



THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH C# 6

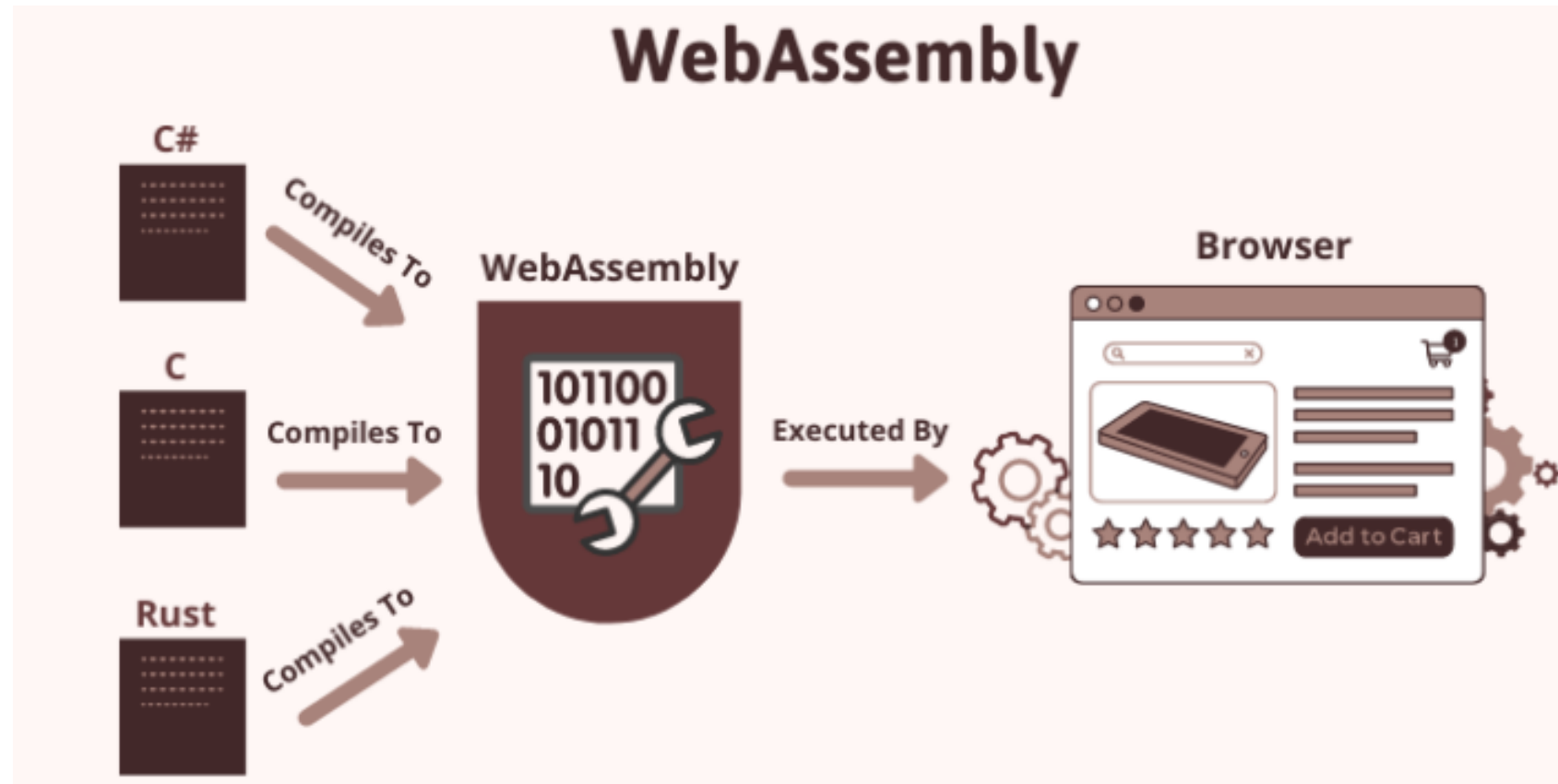
BÀI 2: BLAZOR SERVER

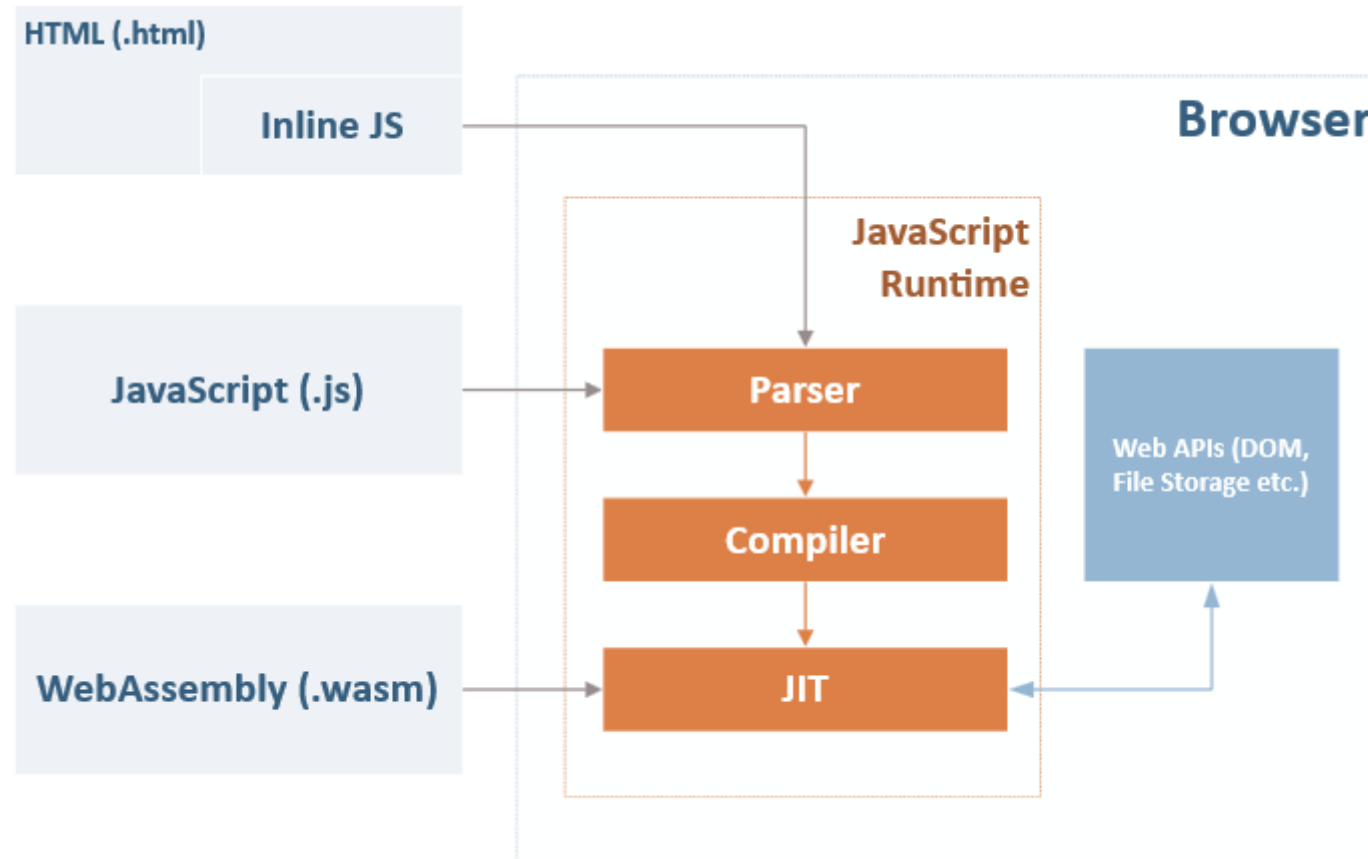
- ◉ WebAssembly
- ◉ WebAssembly và C#
- ◉ Blazor server



