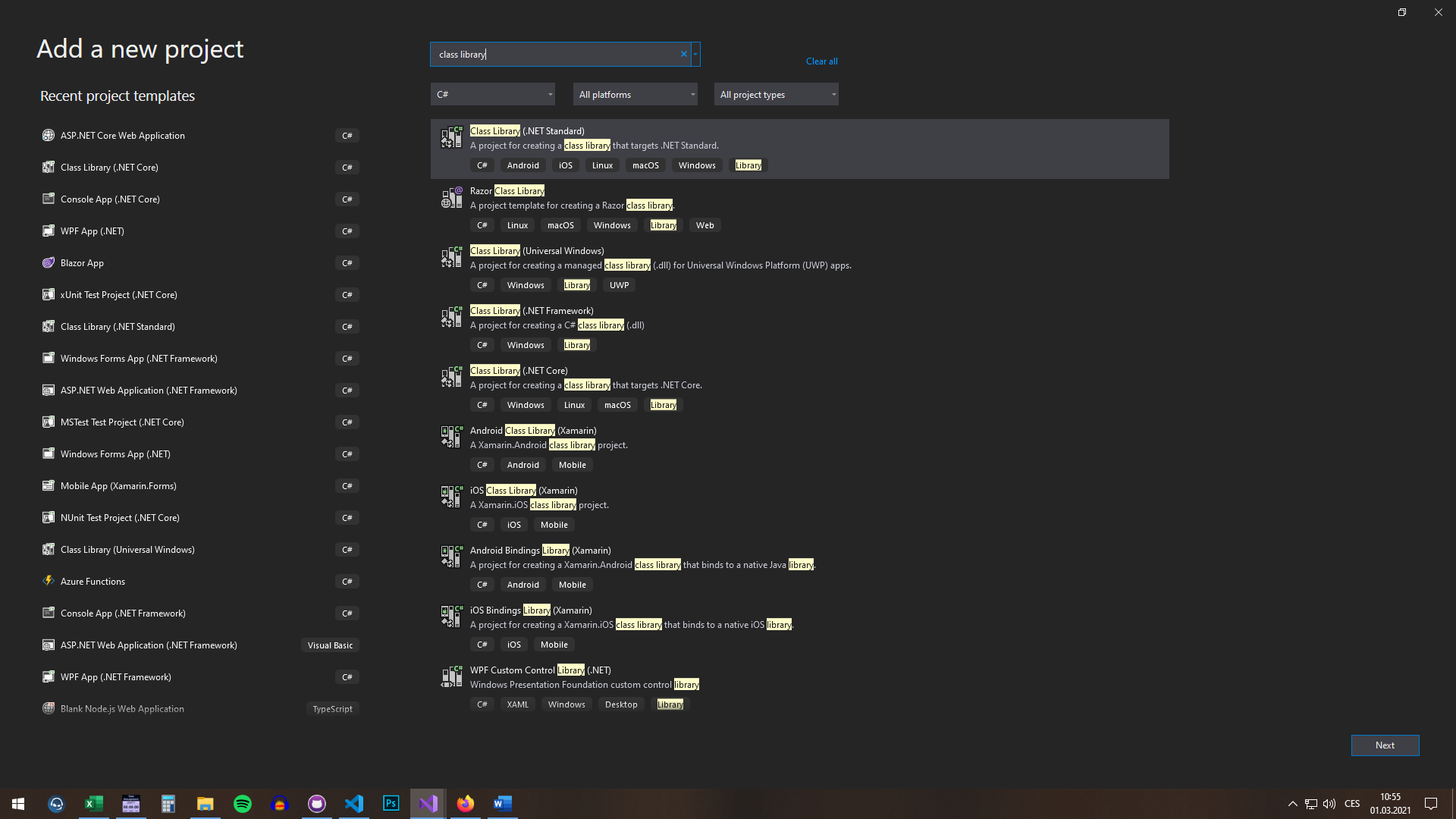
E-shop – ITnetwork

# I. lekce

1. Na začátek kurzu by bylo dobré dát doporučení, že takto rozsáhlý projekt je dobré verzovat po lekcích.
2. U přidávání nových projektů - Class Library - je starý obrázek a není jasné, kterou z mnoha knihoven vybrat (řeší se v komentářích, ale o to větší důvod zapracovat to do lekce).



