



Střední průmyslová škola a Vyšší odborná škola, Písek, Karla Čapka 402, Písek

18-20-M/01 Informační technologie

Maturitní práce

Správa povinných prací

Téma číslo 1

autor:

Richard Kropáček, B4.I

vedoucí maturitní práce:

Mgr. Milan Janoušek

Písek 2020/2021

Anotace

Výstupem mé maturitní práce je fungující systém, jehož úkolem je správa povinných prací odevzdaných za uplynulý školní rok žáky. Učitelé následně mohou práce procházet, stahovat a hodnotit.

Klíčová slova

C#; ASP.NET Core; Entity Framework Core; HTML; JavaScript; CSS; Bootstrap; MySQL; WebApp

Annotation

The output of my graduation thesis is a functioning system, the task of which is the administration of compulsory work submitted by pupils for the past school year. Teachers can then browse, download and evaluate the work.

Keywords

C#; ASP.NET Core; Entity Framework Core; HTML; JavaScript; CSS; Bootstrap; MySQL; WebApp

Poděkování

Tímto bych chtěl poděkovat Mgr. Milanu Janouškovi za vedení mé Maturitní práce, cenné rady a odborný dohled. Děkuji také za pomoc všem, kteří se podíleli na gramatické kontrole práce.

Obsah

1	Úvod	4
2	Teoretický úvod	5
2.1	Programovací jazyk C#	5
2.2	ASP.NET, ASP.NET CORE	5
2.2.1	ASP.NET	6
2.2.2	ASP.NET Core	6
2.3	EF6 a EF Core	6
2.3.1	Entity Framework 6	7
2.3.2	Entity Framework Core	7
2.4	MVC	7
2.4.1	Model	7
2.4.2	View	8
2.4.3	Controller	8
2.5	MySQL	8
3	Uživatelské rozhraní	9
3.1	Student	9
3.2	Učitel	9
3.3	Administrator	9
4	Přihlašování a registrace	10
5	Databáze	11
6	Závěr	12
	Přílohy	16
A	Příloha	17

Kapitola 1

Úvod

Pro realizaci této práce jsem se rozhodl využít ASP.NET Core, což jest otevřený framework pro tvorbu webových aplikací vyvinutý společností Microsoft. Pro databázi jsem se rozhodl využít MySQL, což je velice rozšířený otevřený systém řízení báze dat. Webová aplikace je psána pomocí značkovacího jazyku HTML. Pro vizuální část stránky jsem využil Bootstrap 4.

Kapitola 2

Teoretický úvod

2.1 Programovací jazyk C#

C# je moderní, vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý společností Microsoft. Jazyk C# se řadí mezi typově bezpečné programovací jazyky. To znamená, že nedovoluje provádět operace, které mohou vést k chybám. Tento jazyk umožňuje vývojářům vytvářet mnoho druhů zabezpečených a robustních aplikací, které běží na platformě .NET ¹.

C# je zároveň objektově orientovaný programovací jazyk orientovaný na součásti ². To znamená, že se zaměřuje na vytváření komponent, které jsou tvořeny často se opakujícími částmi kódu. C# také poskytuje jazykové konstrukce pro přímou podporu těchto konceptů, což z něj dělá přirozený jazyk pro tvorbu a používání těchto softwarových komponent. [10]

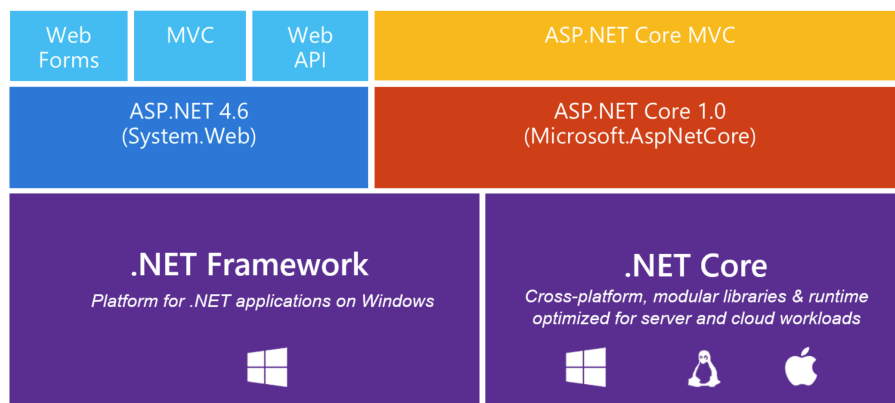
2.2 ASP.NET, ASP.NET CORE

ASP.NET je webový framework obsahující sadu knihoven, které obsahují hotová řešení mnoha základních problémů, které ve webových technologiích vyvstávají. Může se jednat např. o bezpečnost, autentifikaci uživatele, práci s databázemi apod.

Tato technologie je založena na architektuře klient-server. Výstupem ASP.NET aplikace je HTML stránka. ASP.NET tedy běží na serveru a reaguje na dotazy uživatele/klienta. Pro tvorbu ASP.NET aplikace je potřeba znalost především programovacího jazyku C# a značkovacího jazyku HTML. [7]

¹Vývojářská platforma pro vytváření webových, mobilních, desktopových, herních, IoT a dalších aplikací. Je podporovaná v systémech Windows, Linux a macOS.

