**به نام خدا**

به بهانه یک دوره جدید، می‌خواهم کمی تیراندازی کنم! پیشاپیش عذرخواهی می‌کنم!

عنوان دوره جدید:

ASP.NET Core Web API + Blazor Web Assembly **Custom Security**

معرفی Road Map برای NET. کارها

تفاوت زبان‌های برنامه‌نویسی #C و Python و معرفی استراتژی یادگیری آن‌ها

**یک سری مباحث جامعه‌شناسی!**

محدودیت LinkedIn!

اگر نمونه‌گیری بیش از ۱۰۰۰ واحد باشد، یک نمونه‌گیری نرمال بوده و تا حد زیادی، قابل اتکاء و اعتماد می‌باشد.

در طی ۳۳ سال، بیش از شانزده هزار نفر شاگرد داشته‌ام.

**ویژگی مردم ایران به صورت نرم یا نرمال**

**نکته:** ما ایرانی‌های بی‌اندازه! ویژگی‌های مثبت و قابل تحسین داریم! ولی اگر، کمی نقاط ضعف داریم، شایسته است که در صورت امکان این نقاط ضعف را اصلاح کنیم!

ما ایرانی‌ها، **کم و بیش**، مردمانی با **ضریب هوشی** متوسط بوده، که معمولاً بدون اندیشه و تحلیل (**دیدگاه فلسفی**)، **متعصبانه** و غیر منصفانه، نسبت به عقاید خود پافشاری می‌کنیم، و البته می‌توان با کار **Social Engineering**، ما را جوگیر کرده و نظرات / ایدئولوژی‌مان را تغییر دهیم!

**ضریب هوشی!**

متوسط ضریب هوشی ایرانی‌ها ۸۵ بوده و در بین ۲۰۰ و اندی کشور، رتبه ۸۳ را داراست.

**فلسفه!**

فلسفه هر چیزی از خود آن چیز مهم‌تر است!

* فلسفه ازدواج (ازدواج دو ارمنی)
* فلسفه فرزندآوری (برای بهبود زندگی و اصلاح طرف مقابل!)
* فلسفه مذهب (اهمیت حق الناس! در همه ادیان)
* فلسفه مهاجرت (هجرت!)
* فلسفه انتخاب بانک اطلاعاتی! مثلاً اوراکل و SQL Server
* فلسفه انتخاب زبان برنامه‌نویسی! مثلاً #C و Python (من جزء خوارج نشده‌ام!)

**تعصب!**

تعصب در هر موضوعی مجاز نیست! و در جایگاهی هم که الزامی است! تا فلسفه آن چیز را نمی‌دانیم، نباید پرخاشگرانه و غیر منصفانه اظهارنظر کنیم!

* طرفداری از یک تیم فوتبال
* طرفداری از فلسطین و اسرائیل

**جوگیر شدن! و تغییر رویه!**

* دو تا جلسه هیات / دو ساعت مشاهده فیلم‌ها و سخنرانی‌های Instagram و ماهواره
* انتخابات!

**Learn 100**

[[Client]] → Create File : libman.json

[wwwroot] → Update File : index.html

**Learn 200**

[Properties] → Update File : launchSettings.json

[Shared] → Delete File : SurveyPrompt.razor

[Pages] → Delete File : Counter.razor

[Pages] → Delete File : FetchData.razor

[wwwroot] → Delete Folder : sample-data

[Pages] → Update File : Index.razor

[Pages] → Create File : Page1.razor

[Pages] → Create File : Page2.razor

[Pages] → Create File : Page3.razor

[Pages] → Create File : Page4.razor

[Pages] → Create File : Page5.razor

[Shared] → Update File : NavMenu.razor

Link to Pages…

[Pages] → Create Folder : Authentication

[Authentication] → Create File : Login.razor

[Authentication] → Create File : Logout.razor

[Shared] → Update File : MainLayout.razor

Link to Login and Logout

**Learn 300**

[Properties] → Update File : launchSettings.json

**Resources:**

[ASP.NET Core Blazor authentication and authorization](https://learn.microsoft.com/en-us/aspnet/core/blazor/security/webassembly/?view=aspnetcore-7.0)

<https://learn.microsoft.com/en-us/aspnet/core/blazor/security>

Secure ASP.NET Core Blazor WebAssembly

https://learn.microsoft.com/en-us/aspnet/core/blazor/security/WebAssembly

ASP.NET Core Blazor state management

https://learn.microsoft.com/en-us/aspnet/core/blazor/state-management

ASP.NET Core Blazor Hybrid authentication and authorization

https://learn.microsoft.com/en-us/aspnet/core/blazor/hybrid/security

ASP.NET Core Blazor WebAssembly additional security scenarios

https://learn.microsoft.com/en-us/aspnet/core/blazor/security/webassembly/additional-scenarios

**NuGets:**

https://www.nuget.org/packages/Microsoft.AspNetCore.Components.Authorization

[[Client]] → Update File : Client.csproj

Add NuGets…

[[Client]] → Create Folder : Infrastructure

[Infrastructure] → Create File : CustomAuthenticationStateProvider.cs

Anonymous

[[Client]] → Update File : Program.cs

builder.Services.AddOptions();

// If we do not have Policy!

//builder.Services.AddAuthorizationCore();

// If we have Policy!

builder.Services.AddAuthorizationCore(options =>

{

options.AddPolicy("CanBuy", policy =>

policy.RequireClaim("Over21"));

options.AddPolicy("CanDelete", policy =>

policy.RequireRole("Administrator"));

});

builder.Services.AddScoped

<AuthenticationStateProvider,

CustomAuthenticationStateProvider>();

[Shared] → Create File : RedirectToLogin.razor

[[Client]] → Update File : App.razor

[[Client]] → Update File : \_Imports.razor

@using Microsoft.AspNetCore.Authorization; → For [Authorize]

@using Microsoft.AspNetCore.Components.Authorization → For App.razor

Breakpoints:

[Infrastructure] → Create File: CustomAuthenticationStateProvider.cs

[Shared] → Create File: RedirectToLogin.razor

[Pages] → Update File : Page2.razor

@attribute [Authorize]

[Pages] → Update File : Page3.razor

@attribute [Authorize(Roles = "Administrator")]

[Pages] → Update File : Page4.razor

<AuthorizeView></AuthorizeView>

**Learn 400**

References:

Implementing Authentication

https://blazorschool.com/tutorial/blazor-wasm/dotnet6/authentication-and-authorization-724479

https://blazorschool.com/tutorial/blazor-wasm/dotnet6/implementing-authentication-829424

https://blazorschool.com/tutorial/blazor-wasm/dotnet6/implementing-authorization-980267

AuthenticationStateProvider in Blazor WebAssembly

https://code-maze.com/authenticationstateprovider-blazor-webassembly

Custom Authentication in Blazor WebAssembly – Step-By-Step Detailed Guide

https://codewithmukesh.com/blog/authentication-in-blazor-webassembly

[Properties] → Update File : launchSettings.json

[Infrastructure] → Create File : CustomAuthenticationStateProvider.cs

Not Anonymous!

[Authentication] → Update File : Logout.razor