**Common**

You should create a site, which enable to work with Northwind Database ([script with DB structure and test data is attached](https://epam.sharepoint.com/sites/ASPNetCore/Shared%20Documents/Materials/instnwnd.zip?csf=1&web=1))

Base requirements:

* Site should be developed using ASP.Net Core 2+
* Server side should be worked on Microsoft Windows (cross platform is not required)
* MS SQL Server is used as DBMS (you can use any edition, including LocalDB)
* EF Core is used for data access
* If there no special instructions, no requirements apply to the layout and styling of the pages

**Task 1. Base site**

Create site with follow pages:

* Home page, contained welcome message and links to other pages
* Categories page. Show list of categories without images
* Products page. Show table with products
  + Table contains all products fields
  + Instead of references to Category and Supplier should be showed their names

**Note:**

* All configuring parameters (connection strings, …, etc) can keep in code (hardcoded)

**Materials:**

* [Create a web app with ASP.NET Core MVC using Visual Studio](https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/)

**Task 2. Startup and configuration**

Add configuration feature with support 2 parameters:

* Database connection string
* Maximum (M) products count on Product page (show only first M products, others – ignored; if M == 0 show all products)

**Materials:**

* [Application startup in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/startup)
* [Configure an ASP.NET Core App](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/configuration)

**Task 3. Edit forms and Server-side validation**

Add edit forms (New and Update) for products:

* Related entities (such as category) should be presented as dropdown list

Add server-side validation for edited products (not less than 3 different rules)

**Materials:**

* [Introduction to model validation in ASP.NET Core MVC](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation)

**Task 4. Styling and client-side validation**

Add to project 2 client libraries:

* Bootstrap
* jQuery Unobtrusive Validation

For Bootstrap:

* Apply Bootstrap styles to site pages/forms
* Change links to categories and product pages to navigation bar with "hamburger" button

For jQuery Unobtrusive Validation:

* Configure client-side validation by analogy with task 3.

**Materials:**

* [Building beautiful, responsive sites with Bootstrap and ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/client-side/bootstrap)
* [Client-side validation](https://docs.microsoft.com/en-us/aspnet/core/mvc/models/validation#client-side-validation)

**Task 5. Logging and error handling**

Configure logging:

* configure writing logs into logging file
* add writing follow events and information
  + start application (Additional information: application location - folder path)
  + read configuration (Additional information: current configuration values)

Create custom error handler, which:

* logs exception
* return error page with correlation information (for search appropriate records in logs)

**Note**. For get exception in error handler you can use code:

   var error = this.HttpContext.Features.Get<IExceptionHandlerFeature>().Error;

**Materials:**

* [Introduction to Error Handling in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/error-handling)
* [Introduction to logging in ASP.NET Core](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/logging)