**.NET Versions:**

**.NET Framework**: Introduced in 2002, the original .NET platform supports Windows development. It includes a large library and supports languages like C#, VB.NET, and F#. It reached version 4.8 before becoming largely static.

**.NET Core**: Launched in 2016 as a cross-platform, open-source framework designed to run on Windows, macOS, and Linux.

**.NET 5 and Beyond**: In 2020, Microsoft unified .NET Framework and .NET Core under ".NET 5." Subsequent versions (e.g., .NET 6 and .NET 7) added new features like improved performance, support for cloud-native development, and long-term support (LTS) options.

Even version: long-term support(3 years)

Odd version: short-term support(1.5 years)

**Mono/Xamarin**: Supports cross-platform mobile development and is integrated into the unified .NET platform.

**.NET Core:**

.NET Core, now part of the unified .NET platform, introduced modern development paradigms.

**Cross platform**: Supports Windows, Linux, and macOS.

**High Performance**: Optimized for modern hardware, with lower memory usage and higher throughput.

**Lightweight**: Modular design allows developers to include only necessary components(NPM,nuget.org).

**Open source (source browser)**: Developed as an open-source framework under the **MIT License** and **Apache 2.0 License**.  
Why Open Source?

**Transparency**: Developers can inspect how the framework works and report or fix issues.

**Customization**: Freedom to modify the framework for specific needs.

**Faster Evolution**: Open contributions lead to rapid innovation and adoption of modern practices.  
**Community Support**: Access to a global community for support, tutorials, and shared libraries.

**Namespaces in .NET:**

Namespaces are a fundamental feature of .NET, organizing and managing classes, interfaces, and other types to avoid naming conflicts.

**Solutions in .NET**

A **solution** in .NET is a container for organizing and managing related projects in Visual Studio.

**Solution File (.sln)**: Holds metadata about the projects it includes.

**Projects**: Each project can be an application, library, or test module.

**Configurations**: Supports multiple build configurations (e.g., Debug, Release).

Allows managing dependencies between projects.

Simplifies version control by organizing codebases in a single hierarchy.

Supports adding multiple languages within a single solution.