Documentation

1. Upload file in .NET Framework: <https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/sending-html-form-data-part-2https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/sending-html-form-data-part-2>
2. Rest API: <https://clacified.com/tech-science/16/rest-api-put-vs-post-requests-the-difference>
3. Unit test: <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices>
4. IoC: <https://viblo.asia/p/tim-hieu-unity-container-ioc-container-6J3ZgmOA5mB>
5. ASP Net Core : <https://github.com/dotnet/AspNetCore.Docs>
6. Using Autofac: <https://www.c-sharpcorner.com/UploadFile/dacca2/implement-dependency-injection-using-autofac-ioc-container/>
7. Repository pattern: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design#:~:text=of%20Work%20patterns.-,The%20Repository%20pattern,from%20the%20domain%20model%20layer>
8. UnitOfWork: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions/getting-started-with-ef-5-using-mvc-4/implementing-the-repository-and-unit-of-work-patterns-in-an-asp-net-mvc-application>
9. Design patterns: <https://refactoring.guru/design-patterns/csharp>
10. SOLD: <https://www.c-sharpcorner.com/UploadFile/damubetha/solid-principles-in-C-Sharp/#:~:text=SOLID%20design%20principles%20in%20C%23,Dependency%20Inversion%20Principle%20(DIP)>
11. Caching: https://www.c-sharpcorner.com/article/caching-mechanism-in-asp-net-core/
12. Integration test: <https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests?view=aspnetcore-6.0>
13. Implementing domain driven design: <https://docs.abp.io/en/abp/4.0/Domain-Driven-Design-Implementation-Guide>
14. UnitOfWork(MongoDB): <https://docs.abp.io/en/abp/4.0/Domain-Driven-Design-Implementation-Guide>
15. MicrosoftService(eShopOnContainer): <https://github.com/dotnet-architecture/eShopOnContainers>
16. Definition APIGateway: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/direct-client-to-microservice-communication-versus-the-api-gateway-pattern#:~:text=NET%20Core%20lightweight%20API%20Gateway,deploy%20on%20Linux%20or%20Windows>.
17. Definition Microsoft Services: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/architect-microservice-container-applications/direct-client-to-microservice-communication-versus-the-api-gateway-pattern#:~:text=NET%20Core%20lightweight%20API%20Gateway,deploy%20on%20Linux%20or%20Windows>.
18. Implement APIGateway with Ocelet: https://docs.microsoft.com/en-us/dotnet/architecture/microservices/multi-container-microservice-net-applications/implement-api-gateways-with-ocelot