

Optimalizace zobrazení rozvrhu na webových stránkách fakulty

Romana Frýželková

Bakalářská práce
2017



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Romana Frýželková**
Osobní číslo: **A13293**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační technologie v administrativě**
Forma studia: **prezenční**

Téma práce: **Optimalizace zobrazení rozvrhu na webových stránkách fakulty**
Téma anglicky: **An Optimisation of Depiction of the Faculty Teaching Schedule on its Website**

Zásady pro vypracování:

1. Vytvořte aplikaci k optimálnímu zobrazení rozvrhů na webových stránkách FAI pro kombinované studium
2. Seznamte se s programem Rozvrh RNDr. Červený pro tvorbu rozvrhu
3. Zpracujte export databáze pro zobrazení rozvrhu studijních skupin u kombinovaného studia
4. Umožněte přepínání mezi jednotlivými rozvrhy
5. Udělejte spojené zobrazení rozvrhu skupiny, učitele a místnosti

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **MACDONALD, Matthew.** HTML5 the missing manual. Second edition. Sebastopol: O'Reilly Media, 2013. ISBN 9781449373443.
2. **LUBBERS, Peter, Brian ALBERS a Frank SALIM.** HTML5: programujeme moderní webové aplikace. Brno: Computer Press, 2011. ISBN 978-80-251-3539-6.
3. **ODELL, Den.** JavaScript : průvodce programováním ajaxových aplikací. Brno: Computer Press, 2010. ISBN 978-80-251-2733-9.
4. **LECKY-THOMPSON, Ed and Steven D. NOWICKI.** PHP 6: programujeme profesionálně. Přeložil Ondřej GIBL. Brno: Computer Press, 2010. ISBN 978-80-251-3127-5.
5. **FRAIN, Ben.** Responsive web design with HTML5 and CSS3 learn responsive design using HTML5 and CSS3 to adapt websites to any browser or screen size. Birmingham: Packt Publishing, 2012. ISBN 9781849693196.

Vedoucí bakalářské práce:

Mgr. Jaroslav Filo

Ústav matematiky

Datum zadání bakalářské práce:

3. února 2017

Termín odevzdání bakalářské práce:

30. května 2017

Ve Zlíně dne 3. února 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



Ing. Miroslav Matýsek, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 15. 5. 2014

.....
Frýdlova
podpis diplomanta

ABSTRAKT

Bakalářská práce se věnuje optimalizaci zobrazení na webových stránkách fakulty. V teoretické části jsou popsány nástroje potřebné k vytvoření dynamické webové stránky a databáze. Dále pak je stručně popsán program Rozvrh RNDr. Červený a jeho funkce. V praktické části jsou popsány potřebné programy pro vytvoření webové stránky a postup při tvorbě databáze pro následný import dat na stránku. Dále pak je realizace samotného návrhu optimálního zobrazení rozvrhů tříd, vyučujících a učeben na webové stránce fakulty aplikované informatiky.

Klíčová slova: rozvrh, HTML, CSS, SQL, XAMPP, dynamická webová stránka

ABSTRACT

This bachelor thesis focuses on the layout optimization of the Faculty of Applied Informatics web-page. The theoretical part describes the tools required to create a dynamic web-page and database. Furthermore a brief description and of "Rozvrh RNDr. Červený" is provided as well as its function. The practical part describes the programs necessary for the creation of a web-page and the process of creating databases for subsequent import onto the web-page. The practical part also contains application of the proposed optimal layout of classes, tutors and classrooms on the web-page of the Faculty of Applied Informatics.

Keywords: schedule, HTML, CSS, SQL, XAMPP, dynamic website

Chtěla bych poděkovat vedoucímu své bakalářské práce za rady při psaní této práce. Děkuji také rodině a přátelům za velkou podporu, trpělivost a pochopení.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	11
1 HTML	12
1.1 HISTORIE HTML.....	12
1.2 HTML 5.1.....	12
1.2.1 XHTML.....	13
1.3 ZÁKLADNÍ PRINCIPY HTML.....	13
1.4 ZÁKLADNÍ TAGY HTML.....	14
1.4.1 Výhody.....	15
1.4.2 Nevýhody.....	15
2 CSS	16
2.1 FUNKCE CSS.....	16
2.2 PROPOJENÍ S HTML.....	16
2.3 VÝHODY A NEVÝHODY CSS.....	17
2.4 JAVASCRIPT.....	17
2.5 HISTORIE.....	18
2.6 CO OBSAHUJE PROGRAM V JAVASCRIPTU.....	18
2.7 FUNKCE JAVASCRIPTU.....	18
3 PHP	19
3.1 PHPMYADMIN.....	20
3.1.1 MySQL.....	20
4 DYNAMICKÁ WEBOVÁ STRÁNKA	22
4.1 TYPY DYNAMICKÝCH WEBOVÝCH STRÁNEK.....	22
4.1.1 Skriptování na straně klienta.....	22
4.1.2 Skriptování na straně serveru.....	23
4.1.3 Skriptování na straně klienta i serveru.....	23
5 PROGRAM ROZVRH	24
5.1 POPIS PROGRAMU ROZVRH 4.....	24
5.1.1 Výhody programu.....	25
5.1.2 Nevýhody programu.....	25
6 SEZNÁMENÍ S PROGRAMEM ROZVRH 4.1	26
6.1 ZADÁNÍ ZÁKLADNÍCH ÚDAJŮ.....	26
6.2 VYTVOŘENÍ NOVÉHO SOUBORU.....	26
6.3 ÚPRAVA EXISTUJÍCÍHO DOKUMENTU.....	26
6.3.1 Nastavení automatického otevření stávajícího dokumentu.....	26
6.4 UKLÁDÁNÍ SOUBORŮ.....	27
6.5 ROZVRHOVÁ TABULKA.....	27
6.6 SEZNAMY.....	27
6.7 LÍSTKY.....	27
6.7.1 Vazby mezi lístky v rozvrhu.....	28

6.8	ÚVAZKY	28
6.9	ROZVRH	28
6.9.1	Automatická tvorba rozvrhu	28
6.10	EXPORT DAT	29
6.10.1	Export seznamů	29
6.10.2	Export rozvrhů	29
7	ROZVRH FAKULTY APLIKOVANÉ INFORMATIKY	30
7.1	ÚDAJE V ROZVRHU FAKULTY	30
7.1.1	Skupina	30
7.1.2	Předmět	31
7.1.3	Vyučující	32
7.1.4	Učebna	32
II	PRAKTICKÁ ČÁST	33
8	NÁVRH OPTIMALIZACE ZOBRAZENÍ ROZVRHU	34
9	TVORBA WEBOVÝCH STRÁNEK FAKULTY	36
9.1	INSTALACE POTŘEBNÝCH PROGRAMŮ	36
9.1.1	XAMPP	36
9.1.2	PSPad	38
9.1.3	Brackets	39
9.1.4	Komunikace se serverem užitím PHP a JavaScript	40
10	DATABÁZE	44
10.1	DATABÁZOVÝ DIAGRAM	44
10.1.1	Enterprise Architect	44
10.2	TVORBA DATABÁZE	47
10.3	STRUKTURA DATABÁZÍ	49
10.3.1	Struktura databáze „Rozvrhová tabuľa“	49
10.3.2	Struktura databáze „Zoznamy“	50
10.3.3	Struktura databáze „Rozvrh“	50
11	IMPORT DAT	52
12	ZOBRAZENÍ ROZVRHU	54
	ZÁVĚR	58
	SEZNAM POUŽITÉ LITERATURY	59
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	59
	SEZNAM OBRÁZKŮ	65
	SEZNAM TABULEK	67
	SEZNAM PŘÍLOH	68

ÚVOD

Bakalářská práce se celkově zabývá optimalizací zobrazení rozvrhu na stránkách Fakulty aplikované informatiky a má za cíl popsat veškeré prvky a náležitosti, které dopomohou k následné optimalizaci. Optimalizace zobrazení by se měla týkat rozvrhů u kombinovaného studia a to jednak spojeného zobrazování rozvrhu vyučujícího, skupiny a předmětu a poté možnosti uživatele přepínat mezi jednotlivými rozvrhy. Celý tento optimalizovaný rozvrh by se poté měl zobrazovat na simulovaných webových stránkách fakulty, do nichž bude celý soubor s rozvrhem importován. Cílem této optimalizace zobrazení je v první řadě zjednodušení zobrazení rozvrhu třídy, vyučujícího a učebny, kdy uživatel nemusí vyhledávat každý rozvrh zvlášť, ale zobrazí se mu všechny na jedné stránce.

Teoretická část se týká v první řadě značkovacího jazyka HTML, který je nezbytný pro tvorbu jakýchkoli webových stránek. V práci je popsáno, co znamená HTML všeobecně i jeho vývoj během let a desetiletí až k dnešní prozatím nejnovější verzi HTML 5.1, která je popsána se všemi jejími principy, výhodami a nevýhodami. Poté je popsán rozdíl mezi HTML a novým XHTML a nakonec kapitoly jednoduchá tabulka s nejzákladnějšími tagy. Druhá kapitola je pod názvem CSS, který nám určuje vzhled webové stránky. V práci je popsáno co to vlastně CSS je, jak funguje a v čem je jeho využití výhodné, nebo naopak nevýhodné. Krom popisu je v této kapitole také zmínka o propojení stylů a HTML, což je zejména pro práci s webovými stránkami důležité. V kapitole o kaskádových stylech se taktéž nachází podkapitola o JavaScriptu, komplementárnímu jazyku pro webové programování. Třetí kapitola pojednává o jazyku PHP a objasňuje jeho funkce, a za jakým účelem se používá. Dále jsem v této kapitole zmínila ještě populární nástroj phpMyAdmin, který umožňuje správu obsahu relační databáze MySQL, se kterou jsem pracovala na vytvoření kompletních tabulek a seznamů, pro následující práci s rozvrhem. Na tyto tři kapitoly potom navazuje kapitola čtvrtá, která je o samotných dynamických webových stránkách, neboť ty byly jednou z hlavních náplní mé práce. V kapitole je osvětleno, co to dynamické stránky vlastně jsou, jak zhruba fungují a podle čeho se rozdělují. Další dvě kapitoly v teoretické části se věnují programu Rozvrh RNDr. Červený a to konkrétně verzi, kterou aktuálně používá fakulta. Poslední kapitoly v teoretické části obsahují jak teoretický stručný, tak i ten praktický popis programu Rozvrh 4.2. Nakonec jsou popsány všechny prvky, které tvoří rozvrh na FAI a jsou zobrazeny na informačním systému IS STAG nebo webových stránkách Fakulty aplikované informatiky.

