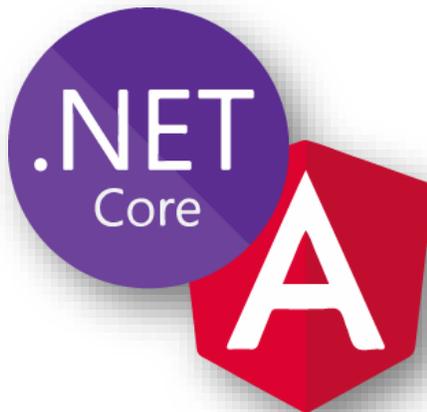


## ASP.NET Core 7.0 and Angular 15.0 with Material

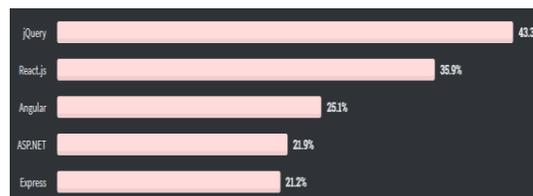


اهداف دوره : فناوری ASP.NET Core پلتفرمی یکپارچه و قابل اجرا روی سیستم عامل های Windows, Linux, Mac می باشد که بر اساس ارزیابی های انجام شده توسط موسسه جهانی TechEmpower، توانسته است بالاترین سطح کارایی را نسبت به سایر فریم ورک های موجود از قبیل Node.js, Java Servlet کسب نماید و دارای کدهای Open-Source می باشد. از طرفی گسترش روز افزون تکنولوژی های فرانت، منجر به ظهور فریم ورک های قدرتمندی در این عرصه گردید که یکی از کامل ترین آنها Angular می باشد که توسط شرکت گوگل توسعه داده شده است. یکی از دلایل محبوبیت آن استفاده از زبان برنامه نویسی TypeScript در توسعه آن می باشد که خود منجر به تولید نرم افزارهای بسیار با کیفیت می گردد.

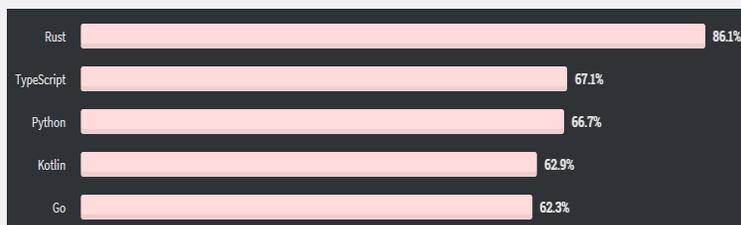
بر اساس ارزیابی هایی که در سال 2022 توسط وب سایت Stack Overflow در خصوص تکنولوژی های برنامه نویسی صورت گرفت نتایج زیر به شرح ذیل حاصل گردید :



ASP.NET Core



Angular



TypeScript

و لذا بدیهی است تلفیق چند تکنولوژی محبوب و قدرتمند می تواند نتیجه بسیار دلپذیر و درخشانی را ایجاد نماید. جهت کسب اطلاعات بیشتر در خصوص ارزیابی به لینک زیر مراجعه فرمایید :

<https://insights.stackoverflow.com/survey/2022>

**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، انتظارات بسیار زیادی را حتی از مبتدیان این رشته دارد که یادگیری همه این مهارت ها نیازمند زمان و هزینه قابل توجه 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 کنیم که با رویکرد پروژه محور و اجتناب از تئوری پردازی محض، در کوتاه ترین زمان مهارت های کدنویسی را آموزش دهد. از طرفی با بکارگیری ابزارهای مدرن از قبیل Node.js, VS-Code, Postman, JSON-Server لذت جدیدی از برنامه نویسی و تولید نرم افزار را تجربه نمایند.

پیشنیاز دوره: گذراندن دوره های (1) Programming In C# 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 فرم های اطلاعاتی با استفاده از انگولار متریال، کار با کنترل های پیشرفته نظیر جدول و دیالوگ، اعتبارسنجی داده ها سمت فرانت و سرور، پیاده سازی سرویس های ارسال و دریافت اطلاعات سمت فرانت، پیاده سازی سرویس های ارسال و دریافت اطلاعات سمت سرور، ایجاد مدل های اطلاعاتی سمت سرور، ساخت دیتابیس با رویکرد Code First، آشنایی با نیازهای امنیتی و در نهایت با ساخت یک برنامه CRUD بسیار زیبا و کاربردی برای مدیریت و ثبت نام کاربران آشنا خواهند شد.

