



# Application Development using .NET Core and Blazor

Course ID #: 7000-446-ZZ-Z

Hours: 35

## Course Content

### Course Description:

Blazor is one of the most exciting web development technologies, that allows for C# to be used both on the server-side and client-side. The client-side C# code is converted to web assembly that is downloaded by the browser. All modern browsers support the web assembly standard, so development under Blazor runs on any browser. The major benefit of Blazor is the elimination of Javascript on the client-side. This leads to higher reliability and also makes more involved user interface development easier. The course covers fundamentals of Blazor including some intermediate level concepts such as components, forms and validation, dependency injection, state management, handling errors and security and Identity implementations. Integrating Blazor with .NET Core web applications is also presented along with some intermediate level .NET concepts.

### Prerequisites:

Some background in web development using ASP.NET or some MVC background. Knowledge of C# language.

### Target Student:

Individuals wanting to gain a good understanding of developing web applications under ASP.NET Core using the new Blazor technology.

### Topics:

#### I. Overview of traditional ASP.Net Core Web Applications

- ASP.Net Core MVC and ASP.Net Core Razor pages architectures
- Developing REST services in ASP.Net Core and consuming these services using traditional Javascript approach

#### II. Blazor Architecture

- Fundamentals behind the Blazor architecture
- Server-side and client-side Blazor
- Introduction to SignalR.

#### III. Client-side Event Handling in Blazor

- Handling DOM events using Blazor.
- Button, keyboard, mouse, drag and drop event handling



# Application Development using .NET Core and Blazor

Course ID #: 7000-446-ZZ-Z

Hours: 35

## IV. Components in Blazor

- Creating medium complexity components
- Integrating components in Razor pages
- Passing parameters to components, data binding, event handling, using lambda expressions with events, routing and route parameters for components, conditional HTML element attributes, templated components

## V. Blazor Forms and Validation

- Data annotations in Blazor
- Using input components to provide default behavior for validating on edit and changing their CSS class to reflect the field state

## VI. Component Libraries

- Creating Razor class library (RCL) to share components across projects
- Consuming a library component

## VII. Blazor Layouts

- Creating custom layouts
- Setting layout for a component
- Nested layouts

## VIII. Security and Identity in Blazor

- Authentication and Authorization in Blazor
- Server authentication and web assembly authentication
- Authentication State Provider service
- Implementing a custom authentication state provider
- Controlling authorization
- Authorization view component
- Role based and policy-based authorization

## IX. State Management and Error Handling

- Different approaches to state management in Blazor including client-side and server-side approaches
- Best practices for error handling and logging in Blazor applications