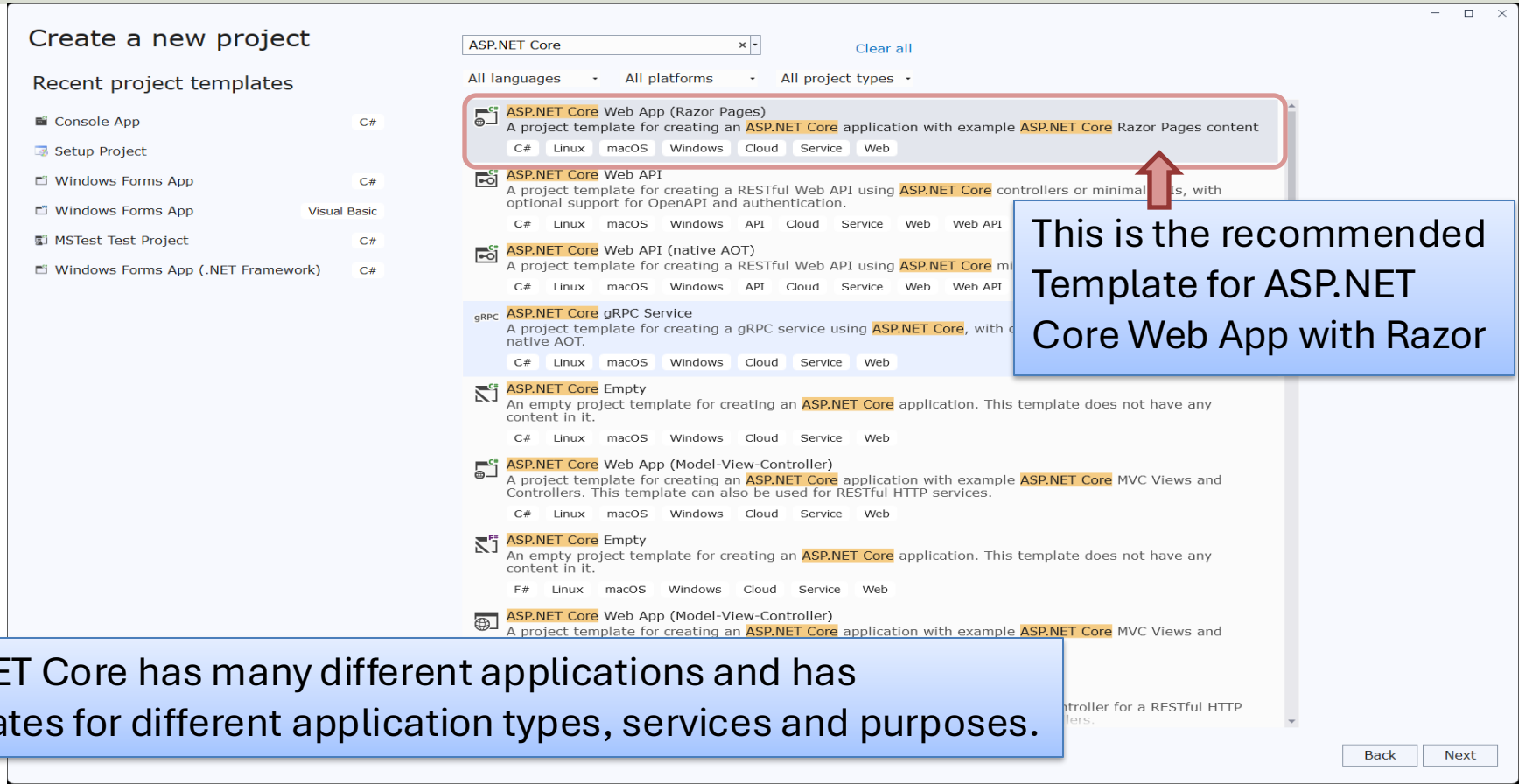
Visual Studio

Create the ASP.NET Core Razor App

Hans-Petter Halvorsen

Table of Contents

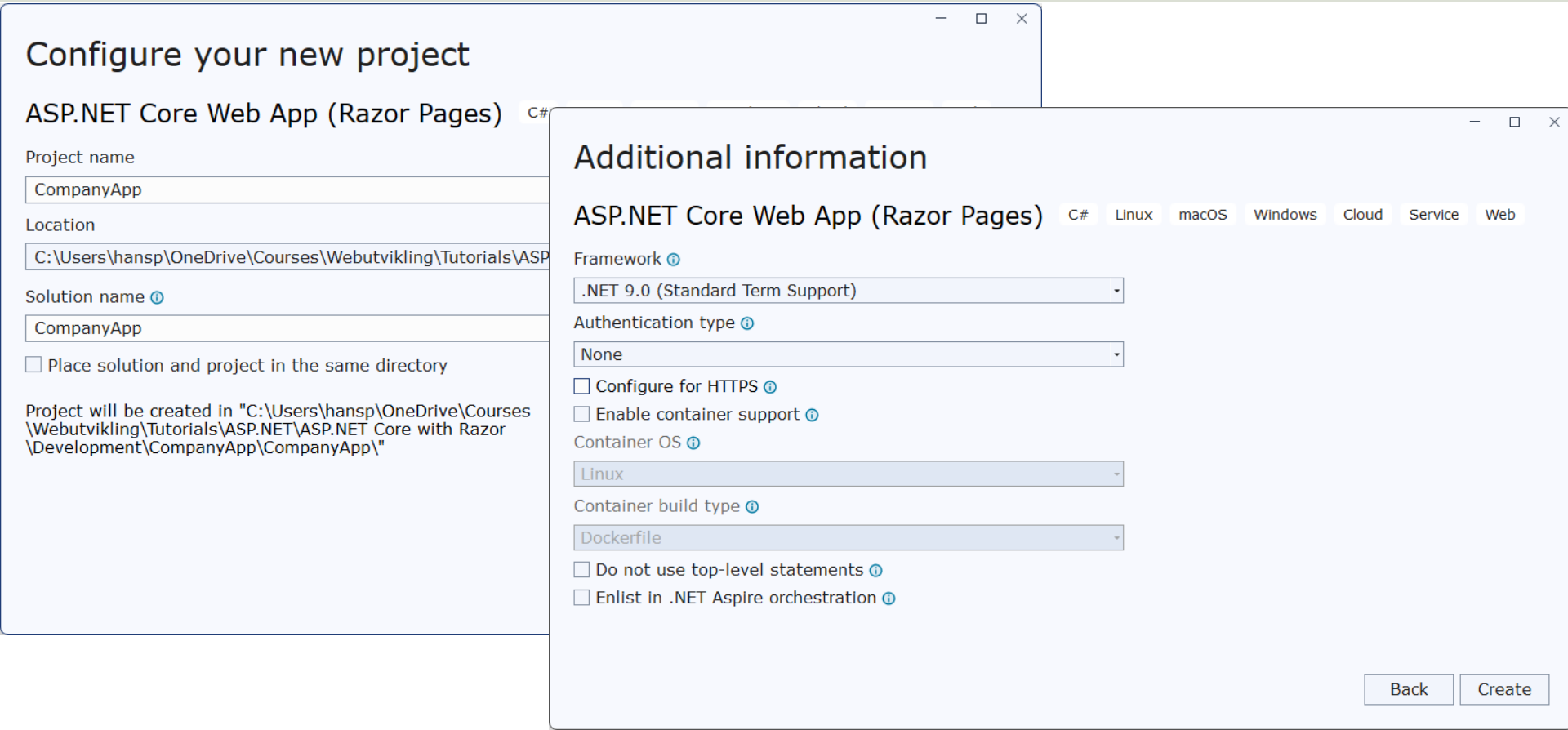
