Mastering ASP.NET Core

DotNetTricks

Mastering ASP.NET Core

About the Course

The Mastering ASP.NET Core course is primarily designed for .NET professionals who want to learn how to develop 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 database using Entity Framework Core. Further Discover to write Unit tests using XUnit and Moq frameworks.

Course objectives

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

- Understand .NET Core architecture and Advantages
- Understand ASP.NET Core middleware
- 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
- Learn to build end-to-end application using ASP.NET Core.
- Application Deployment on IIS and Cloud

Who can do this course?

All .NET Beginner(s)/Professional(s) who are keen to develop modern, light weight 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 2019 or higher, SQL Server 2016

Course Curriculum

Module 1

.NET Core

- Introduction to .NET Core
- .NET Core Features
- .NET Core Framework Architecture
- .NET Core 3.0 vs. .NET 4.5
- .NET Core Supports
- Advantages of .NET Core
- .NET App Model

.NET CLI and Visual Studio Code

- Introduction to .NET CLI
- .NET CLI Commands
- Creating and Running Project using CLI
- VS Code for .NET Core development

ASP.NET Core and MVC5

- Introduction to ASP.NET Core
- ASP.NET Core 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
- IFormCollection

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

- 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 Migration, DB Procedures and Functions

- Entity Framework Code First Migrations
- Updating Database when the Model Changes
- Calling Stored Procedures and functions
- Code First with existing Database

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

- 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

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

Bundling & Minification

- Understanding Bundling & Minification
- Minimize number of static files requests

Error Logging

- Understanding Error Logging Options
- Configuring Error logging provider
- Tracing Error log

Deployment

- Deployment on IIS
- Deployment on Cloud

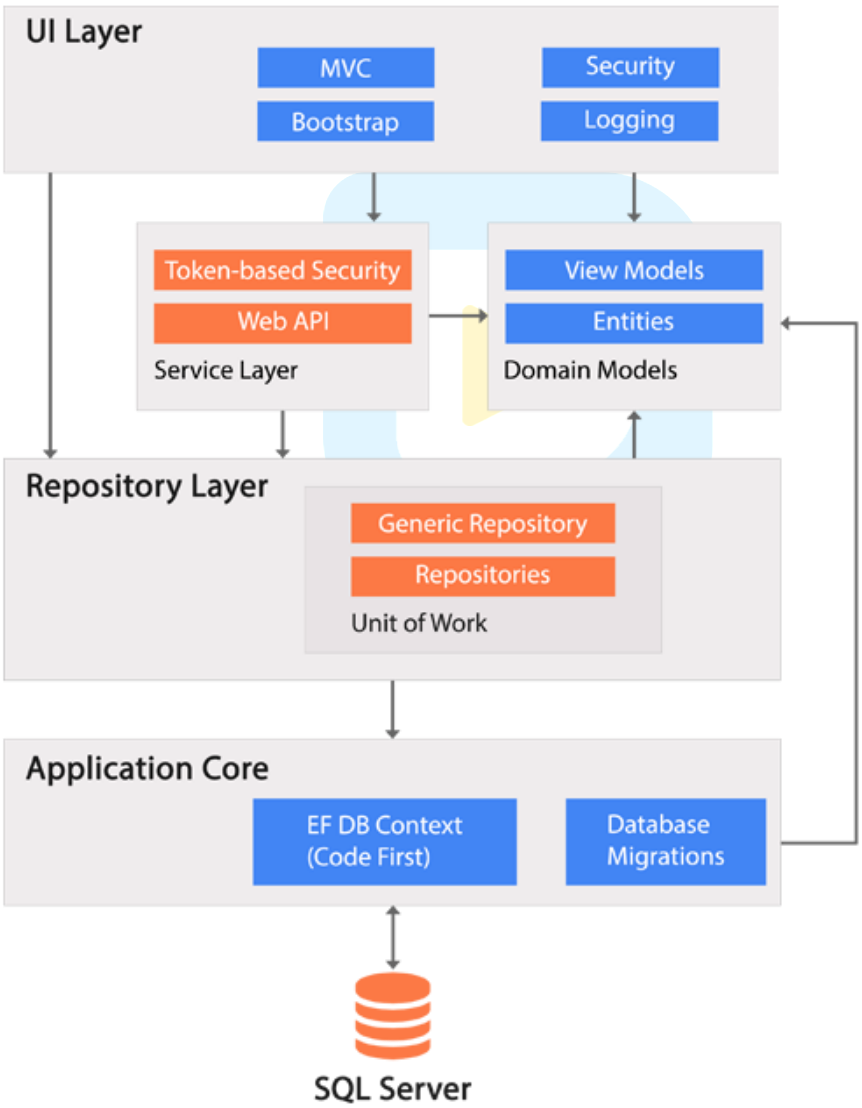
Project Details

Description

You will learn how to develop an online shopping site. The site will show the list of products to User/Customer. A user can purchase products and see the previous order history. Here, admin will be responsible for adding products, categories and approving customer order. The application will use three-tier architecture with extensible approach. We would be using payment gateway to do online payment and finally publishing it on IIS.

Project Architecture

The primary goal of this architecture is to share as much code as possible across the web application and further reuse existing code for mobile apps development if necessary. This architecture will help you to develop web application with the recommended design patterns and practices.



What you will learn through project?

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

- Create DAL layer using Entity Framework Code First approach
- Create BAL layer using Repository design pattern
- Decouple modules using Dependency Injection
- Break a project into multiple modules using ASP.NET Core MVC Areas
- Create Service layer using Web API
- Optimize web pages performance
- Secure application based on user roles
- Authenticate/authorize user using ASP.NET Core MVC custom filters
- Securing ASP.NET Web API using Token based security
- Publish their App on IIS and Cloud

Contact Us

- For more information about the course, visit: www.dotnettricks.com/training/masters-program/aspnet-core
- Feel free to call us at **+91 9999 123 503** or email us at info@dotnettricks.com