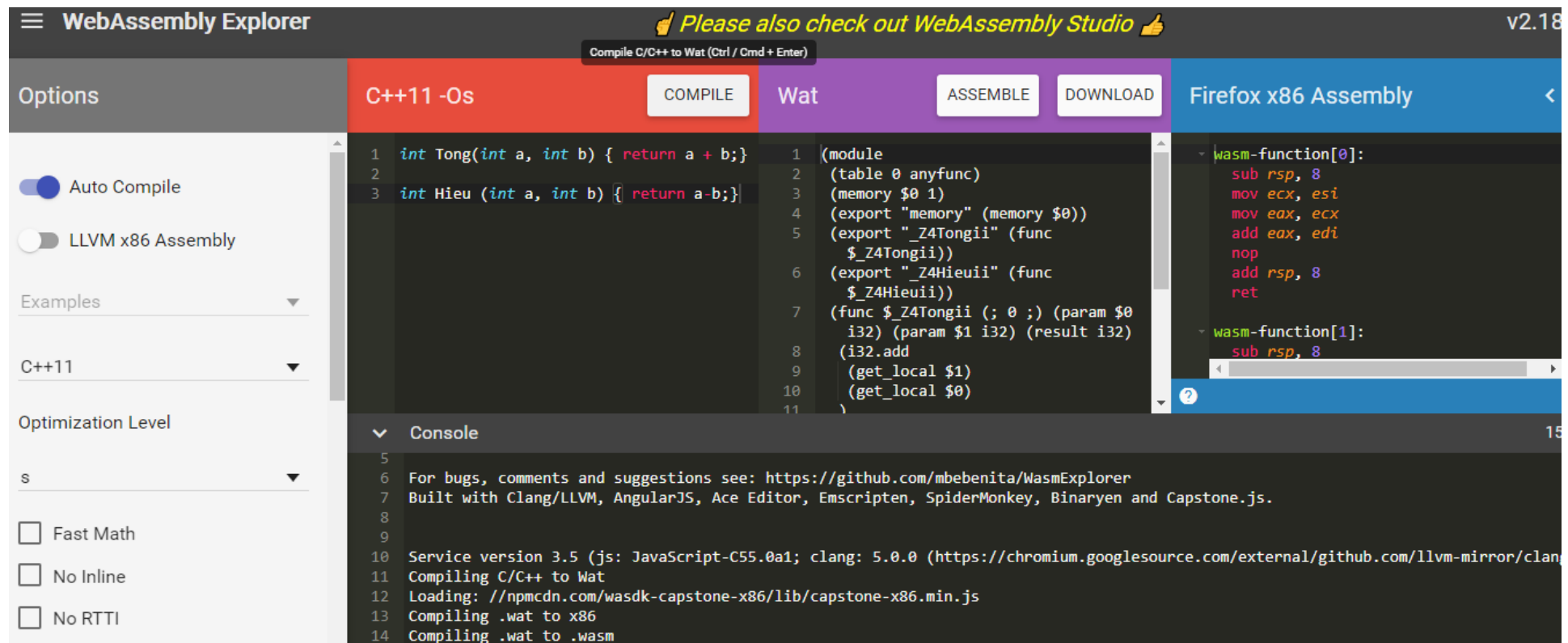


- ❑ Cùng với HTML, CSS và JavaScript, WebAssembly là ngôn ngữ thứ tư cho phép các đoạn mã thực thi trên các trình duyệt (browser).
- ❑ Định dạng nhị phân mức thấp cho web, dịch ra từ ngôn ngữ low-level khác như C/C++, C# và Rust
- ❑ Kết hợp bổ sung cho javascript
- ❑ Các trình duyệt phổ biến như Chrome, Firefox, Safari, Opera, Edge đều hỗ trợ WebAssembly.





❑ <https://mbebenita.github.io/WasmExplorer/>



WebAssembly Explorer v2.18

Please also check out WebAssembly Studio

Compile C/C++ to Wat (Ctrl / Cmd + Enter)

Options

- ☒ Auto Compile
- ☐ LLVM x86 Assembly
- Examples
- C++11
- Optimization Level
- s
- ☐ Fast Math
- ☐ No Inline
- ☐ No RTTI

C++11 -Os **COMPILE** **Wat** **ASSEMBLE** **DOWNLOAD** **Firefox x86 Assembly**

```

1 int Tong(int a, int b) { return a + b;}
2
3 int Hieu (int a, int b) { return a-b;}

```

```

1 (module
2   (table 0 anyfunc)
3   (memory $0 1)
4   (export "memory" (memory $0))
5   (export "_Z4Tongii" (func
6     $ _Z4Tongii))
7   (export "_Z4Hieuii" (func
8     $ _Z4Hieuii))
9   (func $ _Z4Tongii (; 0 ;) (param $0
10     i32) (param $1 i32) (result i32)
11     (i32.add
12       (get_local $1)
13       (get_local $0))
14   )

```

```

wasm-function[0]:
  sub rsp, 8
  mov ecx, esi
  mov eax, ecx
  add eax, edi
  nop
  add rsp, 8
  ret

wasm-function[1]:
  sub rsp, 8

```

Console

```

5
6 For bugs, comments and suggestions see: https://github.com/mbebenita/WasmExplorer
7 Built with Clang/LLVM, AngularJS, Ace Editor, Emscripten, SpiderMonkey, Binaryen and Capstone.js.
8
9
10 Service version 3.5 (js: JavaScript-C55.0a1; clang: 5.0.0 (https://chromium.googlesource.com/external/github.com/llvm-mirror/clang)
11 Compiling C/C++ to Wat
12 Loading: //npmcdn.com/wasdk-capstone-x86/lib/capstone-x86.min.js
13 Compiling .wat to x86
14 Compiling .wat to .wasm

```

❑ Tích hợp webAssembly vào website

scripts

```

1  let tong, hieu;
2  function loadWebAssembly(fileName) {
3      return fetch(fileName)
4          .then(response => response.arrayBuffer())
5          .then(buffer => WebAssembly.compile(buffer))
6          .then(module => {return new WebAssembly.Instance(module) });
7  };
8
9  loadWebAssembly('http://localhost:80/testWebAssembly.wasm')
10 .then(instance => {
11     tong = instance.exports._Z4Tongii;
12     hieu = instance.exports._Z4Hieuii;
13     document.getElementById("tong").innerHTML = "Tong 2 va 5 la: " + tong(2,5);
14     document.getElementById("hieu").innerHTML = "Hieu 2 va 5 la: " + hieu(2,5);
15 });
        
```