1. Možná zkratky DTO a DAO vypsat do závorek, aby člověk nemusel googlit...

*– „předpokládá se masivní googlení studenta“*

# II. lekce

1. **Instalace balíčků NuGet**

U zakládání projektu jsem si zvolila verzi 3.1 (nevím jak ovlivnit, aby to bylo 3.1.7). Když se dívám na své automaticky nainstalované balíčky, tak mají všechny verzi 3.1.10.

Předpokládám, že i další NuGety bych měla mít ve verzi 3.1.10 – ale bod 3. u Instalace balíčků NuGet říká: „vybereme verzi balíčku 3.1.7“ a další odstavec k tomu také nabádá...

Můj **Eshop.Data** vypadá takhle – je to dobře nebo ne?

<ItemGroup>

<PackageReference Include="Microsoft.AspNetCore.Identity" Version="2.2.0" />

<PackageReference Include="Microsoft.AspNetCore.Identity.EntityFrameworkCore" Version="3.1.10" />

<PackageReference Include="Microsoft.AspNetCore.Mvc" Version="2.2.0" />

<PackageReference Include="Microsoft.EntityFrameworkCore.Proxies" Version="3.1.10" />

<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="3.1.10" />

<PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="3.1.10" />

</ItemGroup>

* *V novějším Frameworku se automaticky instalují balíčky ve verzi 3.1.10 což dělá problémy*

1. **U jednoho balíčku se přidávají různé verze** - když už existuje verze 3.1.7, tak proč dávat do jednoho projektu nižší verzi 2.2.0? „Microsoft.AspNetCore.Identity.EntityFrameworkCore, do projektu Eshop.Data ve verzi 2.2.0 a do projektu Eshop ve verzi 3.1.7“

* nevadí, i když si nasinstalujeme 2.2.0 Visual Studio si verzi samo předělá na 3.1.7

1. **HomeController**

„Do kontroleru HomeController vložíme následující kód: ... “

Vygenerovaný HomeController už tenhle kód obsahuje...

* Detail

1. **U obrázku pro kontrolu změn vyplavou 3 problémy:**

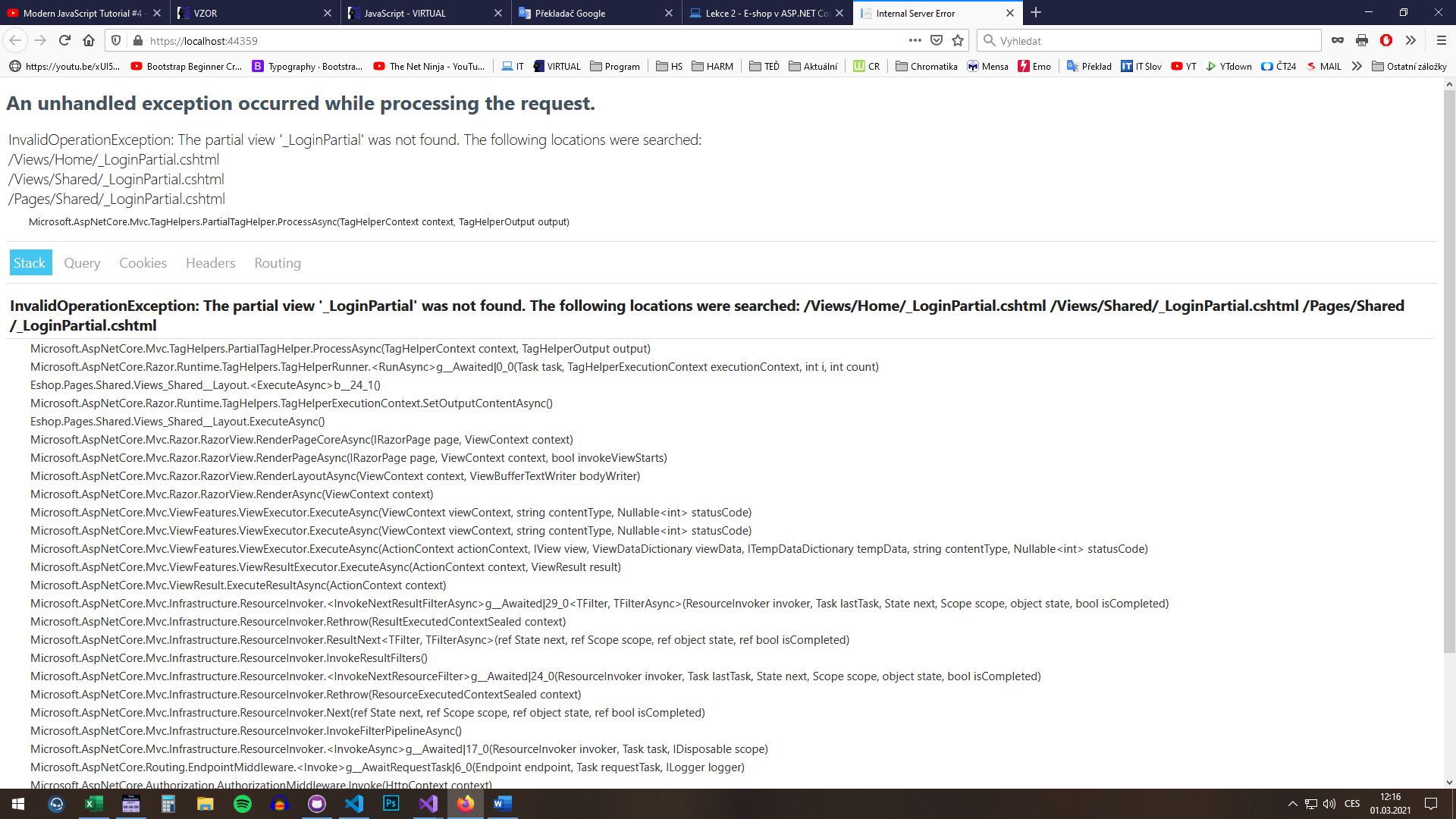
Složka Data je v této fázi pod Eshopem, ne v Eshop.Data. Že se má složka přetáhnout je až dál v textu.