V praktické části je v hned v úvodní kapitole, popsán návrh na optimalizaci zobrazení rozvrhu pro kombinované studium. Optimalizace zobrazení by v první řadě měla spočívat v tom, že uživatel, který si chce rozvrh zobrazit, nemusí jednotlivě vyhledávat rozvrhy vyučujícího, třídy/skupiny nebo místnosti. Jako další kapitola v praktické části pojednává o přípravě na tvorbu samotných webových stránek, neboť rozvrh by se měl zobrazovat v prostředí fakulty. V práci jsou popsány i jednotlivé programy, které byly nezbytné při mé tvorbě, jako jsou XAMPP, PSPad nebo Brackets a poté samotná tvorba. Následující kapitola pojednává o tvorbě databázového diagramu, kde jsem vyznačila všechny důležité prvky, které se budou v rozvrhu objevovat. Hned na to navazuje kapitola s popisem a příklady tvorby samotných tabulek databáze v SQL a vkládání dat do nich. Data do databáze byla čerpána z programu RNDr. Červený, odkud byla importována na již vytvořené stránky fakulty a to importu pojednává také další kapitola v praktické části. Na konec praktické části je popsáno samotné optimalizované zobrazování rozvrhu na stránkách fakulty.

I. TEORETICKÁ ČÁST

1 HTML

HTML nebo HyperText Markup Language je značkovací jazyk, který je určen k vytváření dokumentů, které obsahují pokročilé formátování a hypertextové odkazy. Stále se jedná o jeden z nejoblíbenějších jazyků pro tvorbu jednoduchých webových stránek. Tento značkovací jazyk používá k vytváření a formátování souborů pro webové stránky definované značky, neboli tagy. [1]

1.1 Historie HTML

HTML prošlo během let několika vývojovými fázemi. První verze HTML 1.0, se objevila v roce 1990 a její neoficiální verze HTML+ byla představena v roce 1993. Základní prvky HTML+ zahrnovaly formuláře, tabulky a obrázky s popisem, ale nebylo možné upravovat text nebo formátovat odstavce. Obsahovalo 78 prvků, které v dnešní verzi HTML nenajdeme.

HTML 2.0 bylo představené v roce 1994 a na rozdíl od svých předchůdců mělo již formální specifikaci a stalo se oficiálním standardem. Verze obsahovala 49 prvků.

HTML 3.0 se objevilo v březnu 1995. Tato verze měla již větší možnosti jako matematické elementy nebo obtékání obrázků textem. Životnost této verze ovšem vypršela a dnes již nejsou podporovány.

V květnu roku 1996 potom bylo uvedeno na trh HTML 3.2, paradoxně se bere jako zástupce HTML 2.0 přičemž tzv. ignoruje HTML 3.0. Tato verze přidala k původním ještě 19 nových prvků.

Poté byla vytvořena verze 4.0, které se během jejího vývoje přezdívalo „Cougar“. Přidala podporu prvku OBJECT (možnost vkládání obrázků a multimédií). Podporuje kaskádové styly, rozpoznání jazyků se zvláštními znaky a úpravy formulářů a tabulek.

Předposlední vyvinutou verzí do dnešní doby je verze HTML 5.0, která byla schválena koncem roku 2014. [2]

1.2 HTML 5.1

Zatím nejnovější verzi HTML 5.1, zveřejnila společnost W3C v listopadu 2016 a už se připravuje další verze HTML 5.2, která by měla být zveřejněna na konci letošního roku. Tato verze prakticky přebírá všechny vlastnosti a funkce původní verze, ale je navíc rozší-

řena o další. Jednou z novinek je prvek, který umí vkládat obrázky prostřednictvím interpretace vektorové grafiky. Verze 5.1 dále umožňuje používat takové atributy, které mohou vytvořit nulovou šířku obrazu. Tato funkce může být užitečná zejména pro vývojáře, aby eventuelně skryli některé obrázky před uživateli. Další funkcí je potom možnost přidání kryptografického noncesu do stylů a skriptů. Je to náhodně vygenerované číslo, které lze použít pouze jednou a po každém dalším otevření stránky musí být obnoven. Používá se například pro zabezpečení stránek s politickým obsahem. Dále pak disponuje funkcí, která zobrazí nebo naopak skryje dodatečné informace. Verze 5.1, se také určitým způsobem snaží zabránit „hиту“ dnešní doby a to aby někdo nekompetentní získal o uživateli citlivé informace tzv. phishingu, pomocí jednotlivých kontextů prohlížeče. [3] <https://www.w3.org/TR/html51/>

1.2.1 XHTML

Je to moderní značkovací jazyk, který by měl sloužit jako nástupce HTML a zároveň je aplikací jazyka XML. Jak bylo tedy již zmíněno jazyk XHTML je v podstatě jazyk HTML, který navíc splňuje pravidla jazyka XML jako je například podmínka, že všechny názvy atributů a elementů by měly být zapsány buď malými, nebo velkými písmeny, protože v jazyku XML záleží na velikosti písmen. Dalším pravidlem je to, že atributy musí mít uvedené hodnoty, a ty musí být zapsány do uvozovek a to se týká dokonce i jednoduchých číselných hodnot. Navíc co se týká XHTML, tak na rozdíl od HTML musí být všechny elementy dobře formované - to znamená, že každý element, který byl otevřen musí být také uzavřen. Díky tomu webový prohlížeč lépe analyzuje kod jazyka XHTML, což je výhodou zejména pro mobilní aplikace. [4] [Kniha Javascript](#)

1.3 Základní principy HTML

Aby byl tento jazyk disponoval svou proslulou přehledností a jednoduchostí, musí dodržovat základní principy:

- **Přístupnost** – rozumí se tím především podpora světových jazyků a nezávislost na zařízeních
- **Spolupráce** – řešení chyb a snaha vyhnout se zbytečným složitostem
- **Užitečnost** – zabývat se skutečnými problémy, ctít přání zákazníků a jeden DOM pro všechny

- **Kompatibilita** – přizpůsobení se rozdílným softwarovým prostředím, podpora stávajícího obsahu webu, zlepšovat nastavené parametry[5]<http://html5css3.4fan.cz/html-uvod.php>

1.4 Základní tagy HTML

Tagy se nazývají značky, které dohromady tvoří celý HTML soubor. Strukturu celého dokumentu tvoří **<html>** základní párový znak potom **<head>** hlavička a nakonec **<body>** tělo celého souboru. V tabulce níže se nachází alespoň ty nejzákladnější z nich.

Tabulka 1: Základní tagy HTML souboru

<i>Struktura dokumentu</i>	
<html>, </html>	začátek dokumentu; konec dokumentu
<head>, </head>	je začátek hlavičky; konec hlavičky
<body>, </body>	začátek těla stránky; konec těla stránky
<title>, </title>	hlavní nadpis stránky
<i>Vzhled dokumentu</i>	
<h1>, </h1>	nadpis první úrovně.
<h2>, </h2>	nadpis druhé úrovně
<p>, </p>	vymezení odstavce
, 	tučný text
<i>, </i>	Kurzíva
, 	nějakým způsobem odlišný text
<u>, </u>	podtržený text
<i>Seznamy</i>	
, 	odrážkový seznam
, 	číslovaný seznam
<i>Tabulky</i>	
<table>, </table>	Tabulka
<tr>, </tr>	Řádek
<td>, </td>	buňka

1.4.1 Výhody

Některé z výhod již obsahuje text o pár řádků výše a je to zmiňovaná jednoduchost, přehlednost a stejné zobrazení na téměř všech prohlížečích. Mezi další výhody patří malý a přehledný zdrojový kód a také fakt, že tento jazyk je nejrozšířenější, tedy že valná většina stránek v sobě nese kód html a výjimkou jsou pouze některé flashové stránky.[6](<http://seminarka-web.wz.cz/main.php?page=html&specif=o>). Co se týče novějších verzí, jejich specifikace vybízí k tomu, aby uživatel psal čistější a přehlednější kód a to pomocí sémantických značek. Taktéž umožňuje dostupnost informací o GEO lokaci uživatele nebo lepší podporu pro mobilní zařízení. Výhodou je také to, že uživateli umožní dočasně ukládat data do stálého úložiště, čímž se znatelně zvýší reakční doba stránek z důvodu, že nespolehá na ukládání souboru cookies ze strany uživatele. [7] <http://www.dostupnyinternet.cz/blog/html-5/>

1.4.2 Nevýhody

Paradoxně právě jednoduchost, která je jinak jeho předností zapříčiňuje to, že dynamickou a moderní stránku nelze vytvořit pouze v HTML. Další nevýhodou je to, že nelze oddělit vzhled dokumentu od jeho obsahu. [1]KNIHA HTML 5

2 CSS

CSS neboli *Cascading Style Sheets* se doslovně přeloží, jako *Tabulky kaskádových stylů*. Autorem zcela prvního návrhu byl Håkon Wium Lie z Norska a svůj koncept CSS v roce 1994 publikoval. V roce 1997 jej potom dokončila standardizační organizace W3C, ve které Lie pracoval. Do dnešní doby byly vydány CSS1, CSS2 a CSS3. [8] https://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styl%C3%BD

Hlavním smyslem CSS je umožnit uživatelům oddělit vzhled dokumentu od jeho struktury a obsahu. Stručně bychom mohli říci, že je to určitý zápis, který určuje vzhled HTML dokumentu (barvy, obrázky nebo rozmístění prvků). Upravený soubor má příponu *css* a sám o sobě je nefunkční, jeho funkčnost se ukáže jedině s propojením s HTML souborem. [9] <http://www.pestujemeweb.cz/obsah/css/co-je-css.php>

2.1 Funkce CSS

Jednou z hlavních funkcí je to, že fungují z velké části i automaticky, tudíž se vzhled celého dokumentu deklaruje jedním způsobem. Další funkce:

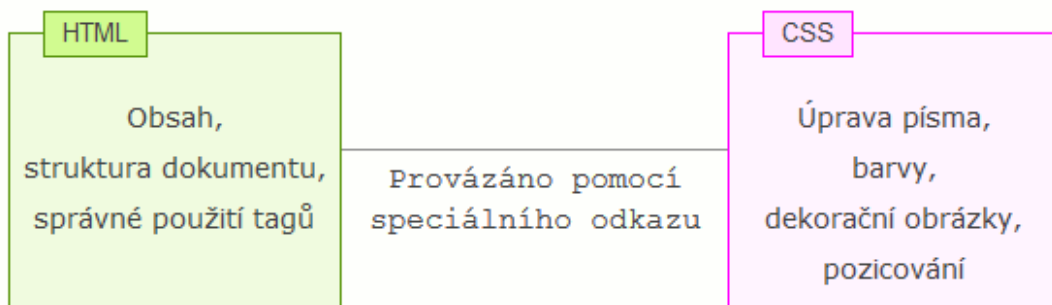
- Nastavení pozadí – umí nastavit pozadí jak stránky, tabulky nebo třeba odstavce
- Nastavení písma – velikost, kapitálky nebo prokládání
- Nastavení okrajů
- Funkce na části textu – průhlednost, neviditelnost nebo skrytí celého textu, nebo jen jeho části
- Umístění objektu kamkoli do stránky – jedná se o text, obrázek nebo cokoli jiného
- Formátování nadpisů – tato funkce je automatická
- Úprava odstavců - odsazení prvního řádku odstavce, vymaže nebo naopak zvětší velikost prázdného místa po odstavci nebo zvětší řádkování
- Změna grafického obsahu běžných tagů
- Grafické odrážky – automatická funkce [10] <https://www.jakpsatweb.cz/css/css-uvod.html>

2.2 Propojení s HTML

Jsou tři možnosti připojení těchto dvou prvků a jako první a nejpoužívanější je přímé **připojení CSS k prvku HTML (viz obr. 1)**, což je ukázáno na obrázku níže. Jako druhý způsob je **vložení CSS zápisu do hlavičky HTML dokumentu** toto připojení je ovšem -

stejně jako první možnost- vhodný v případě že je web tvořen pouze jedním HTML souborem. Nakonec třetí způsob se využívá především při situaci, kdy je web tvořen více HTML soubory a jedná se o **připojení externího CSS k HTML souboru**. [11]

<http://www.pestujemeweb.cz/obsah/css/pripojeni-css-k-html.php>



Obrázek 1: Připojení CSS k prvku HTML [12]

2.3 Výhody a nevýhody CSS

Co se týče výhod, má jich CSS nespočet a mezi hlavní patří v první řadě to, že nabízí rozsáhlejší možnosti formátování, než samotné HTML a týká se to například vlastností *padding* (nastavení vnitřních okrajů prvku) nebo *margin* (šířka vnějších okrajů prvku). Další výhodou je jistě také to, že použitím CSS se dá velikost dokumentu snížit až o polovinu a to sebou nese i výhodu při procházení textu vyhledávacím nástrojem, kdy nástroj nemusí procházet tolik textu. V neposlední řadě je výhodou také to, že CSS dynamicky pracuje se styly ve srovnání s HTML, kdy místo sáhodlouhého vyhledávání všech značek a změn jejich atributů, se s použitím CSS pouze přepíše jediný soubor *css*. K dalším výhodám patří ještě krátká doba načítání stránky, oddělení stylu a struktury, formátování *xml* dokumentů, jednodušší údržba webové prezentace nebo možnost webového prohlížeče, soubor se styly uložit do cache paměti. <http://www.garth.cz/uvod-do-css/vyhody-css-a-srovnani-z-html/>Jednou z hlavních nevýhod je ne vždy dobrá podpora ve většinových prohlížečích, kvůli jejich občasným chybám v implementaci CSS. [13]

2.4 Javascript

Je to multiplatformní skriptovací jazyk, který se používá zpravidla pro webové stránky jako interpretovaný programovací jazyk. JavaScript je v podstatě hovorový název pro spe-

cifikaci formálně nazývanou ECMAScript. Nejčastěji se JavaScript používá k podpoře multimediálního rozsahu, animaci na stránkách nebo měření a analytice webu.

[7]<http://www.dostupnyinternet.cz/blog/html-5/>

2.5 Historie

JavaScript byl vyvinut mezi léty 1995-1996 Bernardem Eichem ve společnosti NetScape. Jeho původní název ovšem nebyl JavaScript, nýbrž LiveScript. Název se změnil až po nátlaku pracovníků firmy na Bernarda Eicha. Bylo to trochu nešťastné rozhodnutí, neboť se JavaScript začal spojovat s Javou, která byt' byla velmi populární, zrovna v době vyvinutí JavaScriptu její popularita začala mírně klesat a tím pádem se původně nedostatky Javy, přičítaly JavaScriptu. Nakonec se podařilo pomocí rozsáhlé kampaně vysvětlit i širší laické veřejnosti jak to je ve skutečnosti a toto špatné propojení s Javou, JavaScript setřásl.

Po počáteční nedůvěře tvůrců webových stránek a programátorů tento jazyk dokázal potvrdit svoji užitečnost a schopnost vytvářet interaktivitu na World Wide Webu. JavaScript se poté brzy zaměřil na práci s obrázky na webových stránkách a ověřování obsahu formulářů na straně klienta.[14]

2.6 Co obsahuje program v JavaScriptu

Program, který je napsán v JavaScriptu obsahuje příkazy, které jsou tvořeny znaky, operátory a identifikátory umístěné za sebou v řadě. [JavaScript krok za krokem Steve Suehring](#)

2.7 Funkce JavaScriptu

Je to z velké části komplementární jazyk proto je velmi nepravděpodobné, že bude celá aplikace napsána pouze v JavaScriptu, aniž by byly použity i jiné jazyky- většinou HTML a také bez prezentace ve webovém prohlížeči. JavaScript je schopen vytvořit webovým stránkám potřebnou interaktivitu prostřednictvím:

- vytvoření vysouvacích navigačních nabídek
- transformací textu na webové stránce
- dynamickým přidáním elementů do stránky
- pomoc se vstupními formulářovými poli

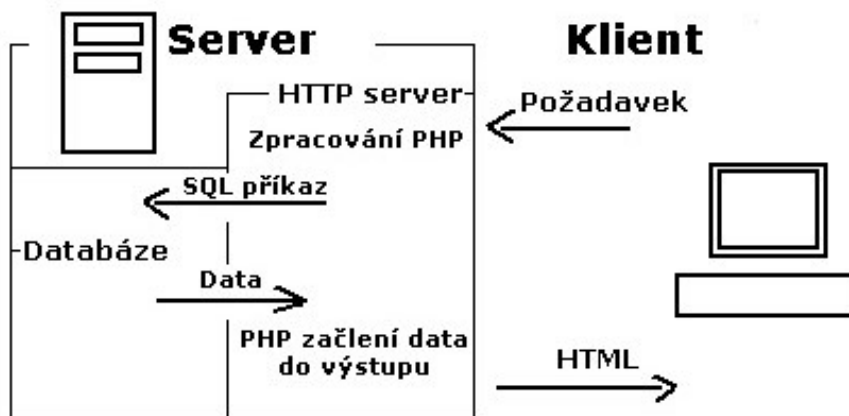
JavaScript podporují některé produkty společnosti Adobe, ale i přesto se tento jazyk používá především ve webovém programování. [4]

3 PHP

Je to skriptovací programovací jazyk pro tvorbu dynamických webových stránek a aplikací a vložený do běžného HTML kodu. Je to velmi všestranný jazyk a data se ukládají většinou do klasických textových souborů nebo do databáze (nejčastěji MySQL). Části PHP kodu se do skriptů vkládají pomocí speciálních značek. PHP není závislé na žádné platformě, může tedy běžet téměř kdekoli a to hlavně díky tomu, že do prohlížeče ve výsledku už dorazí pouze html kod, neboť všechnu práci PHP udělá na serveru [15]. Způsob, jakým PHP pracuje po vyslání požadavku klientem lze vidět i na obrázku č. 2 <https://www.interval.cz/clanky/php-zakladni-informace/>

Jazyk PHP má ale u některých vývojářů aplikací poměrně špatnou pověst, protože se rozšířil názor, že jazyk PHP je určen pouze pro uplatnění ve velmi jednoduchých aplikacích a pro začínající programátory. Důvodem by mohla být jednoduchá syntaxe nebo absence komplexního a rozsáhlého prostředí pro aplikační server. Navzdory tom, má ale tento jazyk spoustu dobrých výrazných vlastností. Jednou z hlavních je **jednoduchost**, kdy syntaxe jazyka PHP je ovlivňována populárními programovacími jazyky (Perl, ASP a C) a z nich PHP využívá a začleňuje pouze ty nedůležitější prvky jejich syntaxe, zatímco zbytečné funkce odstraňuje. Další významnou vlastností PHP je **podpora napříč platformami**, hlavně z toho důvodu, že tento jazyk vyžaduje pouze instalaci nového modulu na webový server a nepožaduje pro svůj chod žádné další procesy nebo služby a tím se snadno přizpůsobí jakémukoli webovému serveru. Tento jazyk má navíc jedinou implementaci šířenou pod open-source licencí a velmi aktivní vývojářskou komunitu. Dalším prvkem, díky kterému je tento jazyk stavěn do předních pozic seznamu možných jazyků pro tvorbu webových stránek je **dostupnost a dlouhověkost**. Každý uživatel má totiž jistotu, že tento jazyk podporuje téměř každý poskytovatel připojení, tudíž aplikace budou fungovat v každém prostředí, ve kterém bude potřeba bez jakýchkoli obav o podporu na straně cílového prostředí. Kvalita dokumentace a její snadná dostupnost celkově ulehčuje práci s PHP. Poslední z těch nejvýraznějších vlastností jazyka PHP je jeho **síla** a to i přes názory některých, že tento jazyk není pro vývoj softwaru na podnikatelské úrovni. PHP s poslední verzí ukázalo, že ve skutečnosti představuje formalizovaný jazyk a prostředí pro vytvoření těch nejrobustnějších aplikací. [16]KNIHA PHP6

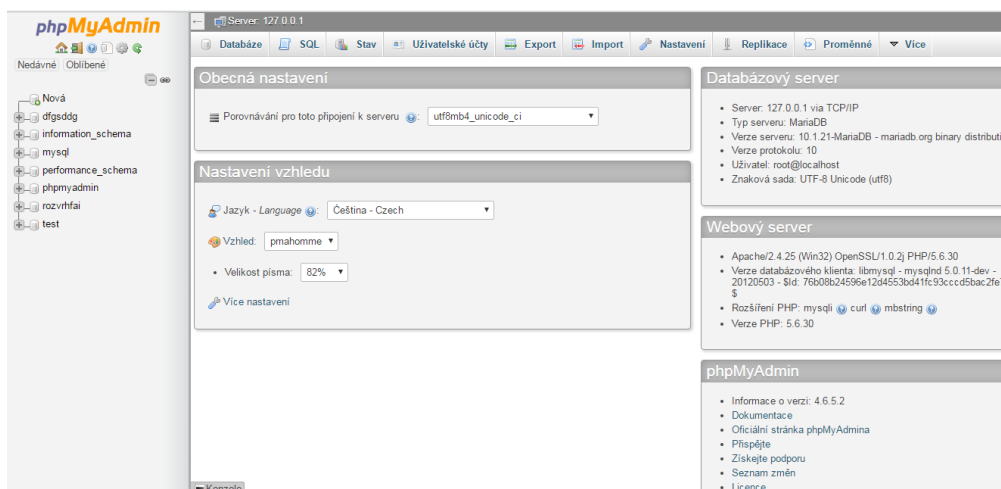
PHP má krom základních prvořadých funkcí ještě plno dalších a umí vytvořit anketu, jednoduchou kalkulačku, graf, knihu návštěv nebo diskuzní forum.



