SEMATEC
IT Training Center

ASP.NET Core 8.0 and Blazor with Material

اهداف دوره : فناوری ASP.NET Core پلتفرمی یکپارچه و دارای کدهای باز Open-Source بوده که قابلیت اجرا روی سیستم عامل های Windows, Linux, Mac را دارد و بر اساس آخرین ارزیابی های انجام شده TechEmpower Framework Benchmarks در لینک https://bit.ly/3SGPI9R این فریم ورک توانسته است بالاترین سطح کارایی را نسبت به سایر فریم ورک های موجود از قبیل Node.js, Java Servlet کسب نماید. همچنین توسعه روز افزون فناوری های فرانت، منجر به ظهور فریم ورک قدرتمند با عنوان Blazor شده که می تواند در بالاترین سطح کیفیت نیازمندی های توسعه نرم افزار را در بخش را فرانت تامین نماید.

SEMATEC
IT Training Center

مخاطبین دوره : تمامی افرادی که علاقه مند به یادگیری و تولید برنامه های Web-Based می باشند، می توانند در این دوره شرکت نمایند. بازار کار امروز بدلیل رقابتی شدن، انتظار داشتن مهارت های بسیار زیادی را حتی از مبتدیان این رشته دارد که یادگیری آنها نیازمند زمان و هزینه قابل توجهی می باشد! پس چه باید کرد ؟ ما بر اساس تجربه و شناخت دقیق بازار کار و محدودیت های اقتصادی در حال حاضر، تلاش کردیم دوره ای را طراحی نمائیم که با رویکرد پروژه محور و اجتناب از تئوری پردازی محض، در کوتاه ترین زمان مهارت های مذکور را آموزش دهد.

پیشنیاز دوره : گذراندن دوره های Programming In C# (1) و Web Fundamental یا تسلط به مطالب دوره های مذکور

مدت دوره : مدت زمان آموزشی این دوره ۵۴ساعت می باشد.

دستاوردها: در انتهای دوره فراگیران توانایی طراحی فرم های اطلاعاتی با استفاده از بلیزور متریال، کار با کنترل های پیشرفته نظیر جدول و دیالوگ، اعتبارسنجی دادها، پیاده سازی سرویس های ارسال و دریافت اطلاعات، ایجاد مدل های اطلاعاتی سمت فرانت اند/بک اند، ساخت بانک اطلاعاتی در محیط SQL Server با رویکرد Code First، آشنایی با بعضی از اصول مهندسی نرم افزار نظیر SOLD, DRY, PI, SoC, IoC، الگوهای طراحی نظیر Factory, Builder, Chain Of Responsibility, Repository, Result، آشنایی با معماری های Monolith Modular, Clean Architecture و پیاده سازی بخشی از نیازمندیهای امنیتی داشته باشند و در نهایت همه موارد فوق را از طریق ساخت یک پروژه عملی تجربه نمایند.

سرفصل مطالب آموزشی :

Introduction

- ASP.NET Core
- TypeScript
- Angular

Development IDE

- Visual Studio Code and Required Plugins
- .NET 8.0 SDK
- Node.js

Part 1 – Backend

ASP.NET Core Fundamentals

- Dotnet-CLI Command
- Create new project
- Overview project structure
- Top-level Statement
- File Scoped Namespace
- Global Namespace
    - Implicit
    - Explicit
    - Custom
- Program

Middleware

- Middleware
- Request delegate
- Async/await
- Register Middleware
- Register Priority
- Create Custom Middleware
- Middleware Pipelining

- Middleware Extension
- Built-In Middleware's

Service Provider

- Service Provider
- Create Service Instance
- Dependency
    - Structural
    - Behavioral
- SOLID Principals
- Dependency Inversion Principal
- Inversion of control
- Dependency Injection
    - Constructor Injection
    - Method Injection
    - Property Injection
- Dependency Graph
- Object life cycle
- Service Extension
- Application Configuration
    - Launch
    - Setting
    - Command line arguments
- Option Pattern
    - Option Setting
    - Option Setup
- Application Environment
    - Development
    - Production