Name

- ..
- index.html
- scripts.js
- testWebAssembly.wasm

localhost

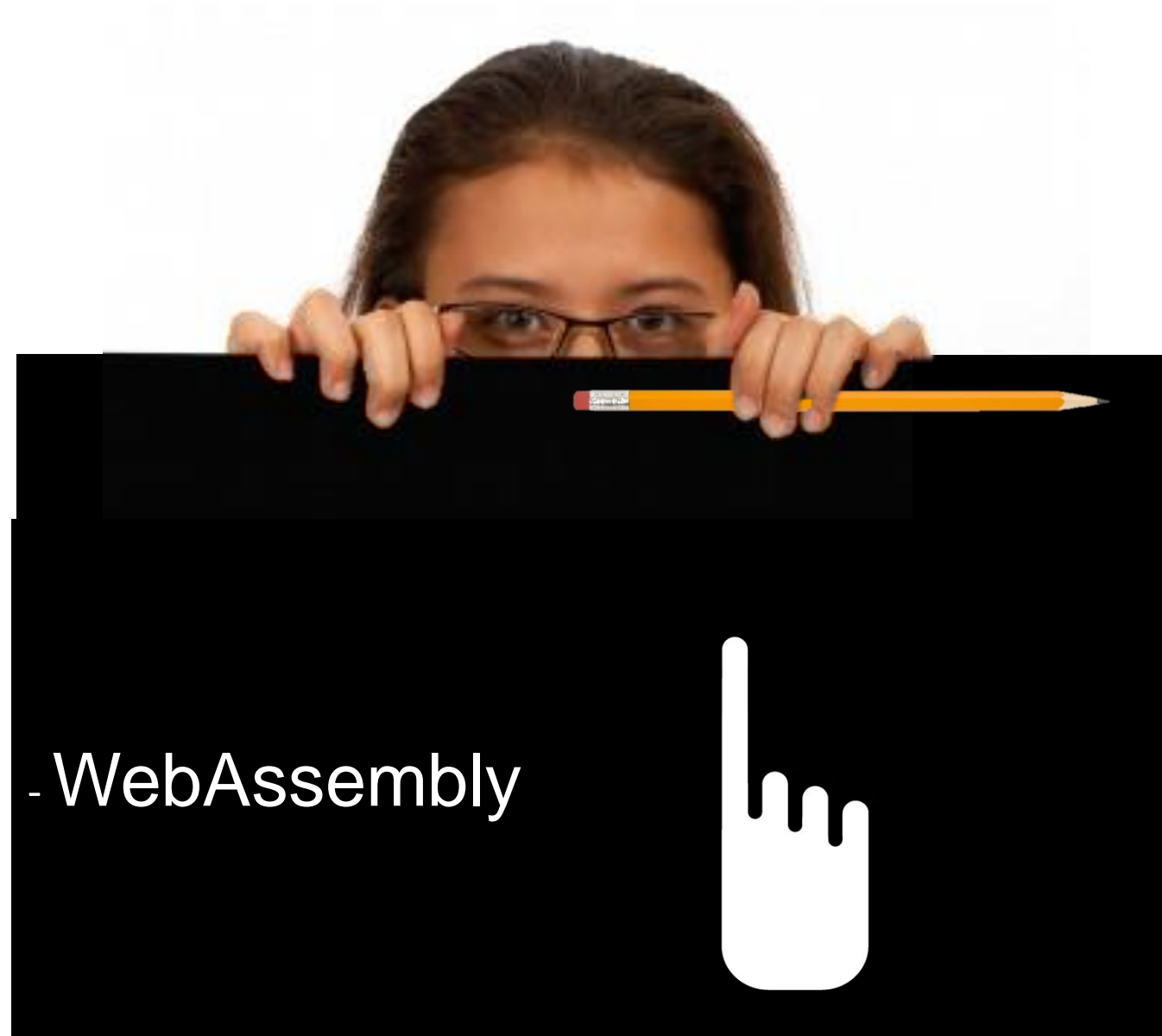
Hi Thepv,Demo WebAssembly!

Tong 2 va 5 la: 7

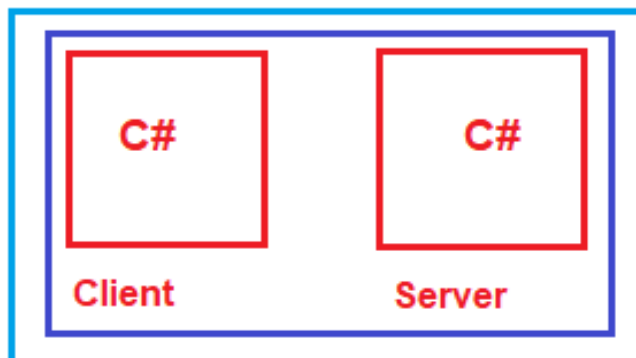
Hieu 2 va 5 la: -3

```

<html>
<head>
  <script src="scripts.js"></script>
</head>
<body>
  <h1>Hi Thepv,Demo WebAssembly! </h1>
  <h3 id="tong"></h3>
  <h3 id="hieu"></h3>
</body>
</html>
        
```



- ❑ C# dịch sang WebAssembly và chạy trên trình duyệt như một ứng dụng native: Blazor