Obrázek 2: Schéma spolupráce PHP s databází [17]

3.1 phpMyAdmin

Tento velmi populární nástroj umožňující správu obsahu databáze MySQL byl založen v roce 2001. Jak je již z názvu zřejmé, je tento Open Source (šířený zdarma) nástroj napsaný v jazyce PHP a umožňuje uživateli vytvořit nebo naopak odstranit databázi, provádět SQL příkazy, správa tabulek (vytvoření, editace nebo odstranění) nebo spravovat klíče. Prozatím je k dispozici v 57 jazycích. [18] Na obrázku níže je vyobrazeno hlavní menu tohoto nástroje. <https://cs.wikipedia.org/wiki/PhpMyAdmin>



Obrázek 3: Hlavní stránka phpMyAdmin[19]

3.1.1 MySQL

MySQL je zkratka z angl. jazyka „*My Structured Query Language*“ a je to typ relační databáze, která vychází z deklarativního programovacího jazyka SQL. Nejčastěji se používá ve spojení s PHP pro webové aplikace menšího rozsahu. [20] <http://www.artic-studio.net/slovnicek-pojmu/databaze-mysql> Její velkou výhodou je rychlost, neboť právě na tu bylo MySQL od počátku optimalizováno a to i za cenu, že až donedávna měla jen jednoduché způsoby zálohování a nepodporovala triggerů nebo uložené procedury. [21] <http://cs.wikipedia.org/wiki/MySQL> Mezi uživatelsky nejoblíbenější patří také díky podpoře všech hlavních platforem, vysokému výkonu, kompatibilitě s jinými systémy (zejména Apache a PHP) nebo své jednoduchosti. MySQL velmi úzce spolupracuje s pokročilým nástrojem phpMyAdmin, kdy tento nástroj kompletně spravuje MySQL systém přes webové rozhraní. [22]

<http://www.adaptic.cz/znalosti/slovnicek/mysql/>

4 DYNAMICKÁ WEBOVÁ STRÁNKA

Je to v podstatě druh webové stránky, která obsahuje aktuální informace pro každé individuální zobrazení. Její obsah se mění v závislosti na čase, konkrétnímu uživateli nebo uživatelské interakci, kontextu nebo na jejich různých kombinacích. [23]

https://cs.wikipedia.org/wiki/Dynamick%C3%A1_webov%C3%A1_str%C3%A1nka Základem tvorby dynamické webové stránky je programovací technologie, tedy software (program) a data uložená v databázi. Program, který je umístěn na internetovém serveru poté dynamicky sestavuje a odesílá do prohlížeče uživatele webovou stránku a to na základě požadavku na její zobrazení. [24] <http://www.shopcentrik.cz/slovník/dynamicke-webove-stranky.aspx>

Dříve se dynamické webové stránky vyznačovaly pouze tím, že umožňovaly pravidelnou aktualizaci a změny v obsahu, ale v dnešní době mají ještě spoustu dalších vylepšení jako je vyhledávání informací uvnitř stránek podle klíčových slov, předkládání individuálního obsahu stránky po přihlášení návštěvníka, reakce na pohyb kurzoru nebo odesílání dat prostřednictvím formulářů. [25] <http://www.helpmark.cz/slovníkpojmu/32-dynamicke-webove-stranky>

Všeobecně vzato jsou dva druhy webových stránek dynamické a statické. Tyto dva druhy webových stránek se liší především tím, že zatímco dynamická stránka dokáže svůj obsah měnit, statická stránka tuto funkci postrádá, alespoň do té doby, než proběhne aktualizace ze strany správce. Statická stránka se tedy klientovi zobrazí přesně tak, jak ji naposled zanechal její správce. Dalším rozdílem mezi dynamickými a statickými webovými jsou data uložená v databázi, protože zatímco dynamický typ stránky je plně využívá a pracuje s nimi, statický typ nedokáže zobrazovat data uložená v databázi. [25] [26]

4.1 Typy dynamických webových stránek

Způsob jakým se mají dané informace na stránce aktualizovat a měnit, určují skripty, které jsou uloženy ve webových stránkách na serveru. Typy těchto stránek se potom odvíjejí od toho, kde zpracování těchto programů nebo skriptů probíhá. [27]

<http://olomoucky.denik.cz/podnikani/vyznam-a-tvorba-dynamickych-webovych-stranek.html>

4.1.1 Skriptování na straně klienta

Jedná se o procesy, které se spouštějí v internetovém prohlížeči-tedy na straně klienta. Po stažení webových stránek do internetového prohlížeče se tyto procesy spustí, což sice znamená minimální zatížení serveru a rychlost, ale nadržbou stranu mohou tyto procesy pracovat pouze s informacemi, které jsou v okamžiku spuštění skriptu stažené

v internetovém prohlížeči. Skriptovacími jazyky na straně klienta jsou například JavaScript nebo ActionScript.

4.1.2 Skriptování na straně serveru

Jak už s názvu vyplývá, veškeré skripty jsou spouštěny přímo na internetovém serveru. Internetový prohlížeč uživatele pošle na adresu serveru příkaz, ve kterém serveru sdělí svůj požadavek na to, co chce zobrazit, server tento požadavek vyhodnotí, provede naprogramovaný příkaz a poté uživateli vrátí výsledek jeho požadavku. Skripty na serveru výsledná data naformátují do HTML jazyka, který přečte všechny internetové prohlížeče. Procesy na straně serveru jsou oproti skriptování na straně klienta pomalejší. Programovacími jazyky na straně klienta jsou PHP, ASP nebo Perl.[27]

4.1.3 Skriptování na straně klienta i serveru

Jasným příkladem kombinace klienta i serveru u skriptování je AJAX technologie, která vyvíjí webové stránky pro dynamickou výměnu obsahu se stranou serveru a to bez výhradní nutnosti znovu aktualizovat webovou stránku (např. Google maps).[28]

https://cs.wikipedia.org/wiki/Dynamick%C3%A1_webov%C3%A1_str%C3%A1nka

5 PROGRAM ROZVRH

Tento úspěšný program pod specifickým názvem „Rozvrh“ slouží především pro tvorbu rozvrhů na všech typech škol a to od základních až po vysoké školy. Byl vytvořen panem RNDr. Lubomírem Červeným, který je expertem na softwary pro vývoj majetku, dále pak také vývoj programů pro všechny typy škol, tvorbu webových stránek, zastupování a inventarizaci. Je také autorem knihy „Zastupovanie“, kterou vydal v roce 1992.[29]

<https://www.zilina.sk/firmy/rndr-lubomir-cervený>

Rozvrh 4 není aktuální verzí programu. Je to verze z roku 2003 a mezitím program stihl projít ještě dvěma inovacemi až do té dnešní s názvem Rozvrh 6. Ovšem věnovat se budeme programu „Rozvrh 4.2“, protože tuto verzi v současnosti používá k tvorbě rozvrhu kombinovaného studia Fakulta aplikované informatiky Univerzity Tomáše Bati ve Zlíně.

Při tvorbě rozvrhu je dbáno jak na dodržování pedagogických požadavků, jako je například počet volných hodin v rozvrhu vyučujícího, tak na různé specifické požadavky každé školy. Program mimo jiné obsahuje účinné nástroje, které umožňují jednoduchým způsobem upravovat stávající rozvrh.

<http://tahaj.sme.sk/software/338#>

5.1 Popis programu Rozvrh 4

Co se týče programu „Rozvrh 4.2“ tvorba rozvrhu jako taková funguje na principu umísťování lístků do rozvrhové tabule. Program pracuje s algoritmem, který vytváří kompletní rozvrh, ale lze pomocí něj vytvořit také pouze určitou část rozvrhu. Uživatel může vytvořit více-týdenní rozvrhy, ale s maximem pěti-týdenního cyklu. Program dává možnost dodatečně přiřadit učebny do již vytvořeného rozvrhu nebo neomezeně slučovat či dělit skupiny zcela libovolně a posléze umožňuje snadný oboustranný přenos dat s programy MS Excel a Word. Uživatel má také možnost řadit nebo filtrovat údaje podle různých kritérií. Po vytvoření celého rozvrhu, program umožňuje tvůrci testování a zpětnou kontrolu částí souboru nebo celého rozvrhu. Export všech údajů ve formátu HTML. [30]

5.1.1 Výhody programu

Výhodou je jednoduché zadávání všech vstupních dat kde patří definování vyučujících, učeben, skupin, tříd, předmětů a budov přičemž program umožňuje neomezeně spojit nebo naopak rozdělit třídy na libovolné skupiny a poté následné testování a kontrola vytvořeného rozvrhu nebo jen jeho částí. Program je také schopen akceptovat individuální požadavky na rozvrh učitele nebo učební místnosti. Dalšími výhodami jsou funkce programu, které umožňují tvorbu rozvrhu na více fakultách a kontrolu času pro přesun mezi jednotlivými pavilony na fakultě, přehledný výpis úvazků a učební osnovy nebo funkce, které umožňují návrh a následnou tvorbu úvazků.

5.1.2 Nevýhody programu

Tato verze programu nemá síťovou verzi s přístupem na lokální nebo vzdálený server a také postrádá lokalizátor textů programu. Uživatel nemá možnost libovolně rozmístit datové a informační panely v okně. S touto verzí programu bohužel není možnost exportu dat do většiny dnešních formátů. Program je pouze ve slovenštině a neprobíhá zde automatická aktualizace programu. [30]

6 SEZNÁMENÍ S PROGRAMEM ROZVRH 4.1

V této kapitole se chci věnovat seznámení se s programem Rozvrh 4.2 RNDr. ČERVENÝ v praxi. Tato konkrétní verze je ve slovenštině.

6.1 Zadání základních údajů

Při úplně prvním použití je logicky nutné si vytvořit zcela nový soubor s projektem rozvrhu. Po vytvoření projektu poté uživatel může každou další změnu nebo aktualizaci rozvrhu upravovat v již existujícím souboru s rozvrhem.

