

**Progressive Web Application Development using EF Core and Blazor**

**Course Number:** ASPNC-114  
**Duration:** 3 days

**Overview**

Blazor is a client-side UI framework used to develop rich web UI experiences using HTML, CSS, and C# instead of JavaScript. Blazor is similar to other JavaScript frameworks but uses WebAssembly to execute C# code in the browser so that the front- and back-end of the web application can be written in the same language.

This Progressive Web Application Development using Entity Framework Core and Blazor training course teaches attendees how to create modern Progressive Web Applications (PWAs) using Blazor. Students learn how to use PWA to create an app-like experience in a website that acts and feels like a native desktop/mobile app.

**Prerequisites**

Attendees must have prior development experience with the .NET framework and C#. Basic knowledge of JavaScript is useful but not necessary.

**Materials**

All Blazor training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* Windows 10 or later with at least 8 GB RAM
* Visual Studio 2019 or later
* .NET Core 3.1 or later SDK
* LocalDB or another version of SQL Server
* Postman application
* Additional lab files that Accelebrate provides

**Objectives**

* Install and use the Entity Framework package to work with databases on the server-side
* Install packages required to develop the client-side Blazor applications
* Use components, services, dependency injection, layout, and routing
* Enable PWA features in an application by creating service workers
* Use PWA features, such as push notifications and storage with IndexedDB

**Outline**

* Introduction to Blazor, Entity Framework Core, and Progressive Web Apps
  + What is Microsoft Blazor
  + What is Entity Framework Core
  + What is Progressive Web Apps (PWAs)
  + Relationship between C#, LINQ, Entity Framework Core, JavaScript, HTML, and Razor.
  + Comparing Blazor to Angular and React
* Installing the Packages and Working with Git
  + Blazor
  + Entity Framework Core
  + ODP.NET for Oracle
  + Working with Git on Azure DevOps Services
* Getting Started with Entity Framework Core
  + Entity Framework Core Overview
  + Installing Entity Framework Core
  + Modelling using Code First
  + Modelling using Database First
  + Using Fluent API
* Querying and Saving Data with Entity Framework Core
  + Basic Queries with LINQ
  + Calling Stored Procedures
  + Using Include and Then Include
  + Controlling the Tracking Behavior
  + Saving changes
  + Performing Bulk Insert
  + Implementing Transactions
* Getting Started with Blazor
  + Creating a Blazor Hello World Application
  + Understanding the Razor fundamentals
* Working with Blazor Components
  + Working with Components in Blazor
  + Using C# in Components
  + Parameterize Components
  + Understanding Component Life Cycle
  + Binding - One-way data binding
  + Binding - Two-way data binding
  + Binding - Event binding
* Advanced Blazor Component Concepts
  + Render Raw HTML
  + Render Child Content
  + Using RenderFragment
  + Using RenderTreeBuilder
  + Using @key and @attribute
* Layout and Routing
  + Creating a Master Layout
  + Implementing Routing
  + Using URL Helpers
* Forms and Validation
  + Using EditForm
  + Implementing Form Validation
  + Using @ref
  + Routing
* Dependency Injection & JSInterop
  + Understanding Dependency Injection (DI)
  + DI with Default Service
  + DI with Custom Service
  + Call a JavaScript Function
  + Call C# from JavaScript
* Debugging & Deployment
  + Debugging a Blazor Application
  + Deploying a Blazor Application
  + Upgrading a Blazor Application
* Securing a Blazor Application
  + Blazor Authentication
  + Using ASP.NET Core Identity and JWT
* Best Practices
  + Project Structure
  + Optimizing the Startup Time
  + Optimizing Rendering Performance
  + Optimizing the Application Download Size
* Progressive Web Apps (PWAs) Overview
  + Understanding the features of a progressive web app
  + Getting started with a Blazor PWA
  + When to create offline apps
  + Using Service Workers
  + Customizing the application appearance
  + Adding Push Notifications
  + Controlling Caching
* Conclusion