



To discuss this course and customizations:  
Call: 434-509-5680 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

## **Build .NET 10 Web Apps with Blazor**

### **Duration**

3 days

### **Target Audience**

Students need HTML, CSS, JavaScript, and C# programming experience.

### **Description**

This course offers a detailed journey into the ASP.NET Core Blazor framework, emphasizing the latest enhancements introduced in .NET 10. Participants will gain proficiency in building sophisticated, composable user interfaces using Razor Components, leveraging the power of Razor syntax for effective data binding and event handling. The curriculum includes hands-on experience with the new QuickGrid component, enabling efficient data display and manipulation. Additionally, learners will explore the intricacies of configuring page routing in Blazor applications and the seamless integration of client-side JavaScript, enriching the functionality and interactivity of their web applications. The course culminates with practical guidance on deploying Blazor applications to production, ensuring participants are equipped with the skills necessary for real-world application development.

### **Objectives**

- Understand the ASP.NET Core Blazor platform including new .NET 10 features
- Build Composable UIs with Razor Components
- Utilize Razor Syntax to Perform Data Binding and Event Handling
- Employ the QuickGrid Component with new .NET 10 enhancements
- Configure Page Routing with Blazor
- Integrate Client-Side JavaScript with Blazor
- Implement Authentication with WebAuthn/FIDO2 Passkeys
- Deploy a Blazor application to production



To discuss this course and customizations:  
Call: 434-509-5680 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

## Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

## Software Requirements

Students will need a free, personal GitHub account to access the courseware. Student will need permission to install Visual Studio 2026 on their computers. Also, students will need permission to install NuGet Packages. If students are unable to configure a local environment, a cloud-based environment can be provided.

## Training Topics

### Getting Started with Blazor

- What is Blazor?
- Comparison to Other Web Frameworks
- Render Modes
- Server-Side & WebAssembly
- What's new for Blazor in .NET 10
  - Persistent State Attribute for Simplified State Management
  - WebAuthn/FIDO2 Passkey Authentication
  - NavigationManager.NotFound() for 404 Handling
  - ReconnectModal Component
  - Blazor.pause()/Blazor.resume() for State Persistence
  - blazor.web.js Size Reduction (76% smaller)

### Running Blazor Code

- Single Page Applications
- What is WebAssembly?
- Browser Compatibility
- WebAssembly vs. JavaScript
- How does .NET Core / C# run in a web browser?

### Blazor Application

- Project Template
- Create a New Application



To discuss this course and customizations:  
Call: 434-509-5680 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

- Hosting Blazor with a ASP.NET Core MVC Server
- Configuration
- Dependency Injection
- Environments
- Logging
- Handling Errors
- Debugging WebAssembly in the Browser

### Razor Components and Data Binding

- What is a Component?
- Creating a Data Model
- Binding the Data Model to the HTML
- Passing Arbitrary Attributes
- Handling Events
- Manually Trigger State Updates and Re-rendering

### Rendering

- Render Modes
- Static Server Rendering
- Interactive Server Rendering
- Interactive WebAssembly Rendering
- Interactive Auto Rendering
- AOT vs. JIT Compilation
- Output Caching
- Streaming Rendering
- Render Modes and Consuming Data
- Render Modes and JavaScript

### Composing Razor Components

- Decompose a Component into Smaller Components
- One-Way Data Binding
- Two-Way Data Binding
- Pass Data from a Parent Component to a Child Component using Parameters
- Pass Data from a Child Component to a Parent Component using Event Callbacks
- Use Refs to Access DOM Elements



To discuss this course and customizations:  
Call: 434-509-5680 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

- Razor Component Libraries
- Razor Component Design Patterns
- Parameters are Immutable
- Required Component Parameters
- Lift State Up
- Managing State in General
- Persistent Component State
- Persistent State Attribute
- Section Components
- QuickGrid Component
  - RowClass Parameter for Conditional Row Styling
  - HideColumnOptionsAsync() Method

### Razor Component Forms

- What is the purpose of Form?
- Collecting Data using a Form, Input, Select, and TextArea Elements
- Explore Form Element Two-Data Binding
- Build Forms with the Blazor Edit Form Razor Component
- Explore the Concept of the Edit Context
- Specialized Edit Form Controls
- Applying Validation to the Form
- Decorating the View Model with Validation Attributes
- Code Custom Validation Attributes
- Nested Form Validation with AddValidation()
- Validatable Type Attribute
- Antiforgery Tokens

### Razor Component Pages

- What is the Page model?
- Differences between Razor Pages and Razor Components
- Using a Razor Component as a Page
- Explore the Router Component
- Configuring Page Routing
- Route to Components from Multiple Assemblies
- Using Route Parameters
- Using the Query String
- Enhanced Navigation for Static Rendered Pages



To discuss this course and customizations:  
Call: 434-509-5680 or Email: [sales@cloudcontraptions.com](mailto:sales@cloudcontraptions.com)

- `NavigationManager.NotFound()` Method
- NavLink Query String and Fragment Handling

### [Authentication, Authorization, and Auditing](#)

- What is Authentication?
- What is Authorization?
- What is Auditing?
- Authentication Models for Blazor
- WebAuthn/FIDO2 Passkey Authentication
- ASP.NET Core Identity Passkey Support
- Component Attribute-based Authorization
- Authorization Components
- Authorization with C# Code
- Roles and Claims
- Blazor Identity UI