iscte
INSTITUTO
UNIVERSITÁRIO
DE LISBOA

emprego
digital

# ASP.NET Core Blazor

A .NET SPA framework

Paulo Vieira 2020

UPskill

# Client-Side Execution

- Most web browsers can run code written in the JavaScript language
  - Because JavaScript is local to the browser, it can respond very quickly to user actions such as clicking, pointing, or dragging
- Many JavaScript libraries are available that help accelerate client code development
  - jQuery library makes it simple to access page elements and manipulate them by changing their style or content

# Single Page Applications (SPA)

- ASP.NET applications can also use the Asynchronous JavaScript and XML (AJAX) technology on the client computer to interact with the web server.
  - You can use AJAX to update a small section of an HTML page, instead of reloading the entire page from the server
  - Such partial page updates help make web pages more responsive and engaging
- One of the most popular types of AJAX-based web application is Single Page Applications (SPA)
  - Angular framework and the React library

# What is Blazor for?

- Build client web apps with C# and Microsoft. NET
- Build front-end and back-end logic for web apps with common languages, frameworks, and tools
- Using the same language for front-end and back-end code can:
- Accelerate app development.
- Reduce the complexity of the build pipeline.
- Simplify maintenance.
- Let developers understand and work on both client-side and server-side code.

https://docs.microsoft.com/en-gb/learn/modules/build-blazor-webassembly-visual-studio-code/

# What is Blazor?

- Blazor apps are composed of reusable web UI components built using C#, HTML, and CSS
- Using C# for all code simplifies sharing data between the front end and back end, code reuse to accelerate development, and maintenance.

# Blazor = Browser + Razor

- Razor:
  - Razor is a markup syntax that uses HTML and C# for writing UI components of Blazor web apps.
  - Can be used to dynamically generate HTML
  - Razor is based on ASP.NET and designed for creating web apps.
  - In ASP.NET Core MVC it is executed at the server-side to generate HTML which is sent to the browser

# Blazor = Browser + Razor

- Razor:
  - Razor is a markup syntax that uses HTML and C# for writing UI components of Blazor web apps.
  - Can be used to dynamically generate HTML
  - Razor is based on ASP.NET and designed for creating web apps.
  - In ASP.NET Core MVC it is executed at the server-side to generate HTML which is sent to the browser
- Razor directives are component markup used to add C# inline with HTML. With directives, developers can define single statements, methods, or larger code blocks.

# What is WebAssembly?

- WebAssembly (WASM) is an open binary standard.
- WebAssembly is a textual assembly language with a compact binary format for fast downloads and near-native performance.
- WebAssembly provides a compilation target for languages such as C, C++, and Rust.
- It's designed to run alongside JavaScript so that both work together.
- WebAssembly also can generate progressive web applications to be downloaded and run offline.

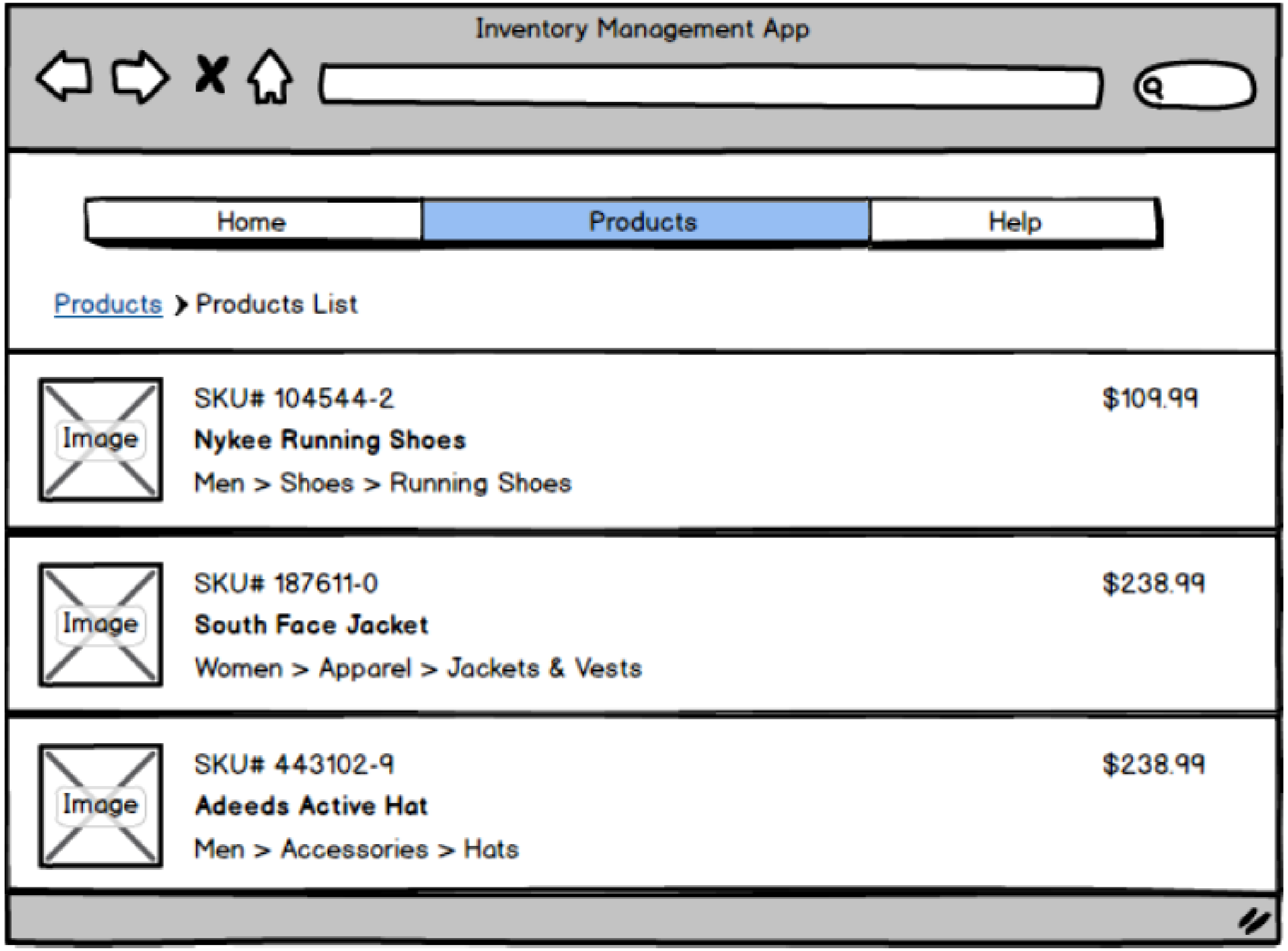
# What is Blazor WebAssembly?

- With Blazor WebAssembly, developers can run .NET code in a browser.
- It's a single-page app framework and uses the WebAssembly open standards without requiring plug-ins or code generation.
- Blazor uses a .NET runtime compiled to a WebAssembly module that is downloaded with an app. The module can execute .NET Standard code included in a Blazor app.
- A Blazor WebAssembly app is restricted to the capabilities of the browser that executes the app. But the app can access full browser functionality via JavaScript interop.