BLAZOR



+



WEBASSEMBLY

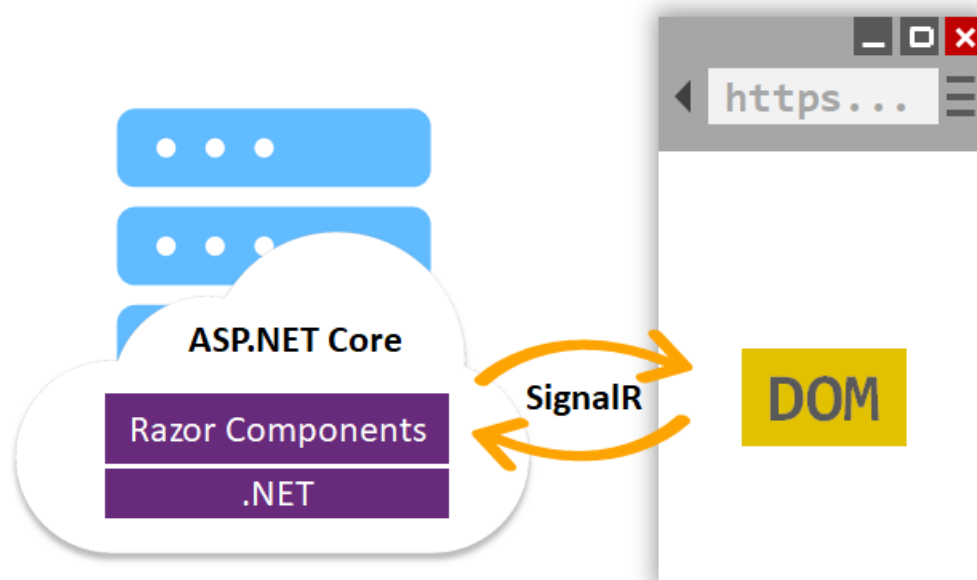
+



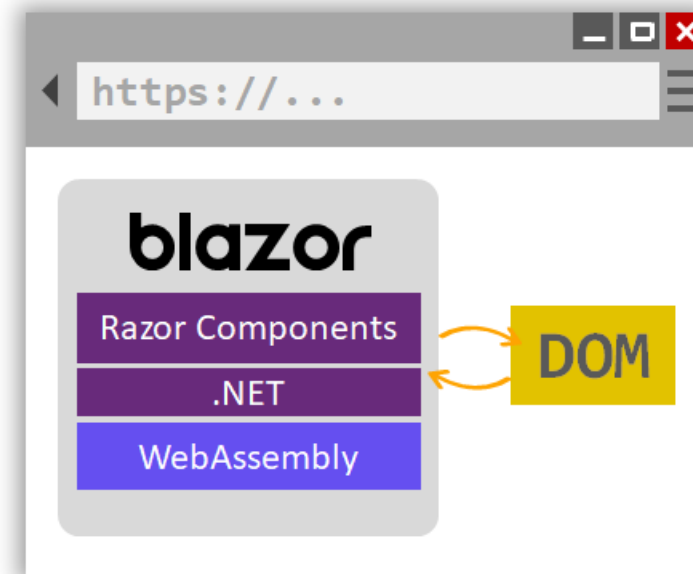


1. Blazor Server
2. Blazor WebAssembly
3. Blazor Electron
4. Mobile Blazor Bindings

Server-side Blazor









Client-Side Blazor



Create a new project

Recent project templates

-  ASP.NET Core Web Application C#
-  NUnit Test Project (.NET Core) C#
-  NUnit Test Project (.NET Core) F#
-  Class Library (.NET Core) C#
-  xUnit Test Project (.NET Core) C#
-  Class Library (.NET Standard) C#

Search for templates (Alt+S)

All languages

All platforms

All project types

Windows, Linux and macOS

Visual Basic

Windows

Linux

macOS

Console



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly (wasm). These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C#

Linux

macOS

Windows

Cloud

Web



ASP.NET Web Application (.NET Framework)

Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.

Visual Basic

Windows

Cloud

Web

Next

Create a new Blazor app

.NET 5.0



Blazor Server App

A project template for creating a Blazor server app that runs server-side inside an ASP.NET Core app and handles user interactions over a SignalR connection. This template can be used for web apps with rich dynamic user interfaces (UIs).



Blazor WebAssembly App

A project template for creating a Blazor app that runs on WebAssembly. This template can be used for web apps with rich dynamic user interfaces (UIs).

Authentication

No Authentication

[Change](#)

Advanced

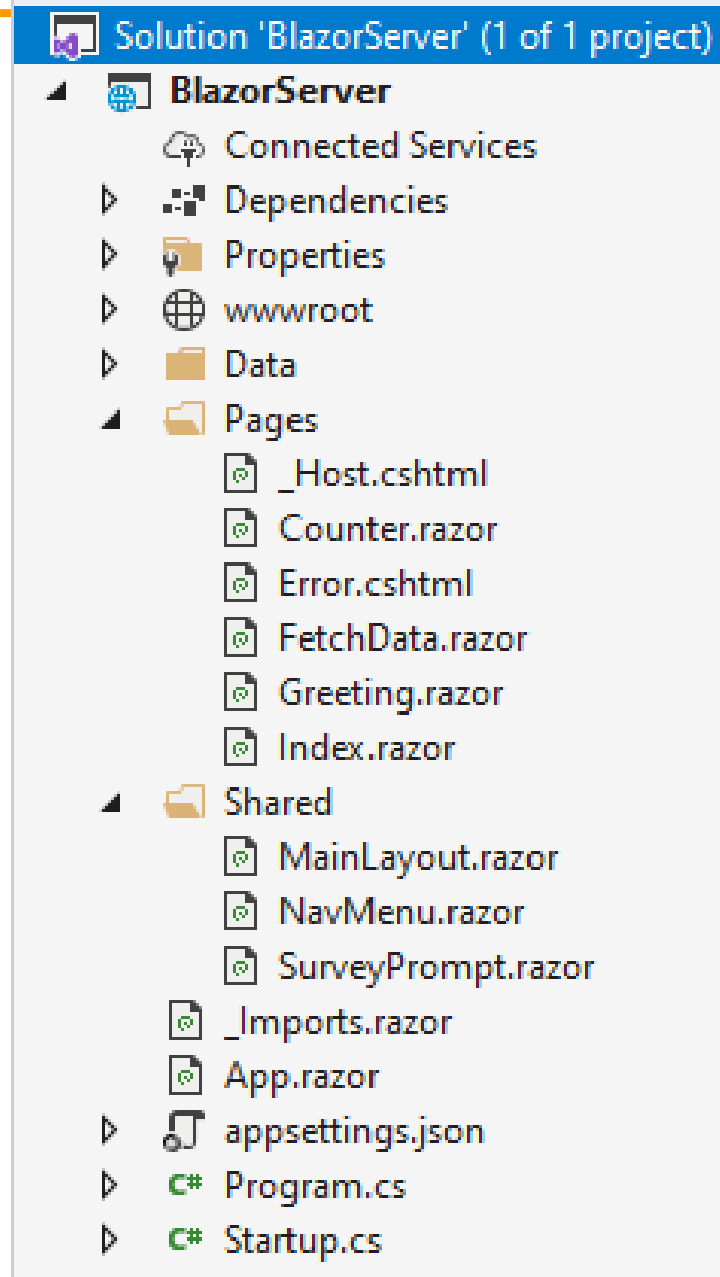
☐ Configure for HTTPS

☐ Enable Docker Support

(Requires [Docker Desktop](#))

Linux

BLAZOR SERVER PROJECT STRUCTURE



□ Program.cs

```
public static void Main(string[] args)
{
    CreateHostBuilder(args).Build().Run();
}

public static IHostBuilder CreateHostBuilder(string[] args) =>
    Host.CreateDefaultBuilder(args)
        .ConfigureWebHostDefaults(webBuilder =>
        {
            webBuilder.UseStartup<Startup>();
        });
```

□ Startup.cs

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddRazorPages();
    services.AddServerSideBlazor();
    services.AddSingleton<WeatherForecastService>();
}
```

```
1 app.UseEndpoints(endpoints =>
2 {
3     endpoints.MapBlazorHub();
4     endpoints.MapFallbackToPage("/_Host");
5 });
```


□ _Host.cshtml

```
<component type="typeof(App)" render-mode="ServerPrerendered" />
```

```
<script src="_framework/blazor.server.js"></script>
```

□ App.razor

```
1 <Router AppAssembly="@typeof(Program).Assembly" PreferExactMatches="@true">
2     <Found Context="routeData">
3         <RouteView RouteData="@routeData" DefaultLayout="@typeof(MainLayout)" />
4     </Found>
5     <NotFound>
6         <LayoutView Layout="@typeof(MainLayout)">
7             <p>Sorry, there's nothing at this address.</p>
8         </LayoutView>
9     </NotFound>
10 </Router>
```

❑ MainLayout.cshtml

