ASHOK IT

Learn Here.. Lead Anywhere..!!

Complete. Net8/9 Core - C#12 Full Stack Web Developer with Azure DevOps, Core GenAI Techniques

.NET 8/9 and C# 12.0

Table of Content

Duration – 5 Months

• Microsoft Visual Studio & Framework History & Background

• Introduction to .NET Framework 4.8

- o What is .NET Platform?
- o What is .NET Framework
- o .NET Framework, Languages, and Tools
- o .NET Framework Major Components
- o Common Language Runtime (CLR)
- o The CLS (Common Language Specification)
- o The CTS (Common Type System)
- o Value Types and Reference Types
- o Compilation and Execution in .NET
- • Understand the .NET Framework 4.8 stack

• Introduction to .NET Core – Net8

- o .NET Core – Overview
- o Characteristics of .NET Core
- o The .NET Core Platform
- o .NET CORE architecture and Advantages
- o Build and run Cross platform apps
- o .NET Core – Environment Setup
- o .NET Core – Code Execution
- o IoC Container & Middleware
- o .NET Core – Modularity
- o .NET Core – Project Files

- o IIS Self Hosting & different cross platform deployments
- o Microservices using .NET Core
- o .NET Core – Windows Runtime and Extension SDKs.
- o .NET Core – Create .NET Standard Library.
- o Comparison between .NET Framework & .NET Core
- o Introduction to Dependency Injection

• Introduction to C#

- o Features of C#
- o C# Compilation and Execution
- o General Structure of a C# Program

• Data Types and Arrays in C#

- o Data Types in C#
- o Value Types and Reference Types
- o Boxing and UnBoxing
- o Single Dimensional, Multi-Dimensional & Jagged arrays
- o Nullable Types
- o Implicitly Typed Local variables
- o Var vs dynamic
- o Is and as operator
- o Ref vs out keywords
- o The ‘object’ base class in .net
- o Equals() vs ==
- o String vs StringBuilder
- o Various String class methods
- o Default parameters, named parameters
- o Parse() vs TryParse() vs Convert Class methods

• Debugging in C#

- o Various Types of .NET Projects
- o Tracing, Debugging, Build

- o Compile Options
- o Using break points
- o Using break conditions
- o Debugging Exception
- o Using watch and output window
- o What are Diagnostics?
- o Debug and Trace Classes
- o Creating multiple projects within one solution
- o Customizing Visual Studio Settings – Extensions, NUGet Package, Environmental Settings
- o Using watch and output window
- o Creating multiple projects within one solution
- o Customizing Visual Studio Settings – Extensions, NUGet Package, Environmental Settings

• OOP with C#

- o Structures and enums
- o The architecture of a class in C#
- o Instance, Class & Reference variables
- o Access Modifier
- o Abstract Classes
- o Constructors, Destructors, The GC
- o .NET Base class library
- o Inheritance in C#
- o Method Overloading
- o Method Overriding
- o Operator Overloading
- o Method Hiding
- o Access modifiers : private, pubic, protected, internal, protected internal, new
- o Anonymous types
- o Abstract classes

- o Sealed classes
- o Creating Interfaces
- o Implementing Interface inheritance
- o Declaring properties within Interfaces
- o Namespaces
- o Creating and using Generic classes
- o Indexers & Properties
- o Auto Implemented properties
- o Static Classes
- o Property Accessors
- o Partial types
- o Extension methods
- o Object Initializer

• Evaluating Regular Expressions in C#

- o RegEx Class
- o Forming Regular Expression
- o Methods for Regular Expression

• Exception Handling

- o Exceptions in C#
- o Exception class hierarchy
- o Try block
- o Multiple catch blocks
- o Finally block
- o Purpose of throw keyword
- o Purpose of inner exception
- o Creating Custom Exception

• Garbage Collection in C#

- o Role of a Garbage Collector
- o Garbage Collection Algorithm

- o Finalize vs Dispose
- • Collections & Generics
- o System.Collections Namespace
- o Collection Interfaces
- o Collection Classes
- o The collection API
- o Working with Generics
- o Creating Generic class, Generic Methods, Interfaces, Delegates
- o Collection Initializers
- o Iterators
- o IEnumerable, IEnumerator, IComparor interfaces
- o Constraints