# Blazor WebAssembly supported browsers

- Blazor WebAssembly requires a modern desktop or mobile browser. The following browsers are currently supported:
  - Microsoft Edge
  - Mozilla Firefox
  - Google Chrome
  - Apple Safari

# Example – Inventory Management App



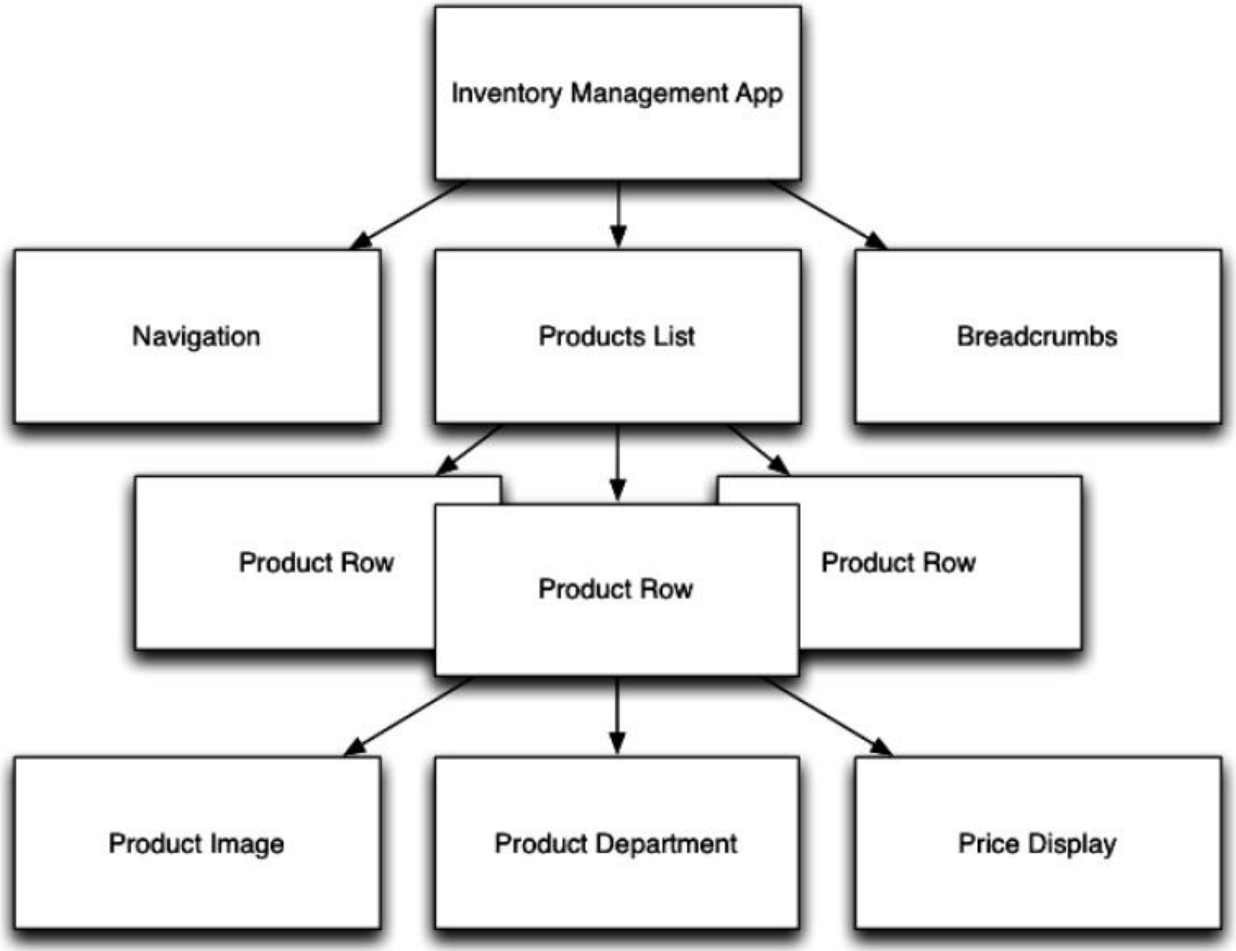
Products ❯ Products List

SKU# 104544-2 $109.99
**Nykee Running Shoes**
Men > Shoes > Running Shoes

SKU# 187611-0 $238.99
**South Face Jacket**
Women > Apparel > Jackets & Vests

SKU# 443102-9 $238.99
**Adeeds Active Hat**
Men > Accessories > Hats

# Example – Inventory Management App



Inventory Management App

Navigation

Products List

Breadcrumbs

Product Row

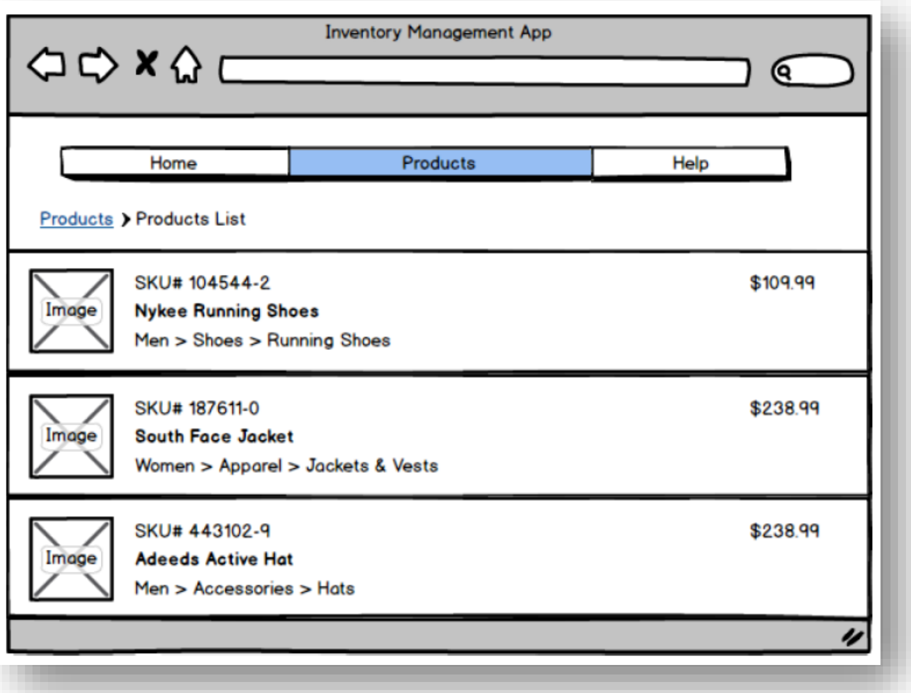
Product Row

Product Row

Product Image

Product Department

Price Display



Inventory Management App

Products › Products List

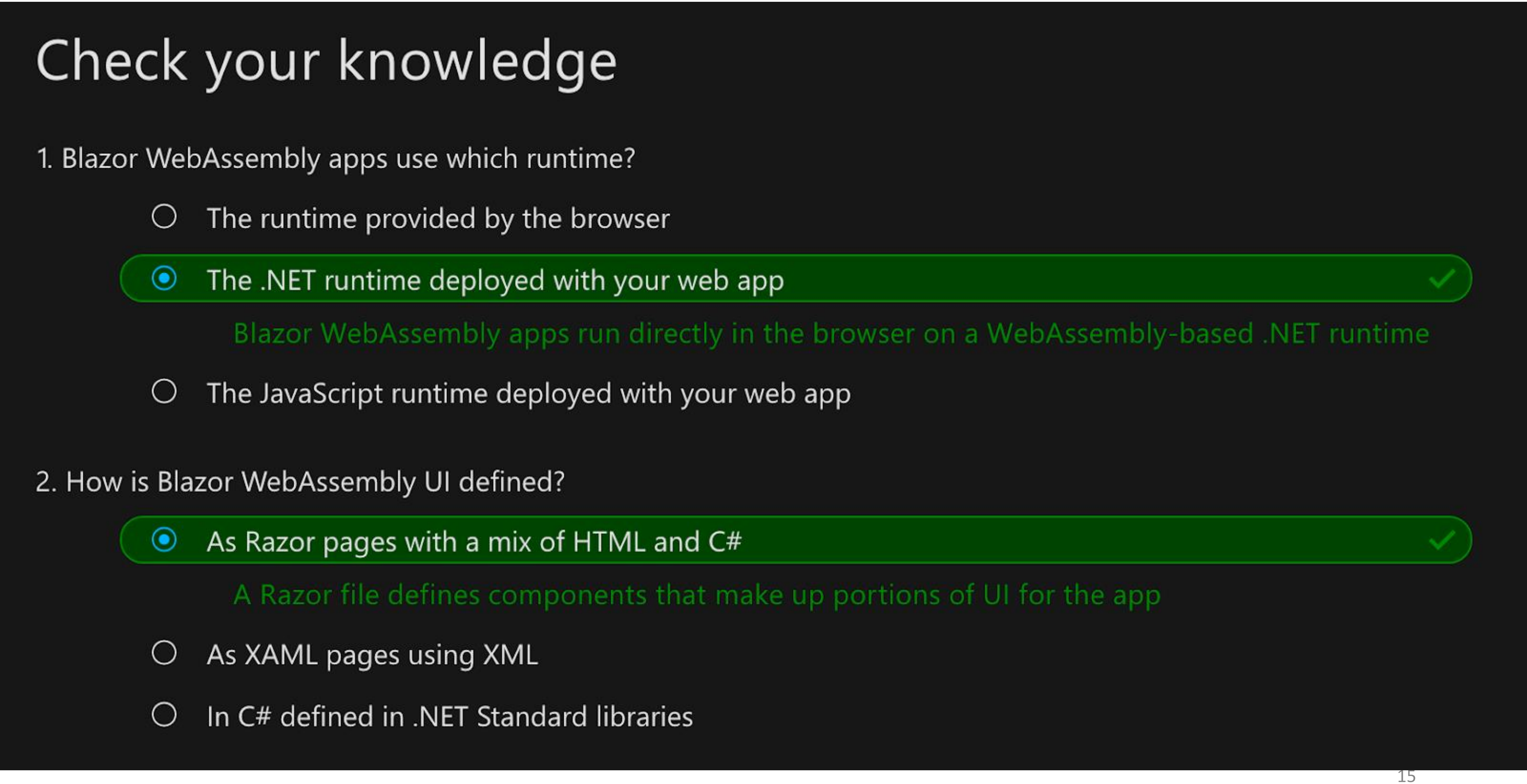
| | | |
|---|---|---|
| Image | SKU# 104544-2<br>**Nykee Running Shoes**<br>Men > Shoes > Running Shoes | $109.99 |
| Image | SKU# 187611-0<br>**South Face Jacket**<br>Women > Apparel > Jackets & Vests | $238.99 |
| Image | SKU# 443102-9<br>**Adeeds Active Hat**<br>Men > Accessories > Hats | $238.99 |