```

1  @inherits LayoutComponentBase
2
3  <div class="page">
4      <div class="sidebar">
5          <NavMenu />
6      </div>
7
8      <div class="main">
9          <div class="top-row px-4">
10             <a href="https://docs.microsoft.com/aspnet/" target="_blank">About</a>
11          </div>
12
13          <div class="content px-4">
14              @Body
15          </div>
16      </div>
17 </div>

```

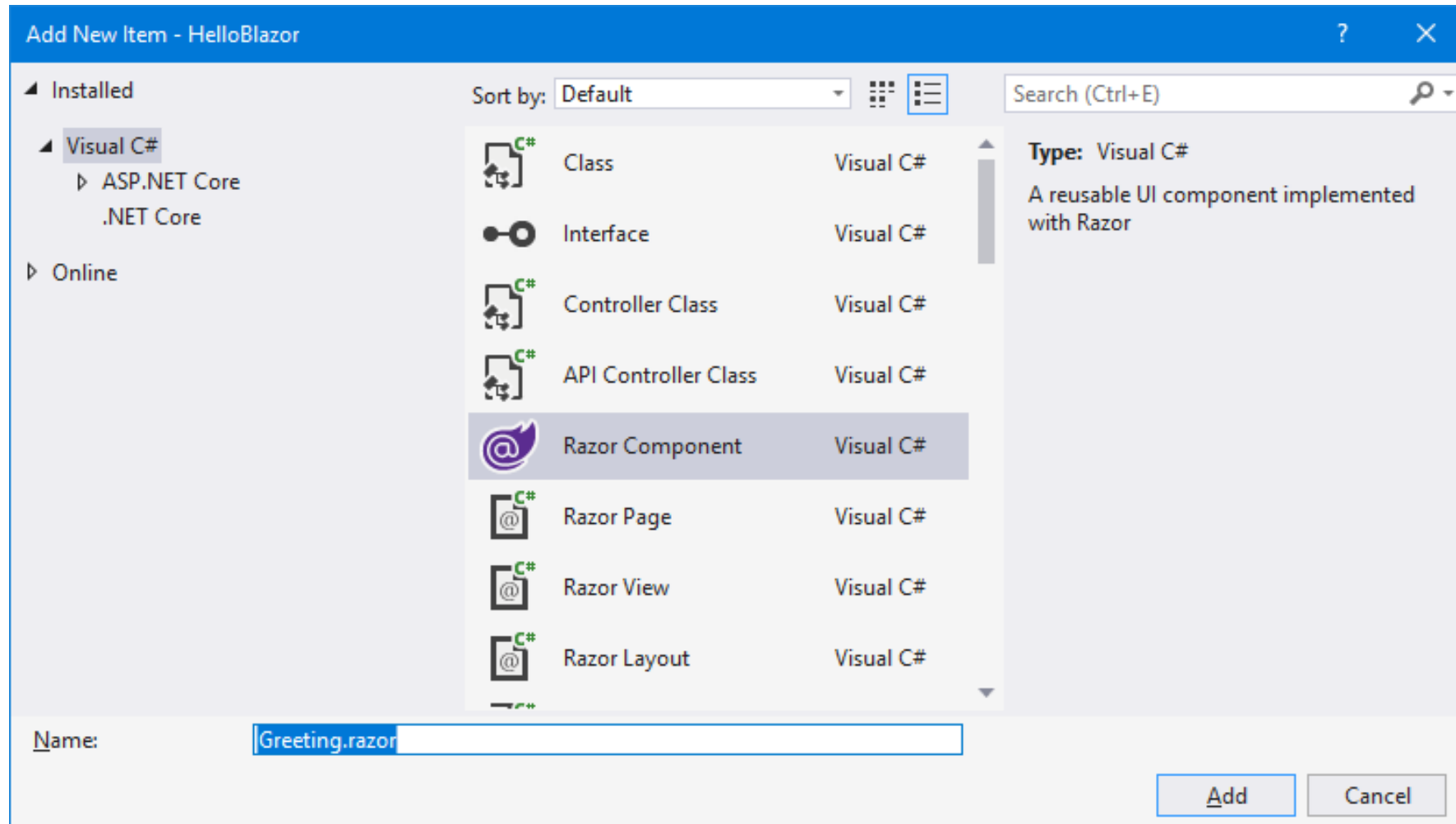
□_Imports.razor

```
@using System.Net.Http
@using Microsoft.AspNetCore.Authorization
@using Microsoft.AspNetCore.Components.Authorization
@using Microsoft.AspNetCore.Components.Forms
@using Microsoft.AspNetCore.Components.Routing
@using Microsoft.AspNetCore.Components.Web
@using Microsoft.AspNetCore.Components.Web.Virtualization
@using Microsoft.JSInterop
```

The screenshot shows a web browser window with the address bar displaying 'localhost:64573'. The page title is 'BlazorServerApp'. The main content area displays 'Hello, world!' and 'Welcome to your new app.' A sidebar on the left contains links for 'Home', 'Counter', and 'Fetch data'. A DevTools Network tab is open, showing a list of resources. A red arrow points to the 'blazor.server.js' file, and another red arrow points to the '_blazor?id=C14mNmKxgwoRS29cJ_RzAw' resource.

Name	Status	Type	Initiator	Size	T...	Waterfall
localhost	200	docum...	Other	1.9 kB	8...	
bootstrap.min.css	304	stylesh...	(index)	259 B	4...	
site.css	304	stylesh...	(index)	259 B	4...	
BlazorServerApp.styles.css	304	stylesh...	(index)	259 B	4...	
blazor.server.js	304	script	(index)	273 B	4...	
open-iconic-bootstrap.min.css	304	stylesh...	site.css	259 B	1...	
open-iconic.woff	304	font	open-icon...	272 B	3...	
negotiate?negotiateVersion=1	200	fetch	blazor.ser...	476 B	5...	
favicon.ico	200	x-icon	Other	5.7 kB	1...	
_blazor?id=C14mNmKxgwoRS29cJ_RzAw	101	webso...	blazor.ser...	0 B	P...	

❑ Tạo mới page trong blazor



❑ Tạo mới page trong blazor

```

Greeting.razor Counter.razor NavMenu.razor BlazorServer
@page "/greeting"
<h3> Hello Blazor server demo</h3>

@code {
    private string title = "Thepv, hi Blazor server";
    private string name;
    private string address;
    private string greeting;

    private void welcome()
    {
        if (!string.IsNullOrEmpty(name) && !string.IsNullOrEmpty(address))
        {
            greeting = $"Welcome {name} from {address} go to blazor app ";
        }
    }
}