V Eshop.Data chybí v této fázi Models a ApplicationUser.cs – informace o přidání této složky a třídy je až dál v textu.

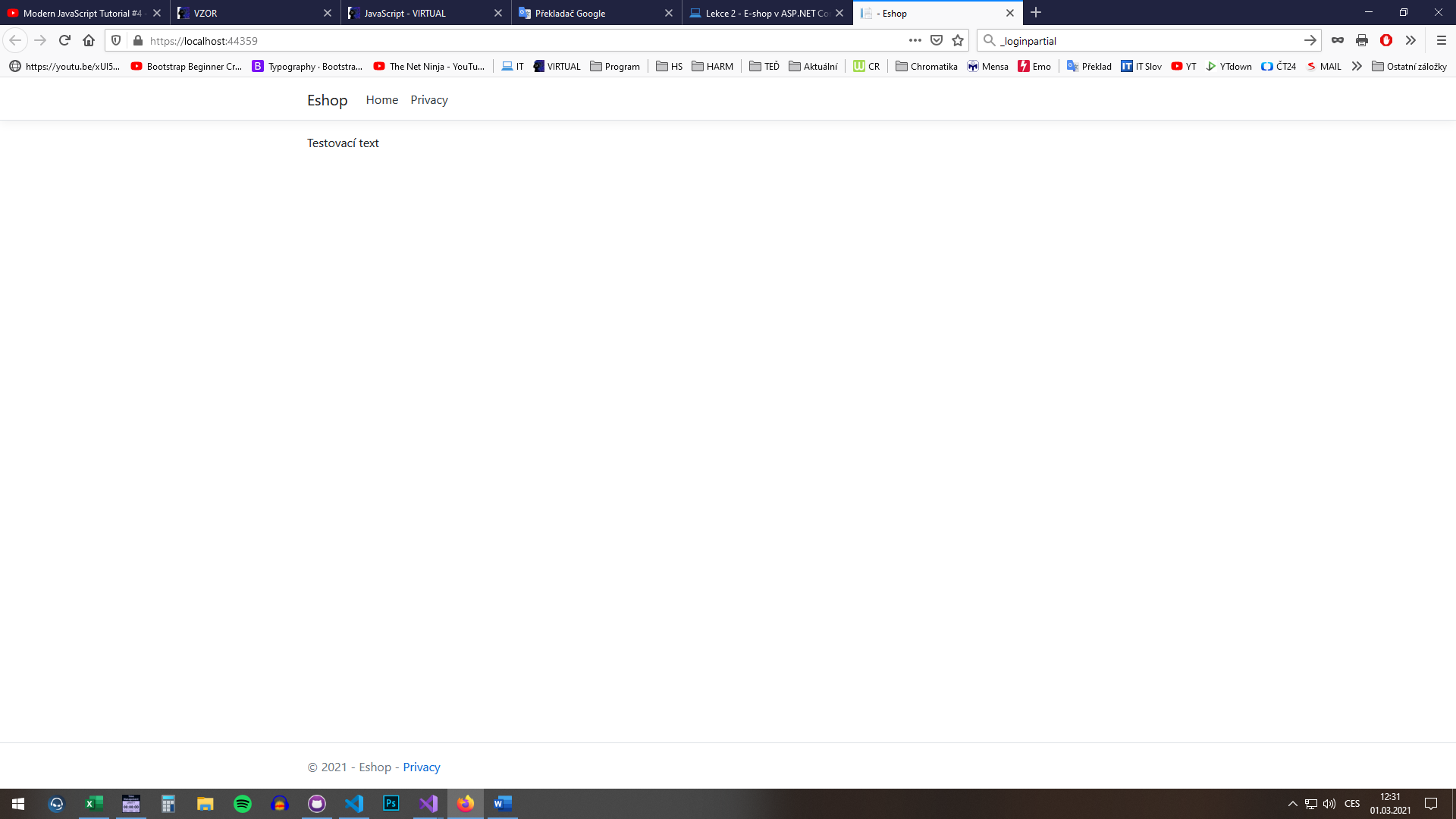
Automaticky vygenerované třídy: Class1.cs z Eshop.Business a Eshop.Data se asi mají smazat? Na obrázku nejsou.