• Anonymous Types, Delegates, Events & Lambda

- o Extension Methods
- o Anonymous Type
- o Var and Dynamic
- o Introduction to Delegates
- o Events in C#
- o Anonymous Methods
- o Lambda Expression
- o Expression Tree

• File I/O and Serialization

- o Using StreamReader, StreamWritter
- o Using BinaryReader, BinaryWriter
- o Using File, FileInfo, Directory, DirectoryInfo
- o Serialization modes: SOAP, XML
- o JSON serialization

• Introduction To Reflection and Attributes

- o What is Reflection?

- o Attributes.
- o Pre-defined Attributes
- o Custom Attributes.

• Threading, Parallel and Async programming with C#

- o Task Parallel Library
- o Threads Vs. Tasks
- o Thread state
- o Task Based Asynchronous Model
- o Async and Await
- o Using Locks

• Packaging and Deployment

- o File System Editor
- o Registry Editor
- o File Types Editor
- o User Interface Editor
- o Custom Actions
- o Launch Condition Editor
- o Creating Uninstall Shortcut

• New Features in C# 10.0

- o Record structs
- o Improvements of structure types
- o Interpolated string handlers
- o global using directives
- o File-scoped namespace declaration
- o Extended property patterns
- o Improvements on lambda expressions
- o Allow const interpolated strings
- o Record types can seal ToString()
- o Improved definite assignment

- o Allow both assignment and declaration in the same deconstruction
- o Allow AsyncMethodBuilder attribute on methods
- o CallerArgumentExpression attribute
- o Enhanced #line pragma
- o Warning wave 6

• C# 11 Features

- o Generic attributes
- o UTF-8 string literals
- o Newlines in string interpolation expressions
- o List patterns
- o File-local types
- o Required members
- o Auto-default structs
- o Pattern match Span<char> on a constant string
- o Extended nameof scope
- o ref fields and scoped ref
- o Warning wave 7

• C# 12 Features

- o Primary constructors
- o Collection expressions
- o Inline arrays
- o Optional parameters in lambda expressions
- o ref readonly parameters
- o Alias any type
- o Experimental attribute
- o Interceptors

Automated Testing with MSTest and Nunit

- o Using Asserts to Pass or Fail Tests
- o Controlling and Customizing Test Execution

- o Creating Data Driven Tests
- o Reducing Code Duplication and Increasing Test Readability
- o Writing Your First NUnit Test
- o Understanding NUnit Tests
- o Asserting on Different Types of Results
- o Controlling Test Execution
- o Creating Data Driven Tests and Reducing Test Code Duplication

DevOps Concepts

- • Introduction to DevOps :
- o What is DevOps
- o Evolution of DevOps
- o Agile Methodology
- o Why DevOps
- o Agile vs DevOps
- o DevOps Principles
- o DevOps Lifecycle
- o DevOps Tools
- o Benefits of DevOps
- o Continuous Integration and Delivery pipeline

Git

- • Getting Started with Git
- o Install the Git Tools
- o Clone an Existing Repository
- o Add Files to a Repository
- o Edit Files in a Git Repository
- o Create and Merge Branches
- o Rewrite History in a Git Repository
- o Resolve Merge Conflicts

RDBMS & SQL Server

- Introduction to RDBMS
  - o Introduction to databases
  - o Data Models in Database
  - o Properties of RDBMS
  - o Normalization
  - o CODD's Relational Database Rules
  - o Data Integrity
  - o T-SQL Language
- Working with Data Types, Tables & Data Integrity covering DDL, DML, DCL statements
  - o Working with Data Types (Only Basics of Data Types)
  - o Working with Schema
  - o Working with Tables
  - o Implementing Data Integrity
- Beginning with Transact-SQL
  - o Transact-SQL
  - o System Functions
  - o Advanced T-SQL Queries`
  - o Advanced T-SQL Statements
  - o Other T-SQL Statements
  - o Set Operators
  - o Transact-SQL
  - o System Functions
  - o Advanced T-SQL Queries
  - o Advanced T-SQL Statements
  - o Other T-SQL Statements
- Working with Joins and Subqueries
  - o What are Joins?
  - o Types of joins
  - o Subqueries