# What are Razor components?

- Self-contained chunk of user interface
- Classes built from razor and C# with one specific purpose – easier to understand, debug, and maintain
- Contain markup and C# code in @code section
- Should be reusable
    - use a component library instead of copy-pasting
- Have a life cycle just like any other .NET object

# What are Razor components?

- A Razor file defines components that make up a portion of the app UI. Components in Blazor are analogous to user controls in ASP.NET Web Forms.
- If you explore the project, you'll see that most files are .razor files.
- At compile time, each Razor component is built into a .NET class. The class includes common UI elements like state, rendering logic, lifecycle methods, and event handlers.



# Check your knowledge

1. Blazor WebAssembly apps use which runtime?

○ The runtime provided by the browser

● The .NET runtime deployed with your web app ✓

Blazor WebAssembly apps run directly in the browser on a WebAssembly-based .NET runtime

○ The JavaScript runtime deployed with your web app

2. How is Blazor WebAssembly UI defined?

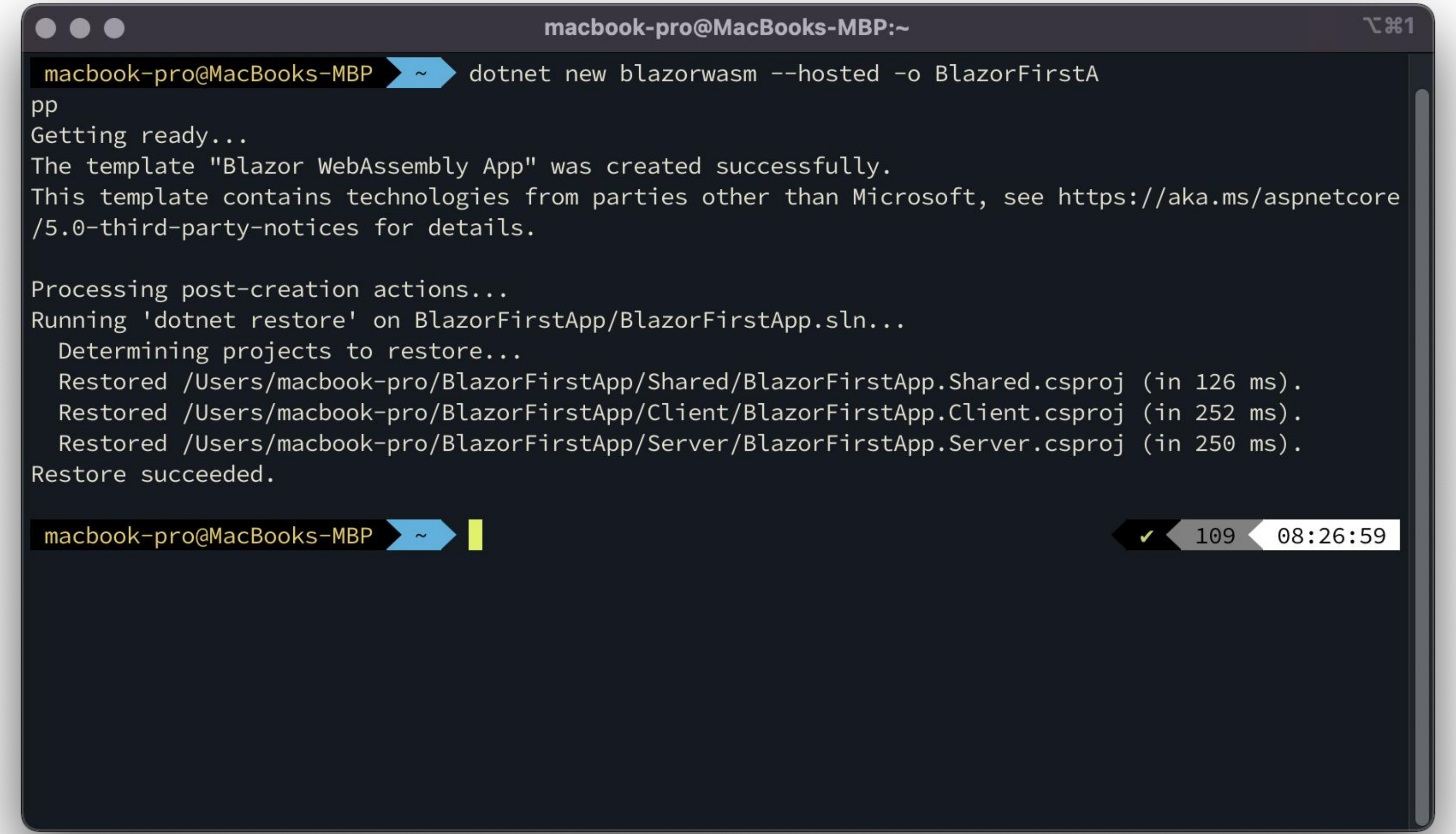
● As Razor pages with a mix of HTML and C# ✓