ASP.NET Core Web App with Razor



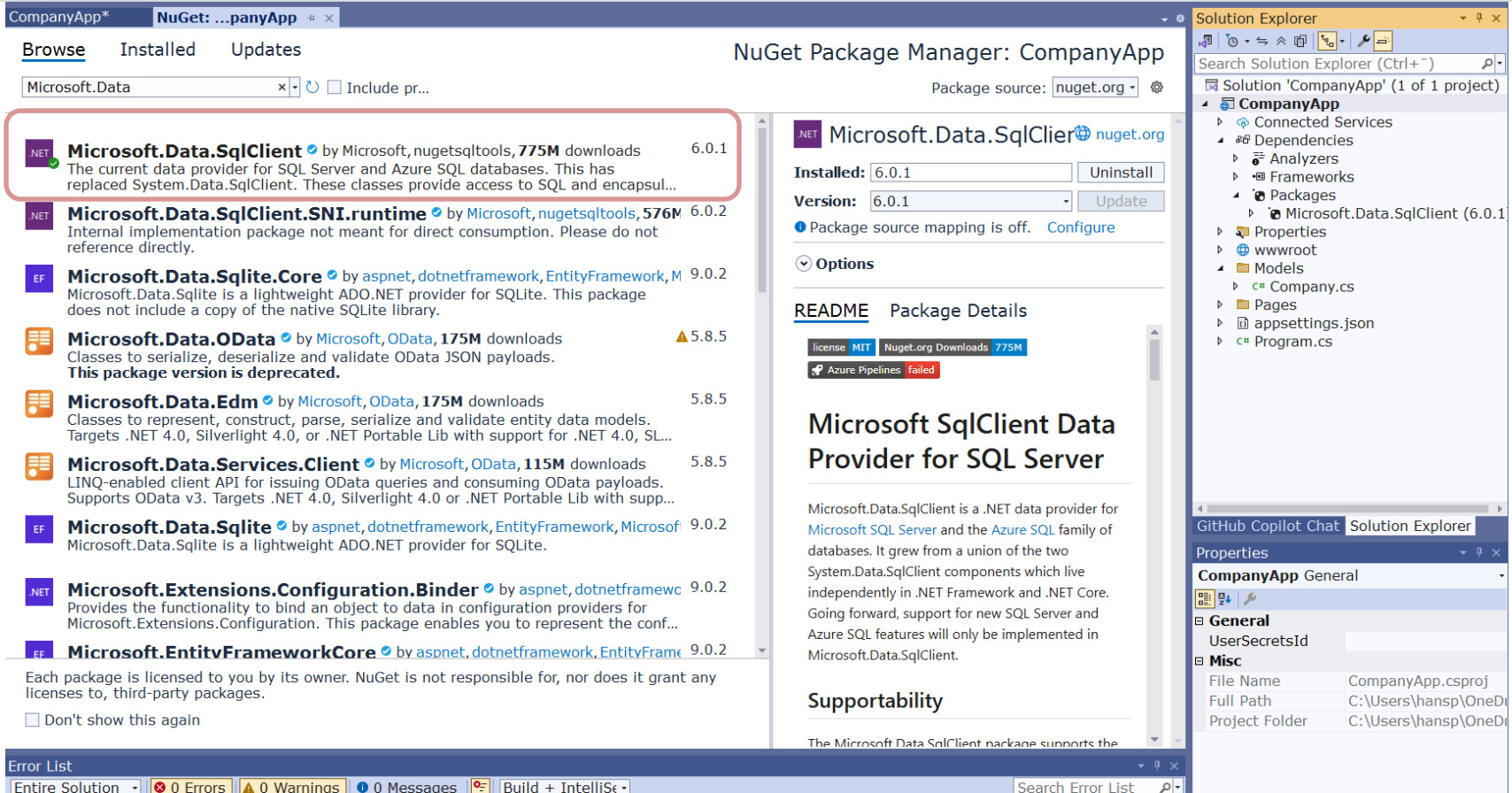
This is the recommended Template for ASP.NET Core Web App with Razor

