# Microservices Architecture Essentials: Exercises

Problems for exercises and homework for the [Architecture of ASP.NET Core Microservices Applications](https://softuni.bg/trainings/2999/architecture-of-asp-dot-net-core-microservices-applications-june-2020) course.

You should create a repository on GitHub, push your source code there, and link it here:

<https://softuni.bg/trainings/2999/architecture-of-asp-dot-net-core-microservices-applications-june-2020#lesson-16427>

## Server-Side Application

Create a web application of your choice by using ASP.NET Core 3.1. It can be anything you like – a blog, an online shop, an ad system, a social network, etc. You can use API + a JavaScript client or just plain MVC.

* The project should have authentication and authorization functionality.
* The project should have at least 5 database tables, including users.
* The project should have at least 10 endpoints.
* The project should have an administration. It may be a separate application.
* Do not overengineer your project’s architecture. Keep it as simple as possible. You may introduce a simple service layer, but it is not required.
* This exercise’s focus is on external architecture, so do not invest too much time making the perfect web app. Do it as an urgent freelance project – working, but without top code quality. 😊

## Client-Side Application

If you know a front-end JavaScript framework (Angular, React, Vue, etc.), it is strongly advised to use a Web API + a front-end JS client. It is the preferred microservices approach in terms of performance and scalability. Keep the front-end application in a separate project and process. If you do not know a front-end framework, just create your project as a single ASP.NET Core MVC server. You are going to use the MVC framework as a presentation layer in front of the microservices. This is a perfectly fine and viable production-ready solution, but it is a bit slower than the other approach in terms of throughput.