- Database Objects: Indexes and Views
    - o Introduction to Index in SQL Server
    - o Introduction to Views in SQL Server
- Stored Procedures
    - o Stored Procedure
    - o Implementing Stored Procedure
    - o Exception handling using TRY-CATCH

ADO.NET + LINQ + EF Core

- ADO.NET Architecture
    - o .NET Data Providers
    - o DB Connectivity Architectures in .NET
    - o Elements of .NET Data Providers
    - o Introduction to SQL Server
    - o Namespaces in ADO.NET
    - o Using server explorer window
    - o Connection class
    - o Command class
    - o Direct Command execution against database
    - o Using Parameters in command
    - o Performing CRUD operations

LINQ

- Language Integrated Query
    - o Introduction , LINQ Syntax
    - o Introduction to System.LINQ.Queryable
    - o Query Operators
    - o Select, from, Where
    - o ofType
    - o OrderBy
    - o ThenBy

- o GroupBy, into
- o Select
- o SelectMany
- o Take, TakeWhile
- o First
- o FirstOrDefault
- o Single
- o SingleOrDefault
- o Aggregate functions Sum, Min, Max, Average, Count
- o Distinct
- o Intersect
- o Except
- o Join
- o LINQ projection
- o Deferred execution vs immediate execution
- o Let keyword
- o LINQ to Object
- o LINQ to DataTable

• Entity Framework Core

- o Overview of ORM Products
- o Entity Framework introduction
- o Using Database first Approach
- o Using Code First approach
- o Implementing Repository Pattern
  - ▪ Introduction & Benefits
  - ▪ Repository Pattern implementation
  - ▪ Setting up Entities in EFCore
- o Using LINQ to Entities to perform CRUD operations
- o SQL Query Logging

- o Migration & Database Update
- o Eager Loading Vs Explicit Loading Vs Lazy Loading
- o Raw SQL And Stored Procedures

ASP.NET Core Web API

- • ASP .Net Core Fundamentals
- o ASP.NET Core - Project.Json
- o ASP.NET Core – Configuration
- o Middleware Pipeline

• Introduction to .Net Core WebAPI

- o Introduction to Web Service
- o Introduction to REST API
- o Introduction to Web API
- o Difference between Web Service, WCF Service and Web API
- o HTTPS Verbs
- o Web API Routing
- o Configuring WebApi
- o Testing the Web API Project with Postman and Swagger
- o Building first ASP.NET Core Web API
- o Fluent Validation

• Working with Relational Data using Entity Framework Core

- o Relationships in EF Core
- o HTTP Response Status Codes
- o Try-Catch-Finally block
- o Throwing custom exceptions
- o Global error handling
- o Custom global error handling
- o DML Manipulation using Repository Pattern

• Controller Action Return Types

- o Introduction to Controller Action Return Types

- o Specific Type
- o IActionResult
- o ActionResult<Type>
- o Custom Return Type

• Web API Versioning

• Web API Logging

• Unit Testing in Web API

• Security on Web API

- o Configuring Identity services
- o Configuring authentication
- o Preventing Cross Site Scripting
- o Enabling Cross-Origin Requests (CORS)
- o JWT Token Authentication

Microservices Fundamentals

- o Microservices Fundamentals
- o Basic
- o ASP.NET Core Microservices
- o Advance
- o Introduction to Docker
- o Choosing Between .NET 6 and .NET Framework for Docker Containers
- o Architecting container and microservice-based applications
- o Development environment for Docker apps
- o Designing and Developing Multi-Container and Microservice-Based .NET Applications
- o Implement reads/queries in a CQRS microservice
- o Implementing resilient applications in .net
- o Implement authentication in .NET microservices and web applications

ASHOK IT

Learn Here.. Lead Anywhere..!!

