Objective:

Learn how to design your web application using the Onion architecture. Learn how DI works and different types of lifetime.

Theory:

1. Dependency injection in ASP.NET Core | Microsoft Docs
2. Dependency injection in .NET | Microsoft Docs
3. Dependency injection guidelines | Microsoft Docs
4. https://martinfowler.com/articles/injection.html

Task:

Deadline: 3-4 days

Requirements:

- Split your brands controller logic into layers according to Onion architecture principles.
- Configure DI for your web application. Use default DI implementation provided by ASP.NET Core framework. Don't forget about the lifetime!

Here we achieve well-organized architecture where we can easily add new controllers, services, repositories, etc.. . So at this stage you will have a full-fledged web application that allows you to work with one of the entities of your domain.

Answer all of these questions to be prepared:

1. What is dependency injection? How would this help us when building web app?
2. What is Service Locator Pattern and what is the difference between this and DI? Why is service locator considered as an anti-pattern?

3. What is the lifetime in terms of DI? What are the options when using ASP.NET Core DI extension? Could you please provide examples when each lifetime might be useful?
4. What are other well-known DI libraries for .NET Core applications? What are their advantages and disadvantages compared to standard ASP.NET Core DI extension?