[ASP.NET CORE](https://dotnettutorials.net/lesson/net-core-vs-net-framework/)[[1]](#endnote-1)

[Middleware](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-8.0)[[2]](#endnote-2)

[Dependency Injection](https://dotnettutorials.net/lesson/asp-net-core-dependency-injection/)[[3]](#endnote-3)

[Static Files](https://dotnettutorials.net/lesson/static-files-middleware-asp-net-core/)

[wwwroot](https://dotnettutorials.net/lesson/wwwroot-folder-asp-net-core/)[[4]](#endnote-4)

Object Relational Mapping

[Dapper](https://www.learndapper.com/)[[5]](#endnote-5)

EF Core

Object Mapping

[AutoMapper](https://dotnettutorials.net/lesson/automapper-in-asp-net-core-web-api/)[[6]](#endnote-6)

App settings and Configuration

1. [.NET CORE](https://dotnettutorials.net/lesson/net-core-vs-net-framework/):

   .NET Core:

   Là phiên bản kế thừa từ .NET Framework

   Đa nền tảng

   Hiệu suất cao

   Mã nguồn mở

   Hỗ trợ microservice, cloud

   .NET Framework:

   Môi trường thực thi CLR

   Thư viện lớp phong phú (nuget package, …)

   Chỉ triển khai trên Windows

   Nhiều ngôn ngữ: C#, VB.NET, F#

   [CLR](https://comdy.vn/c-sharp/common-language-runtime-clr-trong-c-sharp/):

   Môi trường thực thi của .NET, biên dịch

   Thành phần:

   Security Manager

   JIT Compiler

   Memory Manager

   Garbage Collector

   Exception Manager

   Common Language Specification (CLS)

   Common Type System (CTS) [↑](#endnote-ref-1)
2. [Middleware](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-8.0):

   Middleware is software that's assembled(lắp ráp) into an app pipeline to handle requests and responses. Each component:

   Chooses whether to pass the request to the next component in the pipeline.

   Can perform work before and after the next component in the pipeline.

   Request delegates are configured using [Run](https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.builder.runextensions.run), [Map](https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.builder.mapextensions.map), and [Use](https://learn.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.builder.useextensions.use) extension methods

   A diagram of a customization process

   AI-generated content may be incorrect. [↑](#endnote-ref-2)
3. [Dependency Injection](https://dotnettutorials.net/lesson/asp-net-core-dependency-injection/):

   Manage object dependencies within our application

   Design pattern used to “connect”

   Injected dependencies (objects or services) into the class from outside, typically through constructor parameters. [↑](#endnote-ref-3)
4. [wwrwoot](https://dotnettutorials.net/lesson/wwwroot-folder-asp-net-core/) (wwwroot folder):

   The designated Web Root directory where static files are stored and served to clients.

   Located in the project root by default,

   Contains static assets like images, CSS, JavaScript, and other resources accessible to clients via direct URLs. [↑](#endnote-ref-4)
5. [Dapper](https://www.learndapper.com/):

   Micro-ORMs (Object-Relational Mapping)

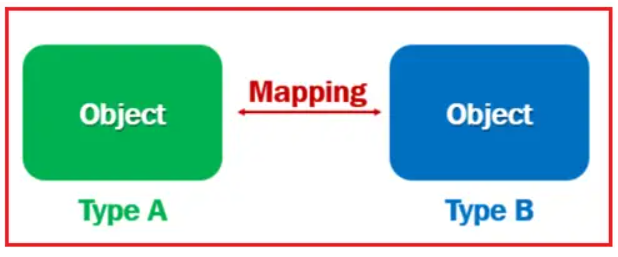
   Such as EF Core

   Speed and simple implementation

   A screenshot of a computer

   AI-generated content may be incorrect. [↑](#endnote-ref-5)
6. AutoMapper:

   AutoMapper is an object-to-object mapping library that automates this process

    [↑](#endnote-ref-6)