A Razor file defines components that make up portions of UI for the app

○ As XAML pages using XML

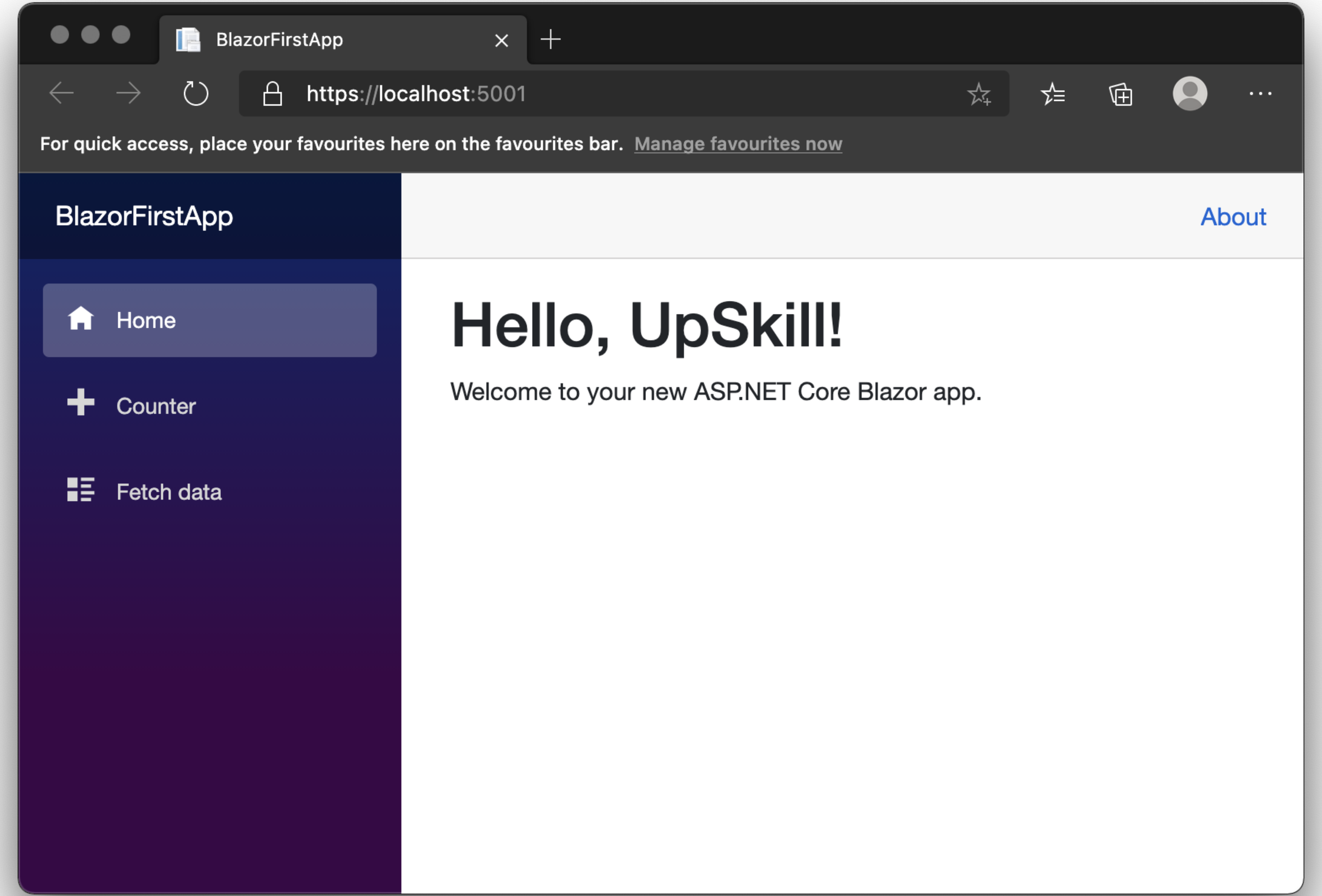
○ In C# defined in .NET Standard libraries

# Generating a project with *dotnet cli*



```
macbook-pro@MacBooks-MBP  ~  dotnet new blazorwasm --hosted -o BlazorFirstA
pp
Getting ready...
The template "Blazor WebAssembly App" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore
/5.0-third-party-notices for details.

Processing post-creation actions...
Running 'dotnet restore' on BlazorFirstApp/BlazorFirstApp.sln...
  Determining projects to restore...
  Restored /Users/macbook-pro/BlazorFirstApp/Shared/BlazorFirstApp.Shared.csproj (in 126 ms).
  Restored /Users/macbook-pro/BlazorFirstApp/Client/BlazorFirstApp.Client.csproj (in 252 ms).
  Restored /Users/macbook-pro/BlazorFirstApp/Server/BlazorFirstApp.Server.csproj (in 250 ms).
Restore succeeded.

macbook-pro@MacBooks-MBP  ~                                                      ✔  109  08:26:59
```

dotnet new blazorwasm –hosted –o BlazorFirstApp



BlazorFirstApp
https://localhost:5001
For quick access, place your favourites here on the favourites bar. Manage favourites now
BlazorFirstApp
About
Home
Counter
Fetch data

# Hello, UpSkill!

Welcome to your new ASP.NET Core Blazor app.

