# Shop Web API summery practice

1. Create ShopDbContext class that consists of Users, Orders, Products data.
   1. apply data annotations for model input validations.

\*[optional] put the ShopDbContext class in a class library project

1. Create ASP.NET Core REST API project that wraps the uses ShopDbContext class library and allows Web API access to the data.
2. Implement CRUD API for Users, Orders, Products in separate controllers, for each controller:
   1. Implement HttpPost method to enable Create operation, receive JSON data from request body
      1. \*when adding new resource return created status code with the location of the resource
      2. \*When trying to Add an existing resource return bad request status code
   2. Implement HttpPut method to enable Update operation, receive JSON data from request body
   3. Implement HttpDelete method to enable Delete operation, receive data from request route
   4. Impelment HttpGet method to enable Read operation, received data from request route
3. [optional] Log any controller action method access to the debug output inside the controller action
4. Implement general exception middleware handler that returns its message in a JSON format only if in Development environment otherwise return "internal server error"
5. Take the connection string value from appsettings.json connectionStrings key and use it on ShopDbContext.OnConfiguring method.
6. [optional - replaces section 4] Log any controllers actions access in the debug output using filters only when in development environment
7. Enable Swagger/OpenAPI interface by installing Swashbuckle.AspNetCore package and configuring it according its [GitHub documentation](https://github.com/domaindrivendev/Swashbuckle.AspNetCore)
8. [optional] Create Shop\_ConsoleClient that uses the Shop\_WebAPI and displays all products in its console. You can use [NSwagStudio · RicoSuter/NSwag Wiki (github.com)](https://github.com/RicoSuter/NSwag/wiki/NSwagStudio) for generating client based on the swagger API