Na začátek je důležité zadat klíčové údaje, aby se rozvrh mohl vytvořit. Údaje se týkají rozsahu rozvrhu na škole a je tedy potřebné zadat:

- počet týdnů
- počet vyučovacích dnů v týdnu
- počet hodin přes den
- začínající vyučovací hodinu

V případě popisu rozvrhu je možné označení hodin, dnů a týdnů.

6.2 Vytvoření nového souboru

V záložce „Súbor“ si otevřete možnost „Nový projekt rozvrhu“ a následně se objeví tabulka, kde musí být zadány potřebné údaje a po vyplnění údajů se soubor uloží na disk.

6.3 Úprava existujícího dokumentu

V záložce „Súbor“ klikneme na možnost „Otvorit“, poté se objeví tabulka, kde si uživatel může vyhledat existující soubor kdekoliv v počítači. Uživatel si také může program nastavit takovým způsobem, aby při spuštění otevřel aktuální rozvrh.

Pokud uživatel změní aktuální název souboru za nový, data v souboru nebudou aktualizována.

6.3.1 Nastavení automatického otevření stávajícího dokumentu

V záložce „Formát“ klikneme na příkaz „Možnosti“ a v případě že chceme otevřít projekt, na kterém jsme naposledy pracovali, zaškrtneme políčko „Otvorit naposledy používaný rozvrh“. Pokud chceme, aby program pokaždé, při spuštění otvíral jeden a tentýž projekt, zaškrtneme políčko „Vždy otvorit rozvrh s názvom:“ a následně jej vyhledat v počítači.

6.4 Ukládání souborů

Program Rozvrh 4.2 je nastaven na automatické ukládání projektů, jako ochrana v případě problémů s programem nebo počítačem. Uživatel si může libovolně nastavit časový interval pro ukládání dat, s ohledem na to, že čím menší interval ukládání, tím více dat může být v případě problému, zachráněno. Pokud dojde například k selhání systému, po opětovném spuštění programu se otevře naposledy uložený záložní dokument. Jakmile uživatel dokument uloží tzv. ručně, záložní automaticky uložený dokument se vymaže.

6.5 Rozvrhová tabulka

Rozvrhová tabulka v programu celkově vymezuje rozsah rozvrhu a určuje způsob označování školního roku. Samotná tabulka může být větší než samotný rozvrh a to samo o sobě umožňuje uživateli do rozvrhu zaznamenávat i kroužky a předměty mimo studijní plán. Maximální rozsah tabulky je:

- 5 týdnů v rozvrhu
- 7 dní v týdnu
- 32 vyučovacích hodin v jednom dni

Každý den při tvorbě rozvrhu může začínat kteroukoli hodinou, s tím že se při tvorbě zadá v poli „Prvá hodina v dni“ a totéž platí i o dnu v týdnu v poli „Prvý deň v týždni“.

6.6 Seznamy

Jeden řádek seznamu ukazuje definici jednoho záznamu ať už je to budova, učitel, třída, skupina nebo předmět. Pro každý záznam lze v jeho sloupcích zadat jeho ostatní vlastnosti, které by mohly být dále potřeba při kontrole nebo celkové tvorbě rozvrhu.

6.7 Lístky

Názvem Lístky se rozumí vyučovací hodina a její typ v konkrétní třídě. Je to vlastně obdobou manuálně tvořeného rozvrhu, kdy se lístek umístil vždy na rozvrhovou tabuli ne příslušnou hodinu. Barva a text jednotlivého lístku, je závislý na údajích, které jsou v lístku zadané.

V jednom lístku může být zadaná výuka pro vícero studijních skupin, pro které probíhá výuka ve stejném čase. V tomto případě se údaje třída, předmět, vyučující, skupina a budova píšou do samostatného řádku v tabulce. V rozvrhu je také možné libovolný lístek uza-

mknout. Potom takto uzamčený lístek není možné jakkoli manuálně nebo automaticky přesunout

6.7.1 Vazby mezi lístky v rozvrhu

Vazba mezi lístky určuje pro dva odlišné lístky podmínky umístění těchto konkrétních lístků do rozvrhu vyučujícího nebo třídy. Jsou čtyři druhy vazeb:

- lístek 1 „**následující hodinu po**“ lístek 2
- lístek 1 „**v jiný den než**“ lístek 2
- lístek 1 „**v ten samý den jako**“ lístek 2
- lístek 1 „**ne následující hodinu po**“ lístek 2

6.8 Úvazky

Jeden řádek ze seznamu úvazky představuje zápis jedné hodiny-lístku. Pokud je v jedné vyučovací hodině zadáno více studijních skupin, pak údaje jako jsou třída, předmět, vyučující, skupina a budova jsou pro každou studijní skupinu oddělené středníkem.

6.9 Rozvrh

V rozvrzích učeben, předmětů a vyučujících, může tvůrce rozvrhu zablokovat hodiny v, průběhu kterých nemůže být v daném rozvrhu vyučování. Požadavek na rovnoměrné odučení předmětu v týdnu je automatická a není potřeba ji dodatečně zadávat. Při jakékoli tvorbě rozvrhu (automatické i manuální) může tvůrce rozvrhu vyloučit kontrolu jednotlivých požadavků, pokud zadá podmínky kontroly.

6.9.1 Automatická tvorba rozvrhu

Automatickou tvorbu rozvrhu lze spouštět i několikrát vždy na libovolnou část rozvrhu. V tomto případě se generuje pokaždé jiná varianta rozvrhu. Pokud generátoru rozvrhu uživatel povolí přístup, tak může přemístit již existující a uložené lístky v rozvrhu, ovšem označit a vybrat jednotlivé lístky se musí manuálně. Automatická tvorba rozvrhu může být spuštěna na všechny lístky současně anebo pouze na určitou část lístků. Př. Pokud je na škole problém se sestavením rozvrhu pro třídu s počítači, lze spustit automatickou tvorbu nejdříve na předměty, ve kterých se pracuje s počítačem.

6.10 Export dat

V programu je možné exportovat seznamy a rozvrhy, nebo po vytvoření kopírovat údaje do Excelu nebo do Wordu.

6.10.1 Export seznamů

Všechny možné seznamy je možno uložit na disk ve formátu HTML. Včetně stránek s jednotlivými seznamy se na disk pokaždé uloží také hlavní stránka pod označením „Zoznam.htm“ a také stránka „Zoznam_menu.htm“ s nabídkami, které umožní uživateli zobrazit jednotlivé seznamy.

6.10.2 Export rozvrhů

Export rozvrhů ve formátu HTML dává možností jednoduchým způsobem ukázat školní rozvrh na webu. Na disk se krom stránek se samotnými rozvrhy uloží ještě Rozvrh.htm (hlavní stránka) a Rozvrh_menu.html a Rozvrh_css.css – tyto stránky obsahují definovaný styl pro zobrazení stránek a odkazy na jejich jednotlivé zobrazení.

Exportovaný rozvrh bude mít stejnou strukturu rozložení řádků a sloupců jako rozvrh zobrazený na monitoru. [30]

7 ROZVRH FAKULTY APLIKOVANÉ INFORMATIKY

Rozvrh je nedílnou součástí školního života. Fakulta aplikované informatiky používá verzi Rozvrhu RNDr. Červený. Odkaz na rozvrh je poté na samotných stránkách fakulty nebo je zobrazen na portále IS STAG. Export rozvrhu z portálu IS STAG je umístěn na stránkách školy v HTML a PDF. Vzhledem k tomu, že u některých prohlížečů je nutné nastavit kodování stránky ručně, doporučuje se prohlížeč Safari. [31]

7.1 Údaje v rozvrhu fakulty

Jako hlavní důležité prvky v rozvrhu jsou samozřejmě dny, ve kterých konkrétní výuka probíhá a dále také hodiny. Každý předmět zobrazený v rozvrhu má určité rozmezí, které musí být v rozvrhu vymezeno. Co se týká kombinovaného studia, je nutné, aby se v rozvrhu zobrazovalo také konkrétní datum a to zejména kvůli nepravidelnosti výuky. Údaje v rozvrhu fakulty jsou popisovány z pohledu studenta.

7.1.1 Skupina

Z počátku byl každý student přiřazen do určité skupiny, která měla daný rozvrh. Počátkem roku 2015 došlo ke změně, kdy v určitý čas byl odstartován na portále IS STAG, tzv. kroužkový předzápis, kdy se každý student přiřadil do skupiny, většinou podle toho jaký konkrétní rozvrh mu vyhovoval. Zkratky skupin zůstaly povětšinou stejné i po změně. Pokud se to týká presenční formy studia, je její název zkrácen do čtyř znaků:

- **ročník**- který v daném akademickém roce studuje
- **začínající písmeno oboru** – A=ITA, B=BTSM, I=IŘT
- **číslo skupiny** – v případě velké kapacity a také z důvodu společné výuky, byly vytvořeny skupiny 1, 2 příp. 3
- **rozdělení skupiny** – dodatečné rozdělení skupiny (X, Y příp. Z), kdy každá skupina je většinou rozdělena na X a Y a v rámci stejného čísla skupiny mají třídy X a Y podobný rozvrh a semináře jsou většinou společné.

U kombinovaného studia platí totéž, s tím rozdílem že se před zkratkou, která se také skládá z počátečních písmen studovaného oboru, vyskytuje kb (kombinované studium).

7.1.2 Předmět

Předměty jsou stavebním prvkem při tvorbě rozvrhu. Každý předmět na fakultě má určitý počet kreditů a řadí se do třech bloků A, B a C. Každý student musí tzv. „nasbírat“ plný počet kreditů aby mohl být připuštěn ke státní zkoušce. V rozvrhu jsou tyto předměty ještě barevně rozděleny a zobrazeny jako:

- a) seminář – na semináři většinou dochází ke skupinové spolupráci mezi studenty i učitelem, může docházet k dialogům nebo různým týmovým pracím. Seminář je povinný.
- b) cvičení – stejně jako seminář, je cvičení povinné. Cvičení je skupinová výuka a jde většinou o řešení příkladů nebo osvojení teoretických znalostí z přednášek – v praxi.
- c) přednáška – jedná se o přednes – spíše teoretické látky- vyučujícího studentům. U některých předmětů chybí. Může se dělit na přednášku v „sudém“ nebo „lichém“ týdnu. Účast na přednáškách je brána jako vlastní zájem studenta a bývá povětšinou nepovinná.

