

AWD

BLAZOR TOPICS



INTRODUCTION TO BLAZOR

- Overview of Blazor and its capabilities.
- Differences between Blazor Server and Blazor WebAssembly.
- Advantages of using Blazor over traditional JavaScript frameworks.
- Installing .NET SDK.
- Setting up Visual Studio/Visual Studio Code for Blazor development.
- Creating a new Blazor project.



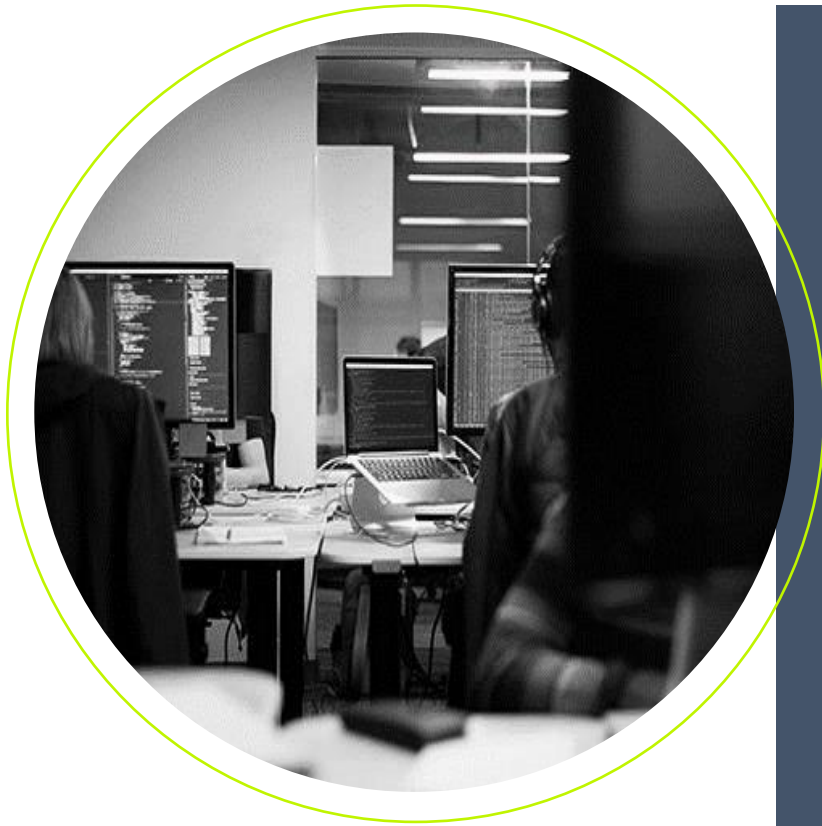
DATA BINDING AND ROUTING

- One-way and two-way data binding.
- Event handling and event callbacks.
- Binding to complex objects and collections.
- Configuring routing in Blazor applications.
- Route parameters and route constraints.
- Nested and layout components.

BLAZOR PROJECT STRUCTURE AND COMPONENTS

- Understanding the project structure and key files.
- Differences between Blazor Server and Blazor WebAssembly projects.
- Creating and using components.
- Component parameters and property binding.
- Lifecycle methods (OnInitialized, OnParametersSet, etc.) and component templates and templated components.





DEPENDENCY INJECTION , FORMS AND VALIDATIONS

- Setting up and using dependency injection in Blazor.
- Registering and injecting services.
- Scoped, transient, and singleton services.
- Building forms with Blazor.
- Data annotations for model validation.
- Handling form submissions and validation messages.



JAVASCRIPT INTEROPERABILITY (JSINTEROP) AND STATE MANAGEMENT

- Calling JavaScript functions from Blazor.
- Invoking .NET methods from JavaScript.
- Passing data between JavaScript and Blazor.
- Managing state within components.
- Application-wide state management.
- Using State Containers and cascading values.
- Understanding the Blazor WebAssembly hosting model.
- Performance considerations and optimizations.
- Working with browser APIs and local storage.
- Working with API calling.
- Understanding the Blazor Server hosting model.
- SignalR and real-time communication.
- Managing connections and scalability considerations.

Other Important Topics

• **Authentication and Authorization**

- Implementing authentication in Blazor applications.
- Role-based and policy-based authorization.
- Protecting routes and components.

• **Error Handling and Diagnostics**

- Handling exceptions and displaying error messages.
- Logging and diagnostics.
- Debugging Blazor applications.

• **Blazor with Entity Framework Core**

- Integrating EF Core with Blazor.
- Performing CRUD operations.
- Asynchronous data access.

• **Testing Blazor Applications**

- Unit testing components and services.
- Integration testing with Blazor.
- Using testing frameworks and tools (e.g., bUnit).

• **Blazor Performance Optimization**

- Optimizing component rendering and re-rendering.

- Lazy loading and pre-rendering components.
- Minimizing JavaScript interop calls.

• **Deployment**

- Publishing Blazor applications.
- Deploying to Azure, Docker, and other hosting environments.
- Configuring CI/CD pipelines for Blazor projects.

• **Localization and Globalization**

- Implementing localization in Blazor applications.
- Using resource files for multi-language support.
- Formatting dates, numbers, and currencies.

• **Advanced Topics**

- Building reusable component libraries.
- Integrating third-party libraries and components.
- Real-time updates with SignalR and Blazor.
- Customizing the Blazor rendering pipeline.