- Staging

RESTfull Service

- Rest Architecture
- Design aspect
- RESTful
- Controller, Action, Routing
- Specific Type, IActionResult, ActionResult<T>
- GET, POST, PUT, DELETE, PATCH
- HTTP Status Code
  - Informational
  - Success
  - Redirection
  - Client Error
  - Server Error
- Minimal API, Endpoint
- Carter Framework
- URI best practice
- API Test Tools
  - Swagger
  - Postman
  - Thunder Client

Architectural Concepts

- Monolithic application
- All-in-one Project
- Layers
- Traditional "N-Layer"
- Onion Architecture

- DRY Principal
- Domain-Driven Design (DDD)
- Clean Architecture
- Project Structure
    - Library Based Pattern
    - Folder Based Pattern
    - Modules
    - Features

Develop Catalog Service

- Overview eShop Application
- Creating Catalog Service
- Add Domain Layer
- Add Application Layer
- Add Infrastructure Layer
- Add Presentation Layer

Domain Layer

- DDD Overview
- Domain Primitives
    - Contracts
        - IEntity
        - IRepository
    - Abstraction
        - Entity
- Domain Features
    - Product Features
        - Product Entity
        - Product Repository

SEMATEC
IT Training Center

Application Layer

- Application Overview
- Contracts
  - IProductManager
  - IEmailService
- Application Features
  - Result Pattern
  - Product Features
    - Implement Manager
    - Implement Mapper by AutoMapper
    - Implement DTOs with C# Record
    - Injecting IProductRepository
    - Check Product Name Uniqueness

Infrastructure Layer

- Persistence
  - Faking with Bogus
  - Implement Product Fake Repository
  - Add Entity Framework Core
  - Implement Product Configuration
  - Implement Catalog Context
  - Implement SQL Repository
  - Implement Product Repository
  - Add Migrations
  - Update Database
- Implement Data Services
  - Paging
  - Sorting

        - Filtering
        - Searching

Presentation Layer

- Add Carter Framework
- REPR Pattern
- Implement Product End Points
    - Implement CRUD Operation
    - Implement Validators with FluentValidation
    - Implement Public Search

Develop Security Service

- Creating Security Service
- Add Domain Layer
- Add Application Layer
- Add Infrastructure Layer
- Add Presentation Layer

Domain Layer

- Security Features
    - User Entity
    - Role Entity

Application Layer

- Contracts
    - IUserManager
    - IIdentityService
- Application Features
    - Login
    - Registration
    - Check User Name Uniqueness

SEMATEC
IT Training Center

Infrastructure Layer

- Add Microsoft Identity Manager
- Persistence
    - Add Entity Framework Core
    - Implement Security Context
    - Add Migrations
    - Update Database
- Authentication
    - Identity Service
    - User Registration
    - Add Claims
    - Finding User by Name/Email
    - User Login
    - JSON Web Token
        - ✓ Implement Token Service
        - ✓ User Claims
        - ✓ Token Handler

Presentation Layer

- Implement User End Points
    - Registration
    - Login

Part 2 – Frontend

Blazor

- Create the First Blazor WebAssembly Project
    - Basic syntax
    - Binds
    - Methods
    - Events and event arguments

- Page and Components
  - Basic
  - Custom Events
  - Custom reference
  - Exchange data between components
  - Parent To Child
  - Child To Parent
  - Sibling

Setup Catalog UI Project

- Create Application Layout
  - Installing Material library
  - Toolbar
  - Sidebar
  - Drawer
  - Dark and Light themes

Catalog Feature

- Product Feature
  - Create Product Manager Page
  - Create Product Service
    - Add Refit Package
    - Create Service Contract
    - Add Endpoints
  - Create Product
    - Create Component
    - Create Request/Response
    - Create Validator
    - Create ViewModel
  - Edit Product
  - Product List
  - Public Search
  - Dialog Service

- Delete and Confirmation Message

Security Feature

- Security Feature
  - Create User Manager Page
  - Create User Service
    - Create Service Contract
    - Add Endpoints
  - Register User
    - Create Component
    - Create Request/Response
    - Create Validator
    - Create ViewModel
  - Login User