Web Basics - HTML, CSS, JavaScript, ES6 & TypeScript

- JavaScript
    - o Introduction to JavaScript
    - o Data Types, Literals, Variables & Constants
    - o Control Flow, Expression & Operators
    - o Functions & Variable Scope
    - o JavaScript Object & Object-Oriented Programming
    - o Exceptions & Error Handling
    - o Iterators & Generators
- HTML
    - o HTML-Introduction
    - o HTML-Basic Formatting Tags
    - o HTML-Grouping Using Div Span
    - o HTML-Lists
    - o HTML-Images
    - o HTML-Hyperlink
    - o HTML-Table
    - o HTML-Form
    - o HTML-Headers
    - o New Form Elements
    - o Understand the new HTML form elements such as date, number, range, email, search and

      datalist
    - o Understand audio, video, article tags
- CSS 3
    - o CSS-Introduction
    - o Syntax
    - o Selectors
    - o Color Background Cursor
    - o Text Fonts

- o Box Model
- o Display Positioning
- o CSS Floats
- o CSS Floats

• Introducing TypeScript

- o TypeScript Syntax
- o Programming Editors
- o The Type System – Defining Variables
- o The Type System – Defining Arrays
- o Type in Functions
- o Type Inference
- o Defining Classes
- o Class Methods
- o Visibility Control
- o Class Constructors
- o Class Constructors – Alternate Form
- o Interfaces

• Working with ES6 Modules

- o var vs let
- o Arrow Functions
- o Arrow Function Compact Syntax
- o Template Strings
- o Generics in Class ,
- o Generics in Function

Angular 18

• Introducing Angular

- o What is Angular?
- o Central Features of the Angular Framework
- o Appropriate Use Cases

- o Building Blocks of an Angular Application
- o Basic Architecture of an Angular Application
- o Installing and Using Angular
- o Anatomy of an Angular Application
- o Running the Application
- o Building and Deploying the Application

• Components & Templates

- o Creating a Component Using Angular CLI
- o The Component Class
- o The @Component Decorator
- o Registering a Component to Its Module
- o Component Template
- o Using a Component
- o Component Hierarchy
- o Component Lifecycle Hooks
- o Template Location
- o The Mustache {{ }} Syntax
- o Setting DOM Element Properties
- o Setting Element Body Text
- o Event Binding
- o Expression Event Handler
- o Attribute Directives
- o Structural Directives
- o Looping Using ngFor
- o Grouping Elements
- o Template Reference Variable
- o @Output() - Child Component
- o @Output() - Parent Component
- o Full Two Way Binding

- o Setting up Two Way Data Binding in Parent

• Template Driven & Reactive Forms

- o Template Driven Forms
- o Importing Forms Module
- o Two Way Data Binding
- o Form Validation
- o Angular Validators
- o Displaying Validation State Using Classes
- o Additional Input Types
- o Reactive Forms Overview
- o Import ReactiveFormsModule
- o Getting Input Values
- o Setting Form Values
- o Validation
- o Using a Custom Validator
- o Sub FormGroups - Component Class
- o Sub FormGroups - HTML Template

• Services & Dependency Injection

- o The Service Class
- o What is Dependency Injection?
- o Injecting a Service Instance
- o Injectors
- o Dependency Injection in Other Artifacts
- o Providing an Alternate Implementation

• Pipes & Data Formatting

- o Built-In Pipes
- o Using Pipes in HTML Template
- o Chaining Pipes
- o Using a Pipe with ngFor

- o A Filter Pipe

• Angular Routing & Angular Modules

- o The Router Component
- o The Angular Router API
- o Creating a Router Enabled Application
- o Passing Route Parameters
- o Anatomy of a Module Class
- o @NgModule Properties
- o Using One Module from Another

• HTTP Client

- o The Angular HTTP Client
- o Importing HttpClientModule
- o Service Using HttpClient
- o Making a GET Request
- o Observable Object
- o Error Handling & Customizing the Error Object
- o Returning an HttpResponse Object
- o Creating New Observables
- o Observable Operators
- o The map and filter Operators

• Observables & RxJS Library

- o Observables Overview
- o Observables in Angular
- o Introduction to RxJS library
- o Angular Authentication with JSON Web Tokens (JWT