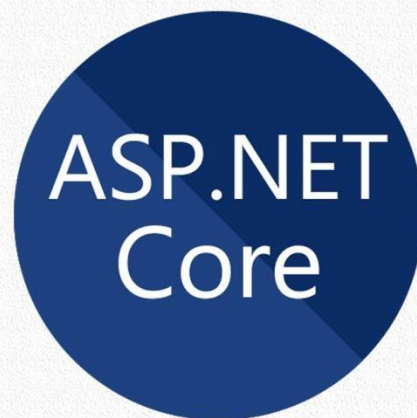


ASP.NET Core Certification Training



ASP.NET Core Certification Training

About the Course

ASP.NET Core certification training is primarily designed for .NET professionals who want to learn how to develop a modern cloud-based app using ASP.NET Core. In this course, you will learn about .NET Core, middleware, MVC pattern, HTML helpers, tag helpers, custom helpers, built-in validation and custom validation, querying the database using Entity Framework Core.

Further, discover to write Unit tests using XUnit and Moq frameworks. Learn to build an end-to-end project with recommended design patterns and practices.

Course objectives

At the completion of this course, attendees will be able to;

- Understand .NET Core architecture and Advantages.
- Understand ASP.NET Core middleware.
- Implement ASP.NET Core Fundamentals.
- Configure ASP.NET Core MVC.
- Use Dependency Injection in ASP.NET Core.
- Work with Entity Framework Core.
- Handle Errors in ASP.NET Core.
- Create REST Service using Web API.
- Understand and Implement Repository, Unit of Work, and Dependency Injection Design Pattern.
- Unit Testing and Mocking using MS Test, xUnit, and Mocking Frameworks.
- Learn to build an end-to-end application using ASP.NET Core.
- Use ASP.NET Core Best Practices to build the enterprise-grade application.
- Application Deployment on IIS and Cloud.

Who can do this course?

All .NET Beginner(s)/Professional(s) who are keen to develop modern, lightweight, and cloud-based web applications should go for this course.

Pre-requisites

Anyone who wants to learn Mastering ASP.NET Core should have a basic knowledge of C# and HTML.

Tools/IDE

Visual Studio 2022, VS Code, SQL Server 2019 or Higher, Postman

Course Curriculum

Module 1

.NET Core

- Introduction to .NET Core
- .NET Core Features
- .NET Core Framework Architecture
- .NET 5.0 vs. .NET 4.5
- .NET Core Supports
- Advantages of .NET Core
- .NET App Model

ASP.NET Core and MVC5

- Introduction to ASP.NET Core
- ASP.NET Core 3.1, 5.0 and 6.0 Features
- ASP.NET Core vs. ASP.NET MVC5
- Advantages of ASP.NET Core

Visual Studio Project Templates

- Understanding Visual Studio ASP.NET Core Templates
- Creating an ASP.NET Core project
- Understanding ASP.NET Core project folder structure
- Understanding configuration files

Module 2

Model, View, Controller & Actions

- Understanding Model, View and Controller
- Types of Views
- Creating Controller
- Understanding Actions
- Actions and Non-Actions Methods
- Understanding Action Results
- Communication between Controller and View

Razor View Engine

- Understanding Razor View Engine
- Razor Syntax
- Razor Statements, Loops etc.

Routes & URLs

- Introduction to Routing
- Defining Routes
- Attribute Routing
- Need of attributes routing

Module 3

Helpers

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

Server-side Data Receiving Ways

- Action Parameters
- View Models/Objects
- ICollection

Reusable UI Components

- Partial View
- View Components

Module 4

ViewModel & Validation

- Creating ViewModel
- Understanding ASP.NET Core MVC Validation
- Need of Server Side and Client-Side Validation
- Validation with Data Annotation
- Custom Validations

Data Passing Techniques

- ViewData
- ViewBag
- TempData
- Session
- Query String
- Cookies

Module 5

Entity Framework and EF Core

- What is ORM
- ORMs used with .NET
- EF6 vs. EF Core
- Advantages of Entity Framework

Database Modeling

- Defining Mapping using Data Annotation
- Implementing CRUD Operations

- Defining Mapping using Fluent API

Database Migrations

- Entity Framework Code First Migrations
- Updating Database when the Model Changes
- Rollback Db changes
- Migrations best practices
- Script Migration

Database First Approach

- Understanding Db First Approach
- Reverse Engineering Approach
- Db First using EF Core Power Tool
- Updating Code using EF Core Power Tool