* *mělo by se řešit*

1. Při pokusu o zobrazení testovacího textu se zobrazí error, protože \_Layout odkazuje na \_LoginPartial, který jsme smazali...

Když odkaz na \_LoginPartial v \_Layoutu zakomentuji: @\*<partial name="\_LoginPartial" />\*@

Dostanu jiný výsledek než je v tutoriálu:



* *mělo by se řešit*

1. **Migrace Identity**

Skončila errorem...

PM> add-migration identity

Build started...

Build succeeded.

An error occurred while accessing the Microsoft.Extensions.Hosting services. Continuing without the application service provider. Error: Some services are not able to be constructed (Error while validating the service descriptor 'ServiceType: Microsoft.AspNetCore.Identity.ISecurityStampValidator Lifetime: Scoped ImplementationType: Microsoft.AspNetCore.Identity.SecurityStampValidator`1[Eshop.Data.Models.ApplicationUser]': Unable to resolve service for type 'Microsoft.AspNetCore.Identity.IUserStore`1[Eshop.Data.Models.ApplicationUser]' while attempting to activate 'Microsoft.AspNetCore.Identity.UserManager`1[Eshop.Data.Models.ApplicationUser]'.) (Error while validating the service descriptor 'ServiceType: Microsoft.AspNetCore.Identity.ITwoFactorSecurityStampValidator Lifetime: Scoped ImplementationType: Microsoft.AspNetCore.Identity.TwoFactorSecurityStampValidator`1[Eshop.Data.Models.ApplicationUser]': Unable to resolve service for type 'Microsoft.AspNetCore.Identity.IUserStore`1[Eshop.Data.Models.ApplicationUser]' while attempting to activate 'Microsoft.AspNetCore.Identity.UserManager`1[Eshop.Data.Models.ApplicationUser]'.) (Error while validating the service descriptor 'ServiceType: Microsoft.AspNetCore.Identity.IUserClaimsPrincipalFactory`1[Eshop.Data.Models.ApplicationUser] Lifetime: Scoped ImplementationType: Microsoft.AspNetCore.Identity.UserClaimsPrincipalFactory`2[Eshop.Data.Models.ApplicationUser,Microsoft.AspNetCore.Identity.IdentityRole]': Unable to resolve service for type 'Microsoft.AspNetCore.Identity.IUserStore`1[Eshop.Data.Models.ApplicationUser]' while attempting to activate 'Microsoft.AspNetCore.Identity.UserManager`1[Eshop.Data.Models.ApplicationUser]'.) (Error while validating the service descriptor 'ServiceType: Microsoft.AspNetCore.Identity.UserManager`1[Eshop.Data.Models.ApplicationUser] Lifetime: Scoped ImplementationType: Microsoft.AspNetCore.Identity.UserManager`1[Eshop.Data.Models.ApplicationUser]': Unable to resolve service for type 'Microsoft.AspNetCore.Identity.IUserStore`1[Eshop.Data.Models.ApplicationUser]' while attempting to activate 'Microsoft.AspNetCore.Identity.UserManager`1[Eshop.Data.Models.ApplicationUser]'.) (Error while validating the service descriptor 'ServiceType: Microsoft.AspNetCore.Identity.SignInManager`1[Eshop.Data.Models.ApplicationUser] Lifetime: Scoped ImplementationType: Microsoft.AspNetCore.Identity.SignInManager`1[Eshop.Data.Models.ApplicationUser]': Unable to resolve service for type 'Microsoft.AspNetCore.Identity.IUserStore`1[Eshop.Data.Models.ApplicationUser]' while attempting to activate 'Microsoft.AspNetCore.Identity.UserManager`1[Eshop.Data.Models.ApplicationUser]'.) (Error while validating the service descriptor 'ServiceType: Microsoft.AspNetCore.Identity.RoleManager`1[Microsoft.AspNetCore.Identity.IdentityRole] Lifetime: Scoped ImplementationType: Microsoft.AspNetCore.Identity.RoleManager`1[Microsoft.AspNetCore.Identity.IdentityRole]': Unable to resolve service for type 'Microsoft.AspNetCore.Identity.IRoleStore`1[Microsoft.AspNetCore.Identity.IdentityRole]' while attempting to activate 'Microsoft.AspNetCore.Identity.RoleManager`1[Microsoft.AspNetCore.Identity.IdentityRole]'.)

Unable to create an object of type 'ApplicationDbContext'. For the different patterns supported at design time, see https://go.microsoft.com/fwlink/?linkid=851728

PM>

* ***vyřešeno*** *– způsobeno jinou verzí balíčků; při použití projektu z 1. lekce - migrace OK*

1. **Dokončení konfigurace**

Upozornění na potřebu přidat using na datovou vrstvu ve Startup.cs mělo být o několik odstavců výš u **Nastavení Identity** – tady už nemá moc význam.

# III. lekce

*V pohodě.*

# IV. lekce

1. **Kontrola databáze**

V lekci se dělá několik migrací a u žádné není připomenuto update-database. Dala bych připomenutí aktualizovat databázi alespoň do lekce „Kontrola databáze“ – jinak se budou žáci divit, že tam žádný dbo.CategoryProduct nemají...

1. **Obrázek** v odstavci Kontrola databáze by měl ukazovat víc ze stromu – zahrnovat i „aspnet-Eshop-...“ Takhle není zřejmé v jaké nadřazené složce se složka Keys může schovávat. (Hlavně pro ty, co neupdatovali databázi a marně hledají i dbo.CategoryProduct...)
2. Na konci lekce chybí vzorový projekt ke stažení (projekt v této fázi)

# V. lekce

1. **Pohledy - Login**

„V dialogu zvolíme Razor View - Empty a přidáme View s názvem Login:“

Následující obrázek odpovídá volbě „Razor View“ – ne „Razor View – Empty“.

(Visual Studio 2019). Buď bych upravila text nebo obrázek nedávala.

1. **Pohled \_Layout**

„upravíme odkaz <a>:...“

V téhle podobě už ho máme (překopírovaný z dřívějška)...

1. **Testování**

„4. Po úspěšné registraci budeme přesunuti zpět na úvodní stránku.“

Po úspěšné registraci nejsme přesunuti na úvodní stránku, ale do administrace účtu.

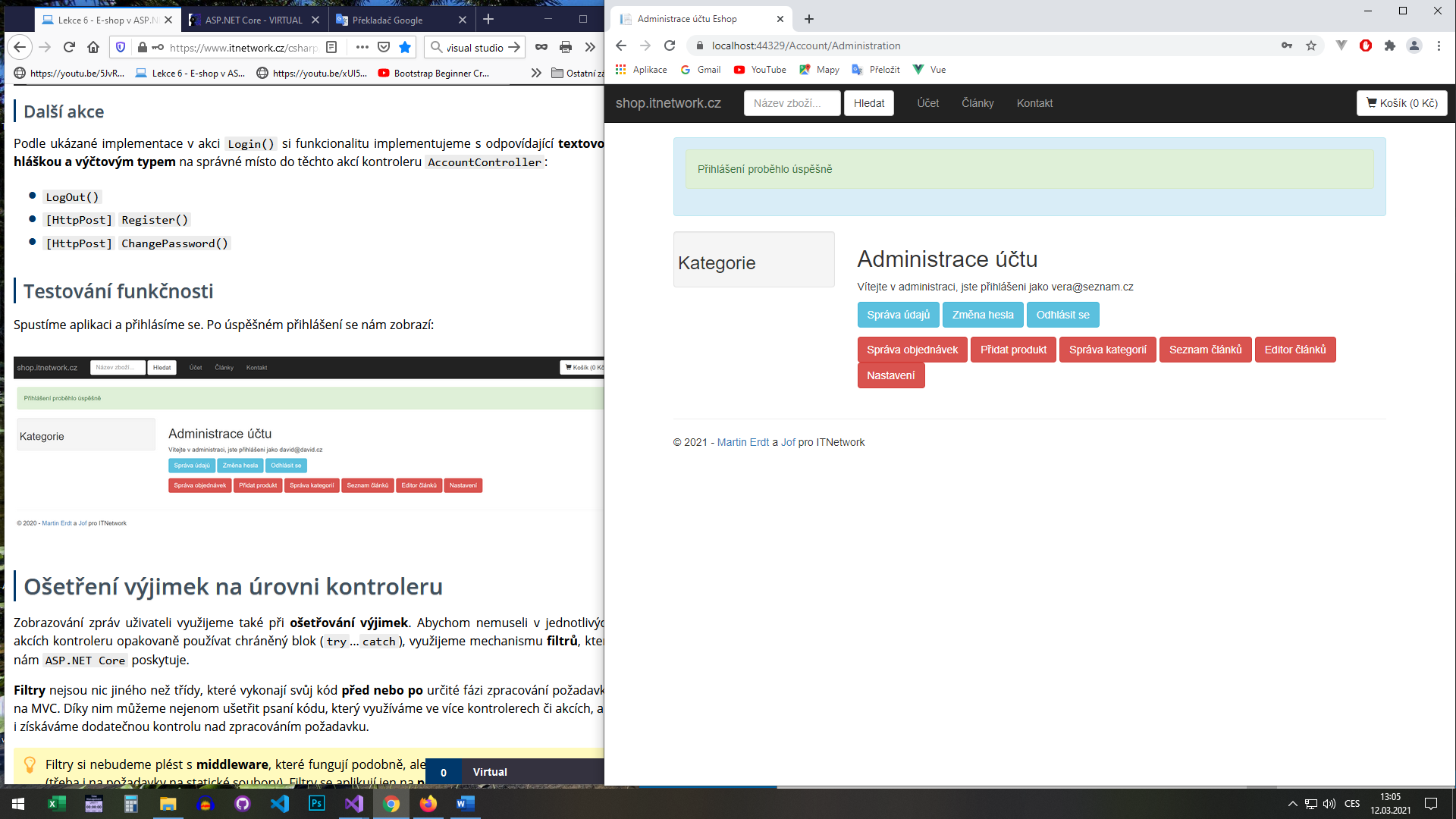
# VI. lekce

1. **Další akce**

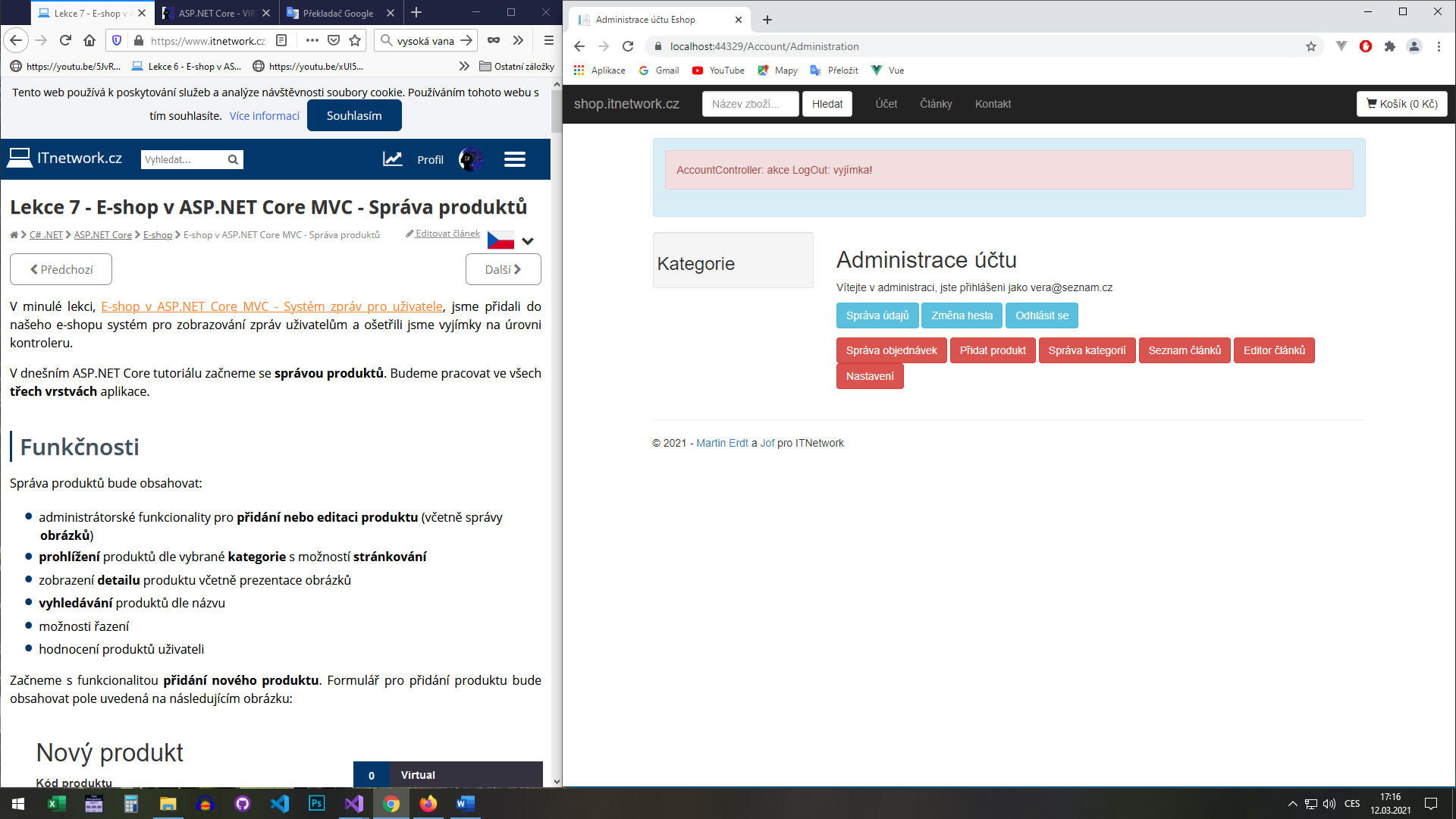
Samostatné doplnění funkcionality je moc fajn, ale ti co tápou, by ocenili pod článkem alespoň jpg s PrintScreenem – s řešením.

1. **Testování funkčnosti**

Success hlášky i výjimky se zobrazují na modré „alert-info“ ploše – na rozdíl od obou ilustračních obrázků v lekci.



(„vyjímka“ – by se správně měla psát „výjimka“)



# VII. lekce

Luxusní, poučné, snadné následovat – není co zlepšit.

# VIII. lekce

1. **Pohled**

první odrážka:

* změňme model na náš připravený viewmodel - @model Eshop.Models.ProductViewModels.ManageProductViewModel

Přidala bych ujištění:

... „Chybové hlášky, které díky této změně vyskočí, v následujícíh krocích opravíme.“

1. **Pohled**

4. odrážka:

Přidala bych TIP: „Abychom nemuseli vypisovat Product osmadvacetkrát, využijeme **multikurzor**. Se stisknutám Ctrl + Alt umístíme kurzory před červeně podtržené vlastnosti a „Product.“ vypíšeme jen jednou.

1. **Pohled**

5. odrážka: naprosto nejasné zadání:

„odstraňme z formuláře pole, která zde nechceme editovat (Hidden), případně je nahraďme skrytými poli (<input ... type="hidden" />), aby se nám při editaci produktu neztratily uložené hodnoty (Stock, ImagesCount)“

V projektu jsou v tuhle chvíli „hidden“ na dvou místech:

Product Id – to by mělo zůstat, ale někdo si může vyložit, že ho má odstranit

a

<div class="form-group form-check">

<label class="form-check-label">

<input class="form-check-input" asp-for="Hidden" /> @Html.DisplayNameFor(model => model.Hidden)

</label>

</div>

Tohle se patrně má smazat...

V závorce zmíněný ImagesCount v „Kompletním kódu pohledu“ nemá label a je hidden...

ale zmiňovaný Stock je beze změny...

Navíc se v „Kompletním kódu pohledu“ najednou objeví nikde nezmiňovaný:

<div class="form-group">

<div class="checkbox">

<label>

<input asp-for="Product.Hidden" type="hidden" />

</label>

</div>

</div>

1. **Pohled**

6. odrážka:

<textarea asp-\_for="Product.ShortDescription" class="form-control"></textarea>

Nadbytečná mezera - kdo si kód překopíruje, musí si ho opravit...

1. **Kompletní kód pohledu**

Dost se liší od toho, co člověk samostatně podle manuálu vytvořil:

* kód v divu s id="product-images-administration" vůbec nebyl zmíněn
* Sekce pro přidávání obrázků byla popsána jako přidejte si vstupní pole:

<input asp-for="UploadedImages" class="form-control" \*\*multiple\*\*/> *(ty hvězdičky tam asi být nemají)*

Bylo by přesnější - přidejte si novou div sekci pro přidávání obrázků:

<div class="form-group">

<label asp-for="UploadedImages" class="control-label"></label>

<input asp-for="UploadedImages" class="form-control" multiple />

<span asp-validation-for="UploadedImages" class="text-danger"></span>

</div>

* + chaos s „hidden“ (viz výše) a nikde nezmíněný:

<div class="form-group">

<div class="checkbox">

<label>

<input asp-for="Product.Hidden" type="hidden" />

</label>

</div>

</div>

* Jiný typ tlačítka – vygenerovaný typ je btn-primary, ve vzorovém kódu je btn-default.

1. **Manažer**

Metodu už v manažeru máme ze 7. lekce, kapitoly SaveProduct()

1. **Doplnění inicializace kolekcí**

Tyto kódy už v Product a Category jsou – kromě jednoho řádku u Product.cs:

ChildCategories = new List<Category>();

Po doplnění do Product.cs vznikne chybová hláška, protože ještě chybí přidat:

public virtual ICollection<Category> ChildCategories { get; set; }

1. **Testování**

Po přihlášení jako administrátor a klepnutí na Přidat produkt se zobrazí chybová hláška:

Tato stránka nefunguje

Pokud problém přetrvává, kontaktujte vlastníka webu.

HTTP ERROR 405

Příčinu jsem nenašla = nemůžu otestovat vkládání produktů.

1. **CleanProduct**

if (!string.IsNullOrEmpty(url))

model.PostedCategories = productManager.FindAssignedCategoriesToProduct(

model.AvailableCategories, model.Product.CategoryProducts.ToList(), model.PostedCategories);

Vložení posledního kódu vyvolá 2 chybové hlášky. Metoda, kterou v ProductControlleru máme nepředává v parametru url, ale model... (jinou metodu tam zatím nemáme)

public IActionResult ManageProduct(ManageProductViewModel model)

{...

# IX. lekce

1. **Manažery**

...„Rovnou si připravíme i metodu...“ – text dělá dojem, že si máme obě metody napsat sami, na základě uvedených pokynů.

Doplnila bych něco jako: „(Kód obou metod bude uveden dále v textu.)“

1. **ImageManager**

Třída dědí z interfacu IImageManager, který ještě nemáme... = zůstává tam error.

Dtto **ProductManager –** **Konstanty**

kód také odkazuje na nevytvořený IImageManager = druhý error v projektu.

1. **GetImageFileName()**

Jméno konstanty (v množném čísle) nesedí se jménem používaným v metodě (jednotné číslo) => vyvolá 3x error

Kód uvedený u Konstant:

private const string ProductImagesPath = "wwwroot/images/products/";

Kód uvedený u metody GetImageFileName()

private string GetImageFileName(int productId, int imageIndex, bool full = true)

{

if (!Directory.Exists($"{ProductImagePath}{productId}"))

Directory.CreateDirectory(ProductImagePath + productId);

string result = $"{productId}/{productId}\_{imageIndex}";

if (full)

result = ProductImagePath + result + ".jpeg";

return result;

}

Kód u konstant je potřeba upravit na jednotné číslo! A to i na dalším řádku:

private const string ProductImagePath = "wwwroot/images/products/";

private IImageManager imageManager = new ImageManager(ProductImagePath);

1. **RemoveProductImage()**

imageManager.ResizeImage(thumbnailFileName, ProductThumbnailSize);

„...does not exist in the current context“

# X. lekce