<div class="form-group">
    <input type="text" placeholder="your name" @bind="name" class="form-control" />
</div>
<div class="form-group">
    <input type="text" placeholder="your address" @bind="address" class="form-control" />
</div>
<div class="form-group">
    <button @onclick="welcome" class="btn btn-primary">Welcome</button>
</div>
<h3>@greeting</h3>
  
```

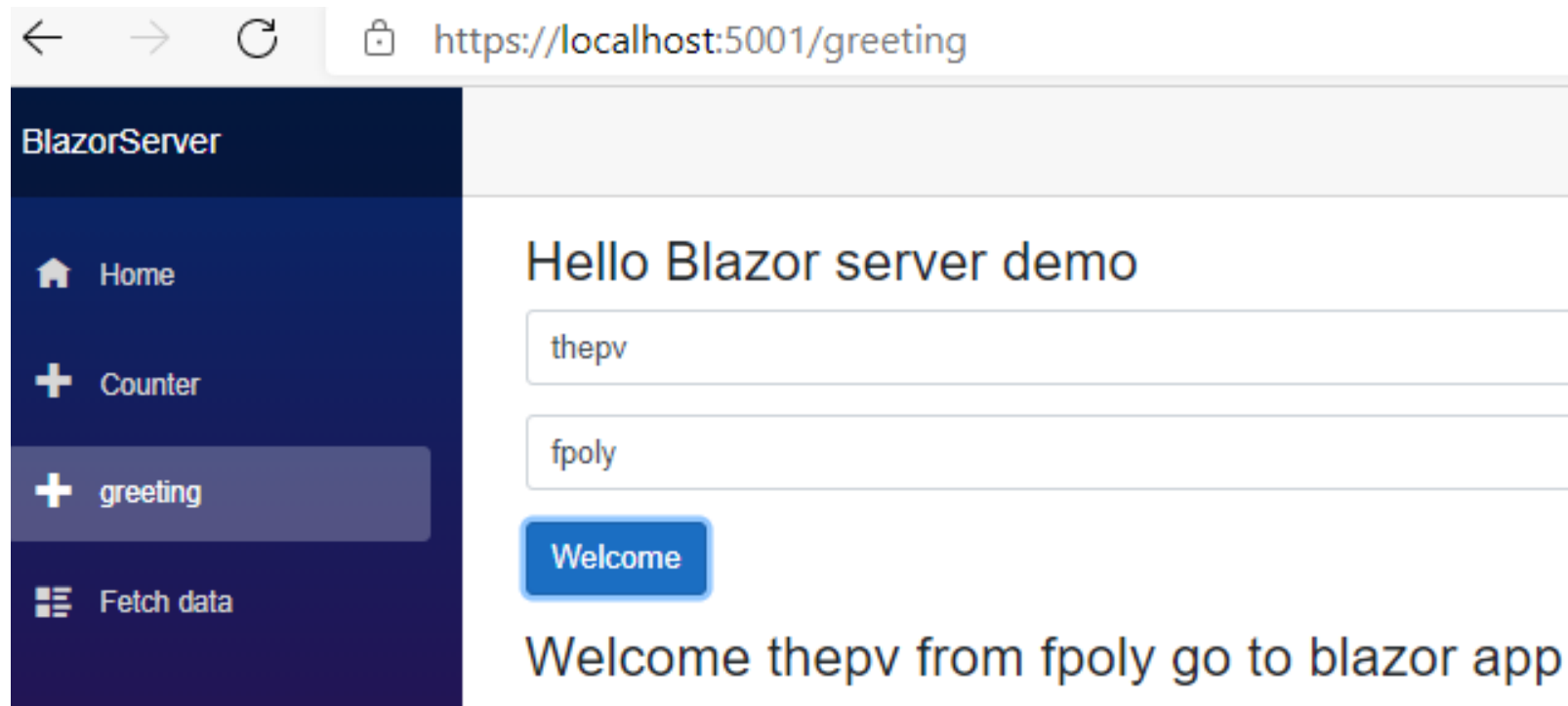
❑ Tạo mới page trong blazor

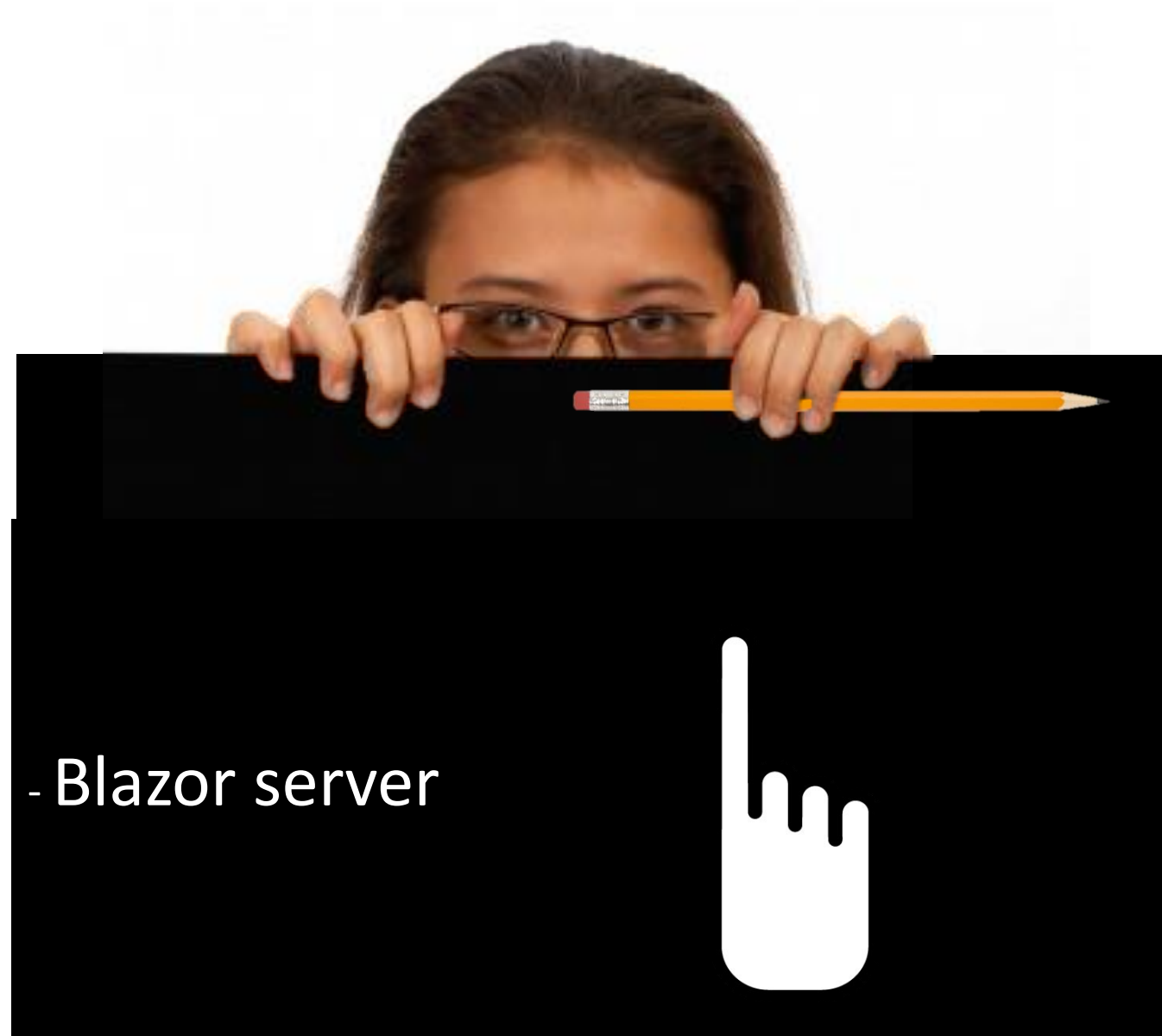
```