DB Procedures and Functions

- Creating and Sps and Functions
- Calling Stored Procedures
- Calling Db functions

Module 6

Repository Design Pattern and Unit of Work Design Patterns

- Understanding Repository and UOW Design Pattern
- Need of Repository Design Pattern
- Need to Unit of Work Design Pattern
- Implementing Repository and UOF Design Pattern

Dependency Injection

- Understanding Dependency Injection
- Need of Dependency Injection
- Implementing DI

Unit Testing (Self-paced)

- Understanding TDD Approach
- Unit Testing Frameworks
- Writing Unit Test Cases

Module 7

Web API

- Introduction to SOA
- Introduction to REST
- REST Principles
- Understanding Web API
- Choosing between WCF and WebAPI

Web API Controller & Actions

- Understanding Controller & Actions
- Creating Controller
- Creating Actions

Database Operations and Postman

- CRUD Operations using Web API
- Using Postman for testing Web API
- Consuming Web API using ASP.NET Core

Advanced Web API

- Implementing Content Negotiation
- Parameter binding
- Web API Versioning
- Securing WebAPI

Module 8

Authentication: ASP.NET Core Identity

- ASP.NET Core MVC Authentication Options
- Introduction to ASP.NET Core Identity
- Implementing Identity

ASP.NET Core MVC Pipeline, Middleware and Filters

- Exploring ASP.NET Core Pipeline
- ASP.NET Core MVC Middleware
- ASP.NET Core MVC Filters
- Extending ASP.NET Core MVC Filters

Securing Web API

- Understanding Token based Security
- Implementing Token based Security
- Implementing Authorization

Module 9 (Project Development)

Discussing Project Architecture

- Understanding Application layers
- Discussing Application Entities Properties
- Discussing Relationship among Application Entities

Developing Project

- Designing Application Architecture
- Developing DAL layer using Entity Framework Core

Implementing Design Pattern

- Developing BAL layer
- Repository Design Pattern

- User Login Process
- Dependency Injection

Module Development

- Creating Admin Module using Area
- Creating User Module using Area

Module 10 (Project Development)

ASP.NET Core MVC Authentication: Identity

- ASP.NET Core MVC Authentication Options
- Introduction to Identity
- Implementing Identity

ASP.NET Core MVC Pipeline, Middleware and Filters

- Exploring ASP.NET Core Pipeline
- ASP.NET Core MVC Middleware
- ASP.NET Core MVC Filters
- Extending ASP.NET Core MVC Filters
- Configuring ASP.NET Core MVC Filters

Module 11 (Project Development)

Securing ASP.NET Core MVC App

- Implementing Authorization using Authorization Filter
- Passing Logged in User Info Across the App

Category and Product Listing

- Category CRUD Operations
- Product CRUD Operations

Securing ASP.NET Web API

- Understanding Token based Security
- Implementing Token based Security
- Customizing principal
- Implementing Authorization

Module 12 (Project Development)

ASP.NET Core Best Practices

- Important Nuget Packages
- Tips to optimize application performance
- Understanding Bundling & Minification
- Minimize number of static files requests
- Understanding Error Logging Options
- Configuring Error logging provider

Deployment

- Preparing code for deployment
- Deployment on Cloud/IIS

Build ASP.NET Core Real-Projects

Apply your coding skills to solve real-world problems. You'll learn how to start a project from scratch by following recommended architecture, practices and deploy to server.



Build Food Delivery Website (Like Swiggy, Dominoz)

Learn to build an online food delivery website like Swiggy, Zomato, Dominoz etc.



Build Music Website (Like Gana.com, JioSaavn)

Learn to build a music website like Gana.com, JioSaavn etc.



Build Online Shopping Website (Like Flipkart, Pepperfry)

Learn to build an online shopping website like Flipkart, Pepperfry, Amazon etc.



Build Job Hunting Website (Like Naukri, Indeed)

Learn to build job hunting website like Naukri, Indeed, Shine etc.



Build Doctor Consultation Website (Like Practo, 1mg)

Learn to build online doctor appointment, consultation like Practo, 1mg etc.



Build Service Offering Website (Like Urbanclap, Helpr)

Learn to build service offering websites like Urbanclap, Helpr etc.



Build Online Education Website (Like DotNetTricks, Coursera)

Learn to build service offering websites like DotNetTricks, Coursera etc.



