

# Build web applications with ASP.NET Blazor

## Module 1: Blazor (ASP.NET Core 6.0) Overview

- What is Blazor?
- Configuring Core 6.0
- Discuss the rapid development and evolution of ASP.NET
- Components
- Blazor Server
- Blazor WebAssembly
- Blazor Hybrid
- JavaScript interop
- Demo: Creating your first Blazor Web Application

## Module 2: Introduction to Blazor Application

- Overview of Blazor
- Blazor and Razor Components
- Installing Blazor Templates
- Debugging Blazor Applications
- Supported Platforms | Browser Compatibility.
- Tooling configuration

## Module 3: Blazor Project Structure

- Describe all files and directories and their purpose.
- Compare the structure of the server and web assembly.
- Demo: Build a Blazor Todo List app
- Demo: Use ASP.NET Core SignalR with Blazor

## Module 4: ASP.NET Core Blazor fundamentals

- ASP.NET Core Blazor routing and navigation
- ASP.NET Core Blazor dependency injection
- ASP.NET Core Blazor logging
- ASP.NET Core Blazor static files
- Demo: Understanding fundamentals using a sample app.

## Module 5: Blazor Razor Components and Data Binding

- Blazor Razor components
- Blazor Split components
- Blazor Model Classes
- Create Employee List Component
- Blazor Loading Indicator
- Blazor data access strategies
- Binding data with Blazor
- Blazor event handling
- Blazor data binding
- Blazor two-way binding
- Component Parameters
- Blazor EventCallback
- ASP.NET Core Razor component lifecycle
- ASP.NET Core Blazor layouts
- Demo: Understanding components and data binding.

### **Module 6: Blazor Component Advance Configuration**

- AutoMapper in ASP.NET Core
- Edit and Update
- Create Operation
- Force reload the component
- Delete Operation
- Component Communication
- Delete Confirmation
- Attribute Splatting
- Arbitrary Attributes
- Cascading Values
- Multiple Cascading Parameters
- Cascading Values Performance
- Demo: Implementing advanced configuration.

### **Module 7: Razor Component Forms**

- What is the purpose of Form?
- Building blazor form
- Blazor Select List
- Custom Input Component

- RCL Project
- Bind Enum to Select
- Blazor DatePicker
- Blazor Form Validation
- Custom Form Validation
- Compare Validation
- Complex Model Validation
- Demo: Creating Blazor form app.

### **Module 8: ASP.NET Core Blazor state management**

- Persist state across circuits
- Where to persist state
- Server-side storage
- URL
- Browser storage
- ASP.NET Core Protected Browser Storage
- In-memory state container service
- Demo: Understanding blazor state management.

### **Module 9: ASP.NET Core Blazor Server with Entity Framework Core (EF Core)**

- Database access
- New DbContext instances
- Scope to the component lifetime
- Enable sensitive data logging
- Executing Raw Sql Queries/ Procedures from EF Core.
- Demo: Accessing database and Executing SQL queries from EF Core

### **Module 10: Security management.**

- Identity Setup in Blazor
- Cookie Authentication
- Authentication - Login Page
- Authentication - Logout Page
- Authorization in Blazor

- Blazor Authorize Attribute
- Blazor Authorization in Code
- Demo: Implementing authentication in Blazor.

### **Module 11: Using RESTful APIs**

- What are RESTful APIs
- Create REST API
- REST API DbContext
- Repository Pattern in REST API
- REST API Get List
- REST API Get By Id
- Post in REST API
- REST API Model Validation
- Put in REST API
- Delete in REST API
- Search in REST API
- Call REST API from Blazor
- Blazor Route Parameters
- Data from multiple tables
- Demo: Implementing REST API and perform CRUD operation.

### **Module 12: Interacting with JavaScript**

- What is the JavaScript Interop?
- When is JavaScript needed?
- Synchronous vs. Asynchronous Calls
- How to call a JavaScript function from a Component
- How to call C# code from JavaScript
- Calling Static Methods
- Calling Instance Methods
- Organizing JavaScript Code within a Blazor WebAssembly App
- Explore JavaScript Ecosystem
- Client-Side Libraries
- NPM & Yarn
- Useful Libraries

- Demo: Calling JavaScript functions from Blazor.
- Demo: Calling Blazor C# functions from JavaScript