Objective:

Learn how to compose different parts of your application and how to reuse existing logic.

Theory:

- https://www.scaleyourapp.com/what-is-swagger-and-why-do-you-need-it-for-your-project/
- https://swagger.io/docs/specification/about/
- https://docs.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-swashbuckle?view=aspnetcore-6.0&tabs=visual-studio
- https://learning.postman.com/docs/getting-started/introduction/
- https://apiguide.readthedocs.io/en/latest/build_and_publish/use_RESTful_urls.html

Task:

Deadline: 8-10 days

Requirements:

- Design and implement your e-store application as a set of controllers that are responsible for different business actions.
- Think a lot about what actions make sense for your domain, please don't make CRUDs for every entity even when it's not needed. Think about the real world actions and try to project them into the set of controllers and actions. Refer to stage 3 document for requirements.
- Application should be designed according to the Onion architecture principles, reuse logic where you can, validate the state of your application. Design endpoints according to REST principles.

- Document your APIs using Swagger (Swashbuckle library). Decorate all you endpoints with returned types and status codes. Take a look at *ProducesResponseType* attribute
- Use Postman for API testing

By the end of this stage we have a robust web application with good dependency management, separation of responsibilities and high extensibility.

Answer all of these questions to be prepared:

1. What is Postman? What are other alternatives to reach your API endpoints? Is it possible to test the API using a browser?
2. What is Swagger? How does documentation of our API help in development?