    NavMenu.razor  X Greeting.razor  Counter.razor  BlazorServer
  </div>

  <div class="@NavMenuCssClass" @onclick="ToggleNavMenu">
    <ul class="nav flex-column">
      <li class="nav-item px-3">
        <NavLink class="nav-link" href="" Match="NavLinkMatch.All">
          <span class="oi oi-home" aria-hidden="true"></span> Home
        </NavLink>
      </li>
      <li class="nav-item px-3">
        <NavLink class="nav-link" href="counter">
          <span class="oi oi-plus" aria-hidden="true"></span> Counter
        </NavLink>
      </li>
      <li class="nav-item px-3">
        <NavLink class="nav-link" href="greeting">
          <span class="oi oi-plus" aria-hidden="true"></span> greeting
        </NavLink>
      </li>
      <li class="nav-item px-3">
        <NavLink class="nav-link" href="fetchdata">
          <span class="oi oi-list-rich" aria-hidden="true"></span> Fetch data
        </NavLink>
      </li>
    </ul>
  </div>
  
```

❑ Tạo mới page trong blazor







THỰC HỌC – THỰC NGHIỆP

Conceive Design Implement Operate

LẬP TRÌNH C# 6

BÀI 2: BLAZOR SERVER P2.3

- **Blazor is a component driven framework**
- **Components can be nested, reused, and shared across multiple projects**
- **Component files have the extension .razor**
- **Razor components or Blazor components**
- **Component name must start with an uppercase character**



Directive

```
@page "/fetchdata"
@using WebApplication33.Data
@inject WeatherForecastService ForecastService
```

Markup / Razor

```
<h1>Weather forecast</h1>

<p>This component demonstrates fetching data from a service.</p>

@if (forecasts == null)
{
    <p><em>Loading ... </em></p>
}
else
{
    <table class="table"> ... </table>
}
```

Component Logic

```
@code {
    WeatherForecast[] forecasts;

    protected override async Task OnInitAsync()
    {
        forecasts = await ForecastService.GetForecastAsync(DateTime.Now);
    }
}
```

- ❑ reuse parts of your code for multiple components
- ❑ readability and maintainability
- ❑ code-behind

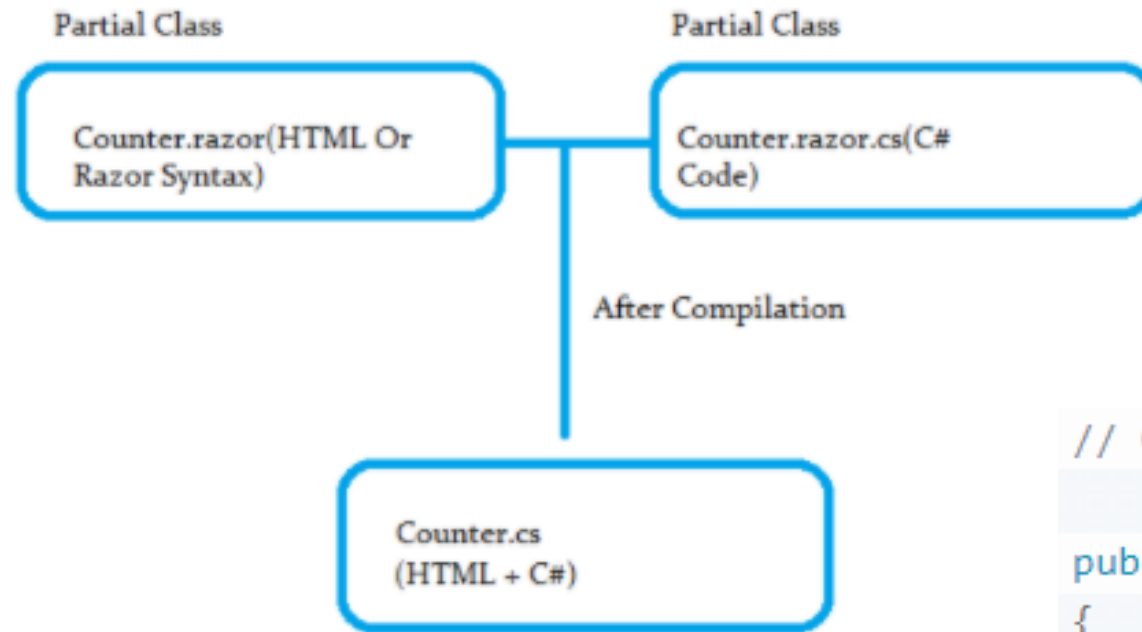
Partial Files

OR

Base Class

SPLITTING RAZOR COMPONENTS IN BLAZOR APPLICATION

❑ Partial files approach

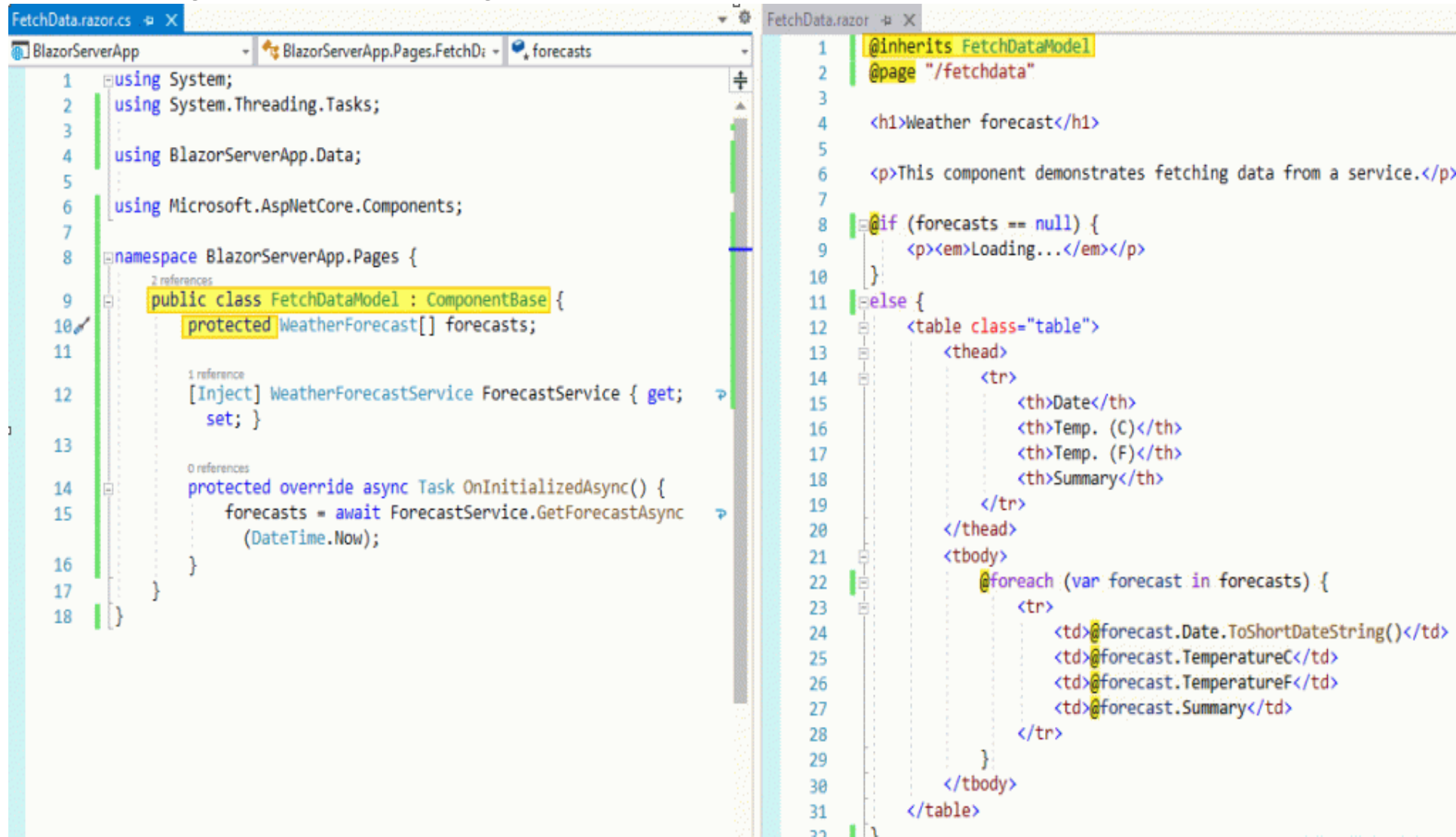


