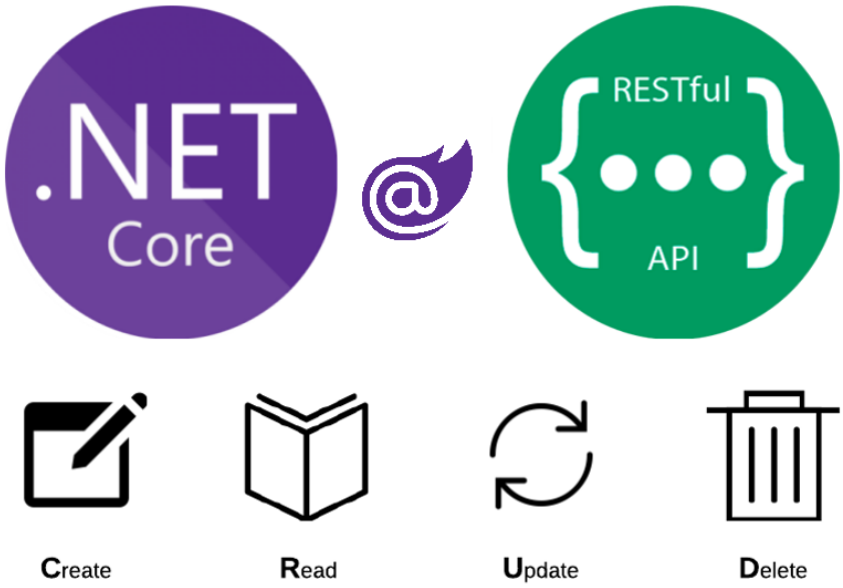
SEMATEC
IT Training Center

ASP.NET Core 7.0, Web API, EF and Blazor with Material



- ***اهداف دوره :*** فناوری ASP.NET Core پلتفرمی یکپارچه و قابل اجرا روی سیستم عامل های Windows, Linux, Mac می باشد که بر اساس ارزیابی های انجام شده توسط موسسه جهانی TechEmpower، توانسته است بالاترین سطح کارایی را نسبت به سایر فریم ورک های موجود از قبیل Node.js, Java Servlet کسب نماید و دارای کدهای Open-Source می باشد. این فناوری برای ساخت کلیه نرم افزارهای مدرن وب با قابلیت رایانش ابری در بستر اینترنت طراحی شده است. همچنین از طریق آن می توان سرویس های توزیع شده، برنامه های IoT و یا بک آفیس برای برنامه های موبایل تولید کرد. همچنین سامانه های تولید شده را نیز می توان روی وب سرورهای IIS, Nginx, Apache میزبانی نمود.
- ***مخاطبین دوره :*** تمامی افرادی که علاقه مند به یادگیری و تولید برنامه های Web-Based می باشند، و دوست دارند فقط با استفاده از استک فناوری مایکروسافت، نرم افزارهای قدرتمندی تولید نمایند می توانند در این دوره شرکت نمایند.

SEMATEC
IT Training Center

- ***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 48 ساعت می باشد.
- ***دستاوردها:*** در انتهای دوره فراگیران توانایی طراحی و تولید سرویس های RESTfull جهت ارسال و دریافت اطلاعات، ارتباط با بانک های اطلاعاتی، فناوری ORM، طراحی و پیاده سازی فرم های اطلاعاتی، اعتبارسنجی دادها آشنا خواهند شد.
- ***سرفصل مطالب آموزشی :***

Backend Modules

Environment Configuration

  - Setup tools and IDE
- Dotnet-CLI
  - Command
  - Create new project
  - Overview project structure
- Application Configuration
  - Program, Startup
  - Launch, Setting, Environment, Command line arguments
- Middleware
  - Request delegate
  - Async/await
  - Type of middleware
    - Content generation
    - Request editing
    - Response editing
    - Short-circuit
  - Pipelining
  - Extension method
- Service
  - Inversion of control
  - Dependency injection
  - Object life cycle
- Web API apps

    - Rest Architecture
    - Design aspect
    - RESTful
    - Controller, Action
    - Endpoint
    - Routing
    - URI best practice
    - GET, POST, PUT, DELETE, PATCH
    - ActionResult, IActionResult
- API Documentation
    - Swagger
    - XML documentation
    - API Versioning
- Repository Pattern
- Entity Framework Core
    - DbContext
    - Data model (Code First)
    - Fluent configuration
    - Id generation strategy (Identity, GUID, HILO)
    - Migration
    - Seed Data
    - Data relationship
        - One to one
        - One to many
        - Many to many
    - Loading related data
        - Eager
        - Explicit
        - Lazy
    - Refactoring repository
- Secret Manager
- AutoMapper
    - ViewModel/Data Transfer Object (DTO)
    - Profile configuration
    - Custom conversion
- Parent/Child resource management service
- Data validation
    - Field level
    - Form level
    - Validators
        - Using built-in
        - Custom validator

- API Versioning
  - By Query
  - By URL
  - By Header
  - Deprecation
- Query Services
  - Paging
  - Filtering
  - Sorting
  - Searching
- Content Negotiation
  - Response
  - Media type
  - Custom format
- Model Binder
- Filters
- Web apps
  - Overview
    - Mode View Controller (MVC)
    - Razor Page
    - Blazor WebAssembly

Frontend Modules

- Blazor
  - Create the First Blazor WebAssembly Project
    - Basic syntax
    - Binds
    - Methods
    - Events and event arguments
  - Page and Components
    - Basic
    - Custom Events
    - Custom reference
  - Layout
    - Custom Layout
    - Navigation
  - Installing Material library
  - Application layout design
  - Adding Side Navigation
  - Adding Drawer Feature
  - Implementing a dark and light theme
  - Introducing CRUD Application (Create, Retrieve, Update, Delete)

- Designing and Implementing Product Manager Page
  - Design Toolbar
  - Design Product List
- Implementing Product List Component
  - Adding Material Grid View
  - Using Material Icon, Button
- Calling REST API
  - Working with HttpClient
  - Creating IDataService Contract
  - Implementing DataService
  - Using JSON.NET
  - Using Http Response Headers
  - Creating Services Models
  - Fetching Data from REST AAPI
- Register Service in Dependency Injection
- Using Service in Product List Component
- Loading and Displaying Data
  - Showing progress bar for data loading
  - Implementing Paging Feature
  - Implementing Sorting Feature
  - Implement Searching Feature
- Creating a New Product Form Component
  - Designing Form
  - Using Material controls
  - Using Client Side Validation Controls
  - Displaying Custom Error Messages
  - Using Dialog Service
  - Using DataService for sending data to the REST API
- Integrating New Form with Product Manager
- Completing Edit and Delete Features
- Full application integration with the REST API