ASP.NET Core has many different applications and has templates for different application types, services and purposes.

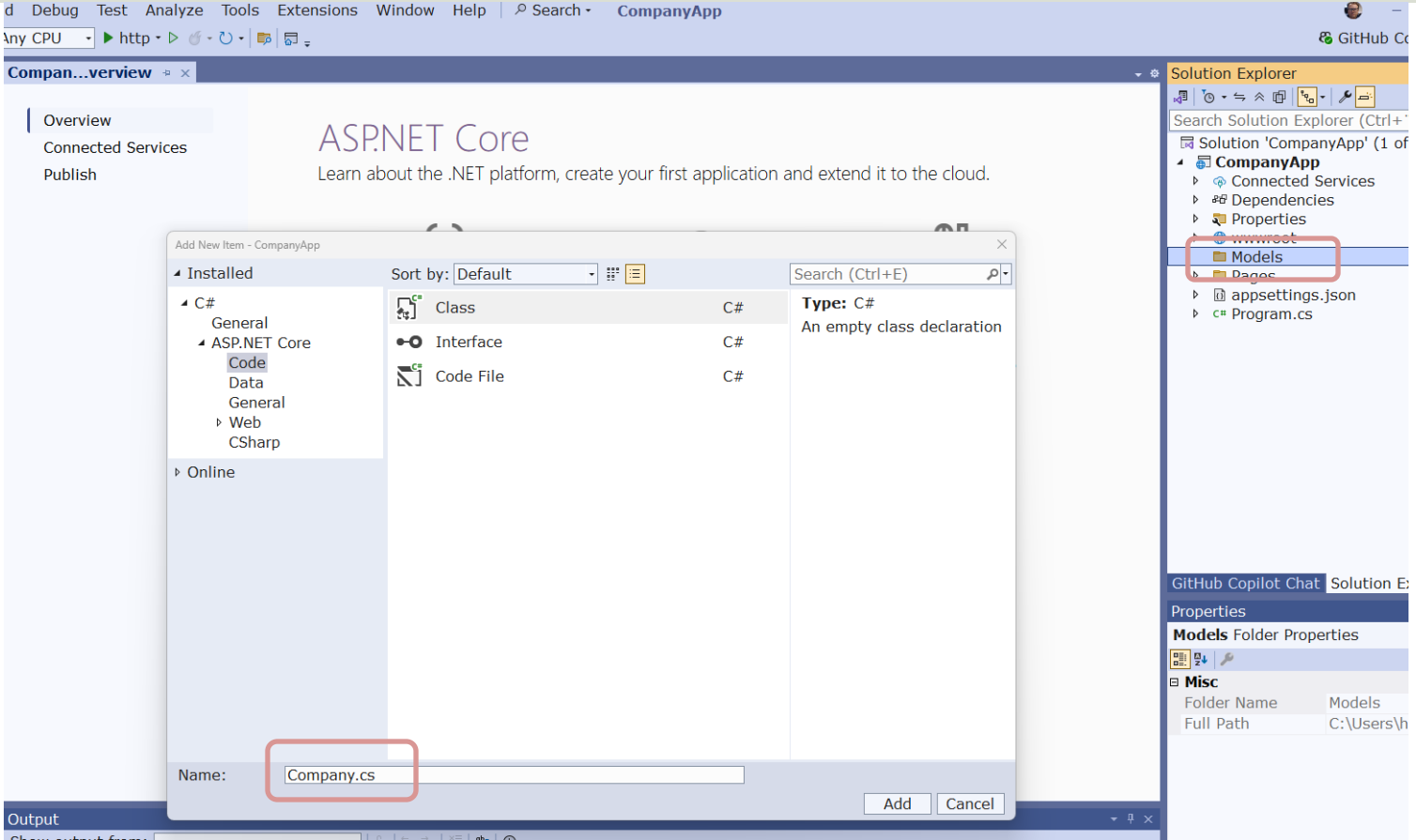
Visual Studio



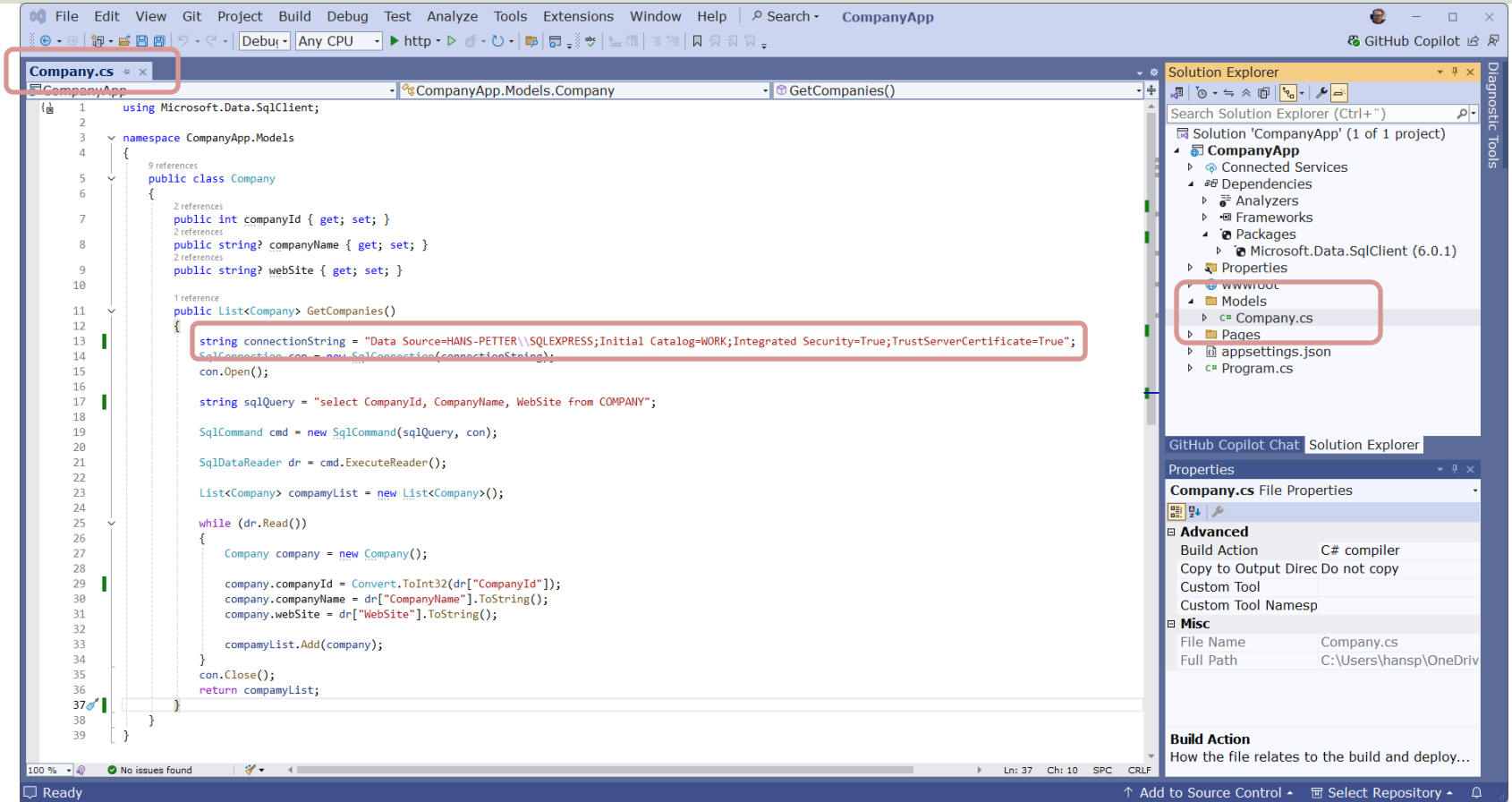
NuGet – Database Communication



Create Class



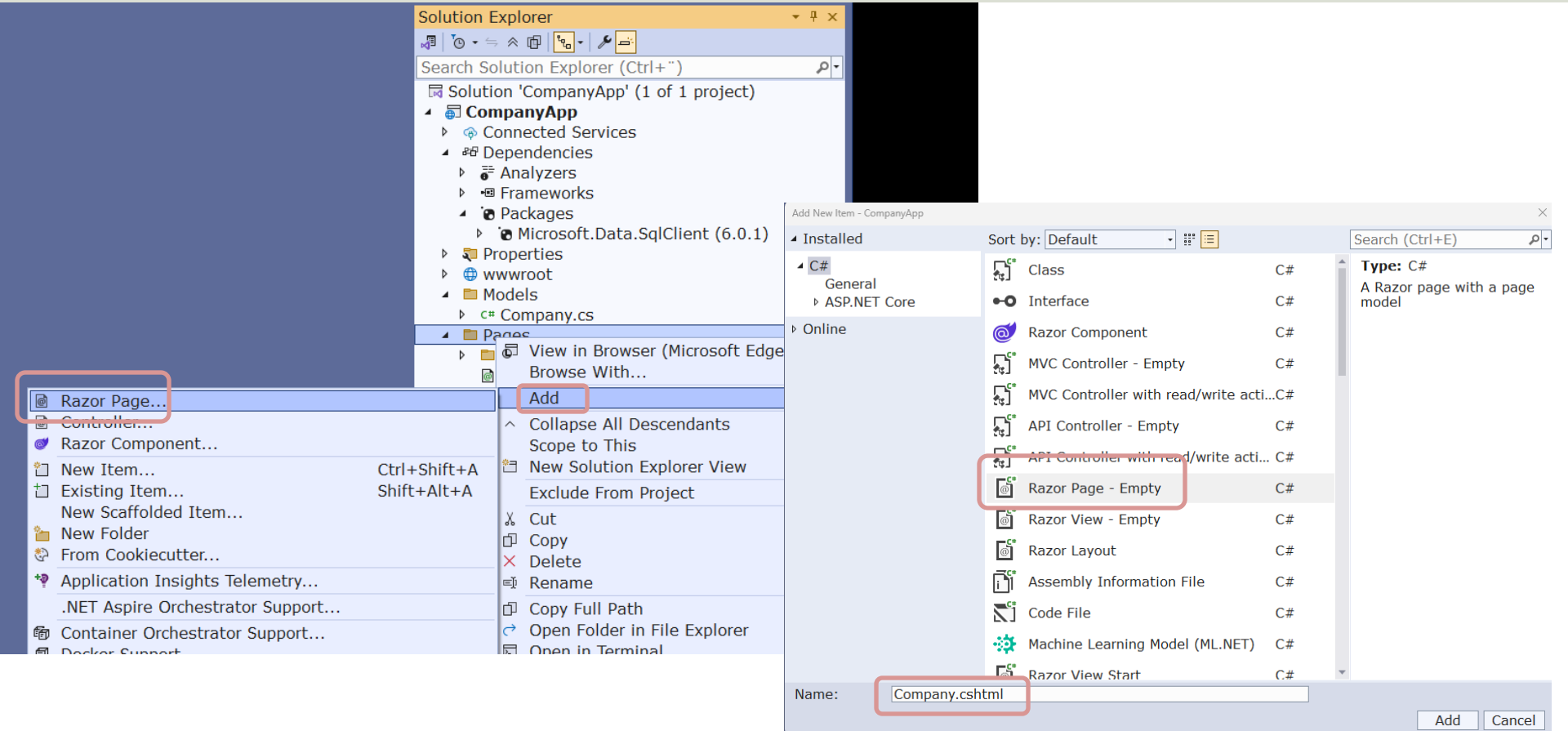
Company Class



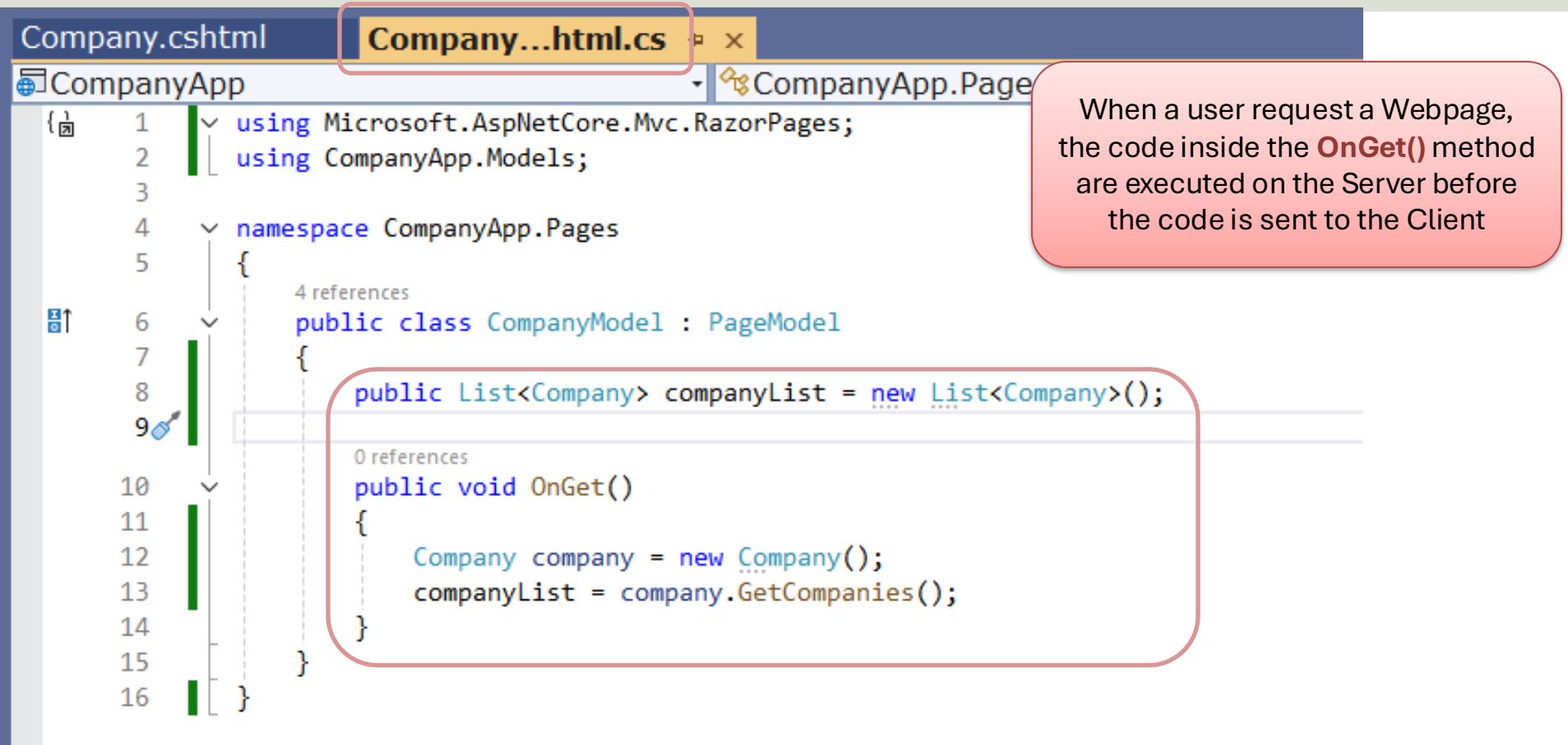
Company.cs

```
using Microsoft.Data.SqlClient;

namespace CompanyApp.Models
{
    public class Company
    {
        public int companyId { get; set; }
        public string? companyName { get; set; }
        public string? webSite { get; set; }

        public List<Company> GetCompanies()
        {
            string connectionString = "Data Source=SERVERNAME\\SQLEXPRESS;Initial Catalog=WORK;Integrated
               Security=True;TrustServerCertificate=True";
            SqlConnection con = new SqlConnection(connectionString);
            con.Open();

            string sqlQuery = "select CompanyId, CompanyName, WebSite from COMPANY";

            SqlCommand cmd = new SqlCommand(sqlQuery, con);

            SqlDataReader dr = cmd.ExecuteReader();

            List<Company> compamyList = new List<Company>();

            while (dr.Read())
            {
                Company company = new Company();

                company.companyId = Convert.ToInt32(dr["CompanyId"]);
                company.companyName = dr["CompanyName"].ToString();
                company.webSite = dr["WebSite"].ToString();

                compamyList.Add(company);
            }
            con.Close();
            return compamyList;
        }
    }
}
```