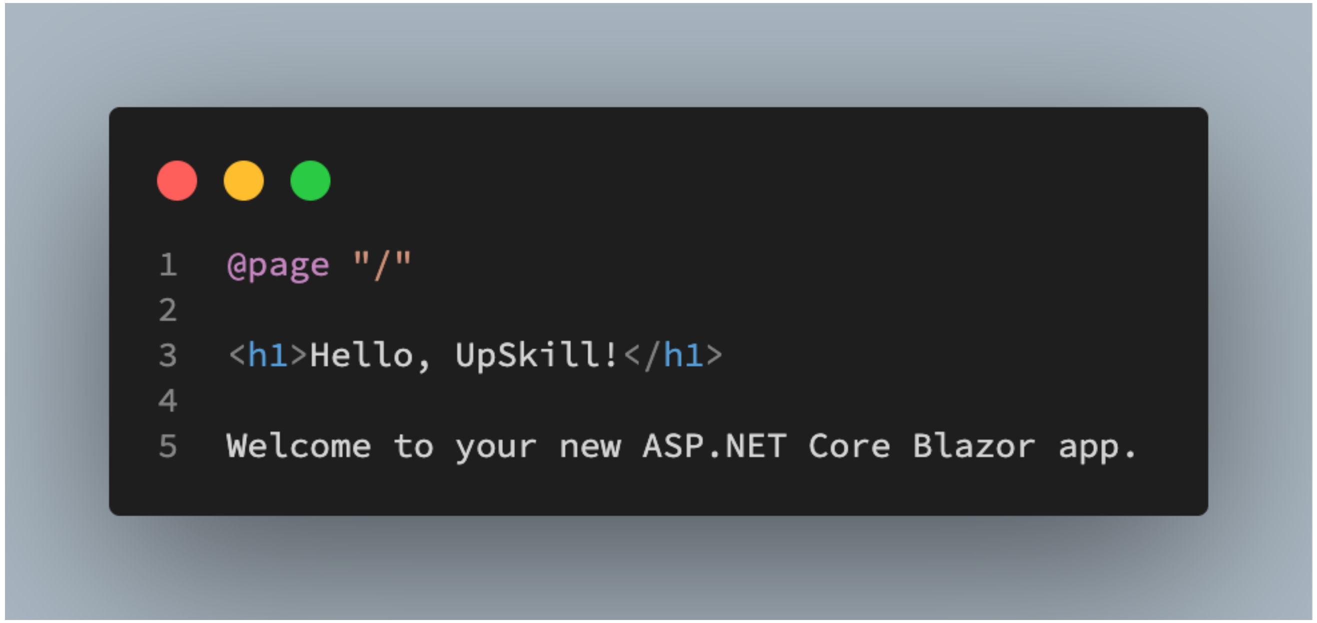
# Blazor WebAssembly project file

- Blazor WebAssembly projects target Microsoft .NET Standard, which is currently version 2.0. Blazor WebAssembly apps are different from Blazor Server apps, which are .NET Core projects.
- Blazor WebAssembly apps target .NET Standard because it runs directly in a browser on a WebAssembly-based .NET runtime that uses Mono. You can't install .NET directly into a browser.

# Blazor WebAssembly entry point

- The entry point for the app is defined in a C# file named Program.cs. When the Program class is instantiated and runs, its Main method is called.
- By default, the Main method configures and creates the .NET WebAssembly object. This object communicates with the WebAssembly host and loads the app's Razor components.

# Client > Pages > Index.razor



```
@page "/"

<h1>Hello, UpSkill!</h1>

Welcome to your new ASP.NET Core Blazor app.
```

# Page directive

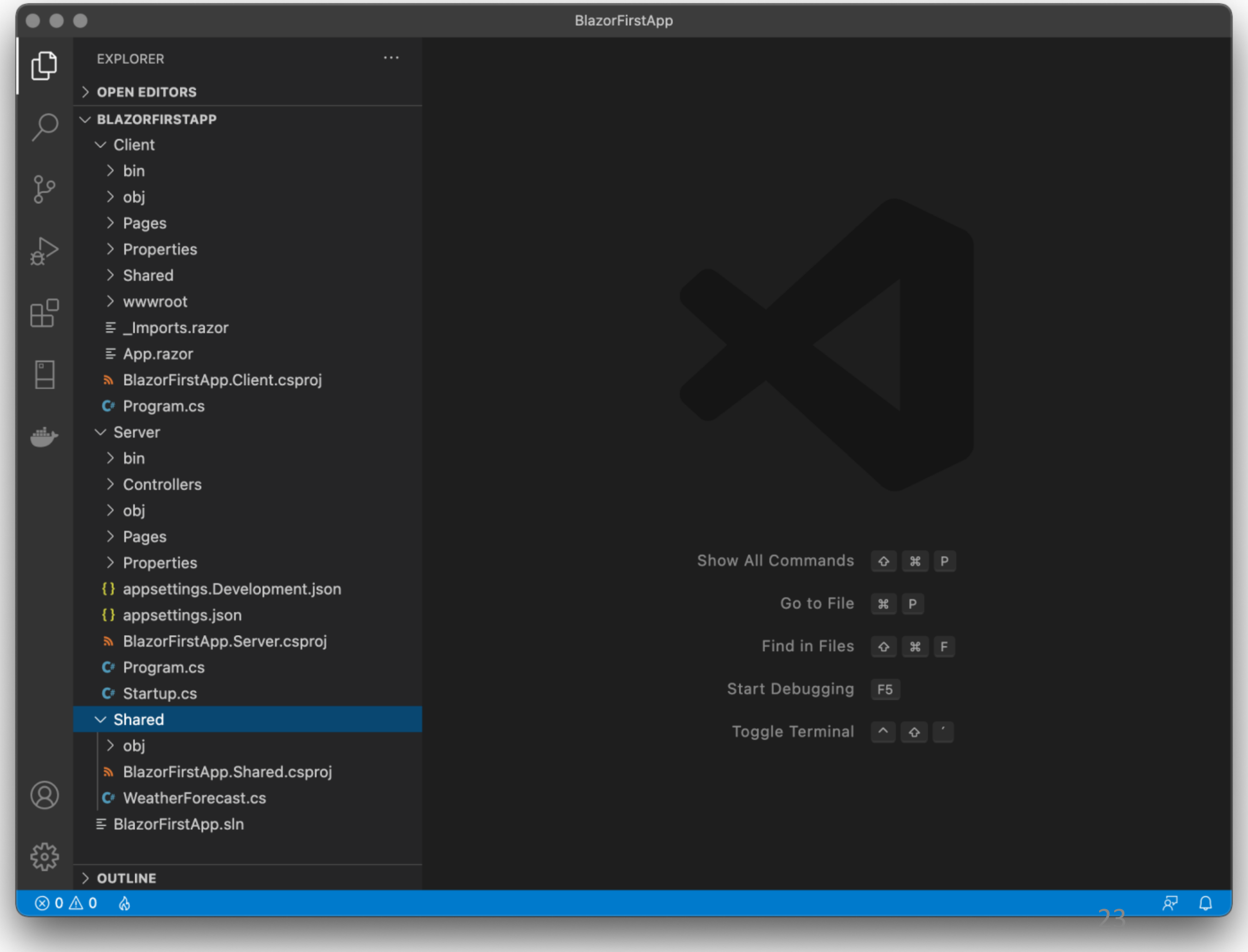
- The @Page directive is special markup that identifies a component as a page. Use this directive to specify a route. The route maps to an attribute route that the Blazor engine recognizes to register and access the page.

# Code directives

- Code directives should be familiar to developers who have used Razor in MVC or Pages.
- You can use @expression() to add a C# statement inline with HTML. If you require more code, use the @code directive to add multiple statements enclosed by parentheses.
- You can also add an @code section to the template for methods and properties. They're added to the top of the generated class, where the document can reference them.

