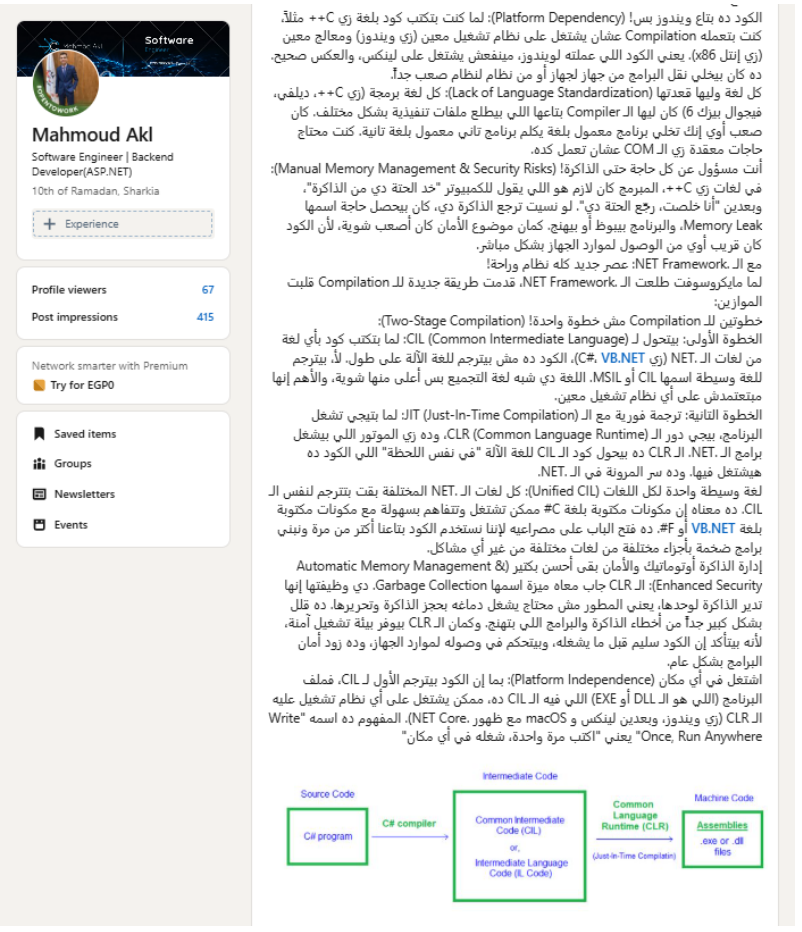
1) LinkedIn article about Compilation With dotnet FW Vs Pre dotnet



2) What has been done to reduce this overhead?

1) Tiered Compilation

- Starts by compiling methods quickly (Tier 0).
- frequently used methods are recompiled with optimizations (Tier 1).

2) ReadyToRun (R2R)

- Pre-compiles IL into native code during publish.
- Assemblies contain both IL and native code.

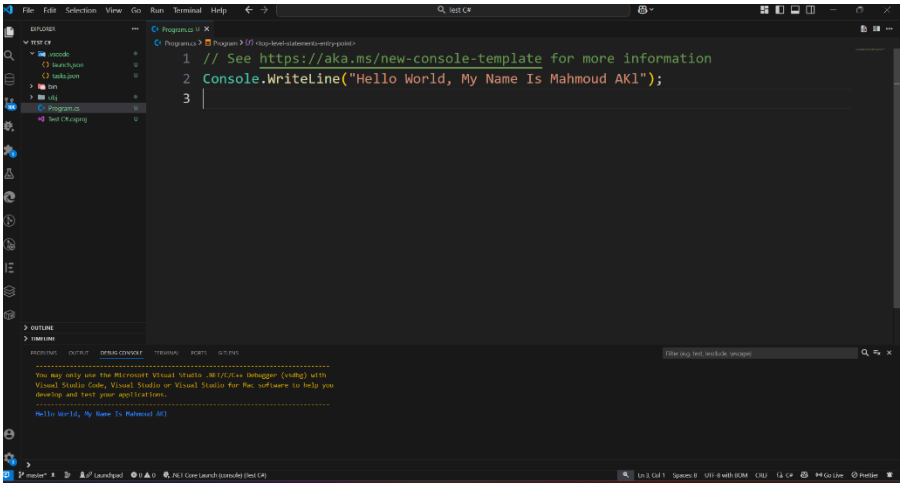
3) Native AOT (Ahead-of-Time Compilation)

- Compiles the entire application to native code before runtime.
- No JIT needed at all during execution.

4) CrossGen / CrossGen2 (Older method before Native AOT)

- Precompiles methods to native code at build time.
- Used by ReadyToRun internally.

3)



4)

1- Dot .NET Versions

- **.NET Framework**: مثال: Windows. 4.8 بدأ من سنة 2002، بيشتغل بس على.
- **.NET Core**: 2016 نسخة خفيفة ومفتوحة المصدر. أول إصدار في. Multi-platform (Windows, Linux, macOS)
- **.NET 5/6/7/8**: تسمى ".NET" بدأت من. فقط، ودي الموحدة لكل الأنواع. .NET 5 2020 في.
- **آخر إصدار**: حاليًا .NET 8 (LTS – دعم طويل الأجل)

2- Namespace

- هي طريقة لتنظيم الكود داخل Projects.
- زي "مجلدات" داخلية Classes، Function، Inertfaces.
- تمنع تعارض الأسماء بين Classes.

```
namespace MyApp.Services
{
    class Getdata { ... }
    class UserName { ... }
    class password { ... }
}
```

3- NET Core

- نسخة خفيفة وسريعة منNET.
- Multi-platform + مفتوحة المصدر
- أداء أعلى منNET Framework.
- مثالي لتطبيقات الويب والـAPIs
- من 2020 بقى فيه دمج في اسم واحد .NET. :يعني (**Unified .NET** = NET Core + Framework).

4 Solution

- الـ Solution هو الملف الرئيسي `.sln` في أي مشروع C# أو .NET.
- بيجمع مشاريع مختلفة (Projects) تحت مشروع واحد كبير
- بيسهل إدارة المشاريع مثل : (Web API + Class Library + Console App)