New Razor Page – “Company.cshtml”



“Company.cshtml.cs”



“Company.cshtml.cs”

```
using Microsoft.AspNetCore.Mvc.RazorPages;
using CompanyApp.Models;

namespace CompanyApp.Pages
{
    public class CompanyModel : PageModel
    {
        public List<Company> companyList = new List<Company>();

        public void OnGet()
        {
            Company company = new Company();
            companyList = company.GetCompanies();
        }
    }
}
```

Company.cshtml | Company.cshtml.cs

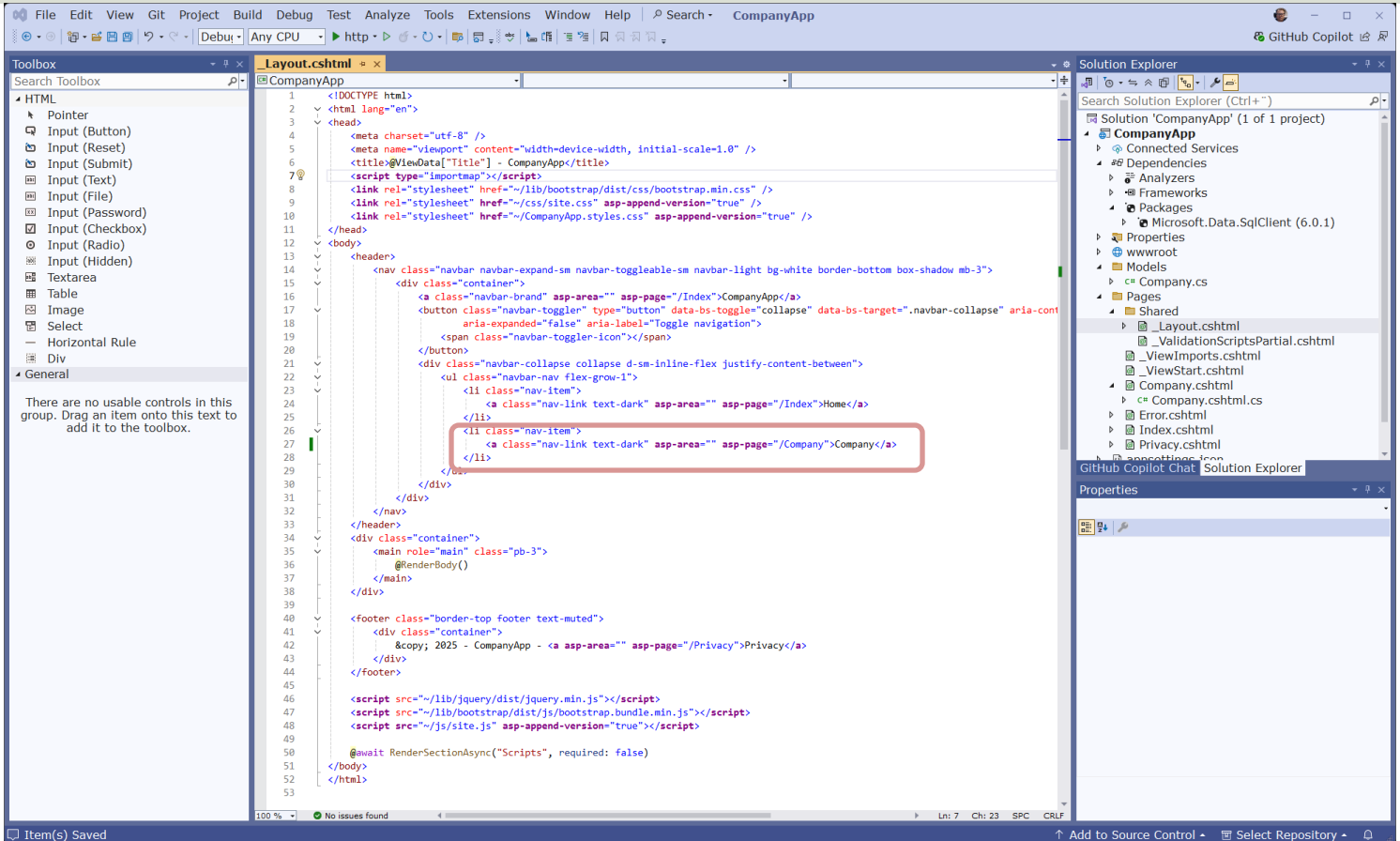
CompanyApp

```
1   @page
2   @model CompanyApp.Pages.CompanyModel
3   @{
4   }
5
6   <div>
7       <h1>Company List</h1>
8       <p>Here you see a list of avalible companies:</p>
9
10      <table class="table">
11          <thead>
12              <tr>
13                  <th>CompanyId</th>
14                  <th>CompanyName</th>
15                  <th>WebSite</th>
16              </tr>
17          </thead>
18          <tbody>
19              @foreach (var company in Model.companyList)
20              {
21                  <tr>
22                      <td> @company.companyId</td>
23                      <td> @company.companyName</td>
24                      <td> @company.webSite</td>
25                  </tr>
26              }
27          </tbody>
28      </table>
29  </div>
```

