

C# Full Stack Development with Angular and ASP.NET

Section 1: HTML 5

- Introduction - Web
- What are the different Types of Web Apps overview?
- Introduction to HTML
- Define HTML Structure
- Difference between Tag vs Element
- Difference between Semantic vs Non-semantic elements
- Difference between Block level elements vs Inline elements
- HTML Elements HTML Forms & its Attributes
- HTML Input Elements
- HTML Global Attributes
- HTML Element Specific Attributes

Section 2: CSS & Bootstrap

- What is CSS?
- Understanding the CSS Syntax.
- CSS Selectors
- How To Add CSS in HTML
- CSS Colors & Backgrounds
- CSS Borders, Margins, Padding, Height and Width
- Responsive Web Design – Media queries
- What is Bootstrap?
- Get Start with Bootstrap
- Containers
- Grid System
- Structure of a Bootstrap Grid

- Bootstrap Colors
- Bootstrap Tables
- Bootstrap Jumbotron
- Bootstrap Alerts, Bootstrap Buttons

Section 3: JavaScript

- Introduction – Javascript
- JavaScript Events
- JavaScript Functions
- Inbuilt methods in JS
- Arrays in JavaScript
- Objects in JavaScript
- Conditional and loops in JavaScript
- HTML DOM Manipulation

Section 4: TypeScript

- TypeScript Introduction
- TypeScript Environment Setups
- Variables in TS
- Datatypes in TS
- OOPS in TypeScript
- Features in TS

Section 5: Angular Introduction

- Angular- Definition
- Difference between Framework & Library?
- History of Angular and its versions.
- Why Angular?

- What are the Features of Angular
- Define Single Page Application
- What is the Difference between SPA & Traditional Application?
- Define MVC
- How MVC works in Client & Server sides?

Section 6: Angular Environment setups

- What is Angular CLI?
- What is the Purpose of the CLI?
- Angular CLI installation.
- CLI vs Without CLI Overview.
- Create an Angular App by using CLI.
- Compiling the Angular App & Open it in a browser.
- Angular app Bootstrapping process.
- About Angular libraries
- Brief explanation about the structure of the Angular App.

Section 7: Main Building Blocks of Angular

- An Overview of the below Main Building blocks of Angular
- Modules
- Components
- Decorator
- Metadata
- Templates
- Data binding - Directives
- Services
- Dependency Injection.

Section 8: Angular Modules

- Angular Module Overview.
- Define the Importance of the Module?
- Why Modules?
- Root Module, Core Module, Feature Module and Shared Module - Overview.
- How to create Angular Modules?
- @NgModule Decorator & its Meta data properties - Overview

Section 9: Components

- Angular Component - Overview.
- @Component decorator & its Meta data properties.
- Component's Structure overview.
- What are the ways to render a Component in the view?
- Component Lifecycle Hooks.
- Nested or Parent & Child Component - Overview

Section 10: Data Binding, Property Binding, Event Binding & 2-way Data Binding

- Data Binding Introduction
- Define String Interpolation.
- Property Binding - Overview.
- What is Custom Property Binding?
- Overview of Event Binding.
- String Interpolation VS Property Binding
- Two-way Data Binding
- Implementing the 2-way Data Binding

Section 11: Services

- Introduction - Service.
- Importance of Service.
- How to create Services in Angular?
- What are the ways to Provide Services in Angular?
- Dependency Injection - Overview.
- How to use Dependency Injection?
- What is @Injectable()?

Section 12: Directives

- Directives - Introduction.
- Component VS Directives
- What are the different kinds of Directives available in Angular?
- Difference between Structural & Attribute Directives.
- Overview of All Structural & Attribute Directives.
- @Input decorator and its methods.
- What is ElementRef and its purpose?

Section 13: Components Communication

- Overview of Components Interaction.
- Component Interaction from Parent to Child.
- Component Interaction from Child to Parent.
- @ViewChild decorator overview
- @Input & @Output decorator overview

Section 14: Reactive Form in Angular

- What is Reactive Form?
- Difference between Template Drive & Reactive Form

- What is form group & form control?
- How to sync view & Reactive form TS?
- What is Patch Values & Set Values?
- How to get Reactive from Values?

Section 15: Service and Dependency Injection in Angular

- Service in Angular
- Create & configure Service in Angular.
- How to do Dependency Injection in Angular?
- Define Singleton Object

Section 16: Routing in Angular

- What is Routing?
- How Routing makes our App into SPA?
- How to configure Routing in an Application?
- Load our components dynamically based on url path.
- How to create Child Route?
- Navigating to other links programmatically.
- Passing Parameter to the Routes.
- Client-side authorization using Route Guard

Section 17: HTTP & Observable in Angular

- HTTP Client in Angular.
- REST API - Overview
- How to establish HTTP request to Server side.
- How HTTP Mechanism works?
- What are the methods available in HTTP?
- Define Observable & Observer

- What are the call back methods available in Observable?
- Creation of a Custom Observable
- Define next(), error() and complete()
- How to send Query Params & Custom Headers?
- How to connect any backend & APIs?

Section 18: Authorization in Angular & JWT

- Client side Authorization vs Server side Authorization.
- Server side Authorization by using JWT Token.
- Set JWT Key Expiry time.
- HTTP interceptors - Overview.
- How to configure HTTP interceptors?

Section 19: C# Fundamentals

- Introduction to .NET in C#
- Variables and Data Types
- Operators and Expressions
- Control Structures (if, switch)
- Loops (for, while, foreach, do-while)
- Error Handling with Try-Catch
- Methods and Parameters
- Namespaces and Assemblies

Section 20: Object-Oriented Programming (OOP)

- Classes and Objects
- Constructors and Destructors
- Encapsulation, Abstraction
- Inheritance and Polymorphism

- Interfaces vs Abstract Classes
- Access Modifiers (public, private, etc.)
- Static vs Instance Members

Section 21: C# Advanced Topics

- Collections (List, Dictionary, Queue, Stack)
- Generics
- Lambda Expressions
- LINQ (Language Integrated Query)
- Exception Handling (Advanced)
- File I/O (Reading/Writing files)
- Nullable Types and Null Safety

Section 22: Working with APIs in C#

- Understanding RESTful APIs (GET, POST, PUT, DELETE)
- Using HttpClient for making API calls
- Sending and Receiving JSON
- Deserialization with System.Text.Json or Newtonsoft.Json
- Handling API Authentication (Bearer Tokens, API Keys)
- Calling External APIs and Consuming Responses
- Creating Unit Tests for API Calls

Section 23: Building APIs with ASP. Net Core

- Introduction to ASP. Net Core
- Project Structure (Controllers, Models, Program.cs, etc.)
- Creating Controllers and Endpoints
- Routing and Attributes
- Model Binding and Validation

- Dependency Injection (DI)
- Middleware and Request Pipeline
- Swagger / OpenAPI Integration
- Error Handling and Logging in APIs

Section 24: Working with Databases and SQL

- Introduction to Relational Databases and SQL Basics
- CRUD Operations with SQL (SELECT, INSERT, UPDATE, DELETE)
- Joins and Query Optimization
- Entity Framework Core (EF Core)
- Code-First and Database-First Approaches
- DbContext and DbSet
- LINQ to SQL
- Migrations
- Connecting C# to SQL Server
- Handling Relationships (One-to-Many, Many-to-Many)

Section 25: Advanced

- Asynchronous Programming with async / await
- Working with Azure/AWS for Deployment
- JWT Authentication in APIs
- Role-Based Authorization
- Versioning in APIs