```
// Counter.razor.cs

public partial class Counter
{
    private int currentCount = 0;

    private void IncrementCount()
    {
        currentCount++;
    }
}
```

Code-behind trong Blazor sử dụng kế thừa



The image shows two side-by-side code editors in Visual Studio. The left editor shows the code-behind file (FetchData.razor.cs) and the right editor shows the code-behind file (FetchData.razor).

FetchData.razor.cs (Left):

```

1 using System;
2 using System.Threading.Tasks;
3
4 using BlazorServerApp.Data;
5
6 using Microsoft.AspNetCore.Components;
7
8 namespace BlazorServerApp.Pages {
9     public class FetchDataModel : ComponentBase {
10         protected WeatherForecast[] forecasts;
11
12         [Inject] WeatherForecastService ForecastService { get; set; }
13
14         protected override async Task OnInitializedAsync() {
15             forecasts = await ForecastService.GetForecastAsync(
16                 DateTime.Now);
17         }
18     }
19 }
  
```

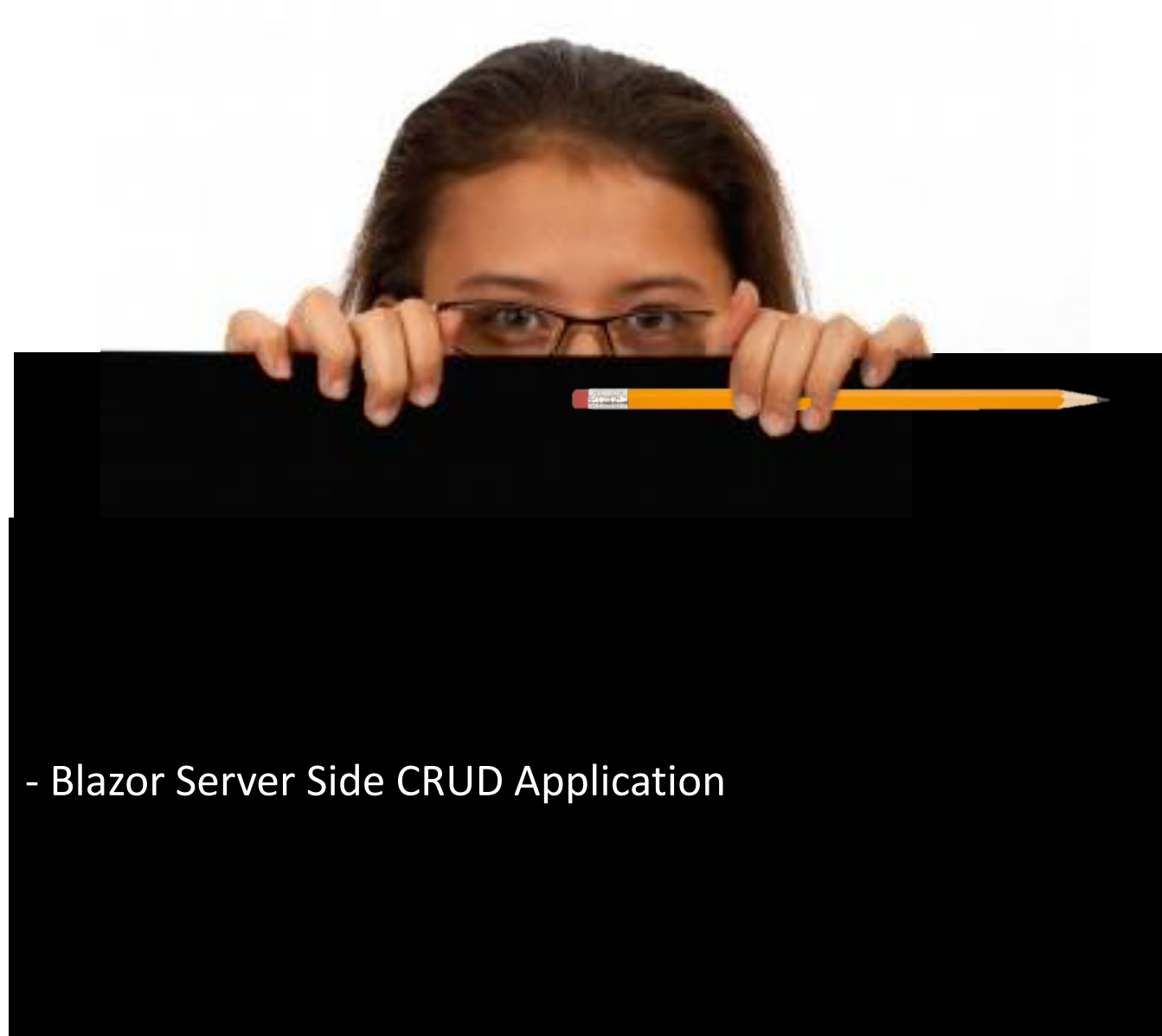
FetchData.razor (Right):

```

1 @inherits FetchDataModel
2 @page "/fetchdata"
3
4 <h1>Weather forecast</h1>
5
6 <p>This component demonstrates fetching data from a service.</p>
7
8 @if (forecasts == null) {
9     <p><em>Loading...</em></p>
10 }
11 else {
12     <table class="table">
13         <thead>
14             <tr>
15                 <th>Date</th>
16                 <th>Temp. (C)</th>
17                 <th>Temp. (F)</th>
18                 <th>Summary</th>
19             </tr>
20         </thead>
21         <tbody>
22             @foreach (var forecast in forecasts) {
23                 <tr>
24                     <td>@forecast.Date.ToShortDateString()</td>
25                     <td>@forecast.TemperatureC</td>
26                     <td>@forecast.TemperatureF</td>
27                     <td>@forecast.Summary</td>
28                 </tr>
29             }
30         </tbody>
31     </table>
32 }
  
```



- Splitting Razor Components in Blazor



- Blazor Server Side CRUD Application

Tổng kết bài học

- ◎ WebAssembly
- ◎ WebAssembly và C#
- ◎ Blazor server





KẾT THÚC

V n