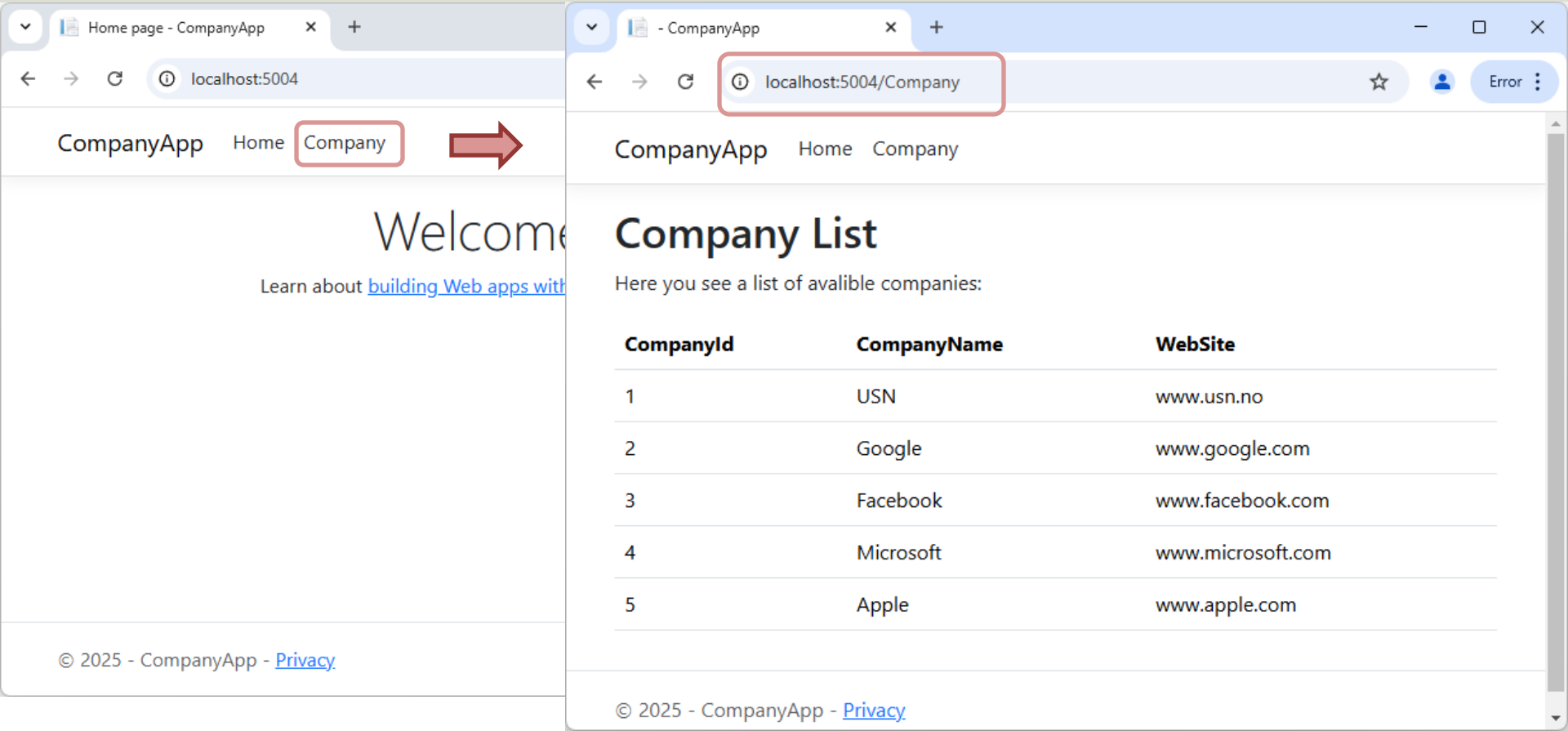
Company.cshtml

```
@page
@model CompanyApp.Pages.CompanyModel
@{
}

<div>
    <h1>Company List</h1>
    <p>Here you see a list of avalible companies:</p>

    <table class="table">
        <thead>
            <tr>
                <th>CompanyId</th>
                <th>CompanyName</th>
                <th>WebSite</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var company in Model.companyList)
            {
                <tr>
                    <td> @company.companyId</td>
                    <td> @company.companyName</td>
                    <td> @company.webSite</td>
                </tr>
            }
        </tbody>
    </table>
</div>
```