# Examining the Project's Parts

- Single solution with three projects:
    - Server
    - Shared
    - Client

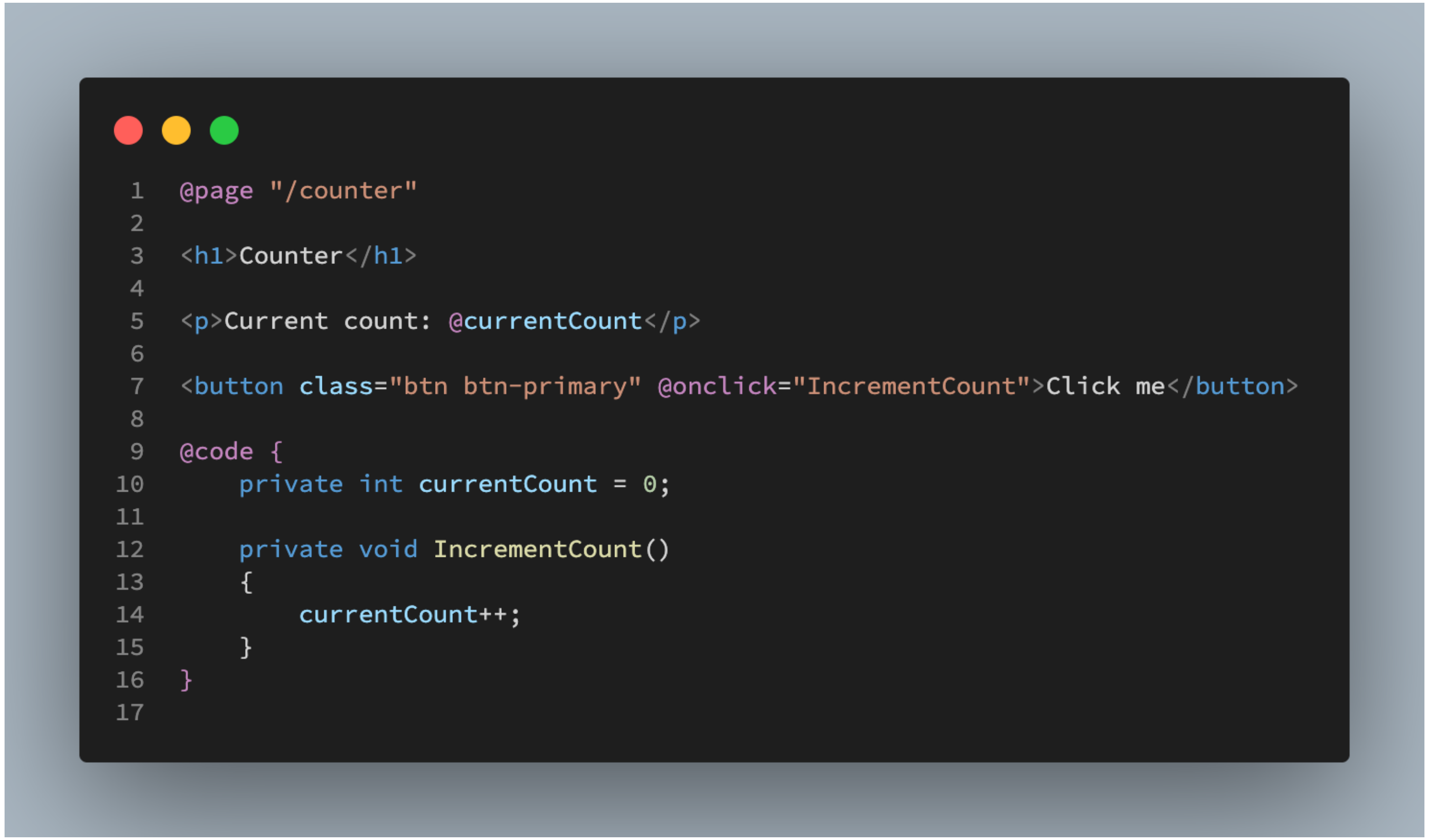


BlazorFirstApp

EXPLORER

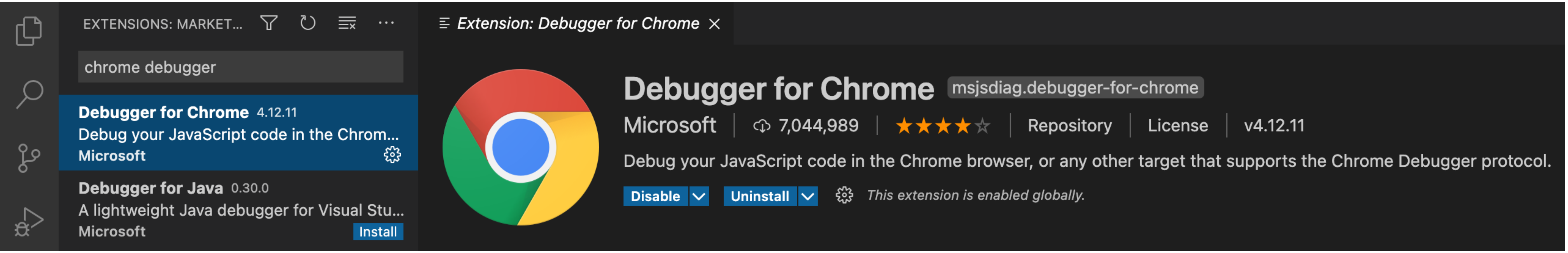
- OPEN EDITORS
- BLAZORFIRSTAPP
    - Client
        - bin
        - obj
        - Pages
        - Properties
        - Shared
        - wwwroot
        - _Imports.razor
        - App.razor
        - BlazorFirstApp.Client.csproj
        - Program.cs
    - Server
        - bin
        - Controllers
        - obj
        - Pages
        - Properties
        - appsettings.Development.json
        - appsettings.json
        - BlazorFirstApp.Server.csproj
        - Program.cs
        - Startup.cs
    - Shared
        - obj
        - BlazorFirstApp.Shared.csproj
        - WeatherForecast.cs
    - BlazorFirstApp.sln
- OUTLINE

# Where is my client code?



```
@page "/counter"

<h1>Counter</h1>

<p>Current count: @currentCount</p>

<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>

@code {
    private int currentCount = 0;

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

```

# How can I debug?



chrome debugger

**Debugger for Chrome** 4.12.11
Debug your JavaScript code in the Chrom...
Microsoft

**Debugger for Java** 0.30.0
A lightweight Java debugger for Visual Stu...
Microsoft

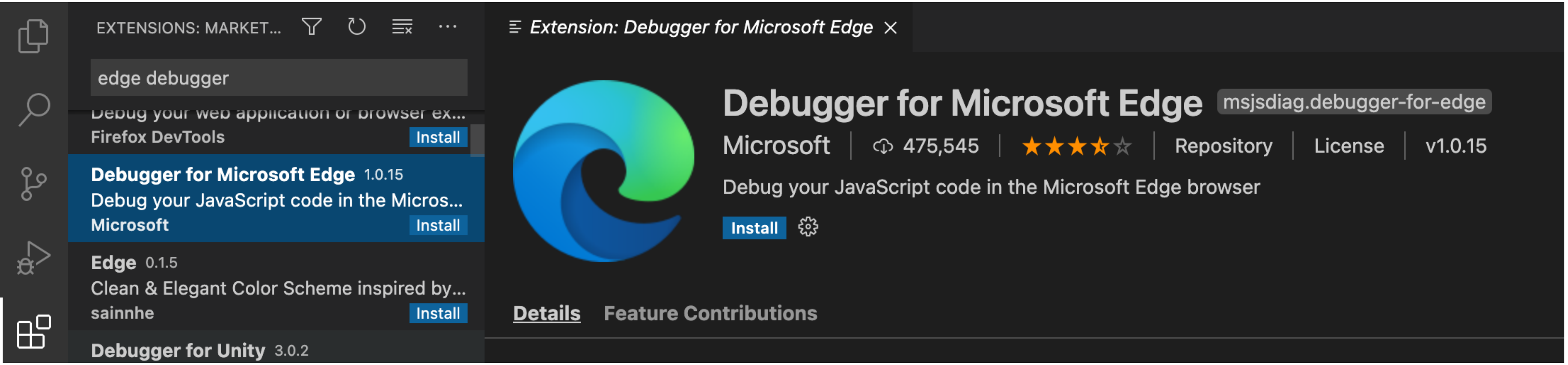
*Extension: Debugger for Chrome*

**Debugger for Chrome** msjsdiag.debugger-for-chrome

Microsoft | 7,044,989 | ★★★★☆ | Repository | License | v4.12.11

Debug your JavaScript code in the Chrome browser, or any other target that supports the Chrome Debugger protocol.