سرفصل مطالب آموزشی:

## **Introduction**

- ASP.NET Core
- TypeScript
- Single Page Application (SPA)
- Angular

## **Installing development tools**

- Visual Studio 2019
- Visual Studio Code and Required Plugins
- .NET 5.0 SDK
- Postman
- Node.js
- Angular CLI

## **Part 1 - ASP.NET Core**

### **Fundamental Blocks**

- Dotnet-CLI Command
- Create new project
- Overview project structure
- Application Configuration
- Program
- Middleware
- Request delegate
- Async/await
- Type of middleware
  - Content generation
  - Request editing
  - Response editing
  - Short-circuit
- Middleware Pipelining
- Middleware Extension
- Service
- Dependency Inversion
- Inversion of control
- Service Locator
- Dependency injection
- Object life cycle
- Service Extension
- Web API apps

### **RESTfull Service**

- Rest Architecture

- Design aspect
- RESTful
- Controller, Action
- Endpoint
- Routing
- URI best practice
- GET, POST, PUT, DELETE, PATCH
- ActionResult, IActionResult
- API Documentation
- Swagger
- XML documentation

## **Data Persistence**

- Repository Pattern
- Fake Repository
- Entity Framework Core
- DbContext
- Data model (Code First)
- Fluent configuration
- Id generation strategy (Identity, GUID, HILO)
- Migration
- Seed Data
- Refactoring repository with EF
- AutoMapper
- Data Transfer Object (DTO)
- Profile configuration
- Data validation
  - Field level
  - Form level

- Validators
- Using built-in
- Custom validator

## Security

- Authentication
- JWT
- Service Token Handler
- Registration
- Login

## Part 2 - Angular

### TypeScript

- Basic Syntax
- Object Oriented Programming
  - Class
  - Interface
  - Inheritance
  - Generic
- Modular Programming
  - Class, Function, Variable, Constant Named or Default Export
  - Import Module Item
  - Namespace
- Webpack
  - Configuration
  - Building TypeScript Library
  - Publish Package
  - Creating NPM Repository
  - Publish Package to NPM

### Angular

- Project structure
- Building blocks overview

- Module
  - Component
  - Directive
  - Service
  - ...
- Basic syntax
- Template-Driven form
- Start CRUD application
- Validation from input
- Service
- Dependency Injection
- HttpClient
- Reactive-Driven form
  - FormControl
  - FormGroup
- Material
  - Field
  - Checkbox
  - Radio
  - Dropdown
  - Validator
  - Table
  - Dialog
  - toastr
- Routing

## **Building CRUD Application**

### **Getting Started with Web API**

- Creating the entry model
- Configuring DbContext
- Configuring a database
- Creating a database
- Getting data from the database

## **Getting Started with Angular**

- Adding a component
- Adding a service
- Configuring routing in Angular
- Requesting data from Web API
- Designing a view with Material Design
- Creating a Material Design table
- Showing data to the users

## **Saving Data to Database**

- Creating a Web API HttpPost endpoint
- Configuring an Angular reactive form
- Creating an Angular reactive form
- Validating an Angular reactive form
- Designing a header and footer
- Creating an HttpPost request in Angular

## **Updating Existing Data**

- Creating a Web API HttpPut endpoint
- Setting up a dialog
- Configuring the dialog with data
- Populating the dialog with data
- Creating an HttpPut request in Angular

## **Deleting Data**

- Creating a Web API HttpDelete endpoint
- Adding a confirmation step
- Getting a single entry
- Designing a confirmation view
- Creating an HttpDelete request in Angular

## **Authentication Using Tokens**

- Setting up AuthController
- Setting up AuthService
- Creating a register component
- Registering users
- Generating a token
- Requesting a token from Angular
- Authentication check via tokens
- Logging out users
- Creating an Angular login component
- Getting data from the login form
- Logging in users
- Handling errors in angular