Build Social Media Website (Like Twitter, Facebook)

Learn to build social media websites like Twitter, Facebook etc.



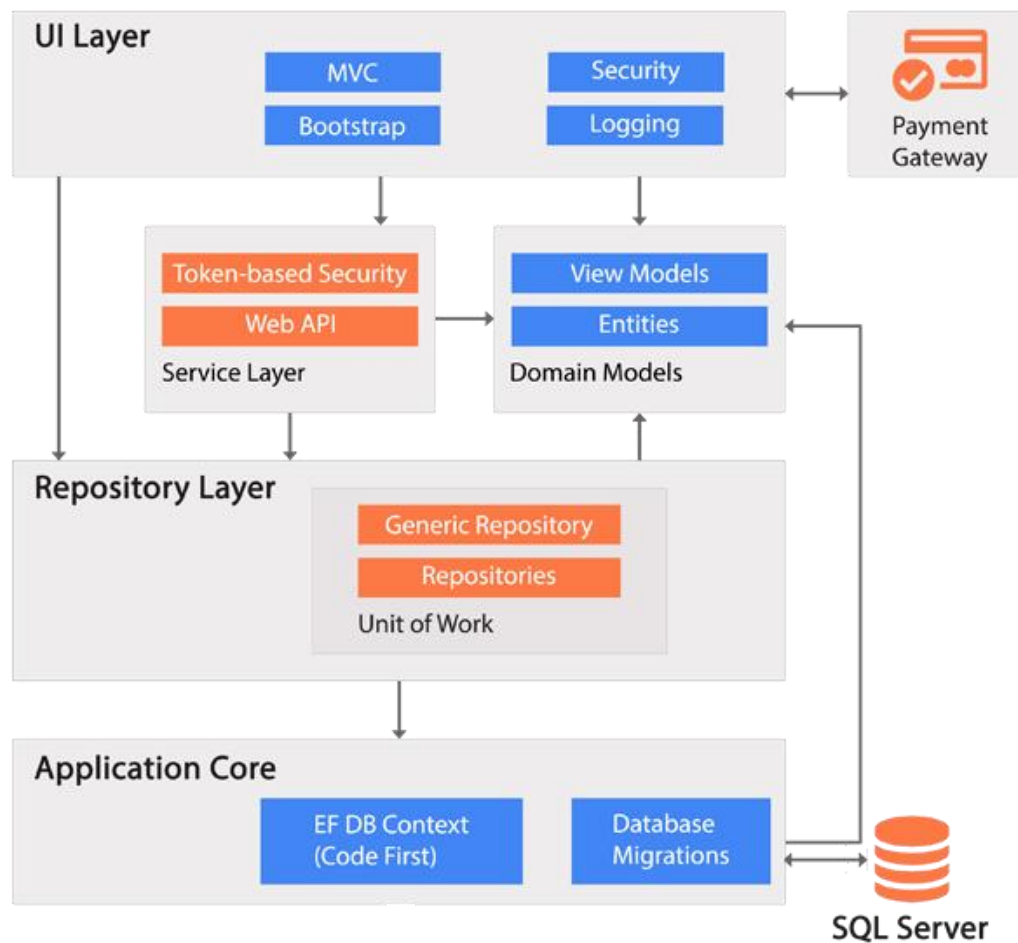
Build Online Booking Website (Like MakeMyTrip, OYO)

Learn to build online booking websites like MakeMyTrip, OYO etc.

ASP.NET Core Project Architecture

The primary purpose of an application architecture is to support the development life cycle of the system. Good architecture makes the system easy to understand, easy to develop, easy to maintain, and easy to deploy. The ultimate benefit of architecture is to minimize the lifetime cost of the system and to maximize programmer productivity.

The goal of the given architecture is to learn and implement important principles and design patterns to build various domains applications like eCommerce, EdTech, Food Delivery or other Service based industries applications.



What you will learn through project?

At the completion of project, attendees will be able to;

- Setup and build project using Clean Architecture.
- Decouples Application modules using Dependency Injection.
- Integrate RazorPay/Paypal payment gateway.
- Implement Repository pattern to perform database operations.
- Build responsive UI using Bootstrap.
- Implement Errors logging and save into Database, File system etc.
- Optimize web pages to boost-up performance.
- Implement Login/Signup workflow using ASP.NET Core Identity.
- Implement Authentication/Authorization for unauthorize access.
- Implement ASP.NET Core best practices.
- Deployment project to cloud or IIS.