Prvek „Předmět“ se v rozvrhu i všeobecně dělí na:

- **Povinné** – tyto předměty jsou bez výjimky povinné a každý student je musí mít splněny, nebo nebude připuštěn ke státní závěrečné zkoušce. Povinné předměty jsou sice i „Cizí jazyk“ a „Sportovní aktivity“, ale pro pořádek, se tyto skupiny rozdělily podle bloků, kdy zcela povinné předměty jsou jednoznačně blok „A“.
- **Cizí jazyk** – požadavky na splnění cizích jazyků může mít každý studijní obor na fakultě jiná. Každopádně každý cizí jazyk je v rozvrhu označen zkratkou „CJV/PA“ poslední dvě písmena se odvíjejí od toho, jaký je to jazyk – tedy obvykle se poslední dvě písmena skládají ze začátečního písmene dotyčného jazyka a číslice, která symbolizuje úroveň např. CJV/PAS1 = španělština-úroveň 1
- **Sportovní aktivity – povinně volitelné** – mezi povinně volitelné předměty se řadí na Fakultě aplikované informatiky pouze sportovní aktivity, které musí mít každý student vyše druhého ročníku splněny (4 semestry včetně povinného plavání). Tyto předměty jsou skupiny B a mají hodnotu jednoho kreditu. Sportovní aktivity jsou všechny v rozvrhu zapsány s předponou MUTV/PS...konečná tři písmena má již každý sport jiná. Př. MUTV/PSAAE – Aerobik.

- **Nepovinně volitelné** – nepovinně volitelné předměty jsou zcela a jen na rozhodnutí každého studenta. Jedná se většinou o předměty směřované na větší zájem studenta o konkrétní předmět, který je v nabídce nebo na přípravu studenta. Bonusem po absolvování nepovinně volitelného předmětu může být i pět kreditů.

7.1.3 Vyučující

Vyučující se v rozvrhu zobrazuje většinou pouze pod příjmením a po označení nebo kliknutí kurzorem myši se objeví detail, který zobrazuje jméno, příjmení a tituly daného kantora.

7.1.4 Učebna

Dalším velmi důležitým údajem v rozvrhu je samozřejmě informace o učebně, kde se bude daná výuka nacházet. Většina učeben, ve kterých v rámci fakulty aplikované informatiky probíhá výuka, se nacházejí na půdě fakulty, ovšem na škole vyučuje spousta externích učitelů a někdy se výuka odehrává na půdě jiné školy. Rozvrh má povinnost informovat studenta o tom, kde se daná učebna nachází (patro, fakulta, příp. adresa). [32]

II. PRAKTICKÁ ČÁST

8 NÁVRH OPTIMALIZACE ZOBRAZENÍ ROZVRHU

Návrh na optimalizaci zobrazení rozvrhu pro kombinované studium by měl odpovídat požadavku na spojené zobrazení rozvrhu skupiny, učitele a místnosti. To vše by se mělo zobrazovat na simulovaných webových stránkách fakulty aplikované informatiky. Cílem požadavku je zjednodušení zobrazování rozvrhů bez dlouhého vyhledávání rozvrhů skupiny, učitele a místnosti.

Na obrázku č. 4 je zobrazen jeden z rozvrhů pro studijní třídu kombinovaného studia. Pokud je požadováno zobrazení rozvrhu pro konkrétní studijní skupinu/třídu, nese v sobě každé pole informaci o názvu předmětu, vyučujícím a třídě kde bude výuka probíhat.

Rozvrhy tříd 1.BT 1. roč. - Bezpečnostní technolo

1.BT 1. roč. - Bezpečnostní technologie, systémy a management

Letní semestr 2016/17 - Kombinované studium, verze 011

LS - 2015/16	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21
pátek	3.2-5													
sobota	4.2-5													
pátek	10.2-6	C.JV/KA2A Outěhcká 51/107 C.JV/KA2N Voháňková 51/119 C.JV/KA2R Záhorská 54/108	Sližová	TUFMI/AQMET 51/220				AUM/AQMAA Sedláček 51/107						
sobota	11.2-6	Macků	AUBI/AQELO 51/219				AUM/AQMAA Sedláček 51/107							
pátek	17.2-7													
sobota	18.2-7													
pátek	24.2-8.	Ivanka	AUBI/AQMZS 51/220				TUFMI/AQMET Sližová 51/220							
sobota	25.2-8	Koňálová	AUPKS/AQPRO 51/218				AUBI/AQELO Macků 51/219							

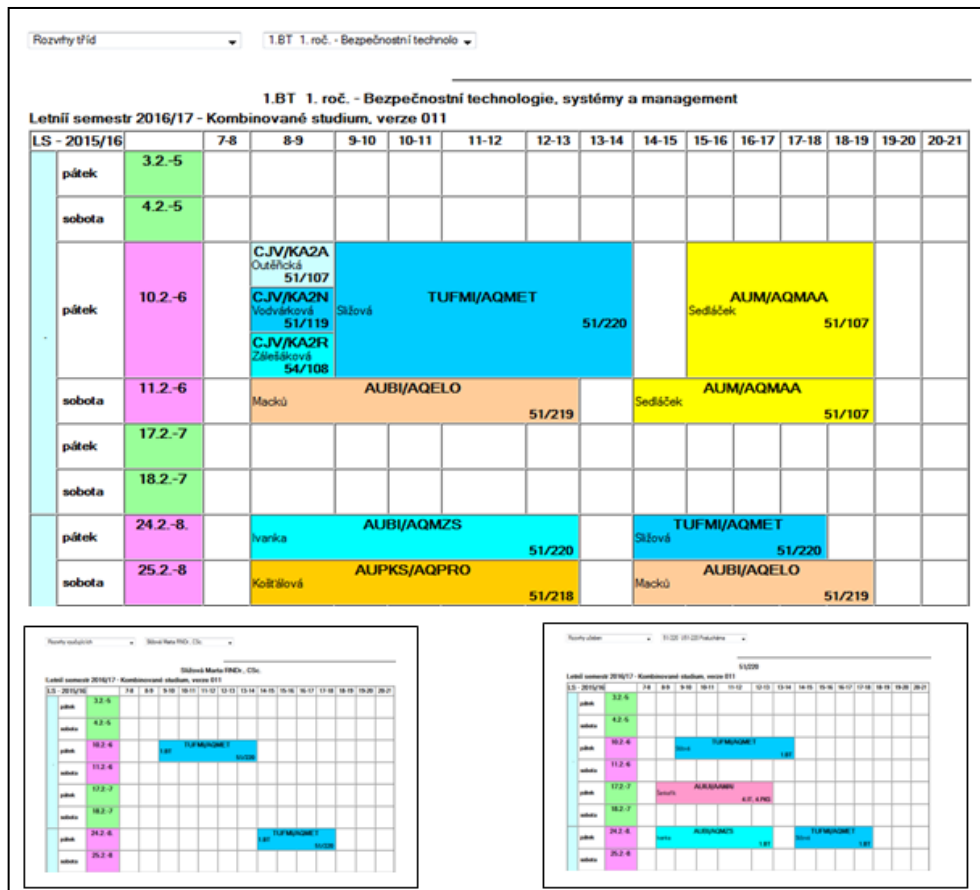
Obrázek 4: Rozvrh kombinovaného studia[19]

V tomto rozvrhu nás zajímá např. pole vyznačené na obrázku č. 5 s předmětem, který vyučuje RNDr. Marta Sližová, CSc. v místnosti č. 51/220. Pokud máme jako hlavní zobrazený rozvrh třídy/skupiny, tak pro optimalizaci zobrazení nás tyto dva údaje budou zajímat nejvíce.



Obrázek 5: Konkrétní pole rozvrhu [19]

Optimalizace zobrazení v tomto případě spočívá v tom, že po kliknutí kurzorem na příjmení vyučujícího (v tomto případě paní Sližové) se pod hlavním rozvrhem skupiny objeví její rozvrh. Totéž platí u kliknutí na číslo učebny, kdy se nám ukáže veškerá výuka, která bude v této místnosti probíhat. Zobrazení bude tvořit vždy jeden hlavní rozvrh, který byl uživatelem zvolen na první výběrové liště tedy „Rozvrhy tříd, „Rozvrhy vyučujících“ nebo „Rozvrhy učeben“ a pod ním se vedle sebe budou zobrazovat dva další rozvrhy v závislosti na volbě hlavního rozvrhu. Na obrázku č. 6 je konečný návrh optimalizace zobrazení rozvrhu. Je zde znázorněn tzv. hlavní rozvrh (v tomto případě rozvrh skupiny) a pod ním je konkrétní zobrazení rozvrhů vyučujícího a třídy, které jsme označili z libovolného pole v tomto rozvrhu.



Obrázek 6: Návrh zobrazení rozvrhů [19]

Pokud si v horní liště zvolíme jako hlavní zobrazení rozvrhu učebny, pod ním se ukáží rozvrhy vyučujícího a studijní skupiny/třídy. Ten samý princip platí také u hlavního rozvrhu vyučujících.

9 TVORBA WEBOVÝCH STRÁNEK FAKULTY

9.1 Instalace potřebných programů

Jako základ pro tvorbu webové stránky jsem si potřebovala vytvořit server, zvolila jsem si program XAMPP, hlavně kvůli jeho jednoduchosti a offline přístupu.

9.1.1 XAMPP

Název tohoto multiplatformního balíku je odvozen od jeho vlastností a hlavních komponentů, které obsahuje (X-multiplatformní, Apache, MySQL, PHP a Perl).