²Component-oriented programming language



Obrázek 2.1: Porovnání ASP.NET Core a ASP.NET [1]

2.2.1 ASP.NET

ASP.NET je webový framework s otevřeným zdrojovým kódem, vytvořený společností Microsoft, pro vytváření moderních webových aplikací. ASP.NET rozšiřuje platformu .NET o nástroje a knihovny určené právě pro vytváření webových aplikací. [3]

2.2.2 ASP.NET Core

ASP.NET Core je open-source a multiplatformní verze ASP.NET. Tato platforma je navržena tak, aby umožnila rychlé vyvíjení runtime komponent, API, překladačů apod. a zároveň poskytovala stabilní a podporovanou platformu pro udržení běhu aplikací.

Aplikace v ASP.NET Core lze, oproti dřívější verzi ASP.NET Windows-only version, vyvíjet a spouštět v systémech Windows, Linux, macOS a Docker. [4]

2.3 EF6 a EF Core

Entity Framework je Object–relational mapping (ORM) ³. To znamená, že se databázové tabulky přímo mapují na C# třídy. V projektu následně pracujeme pouze s objekty a framework za nás na pozadí generuje SQL dotazy. Díky tomu je výsledná aplikace tvořena především pomocí objektů. [8]

³Objektově-relační mapování

2.3.1 Entity Framework 6

Entity Framework 6 (EF6) je ORM, primárně navržený pro .NET Framework, ale zároveň s podporou pro .NET Core. EF6 je stabilní, podporovaný produkt, ale již se aktivně nevyvíjí. [9]

2.3.2 Entity Framework Core

Entity Framework Core (EF Core) je moderní ORM pro .NET. Podporuje dotazy LINQ, sledování změn, aktualizace a migrace schématu.

EF Core pracuje s SQL Server/SQL Azure, SQLite, Azure Cosmos DB, MySQL, PostgreSQL a mnoha dalšími databázemi prostřednictvím modelu modulů plug-in zprostředkovatele databáze. [9]

2.4 MVC

Model-View-Controller (MVC) je návrhový vzor používaný k rozdělení webové aplikace na 3 komponenty.

- Model - Data
- View - Uživatelské rozhraní
- Controller- Logická část aplikace

Pomocí vzoru MVC pro webové aplikace jsou požadavky směřovány na controller, který je zodpovědný za práci s modelem. Model provádí akce a načítá data z databáze. Controller poté zvolí zobrazení (view), které se má zobrazit, a poskytne mu model. View už pouze vykreslí stránku na základě dat získaných z modelu. [2]

2.4.1 Model

Jedná se o dynamickou datovou strukturu aplikace, nezávislou na uživatelském rozhraní. Jejím úkolem je spravovat data, pravidla a logiku aplikace.

V ASP.NET Core MVC má model podobu třídy, která je provázaná s databází. Pomocí atributů definuje pravidla, která se aplikují na klienta a server. [6]

2.4.2 View

Převádí data reprezentovaná modelem do podoby vhodné k interaktivní prezentaci uživateli.[5] Může se jednat o jakoukoliv reprezentaci dat (graf, diagram, tabulka, atd.).

V ASP.NET Core se využívá syntaxe Razor. Ta poskytuje jednoduchý, čistý a lehký způsob vykreslení obsahu HTML stránek na základě view. Razor umožňuje vykreslit stránku pomocí C# a vytvářet webové stránky plně kompatibilní s HTML5. [6]

2.4.3 Controller

Controller reaguje na vstup uživatele a provádí interakce s objekty datového modelu. Zjednodušeně to znamená, že řadič přijme vstup, ověří jej a následně ho předá modelu. [6]

2.5 MySQL

Kapitola 3

Uživatelské rozhraní

3.1 Student

3.2 Učitel

3.3 Administrator

Kapitola 4

Přihlašování a registrace

Kapitola 5

Databáze

Kapitola 6

Závěr

Seznam tabulek

Seznam obrázků

2.1 Porovnání ASP.NET Core a ASP.NET [1]	6
--	---

Literatura

- [1] ASP.NET vs ASP.NET Core. 2017, [cit. 2021-4-4].
Dostupn <https://thienn.com/images/aspnetcore.png>
- [2] Microsoft Corporation: *ASP.NET MVC Pattern*. 2021, [Online; cit. 2021-4-5].
Dostupn <https://dotnet.microsoft.com/apps/aspnet/mvc>
- [3] Microsoft Corporation: *What is ASP.NET?* 2021, [Online; cit. 2021-4-4].
Dostupn <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>
- [4] Microsoft Corporation: *What is ASP.NET Core?* 2021, [Online; cit. 2021-4-4].
Dostupn <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core>
- [5] Model-view-controller. 2021, [cit. 2021-04-05].
Dostupn <https://cs.wikipedia.org/wiki/Model-view-controller>
- [6] Model-view-controller. 2021, [cit. 2021-04-05].
Dostupn <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [7] ČÁPKA, D.: *Lekce 1 - Úvod do ASP.NET*. 2012, [cit. 2021-4-4].
Dostupn <https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net>
- [8] ČÁPKA, D.: *Lekce 8 - Scaffolding a Entity Framework v ASP.NET MVC*. 2014, [cit. 2021-4-5].
Dostupn <https://www.itnetwork.cz/csharp/asp-net-mvc/zaklady/asp-dot-net-mvc-tutorial-scaffolding-entity-framework-editor-clanku>
- [9] VICKERS, A.; OLPROD: *Porovnání EF Core a EF6*. Microsoft Corporation, 2019, [Online; cit. 2021-4-5].
Dostupn <https://docs.microsoft.com/cs-cz/ef/efcore-and-ef6/>

- [10] WAGNER, B.; OLPROD: *Prohlídka jazyka C#*. Microsoft Corporation, 2021, [Online; cit. 2021-3-25].

Dostupn <https://docs.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>

Příloha A

Příloha