*This extension is enabled globally.*



edge debugger

Debug your web application or browser ex...
Firefox DevTools

**Debugger for Microsoft Edge** 1.0.15
Debug your JavaScript code in the Micros...
Microsoft

**Edge** 0.1.5
Clean & Elegant Color Scheme inspired by...
sainnhe

**Debugger for Unity** 3.0.2

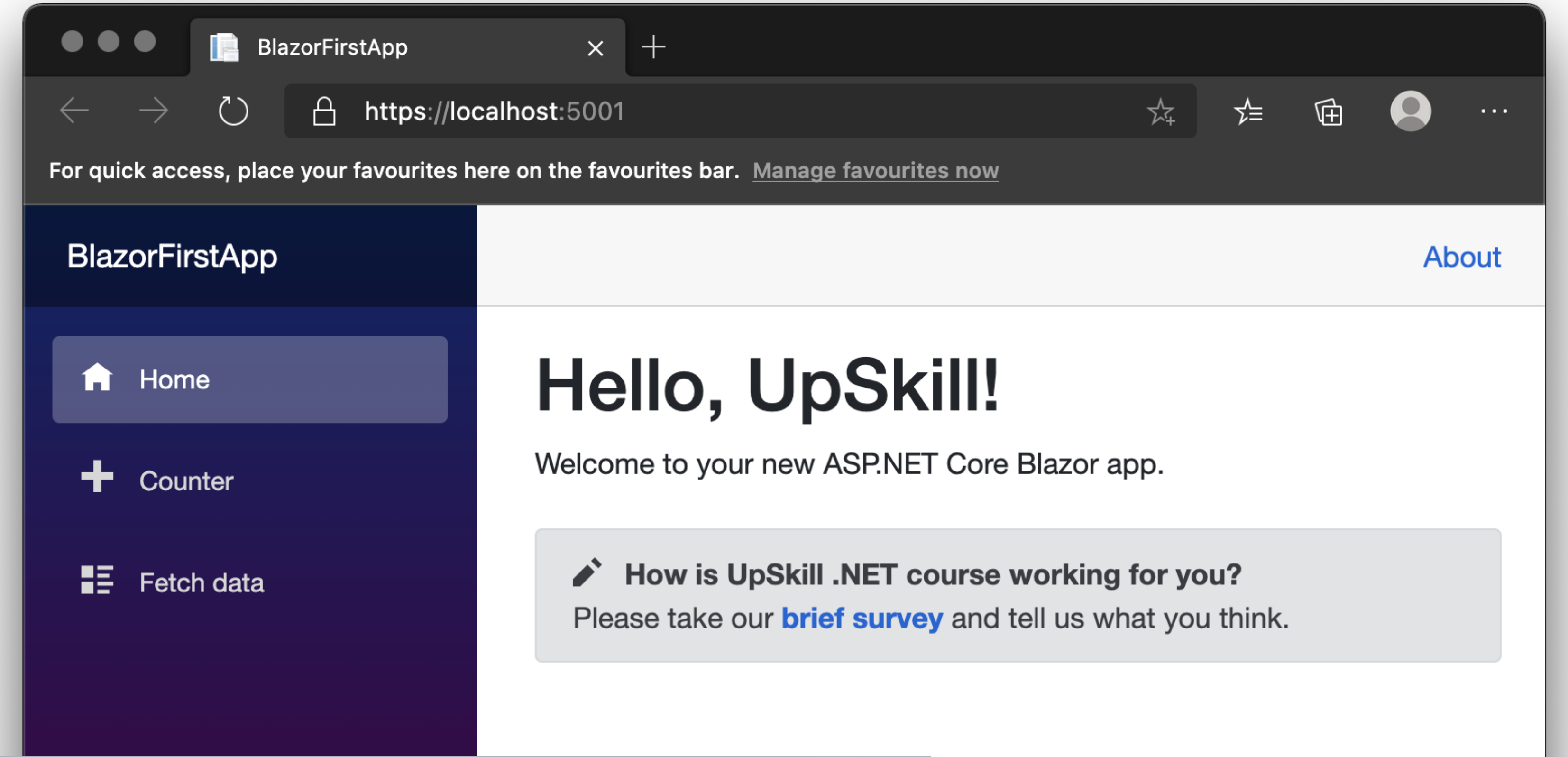
*Extension: Debugger for Microsoft Edge*

**Debugger for Microsoft Edge** msjsdiag.debugger-for-edge

Microsoft | 475,545 | ★★★★☆ | Repository | License | v1.0.15

Debug your JavaScript code in the Microsoft Edge browser

# Components



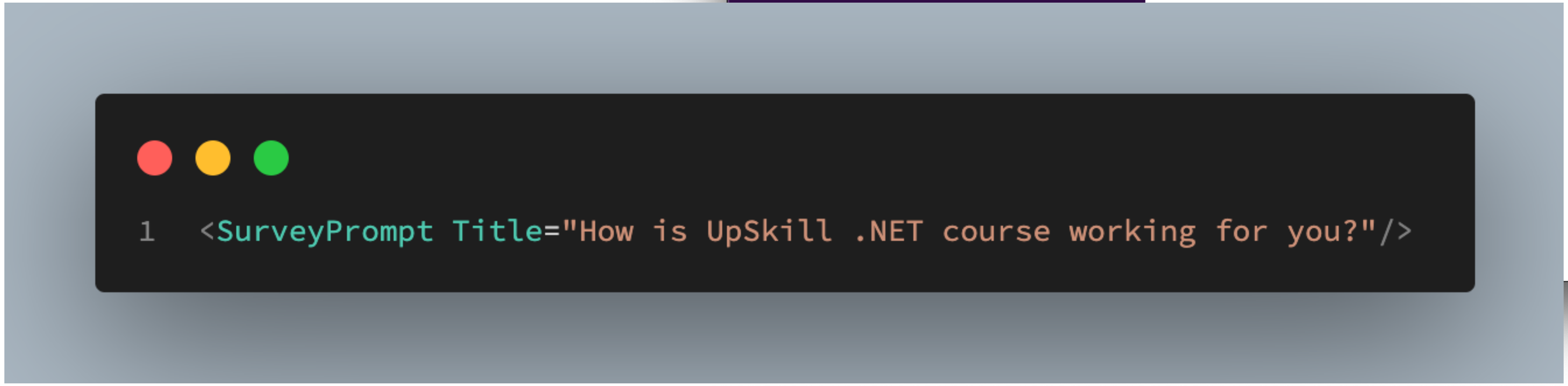
BlazorFirstApp
https://localhost:5001
For quick access, place your favourites here on the favourites bar. Manage favourites now
BlazorFirstApp
Home
Counter
Fetch data
About

## Hello, UpSkill!

Welcome to your new ASP.NET Core Blazor app.