[33] <http://www.itbiz.cz/slovník/informacni-technologie-it/xampp>

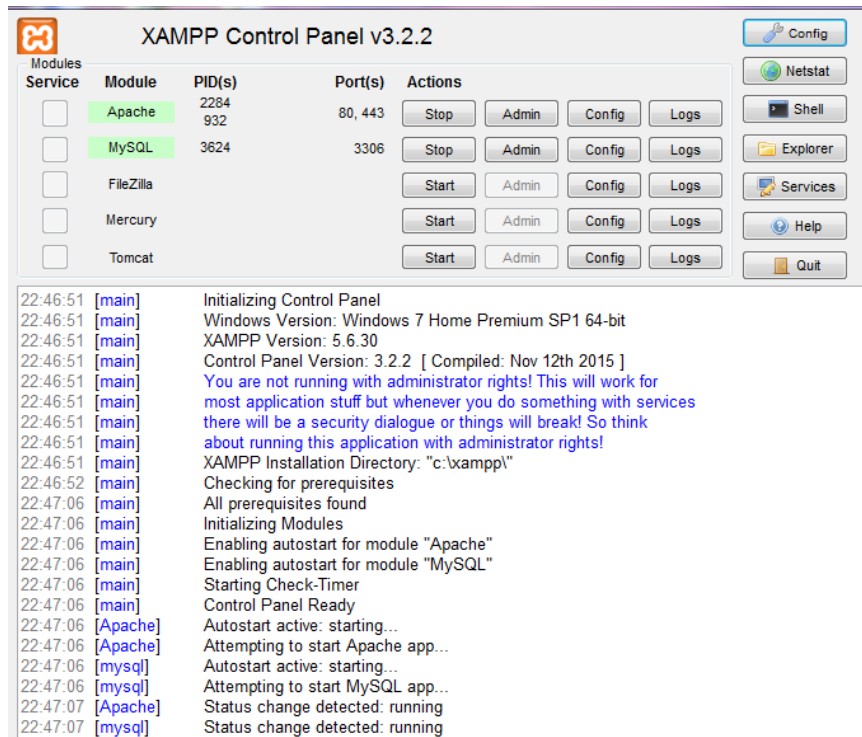
XAMPP umožňuje uživatelům velmi jednoduchou cestou a hlavně rychle vytvořit server z kteréhokoliv počítače a přímo na něm testovat bez jakýchkoli omezení uživatele nebo operačního systému webové aplikace, weby nebo redakční systémy. Tento multiplatformní balík vyvinutý společností Apache umožňuje velmi jednoduchou a rychlou instalaci následujících komponentů: [34] <http://www.instaluj.cz/xampp>

- PHP
- Apache HTTPD, MariaDB,
- phpMyAdmin
- XAMPP Control Panel, Webalizer,
- MiniPerl, Openssl,
- Mercury Mail Transport System, FileZilla FTP Server,
- SQLite, ADODB,
- Zend Optimizer,
- XAMPP Security a Ming,
- další komponenty

Důvod úspěšnosti XAMPPu je v první řadě jeho jednoduchost a offline server. V neposlední řadě také uživatelům jistě vyhovuje to, že požaduje minimální nároky na nastavení a také, že se uživatel vyhne zdoluhavému instalování jednotlivých komponentů, které v sobě XAMPP má obsaženy. [35] <https://en.wikipedia.org/wiki/XAMPP>

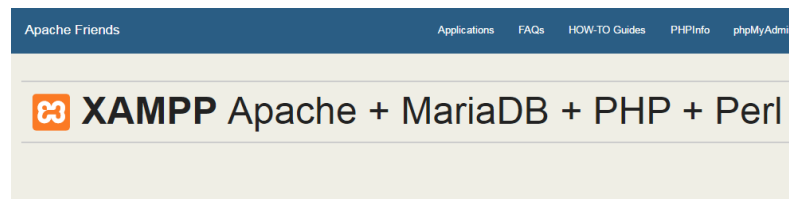
Po dokončení instalace se program spustí kliknutím na ikonu „xampp-control.exe“, která se nachází v nainstalované složce XAMPPu. Hned na to se objeví na obrazovce okno že byl XAMPP spuštěn, důležité je aby nástroje Apache a MySQL byly v zeleném políčku (viz obr. 7) to znamená, že byly spuštěny spolu s programem XAMPP. Tímto spuštěním to

ale nekončí, uživatel musí ještě ve svém prohlížeči zadat adresu **http://localhost/** aby se přesvědčil, jestli instalace proběhla úspěšně.



Obrázek 7: Kontrolní panel XAMPPu [19]

Pokud byla instalace úspěšná, objeví se po zadání **http://localhost/** úvodní stránka s menu a možnostmi pro další práci (viz obrázek č. 8). Toto menu je klíčové a pro správné konečné zobrazení rozvrhů na webové stránce důležité.



Welcome to XAMPP for Windows 5.6.30

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.

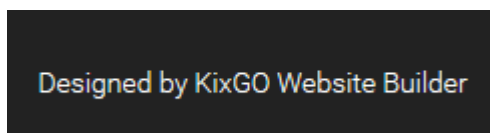
Obrázek 8: Menu programu XAMPP [19]

Jako další důležitá věc je výběr a stažení vhodné šablony (viz obr. 9) pro webovou stránku, která se následně otevře a nadále se upravuje.



Obrázek 9: Šablona pro tvorbu webové stránky [19]

Šablona byla vytvořena tvůrcem webu KixGO a v licenci pro tuto šablonu je uvedeno, že požaduje, aby byl na stránkách zachován odkaz na stránky www.kixgo.com. Při tvorbě webových stránek jsem poukázala na tuto skutečnost, zmínkou ve spodní části webových stránek což je zobrazeno na obrázku č. 10.



Obrázek 10: Zmínka o tvůrci šablony na webových stránkách [19]

Tato šablona se po stažení otevře v programu 7- Zip, všechna data se tedy musí importovat. Importovala jsem si je do nově vytvořené složky v souboru Xampp, abych je následně mohla otevřít pro úpravu.

9.1.2 PSPad

Pro práci s HTML kódem jsem zvolila program PSPad. Program vypadá jako obyčejný textový řádek, s tím že umožňuje o něco důmyslnější zadávání tagů. Je to původní český editor a zároveň také jeden z nejpoužívanějších strukturních editorů. Využívá se především

k tvorbě dynamických nebo jinak specifických stránek. Výhodou je mnoho užitečných nástrojů. [36] <http://www.pspad.com/cz/>

Ve složce se nacházely čtyři hlavní html soubory, které jsem následně otevřela v PSPadu a upravovala jejich obsah (viz obr. 11).

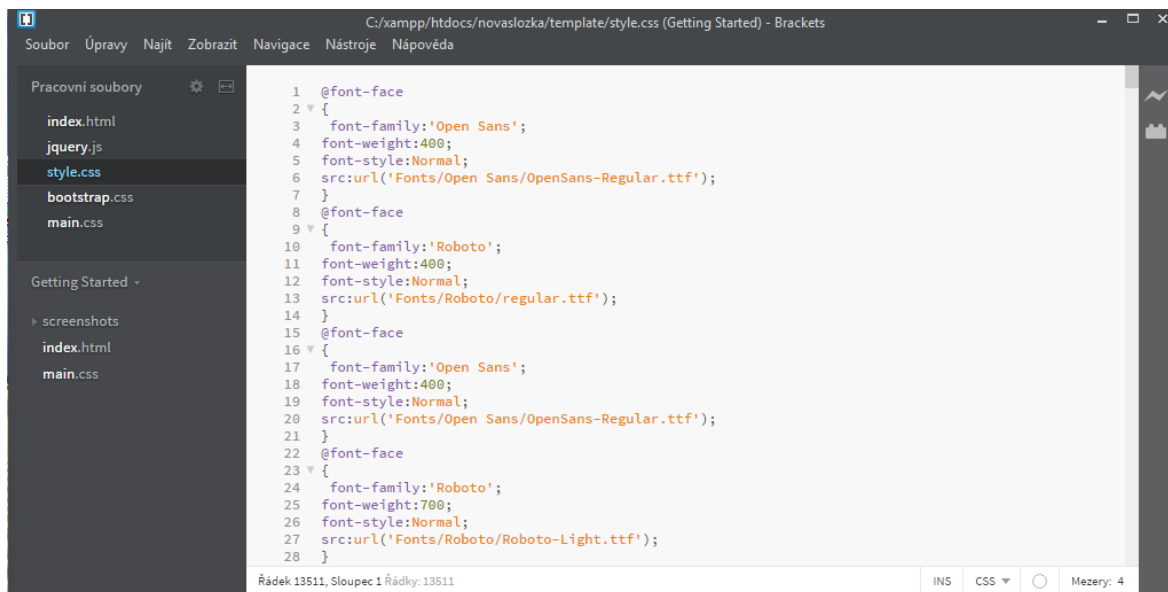
```
</head>
<body class="Home">
  <div class="totopshow">
    <a href="#" class="back-to-top">
      </a>
    </div>
  <div id="ttr_page" class="container">
    <header id="ttr_header">
      <div id="ttr_header_inner">
        <div class="ttr_headershape01">
          <div class="html_content">
            <p style="margin:0em 0em 0em 0em;line-height: normal;"><br style="font-size:0.571em;" />
            </p>
            <p style="margin:0em 0em 0em 0em;text-align:center;line-height: normal;">
              <span style="font-family:'Roboto','Arial';font-weight:300;color:rgba(153,153,153,1);">Nad Stráněmi 4511, 760 01, Zlín
            </span>
            </p>
          </div>
        </div>
      <div class="ttr_headershape02">
        <div class="html_content">
          <p style="margin:0em 0em 0em 0em;text-align:center;line-height: normal;"><br style="font-size:0.571em;" />
          </p>
          <p style="margin:0em 0em 0em 0em;text-align:center;line-height: normal;">
            <span style="font-family:'Roboto','Arial';font-weight:300;color:rgba(153,153,153,1);">Tel.: +420 57-603-5221
          </span>
          </p>
        </div>
      </div>
    </div>
  </div>
</body>
```

Obrázek 11: Prostřední v programu PSPad [19]

9.1.3 Brackets

Je to open source editor pro úpravu vzhledu webu, který byl původně vytvořen Adobem. Jeho vývoj je od počátku založen na takových technologiích, jako jsou JavaScript, HTML nebo CSS. Brackets je dostupný pro operační systémy Linux, Windows a Mac OS X. Brackets má výhodu rychlého editačního modu anebo možnost přenést a zobrazit změny okamžitě v prohlížeči, aniž by byly uloženy změny nebo byl stav jakkoli aktualizován. [37]

Tento editor se mi zdál jako neoptimálnější neboť je open source a upravený kod lze ihned přenést do webového prohlížeče. Po instalaci jsem upravovala vzhled prostřednictvím tohoto editoru jak souborech *css* tak *html* nebo JavaScript. Na obrázku č. 12 je pro ukázkou vyobrazeno prostředí editoru Brackets.



