Websbaba
Technologies Pvt. Ltd.

www.websbaba.in   info@websbaba.in

.NET CORE

MODULE 1: INTRODUCTION TO CORE.NET

.NET CORE INTRODUCTION

- Introduction
- What is .NET Core?
- .NET Core Features
- Why Choose .NET Core
- MVC Pattern
- Compare .NET Core Framework and .Net Framework

.NET CORE FIRST CONSOLE APPLICATION

- Introduction of DOTNET CLI
- Working with DOTNET CLI Commands (new, restore, build, publish, run .. etc)
- Create First Console Application using DOTNET CLI.
- Understand Dependencies
- Understand Code execution flow
- Add & Removing Dependencies
- Working with Visual Studio Code Tool
- Modularity

BUILDING YOUR FIRST ASP.NET CORE APPLICATION

- Environment Setup and Configuration options for ASP.NET Core Application.
- Understanding of ASP.NET Core Project Layout.
- Introduction to .NET Command Line Interface (CLI).
- Understanding of .NET CLI Commands.
- Creating and Running Project using .NET CLI.
- Understanding of .NET Core development using Visual Studio Code Editor.
- Understanding steps involved in Request Life Cycle of ASP.NET Core.
- Working with Multiple Environments.

STARTUP AND MIDDLEWARE

- Introduction
- How Middleware Works
- Using IApplicationBuilder
- Showing Exception Details
- Middleware to Match the Environment
- Serving Files
- Setting up ASP.NET MVC Middleware
- Summary

CONTROLLERS IN THE MVC FRAMEWORK

- Introduction
- The Model View Controller Design Pattern
- Routing
- Conventional Routes
- Attribute Routes
- Action Results
- Rendering Views
- Summary

MODELS IN THE MVC FRAMEWORK

- Introduction
- Models and View Models
- Understanding ASP.NET Core MVC Validation
- Need of Server Side and Client-Side Validation
- Accepting Form Input
- POST Redirect GET Pattern
- Model Validation with Data Annotations
- Summary

RAZOR VIEWS

- Introduction
- Understanding of Razor View Engine.
- Difference between Razor View Engine and Web Form Engine.
- Understanding of Razor Syntax.
- Understanding of passing data from View to Controller.
- Introduction to Razor Pages in ASP.NET Core
- Understanding and Creating View Component.
- Layout Views
- _ViewStart
- _ViewImports
- Razor Pages
- An Edit Form
- Partial Views
- Summary

HELPERS

- Understanding Html Helpers
- Types of Html Helpers
- Built-In Html Helpers
- Tag Helpers
- Inline Helpers
- Custom Helpers Url helpers

USING THE ENTITY FRAMEWORK

- Introduction
- SQL Server LocalDB
- Installing the Entity Framework
- Implementing a DbContext
- Configuring the Entity Framework Services
- Entity Framework Migrations
- Up and Running
- Summary

DEPENDENCY INJECTION IN ASP.NET CORE

- Understanding Dependency Injection.
- Advantages of Dependency Injection.
- Using of Dependency Injection in ASP.NET Core.

ASP.NET CORE AUTHENTICATION AND AUTHORIZATION

- Introduction
- An Overview of ASP.NET Identity
- An Overview of OpenID Connect
- Using Secure Connections and Registering the App
- OpenID Connect Configuration
- Authentication Services and Middleware
- User Authorization
- Identities and Claims
- Summary

HOSTING AND DEPLOYMENT

- ASP.NET CORE Application deployment on Windows IIS
- ASP.NET CORE Application deployment on Cloud
- ASP.NET Core Application deployment on Cloud Using Docker

ASP.Net Core WEBAPI

- What is WebAPI?
- Why select Web API?
- Advantage of WebAPI
- Understanding Http Request Verbs & Constraints
- Http Response Http Status Code
- Creating a Web API Project
- Implementing GET, PUT, POST & DELETE method
- ASP NET Web API and SQL Server
- Web API Content Negotiation
- Web API MediaTypeFormatter
- Web API query string parameters
- FromBody and FromUri in Web API
- Call ASP NET Web API from jQuery
- Calling Web API service in a cross domain using jQuery ajax
- Cross origin resource sharing ASP NET Web API
- Enable SSL in Visual Studio Development Server
- ASP NET Web API enable HTTPS
- Implementing basic authentication in Web API
- Call web api service with basic authentication using jquery ajax
- Web API token authentication
- Web API user registration
- Using asp net identity with Web API
- Using fiddler to test Web API token based authentication
- Web API login & logout page
- How to get authenticated user identity name in asp net web api
- ASP NET Web API Google & Facebook authentication
- Web API attribute routing constraints
- Generating links using route names in asp net web api
- IHttpActionResult vs HttpResponseMessage
- Web API versioning using querystring parameter
- Web API versioning using a custom header & accept header
- Web API versioning using Web API versioning using custom media types

Angular

TypeScript

- Basic Types
- Variable Declarations
- Classes
- Functions
- Interfaces
- Generics
- Enums
- Type Inference
- Namespaces
- Modules
- Mixins
- Iterators and Generators
- TypeScript 2.X Features

ASP.Net Core AngularCLI

- What is Angular CLI
- Installing Angular CLI
- Customize Command Prompt
- Configuration of Angular CLI file
- Understanding Angular CLI project structure
- Angular CLI component
- Angular CLI Service
- Angular CLI Module
- Angular CLI generate directives, pipes and routing guards
- Angular CLI generate class, interface and Enum
- Introduction of Linting TypeScript
- Implementing routing in separate module in angular CLI

ASP.Net Core Angular

- Introductions of Angular
- New Features in Angular
- Install Bootstrap for Angular
- Angular form control and form group
- Angular setvalue and patchvalue methods
- Angular formbuilder example
- Implementing Form validation in Angular
- Angular reactive forms validation
- Using reactive form custom validator with parameter
- Angular reactive forms cross field validation
- Creating formarray of formgroup objects in Angular
- Working with Angular dynamic forms
- Remove dynamically created form controls in angular
- Angular reactive forms post ,put & edit, example
- Lazy loading in angular
- Preloading angular modules
- Angular custom preloading strategy

ASP.Net Core Angular CRUD

- Angular project setup
- Reading data in angular
- Understanding Angular routing and navigation
- Working of Angular base href
- Difference b/w Angular value vs ngvalue
- Pass data from parent to child component in angular
- Angular route params
- Implementing Angular optional route parameters
- Working with Data filtering in angular component
- Create observable from array
- Implementing Edit form in angular
- Implementing Delete form in angular
- Understanding Angular content projection

- Understanding Angular client server architecture
- Implementing Angular http client get example
- Working with Angular http client error handling
- Working with Angular http client Post
- Working with Angular http client Put
- Working with Angular http client Delete

Technical Design & Development (Project Based)

- Working with Project
- Programming Language: C# (MVC Razor)
- WebApi Service
- Designing Tools
- CSHTML
- Using CSS
- Using Ajax
- Using JQuery
- Using Angular

Interview Question and Prepartion for placement

- Technical Interview Preparation
- Mock Interview preparation
- HR Session