**How is UpSkill .NET course working for you?**

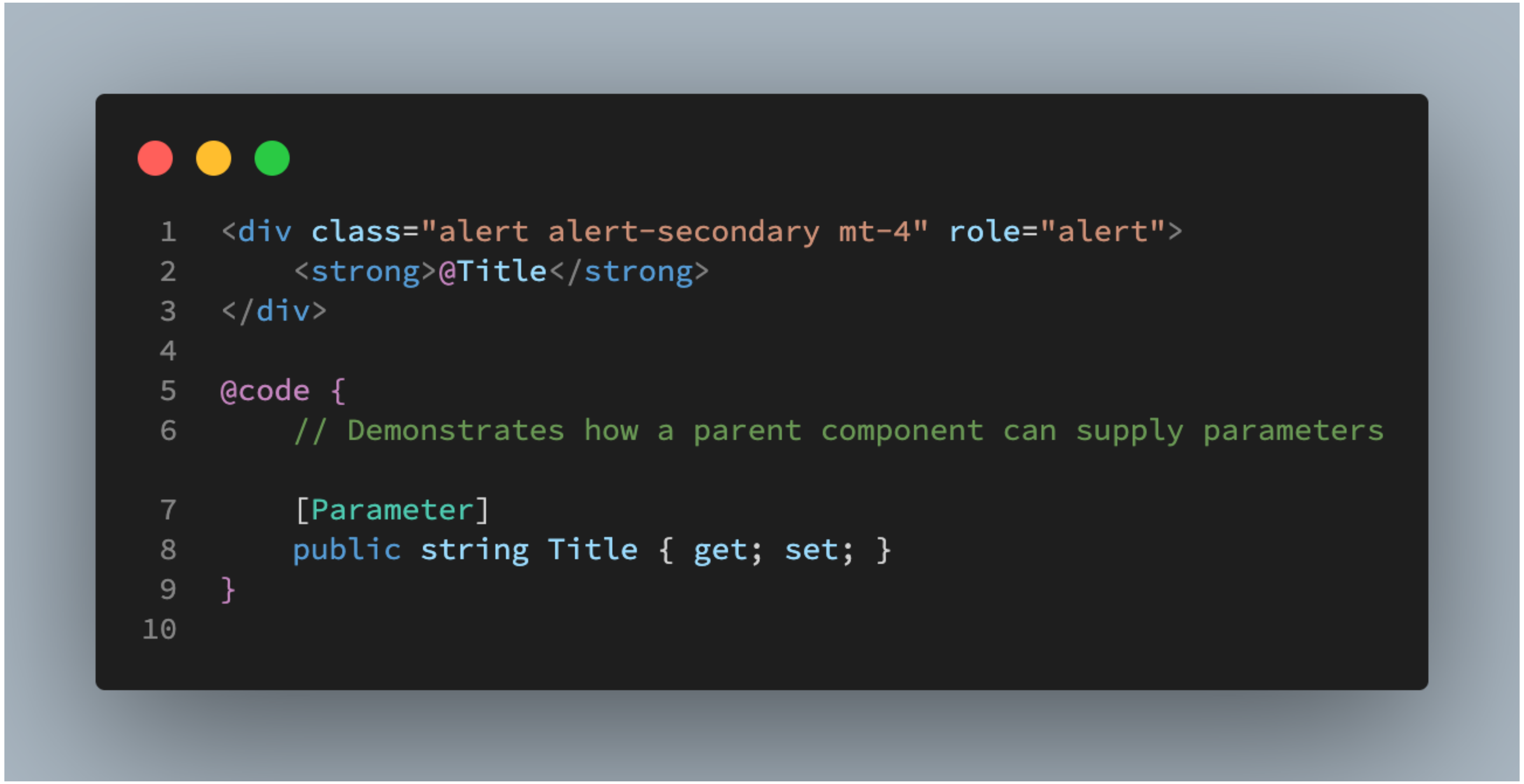
Please take our **brief survey** and tell us what you think.



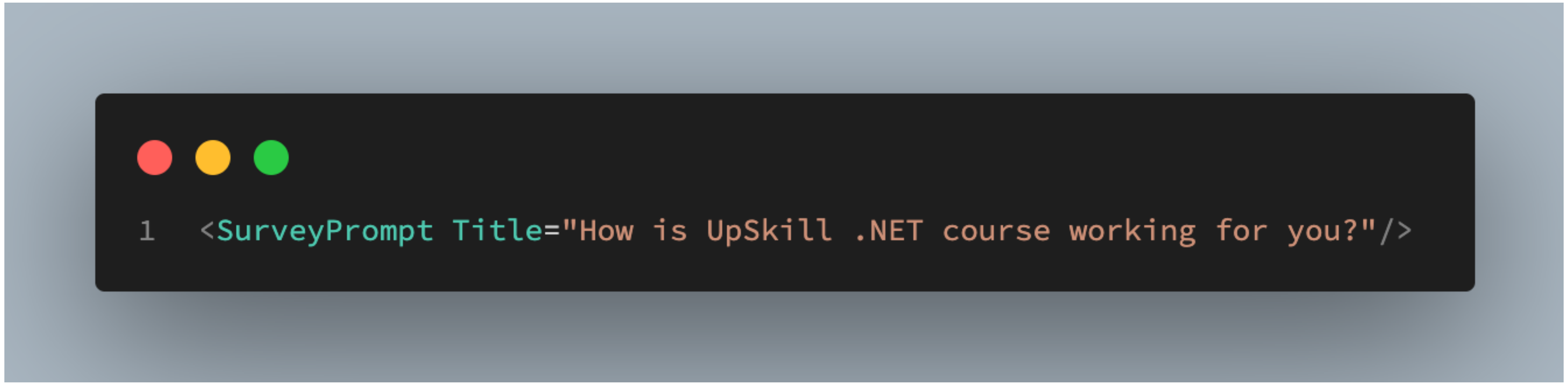
```
<SurveyPrompt Title="How is UpSkill .NET course working for you?"/>
```

# Razor data binding

- Within Razor components, you can data bind HTML elements to C# fields, properties, and Razor expression values. Data binding allows two-way synchronization between HTML and Microsoft .NET.
- Data is pushed from HTML to .NET when a component is rendered. Components render themselves after event-handler code executes. That's why property updates are reflected in the UI immediately after an event handler is triggered.
- Use @bind markup to bind a C# variable to an HTML object. You define the C# variable by name as a string in the HTML. You'll see an example of data binding in the following exercise.



```
<div class="alert alert-secondary mt-4" role="alert">
    <strong>@Title</strong>
</div>

@code {
    // Demonstrates how a parent component can supply parameters

    [Parameter]
    public string Title { get; set; }
}

```



```
<SurveyPrompt Title="How is UpSkill .NET course working for you?"/>
```

# References

- Himschoot, P., “**Microsoft Blazor: Building Web Applications in .NET**”, 2020, Apress, ISBN: 978-1-4842-5927-6
- Sainty, C., **“Blazor in Action”**, October 202 MEAP Version, Manning ISBN 9781617298646
- [Blazor documentation]()

# Further information

- https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor
- https://docs.microsoft.com/en-us/aspnet/core/blazor/host-and-deploy/webassembly?view=aspnetcore-5.0
- https://docs.microsoft.com/en-us/aspnet/core/blazor/progressive-web-app?view=aspnetcore-5.0&tabs=visual-studio

# Practice…

- https://docs.microsoft.com/en-us/aspnet/core/tutorials/build-a-blazor-app?view=aspnetcore-5.0

- https://github.com/dotnet-presentations/blazor-workshop/

*Thank you*

# O futuro profissional começa aqui

iscte
INSTITUTO
UNIVERSITÁRIO
DE LISBOA

emprego
digital

UPskill