Objective:

Learn different approaches about how to design web applications. Learn common concepts about how to organize client-server interaction with HTTP. Learn best practices for building microservices.

Theory:

1. https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/architectural-principles
2. https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures (Ignore info about Docker)
3. https://restfulapi.net/
4. https://en.wikipedia.org/wiki/Representational_state_transfer
5. https://www.c-sharpcorner.com/article/microservice-using-asp-net-core/
6. https://microservices.io/patterns/microservices.html (very useful, save for future reference)
7. https://docs.microsoft.com/en-us/azure/architecture/microservices/ (save for future)
8. https://docs.microsoft.com/en-us/previous-versions/msp-n-p/dn568099(v=pandp.10) (save for future)
9. CQRS
10. https://habr.com/ru/post/427739/
11. https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice
12. https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/

Task:

Deadline: 3-4 days

Requirements:

- Learn about Onion, Layered and DDD service architectures.
- Learn about how REST principles help to build service API.
- Learn about CQRS usage in web applications.
- Learn about microservices.

In our project we will go with the Onion architecture but it’s important to understand different approaches and situations where you would like to use them.

Answer all of these questions to be prepared:

1. What is the point of splitting your code into different layers? How does dependency inversion help?
2. Please describe the main principles of N-layer architecture.
3. Please describe the main principles of Onion(Clean) architecture.
4. What are the main differences between N-layer and onion architectures?
5. Please describe the main principles of DDD. What are the main terms and concepts that are used in DDD?
6. How does CQRS help to build loosely coupled layers and applications?
7. What is REST API? What are the main principles behind this concept? How does this help to build robust web applications? What is the connection between REST and HTTP?
8. What is microservice architecture? What are the pros and cons compared to monolithic architecture?