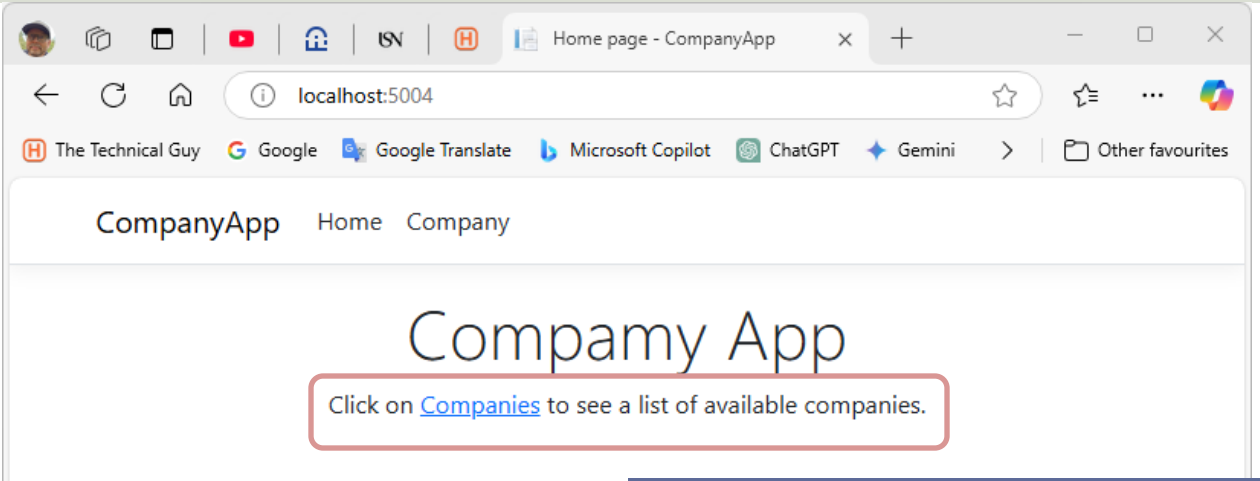
Add Web Page to Menu



Testing the Web Application



Create Hyperlink to “Company.cshtml”



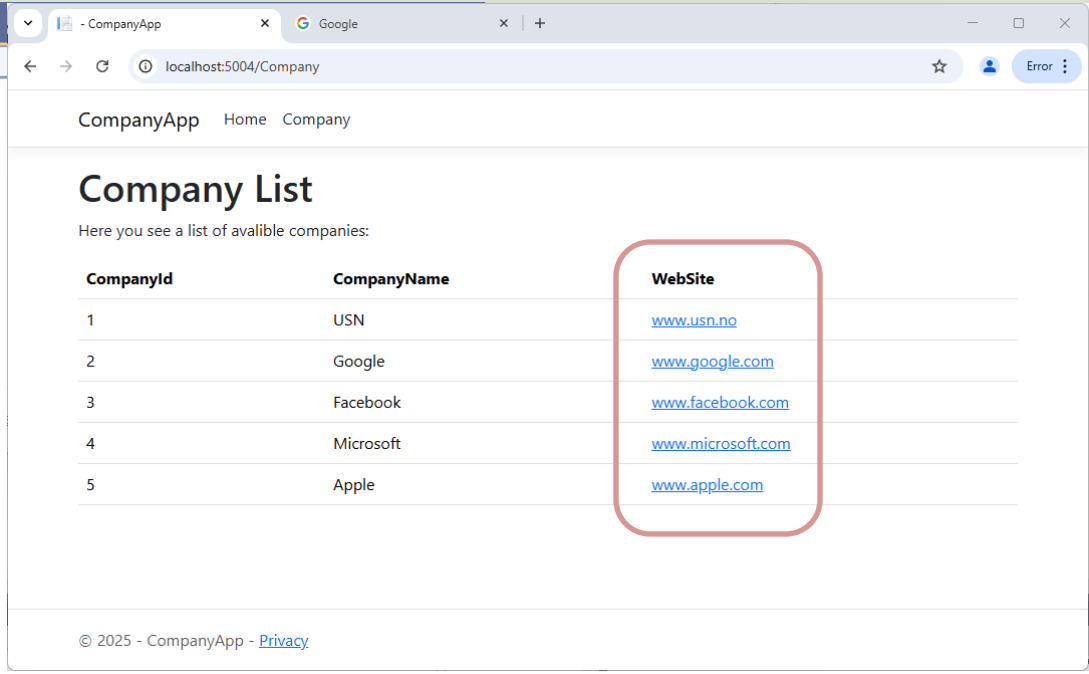
Index.cshtml

Create WebSite Hyperlinks

Company.cshtml

CompanyApp

```
1   @page
2   @model CompanyApp.Pages.CompanyModel
3   @{
4   }
5
6   <div>
7       <h1>Company List</h1>
8       <p>Here you see a list of avalible companies:</p>
9
10      <table class="table">
11          <thead>
12              <tr>
13                  <th>CompanyId</th>
14                  <th>CompanyName</th>
15                  <th>WebSite</th>
16              </tr>
17          </thead>
18          <tbody>
19              @foreach (var company in Model.companyList)
20              {
21                  <tr>
22                      <td> @company.companyId</td>
23                      <td> @company.companyName</td>
24                      <td><a href="https://@company.webSite" target="_blank">@company.webSite</a></td>
25                  </tr>
26              }
27          </tbody>
28      </table>
29  </div>
```



Resources and References

- Tutorial: Get started with Razor Pages in ASP.NET Core: https://learn.microsoft.com/en-us/aspnet/core/tutorials/razor-pages/razor-pages-start
- Learn Razor Pages: https://www.learnrazorpages.com
- Introduction to ASP.NET Core Razor Pages: https://www.csharp.com/article/introduction-to-asp-net-core-razor-pages/

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog