

**Gymnázium, Praha 6, Arabská 14**  
Programování  
**ROČNÍKOVÁ PRÁCE**



Vytvořil: Nguyen Tien Dung

**Gymnázium, Praha 6, Arabská 14**  
**Arabská 14, Praha 6, 160 00**  
**ROČNÍKOVÁ PRÁCE**

**Předmět:** Programování  
**Téma:** Produktový katalog  
**Autor:** Nguyen Tien Dung  
**Třída:** 4.E  
**Školní rok:** 2019/2020  
**Třídní učitel:** Mgr. Jana Urzová

## **Prohlášení**

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne .....

## **Anotace**

Ve své ročníkové práci jsem vytvořil katalog produktů. Jedná se o všeobecný katalog dostupný přes web. Podporuje všechny známé prohlížeče jako Google Chrome, Microsoft Edge, Mozilla Firefox Browser. Jako vývojové prostředí jsem používal Visual Studio 2019 Enterprise verzi. Aplikace je naprogramována v C#. Využil jsem ASP.NET MVC a .NET Framework 4.7.2 Jako databázi jsem využil Microsoft SQL Azure. Aplikace je implementována cloudu - Microsoft Azure.

## **Annotation**

In my term work I worked on a web application. To do that I used Visual Studio 2019 IDE, framework ASP.NET MVC and programming language C#. To work with data I chose Microsoft SQL Server.

## Obsah

1	Úvod .....	6
2	Návrh .....	7
2.1	Funkční požadavky .....	7
2.2	Nefunkční požadavky .....	7
2.3	Design databáze .....	7
3	MVC .....	8
3.1	Modely .....	8
3.2	View .....	10
3.2.1	Razor .....	10
3.3	Controller.....	10
4	CRUD .....	10
5	Dapper ORM.....	11
5.1	Konkrétní použití Dapper ORM .....	11
6	Nahrávání obrázku .....	12
7	Přihlašovací systém .....	13
8	Microsoft Azure .....	13
9	Závěr.....	13
10	Seznam použitých zdrojů.....	13

## 1 Úvod

Cílem mé ročníkové práce je vytvořit obecný katalog pro web, kde uživatel (návštěvník) může prohlížet produkty vyhledávat v produktech dle klíčů, porovnávat produkty. Produkty budou mít popis v textové podobě a taky bude možné umístit obrázky či videa k produktům. U produktů bude kromě popisu i cena a další odkazy například na web výrobce. Uživatele v roli administrátora bude moci katalog spravovat tj. přidávat, opravovat a zrušit produkty. Produkty budou rozděleny do různých kategorií, a podle kategorie bude moci administrátor zadat povinné parametry produktu. Produkty stejné kategorie se budou moci porovnávat mezi sebou podle parametrů. Role administrátora bude chráněna přihlášením. Uživateli také bude umožněno používat search engine, který bude vyhledávat podle rozsahu ceny, názvu nebo parametru.

## **2 NÁVRH**

Návrh aplikace je vytvořen na základě těchto požadavků. Požadavky dělíme do dvou skupin funkční a nefunkční.

### **2.1 FUNKČNÍ POŽADAVKY**

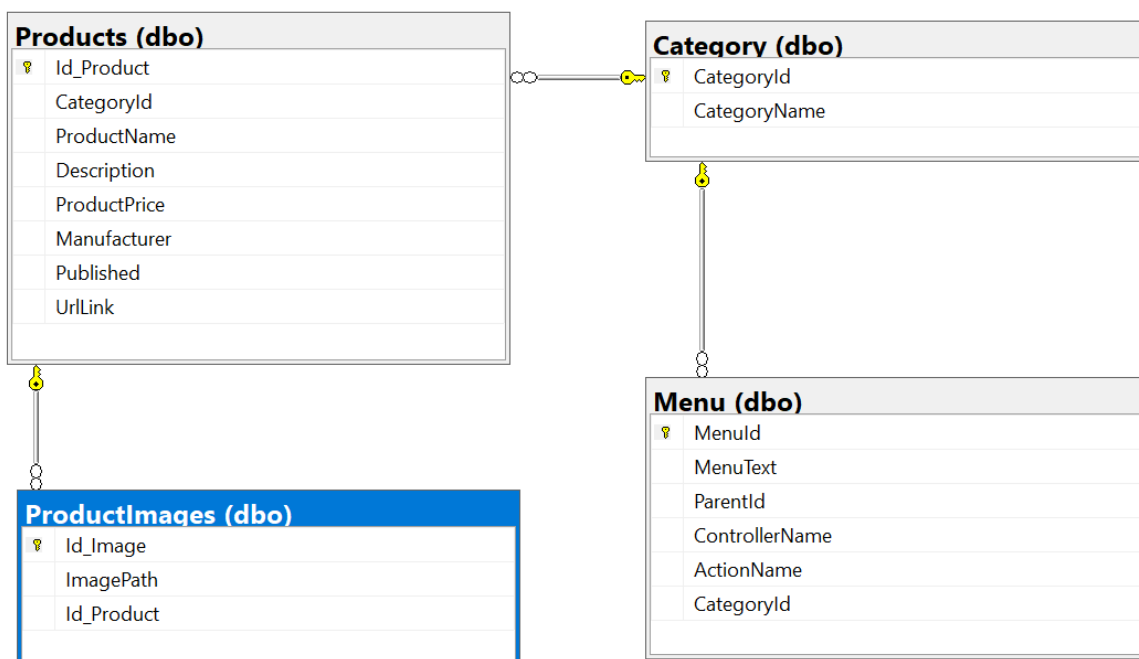
- Produkty jsou charakterizovány vlastnostmi, kategorií a cenou
- K produktu je možno připojit obrázek max. velikost 5 MB
- Úvodní stránka má obsahovat nové produkty
- Aplikace musí umožnit navigaci mezi kategoriemi
- Aplikace musí umožnit vyhledávání produktů dle názvů nebo části názvu
- Aplikace umožňuje přidávání a editaci produktů i kategorií
- Zadávaní a editace produktů má být umožněno jen uživatelům v roli administrátora
- Návštěvníkovi katalogu je umožněno jen prohlížet, vyhledávat a porovnávat produkty
- Uživatel může do seznamu na porovnání přidat produkty nejvýše tři

### **2.2 NEFUNKČNÍ POŽADAVKY**

- Aplikace bude hostována v Microsoft Azure
- K tvorbě projektu bylo použito vývojové prostředí Visual Studio 2019
- Bude se používat programovací jazyk C#
- Framework ASP.NET MVC
- Podporované prohlížeče jsou Google Chrome, Microsoft Edge, Mozilla Firefox Browser
- Databáze-SQL Server 2019

### **2.3 DESIGN DATABÁZE**

První důležitý krok je navrhnout databázi, podle níž se řídí struktura, podle struktury model a podle modelu se pak píšou metody. V prvním návrhu databáze obsahovala pouze 2 tabulky-tabulku produktu a tabulku obrázku. Návrh se vyvíjel podle aktuálního úkolu. Finální design jde vidět na obrázku číslo 1.



Obrázek 1

### 3 MVC

MVC nebo také model-view-controller je designový návrh běžný pro vytváření uživatelských rozhraní, které dělí logiku programu na 3 propojené části. MVC se využívá k rozdělení vnitřní reprezentace informace od vnější. Oddělit jak je informace prezentována uživateli a jak je informace zpracována programem.

#### 3.1 MODEL

Modely reprezentují strukturu dat. V této práci to jsou C# třídy. Zde to jsou dvojího typu. První typ, který uchovává data z back-endu, druhý typ, který uchovává data z front-endu. Na obrázcích jde vidět model produktu. Na prvním obrázku je model produktu pro front-end a na druhém pro back-end. Obrázek s číslem 2 a 3 slouží jako ukázka rozdílu mezi front-end a back-end modelem.



```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.Linq;
5  using System.Web;
6  using System.Web.Mvc;
7
8  namespace CatalogOfProducts2.Models
9  {
10     Počet odkazů: 33
11     public class ProductModel
12     {
13         Počet odkazů: 9
14         public int? ProductId { get; set; }
15
16         [Display(Name = "Product Name")]
17         [Required(ErrorMessage = "Product name required.")]
18         Počet odkazů: 14
19         public string ProductName { get; set; }
20
21         [Display(Name = "Description")]
22         [Required(ErrorMessage = "Description required.")]
23         Počet odkazů: 8
24         public string Description { get; set; }
25
26         [Display(Name = "Category")]
27         [Required(ErrorMessage = "Category required.")]
28         Počet odkazů: 10
29         public string Category { get; set; }
30     }
31 }

```

Obrázek 2

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace DataLibrary.Models
8  {
9     Počet odkazů: 12
10     public class ProductModel
11     {
12         Počet odkazů: 7
13         public int? ProductId { get; set; }
14         Počet odkazů: 9
15         public string ProductName { get; set; }
16         Počet odkazů: 8
17         public string Description { get; set; }
18         Počet odkazů: 7
19         public int CategoryId { get; set; }
20         Počet odkazů: 8
21         public int ProductPrice { get; set; }
22         Počet odkazů: 1
23         public DateTime Date { get; set; }
24         Počet odkazů: 3
25         public string UrlLink { get; set; }
26     }
27 }

```

Obrázek 3

## 3.2 VIEW

View je v MVC uživatelské rozhraní. Zobrazuje data modelu uživatelům a umožňuje jim je upravovat. View ve frameworku ASP.NET MVC je HTML, CSS a speciální syntax Razor, který usnadňuje komunikaci modelu s controllerem.

### 3.2.1 Razor

Razor umožňuje pomocí znaku „@“ a složených závorek psát C# kód do html souborů. Na obrázku číslo 4 můžete vidět názornou ukázkou použití Razor syntaxu ke generování tabulky do které jsou vloženy data modelu.

```
@foreach (var item in Model)
{
    <tr>
    <td>
        <a href="@Url.Content("~/Images/"+item.ImagePath)">
            
        </a>
    </td>
    <td>
        @Html.ActionLink(item.ProductName, "ProductDetails", "Home", new { id = item.ProductId }, new { })
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Category)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.ProductPrice) Kč
    </td>
    @if (Request.IsAuthenticated)
    {
        <td>
            @Html.ActionLink("Edit", "Edit", new { id = item.ProductId })
        </td>
        <td>
            @Html.ActionLink("Delete", "Delete", new { id = item.ProductId })
        </td>
    }
}
```

Obrázek 4

## 3.3 CONTROLLER

Obecně controller zpracovává požadavky uživatele. Ve frameworku ASP.NET MVC je to sada mnoha tříd, která ovládá jak se aplikace bude chovat. V různých třídách jsou různé metody, které zpracovávají data buď z databáze nebo z front-end-inputů uživatelů.

## 4 CRUD

CRUD je anglická zkratka pro create(česky vytvořit), read(česky číst), update(česky upravování) a delete(česky smazat). Je to taky vzor na tvoření webových aplikací,

vzor funkcionality, kterou má rozumná webová aplikace mít. Na implementaci CRUD je použit framework ASP.NET MVC, framework Dapper ORM a programovací jazyk C#. Tento koncept byl v projektu používán pořád a všude. Jeden konkrétní příklad je produkt. Na katalog je potřeba produkt vytvořit, staré produkty vymazat, případné chyby upravit a hlavně si produkty prohlížet. Veškeré akce, které byly popsány, jsou koncept CRUD.

## 5 DAPPER ORM

Dapper ORM je framework, který umožňuje mapovat objektově orientované modely na relační databáze. Tento framework dává možnost programátorovi psát vlastní query s parametry na rozdíl od Entity Frameworku. Dapper je taky rychlejší než Entity Framework, když se pracuje s velkým počtem záznamů. Jeden menší problém, který nastal během používání Dapper ORM, bylo, že Dapper ORM vyžadoval aby jména modelu byla identická s jmény položek v databázi, jinak databáze nevracela žádná data. Řešení bylo v query použít „as“ mezi jménem modelu a položky databáze a tím se problém vyřešil. (1)

### 5.1 KONKRÉTNÍ POUŽITÍ DAPPER ORM

Jedním z konkrétních využití je umožnění uživateli nebo-li adminovi stránky přidat produkty na web a veškeré ostatní akce týkající se produktu, neboli akce, které již byly zmíněny v kapitole CRUD. Uživatel může taky k produktu přidat obrázek.

Většina požadavků na server jsou řešeny přes formuláře. Uživatel vyplní formulář produktu, vyplněný formulář je poté poslán na server. Server zkontroluje zda je požadavek validní a pokud ano, je produkt zapsán do databáze, pokud ne, je požadavek zamítnut. Ostatní funkce jako mazání produktu a upravování produktu jsou také řešeny přes formuláře. Jediná výjimka je prohlížení. Požadavek server zpracuje a data o produktu server načte z databáze a zobrazí.

Na obrázku číslo 5 je v zeleném obdélníku ukázka syntaxe Dapper ORM a na obrázku číslo 6 je vzhled formuláře k vytvoření nového produktu.

```
Počet odkazů: 1
public static int CreateProduct(string productName, int categoryId, string description, int price, DateTime date, string urlLink)
{
    ProductModel data = new ProductModel
    {
        ProductName = productName,
        CategoryId = categoryId,
        Description = description,
        ProductPrice = price,
        Date = date,
        UrlLink = urlLink
    };

    string sql = @"insert into dbo.Products(ProductName, CategoryId, Description, ProductPrice, Published, UrlLink)
                  values (@ProductName, @CategoryId, @Description, @ProductPrice, @Date, @UrlLink); SELECT CAST(SCOPE_IDENTITY() as int)";

    return SqlDataAccess.SaveDataGiveID(sql, data);
}
```

Obrázek 5

# New Product

Product Name \*

Description \*

Category \*

Price \*

Kč

Link

Image \*

 Upload

Obrázek 6

## 6 NAHRÁVÁNÍ OBRÁZKU

Nahrávání obrázku souvisí s přidáváním nového produktu, ve formuláři, který jde vidět na obrázku číslo 6, je tlačítko pro nahrání. Obrázky se někde budou muset ukládat, tzn. že je třeba jako první udělat tabulku v databázi. Tabulku lze vidět na obrázku číslo 1.

Daný úkol se dá řešit dvěma způsoby. Jeden způsob je obrázek uložit do databáze jako BLOB, což je zkratka pro binary large object (česky velký binární objekt), druhý je do databáze ukládat pouze název obrázku a samotný obrázek uložit do složky. U prvního způsobu je o krok méně a nemohlo by se stát, že by se uložil jenom název obrázku, ale samotný obrázek by se neuložil. I přes zmíněnou nevýhodu byl vybrán druhý způsob a to z důvodu jednoduchosti implementace. (2) (3)

Nahrávání obrázku s sebou přináší dva podúkoly. První z nich je validace obrázku, jelikož i když se očekává obrázek tak technicky vzato se může nahrát jakýkoli soubor, který server povolí. Validování souboru je podle 2 kritérií, typ souboru a velikost

souboru. Soubor musí být buď typu PNG, nebo JPEG a nesmí být větší než 5MB. (4) Druhý podúkol, spíše pro pohodlí uživatele, je náhled obrázku. Úkol se řeší pomocí Javascriptu. (5)

## 7 PŘIHLAŠOVACÍ SYSTÉM

Původně bylo v plánu vytvořit vlastní přihlašovací systém, ale po hlubším zkoumání z toho sešlo, jelikož by realizace plánu byla moc komplikovaná. Přihlašovací systém byl nakonec vygenerován pomocí Visual Studia, které to má rovnou jako template. Problém je, že vygenerovaný přihlasovací systém používá jiný framework na databázi a to Entity Framework tzn., že projekt má 2 frameworky na databázi, což zhoršuje údržnost kódu.

## 8 MICROSOFT AZURE

Microsoft Azure je cloudová služba od společnosti Microsoft. Finálním krokem tohoto projektu bylo hostovat aplikaci pomocí cloudových služeb. Důvod, proč byl vybrán Microsoft Azure je, že společnost Microsoft nabízí studentům 100 Amerických dolarů na veškeré jejich cloudové služby.

## 9 ZÁVĚR

Aplikace je hostována na cloudových službách a umí požadované funkcionality, jedná z věcí na, které by se dalo ještě zapracovat je vzhled stránky. Aplikace byla vyvíjena na třinácti palcovém počítači, takže škálování nefunguje tak, jak by měla na jiných zařízeních. Dalším menším nedostatkem je možnost nahraní pouze jednoho obrázku, což pro katalog není úplně ideální. Poslední a hlavní nedostatek je struktura aplikace. Standardní webová aplikace má tři vrstvy front-end, back-end a prostřední vrstvu, která slouží jako médium pro front-end a back-end. Tato aplikace se skládá pouze ze 2 vrstev front-end, back-end, které na mezi sebou přímo interagují, což je z bezpečnostního hlediska nesprávný přístup. Důvod vybrání tohoto postupu je jednodušší vývoj.

Díky projektu se autor naučil základy pro vývoj webové aplikace, front-end a hlavně back-end, framework ASP.NET MVC, základy SQL, základy frameworku Dapper ORM a programovací C#.

## 10 SEZNAM POUŽITÝCH ZDROJŮ

1. **M, Risto.** *Stack Oveflow*. [Online] 30. 3 2019. [Citace: 5. 2 2020.] <https://stackoverflow.com/questions/55433731/dapper-query-result-sets-pk-of-table-as-null-in-objects?fbclid=IwAR30zWuRUUzWTL3PrPhWw-ESadQC7opC2BT5jnGn6vxhV8w4RCgX1lo9RnI>.

2. **Aurigma.** *Aurigma*. [Online] [Citace: 9. 1 2020.] <https://www.aurigma.com/upload-suite/developers/aspnet-mvc/how-to-upload-files-in-aspnet-mvc?fbclid=IwAR0stTsDR9UMrgH0MCmNVKs3LQsXpnAdtHs5ExVdSgLOdZTPDU3qaKKxw0A>.
3. **Dreamer, DotNet.** *Stack Overflow*. [Online] 27. 4 2013. [Citace: 9. 1 2020.] <https://stackoverflow.com/questions/16255882/uploading-displaying-images-in-mvc-4?fbclid=IwAR1mcGuO8j6Yogzn6wbgsoowJ1i6liS6hfQEzoELEg1vXAZQTrAwhA58Y4k>.
4. **Dimitrov, Darin.** *Stack Overflow*. [Online] 11. 6 2017. [Citace: 20. 1 2020.] <https://stackoverflow.com/questions/6388812/how-to-validate-uploaded-file-in-asp-net-mvc/6388927?fbclid=IwAR1UP6YqCuspeEgJ0WWaVZ9KbbqNzF2zsvmSExDV1lZFWAjWy0ZNamz3tX4#6388927>.
5. **cnlevy.** *Stack Overflow*. [Online] 18. 11 2014. [Citace: 29. 1 2020.] [https://stackoverflow.com/questions/4459379/preview-an-image-before-it-is-uploaded?fbclid=IwAR3A4Hse-CRYqhVaYMpVONZfq0Ie-ZKRAK0\\_n\\_r4J4hue\\_wn2lgrTKfcZE](https://stackoverflow.com/questions/4459379/preview-an-image-before-it-is-uploaded?fbclid=IwAR3A4Hse-CRYqhVaYMpVONZfq0Ie-ZKRAK0_n_r4J4hue_wn2lgrTKfcZE).
6. **B, Senthil Kumar.** *developerpublish*. [Online] 10. 2 2015. [Citace: 4. 2 2020.] [https://developerpublish.com/dapper-net-guide-get-single-record-using-dapper/?fbclid=IwAR0TmHs8Ilm\\_KBUq9hUQqhVD3bfRJe3\\_u283qfbHulJZwbJoBI0smsPdO7k](https://developerpublish.com/dapper-net-guide-get-single-record-using-dapper/?fbclid=IwAR0TmHs8Ilm_KBUq9hUQqhVD3bfRJe3_u283qfbHulJZwbJoBI0smsPdO7k).
7. **Kingry, Joseph.** *Stack Overflow*. [Online] 14. 10 2008. [Citace: 4. 2 2020.] [https://stackoverflow.com/questions/200476/html-actionlink-method?fbclid=IwAR3MTDzWBPi5hr0ME5epm64heb4jr5aAl91MjT\\_1F\\_IDd9-uucTM8NZCcssY](https://stackoverflow.com/questions/200476/html-actionlink-method?fbclid=IwAR3MTDzWBPi5hr0ME5epm64heb4jr5aAl91MjT_1F_IDd9-uucTM8NZCcssY).
8. **er-sho.** *Stack Overflow*. [Online] 16. 8 2016. [Citace: 11. 2 2020.] [https://stackoverflow.com/questions/51875938/display-dynamic-menu-in-mvc-layout-page?fbclid=IwAR0LtFWY9GDVpf9EGhEkxy8ZAGvtZSBt8qlzLRGZh\\_gSkR5f8qmeuoe3E4I](https://stackoverflow.com/questions/51875938/display-dynamic-menu-in-mvc-layout-page?fbclid=IwAR0LtFWY9GDVpf9EGhEkxy8ZAGvtZSBt8qlzLRGZh_gSkR5f8qmeuoe3E4I).
9. **Romias.** *Stack Overflow*. [Online] 27. 11 2013. [Citace: 27. 2 2020.] <https://stackoverflow.com/questions/20242981/asp-net-mvc-dropdown-list-from-selectlist?fbclid=IwAR3C-xpCMDiqYiWt4zHqSfzFNnVu8DVKLZ0BDuV35jXII5xyJECOo3RHkVQ>.
10. **IUnknown.** *Stack Overflow*. [Online] 1. 8 2013. [Citace: 15. 3 2020.] [https://stackoverflow.com/questions/815229/how-do-i-redirect-to-the-previous-action-in-asp-net-mvc?fbclid=IwAR28pTleYZIfwPwiSU-MSoonXP-yUbIBJUqZGDsevN5m\\_tnFcvCMc\\_XCBo0](https://stackoverflow.com/questions/815229/how-do-i-redirect-to-the-previous-action-in-asp-net-mvc?fbclid=IwAR28pTleYZIfwPwiSU-MSoonXP-yUbIBJUqZGDsevN5m_tnFcvCMc_XCBo0).