Obrázek 12: Prostředí v programu Brackets [19]

9.1.4 Komunikace se serverem užitím PHP a JavaScript

Na obrázku č. 13 lze vidět funkci napsanou v PHP, která vrací hodnoty pro vytvoření rozvrhové tabulky podle učebny. Jako vstupní argumenty se očekává název databáze, připojení klienta k MySQL serveru a skratka učebny, která slouží k vyselektování správných dat z databáze. Na řádku 118 až 143 je zapsán příkaz SQL, který bude odeslán serveru, na což vrátí požadované data, definované na řádcích 118 až 126 (datum, název předmětu, třídy, vyučujícího, atd.). Tento příkaz je vykonán příkazem „mysql_query“ (řádek 144), který vrací výsledek do proměnné „result“. Pokud by server nevrátil výsledek, nebo by nastala nějaká chyba, vrátí funkce chybovou hlášku (řádky 145 až 148). Pokud funkce obdržela požadovaný výsledek, je třeba, abychom data získaly v požadované formě. Na řádku 150 začíná cyklus „while“, který prochází všechny obdržené výsledky řádek po řádku. Pro jeden záznam (řádek) je použita proměnná „row“, do které pomocí funkce „mysql_fetch_assoc“ jsou brány řádky z výsledků (z proměnné result). Následně použitím funkce echo vracíme hodnoty, které se ve výsledku (v js souboru) projeví, jako textový řetězec. Abychom z řádku dostali požadované záznamy, jako je třeba datum v daném sloupci, je třeba získat tuto hodnotu následujícím příkazem: „\$row[„datum“]“, tedy stejným názvem, jako jsme si pojmenovali ve funkci SELECT (řádek 118).


```
116 function getRozvrhUcebna($mysql_database, $bd,$skratkaUcebna){
117     mysql_select_db($mysql_database, $bd) or die("Oops some thing went wrong");
118     $sql = "SELECT datum.datum AS datum,
119         hodiny.skratka AS hodinaSkratka,
120         datum.den AS nazevDne,
121         predmety.skratka AS predmet,
122         predmety.nazov AS nazevPredmetu,
123         tridyZoznam.nazov AS nazevTridy,
124         tridyZoznam.skratka AS zkratkaTrida,
125         vyucujici.meno AS menoVyucujici,
126         vyucujici.priezvisko AS priezviskoVyucujici
127 FROM rozvrh.rozvrh_triedy tridy
128 JOIN rozvrh.rozvrh_akce akce
129 ON akce.rozvrh_akceID=tridy.rozvrh_akceID
130 JOIN zoznamy.zoznam_triedy tridyZoznam
131 ON tridy.zoznam_triedyID=tridyZoznam.zoznam_triedyID
132 JOIN rozvrhova_tabula.datum datum
133 ON datum.datumID=akce.datumID
134 JOIN rozvrhova_tabula.hodiny hodiny
135 ON hodiny.hodinyID=akce.hodinyID
136 JOIN zoznamy.zoznam_predmety predmety
137 ON predmety.zoznam_predmetyID=akce.zoznam_predmetyID
138 JOIN zoznamy.zoznam_vyucujuci vyucujici
139 ON vyucujici.zoznam_vyucujuciID=akce.zoznam_vyucujuciID
140 JOIN zoznamy.zoznam_ucebne ucebna
141 ON ucebna.zoznam_ucebneID=akce.zoznam_ucebneID
142 WHERE ucebna.skratka= '$skratkaUcebna'
143 ORDER BY datum.datum";
144     $result=mysql_query($sql);
145     if (!$result) {
146         $message = '<b>Invalid query:</b> ' . mysql_error() . "<br>";
147         $message .= '<b>Whole query:</b> ' . $sql. "<br>";
148         die($message);
149     }
150     while($row = mysql_fetch_assoc($result)) {
151         echo $row["datum"];
152         echo "<t>";
153         echo $row["hodinaSkratka"];
154         echo "<t>";
155         echo $row["nazevDne"];
156         echo "<t>";
157         echo $row["predmet"];
158         echo "<t>";
159         echo $row["zkratkaTrida"];
160         echo "<t>";
```

Obrázek 13: Příklad funkce vytvořené v PHP [19]

Funkce „getRozvrhUcebna“ je volána ze souboru „getRozvrhUcebna.php“. Kód tohoto souboru je na obrázku č. 14. Jako argument získávaný pro další zpracování je zkratka učebny (viz přechodí popis funkce). V tomto souboru jsou uložena přístupová práva k serveru s databázemi. Z bezpečnostního hlediska, by zde tyto informace být neměli a bylo by vhodné užití přihlašovacího formuláře. Domnívám se však, že v tomto konkrétním případě, jelikož nepracujeme s citlivými daty, není problém, aby zde byl uveden konkrétní uživatel. Pro zvýšení bezpečnosti však lze upravit práva pro vybraného uživatele na serveru (myPhpAdmin). Připojení k serveru zajišťuje funkce „mysql_connect“. Tato funkce požaduje zadání názvu hostu, na kterém se nalézá databáze, dále uživatelské jméno a jeho heslo. Na řádce 14 je pak nastavena znaková sada utf8, abychom zachovali používanou diakritiku pro češtinu.

```
1 <?php
2 $mysql_database = "zoznamy";
3 $xmlDoc = new DOMDocument();
4 $mysql_hostname = "localhost"; // Example : localhost
5 $mysql_user = "root";
6 $mysql_password = "";
7 $ucebna = "";
8 if(isset($_POST['arg0']) && !empty($_POST['arg0'])) {
9     $ucebna = $_POST['arg0'];
10 }
11 include 'myFunctions.php';
12
13 $bd = mysql_connect($mysql_hostname, $mysql_user, $mysql_password) or die("Oops some thing went wrong");
14 mysql_set_charset('utf8', $bd);
15
16 echo getRozvrhUcebna($mysql_database, $bd, $ucebna);
17
18 ?>
```

Obrázek 14: Kód souboru „getRozvrhUcebna.php“ [19]

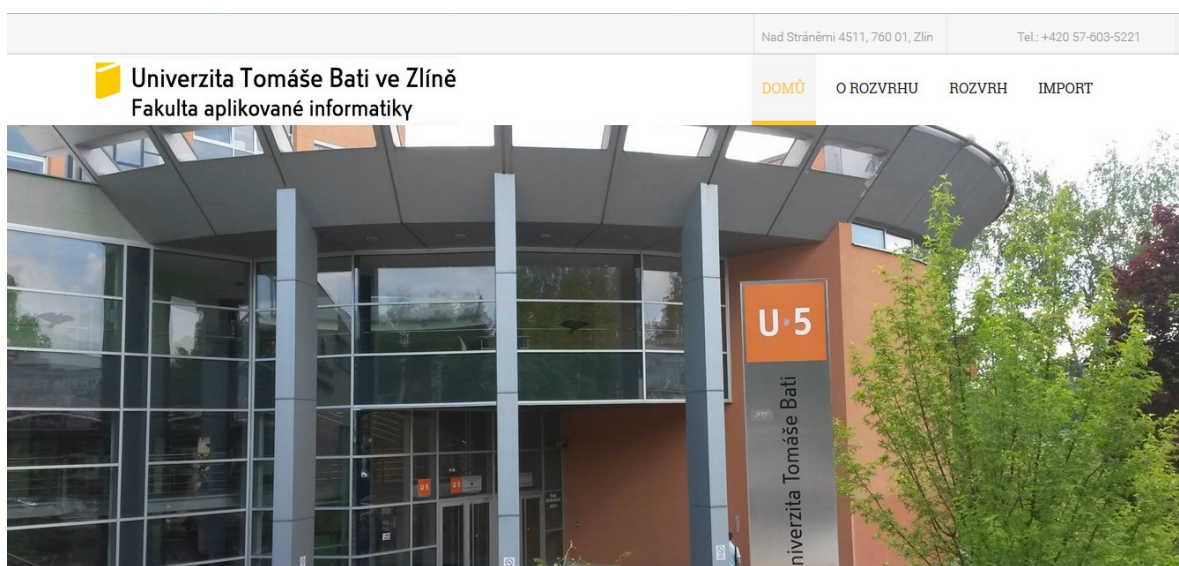
Pro získání takových dat přímo pomocí JavaScriptu, je možné využít například ajax, který byl využit i v mé práci (viz obr. 15). Jako parametr url, je název PHP souboru, který chceme volat. Jako data je použit objekt s argumenty. V mém případě „selectedTrida“ je textový řetězec se zkratkou učebny. Jako typ je zvolen „POST“, čímž očekáváme načtení dat ze serveru. Jelikož pro náš případ není vhodné, aby tento příkaz běžel asynchronně, je „async“ nastavena na hodnotu „false“. Jako parametr „complete“ je použita funkce, která se spustí po obdržení výsledků z databáze. V případě chyby je jako parametr „error“ nastavena funkce pro vyskočení okýnka se zprávou. kód

```
92     $.ajax({
93         url: 'getRozvrhUcebna.php',
94         data: { arg0: selectedTrida },
95         type: 'POST',
96         async: false,
97         complete: function (response) {
98             stringRozvrh = response.responseText;
99         },
100        error: function () {
101            alert('Bummer: there was an error!');
102        }
103    });
```

Obrázek 15: Příklad využití ajaxu [19]

9.2 Vizualizace webových stránek fakulty

Pro zobrazení webových stránek fakulty (viz obr. 16) postupujeme klasicky spuštěním kontrolního panelu programu XAMPP a zadáním adresy **http://localhost/template**.



Obrázek 16: Webové stránky fakulty [19]

Na webové stránce jsou zobrazovány fotky fakulty z vlastního zdroje, které se v průběhu dvou sekund vždy obměňují. V pravém horním rohu je umístěna adresa fakulty s kontaktem na sekretariát. V navigační liště se jako další nachází záložka „O rozvrhu“ kde je stručně popsán rozvrh RNDr. Červený jeho ovládání a typy. Vedle se nachází záložka „Rozvrh“, která bude zobrazovat samotný rozvrh ihned po úspěšném importu, který bude probíhat na vedlejší záložce s názvem „Import“.

10 DATABÁZE

Jako další krok při tvorbě práce, je vytvoření SQL databáze, která bude tvořena základními prvky rozvrhu (předmět, vyučující, čas, datum, třídní skupina a učebna). Celek budou tvořit tři hlavní databáze „Rozvrh“, „Rozvrhová tabuľka“ a „Zoznamy“ a ty budou obsahovat tabulky, zaplněné konkrétními daty.

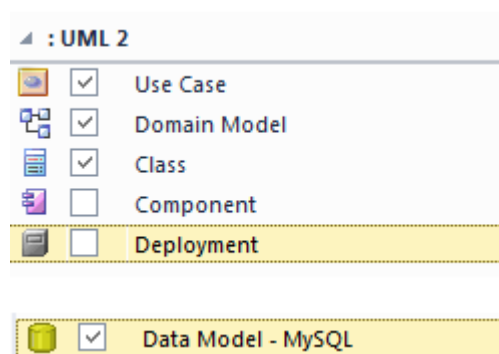
10.1 Databázový diagram

Ještě před samotnou tvorbou databáze jsem si vytvořila databázový diagram, abych posléze měla o něco jednodušší tvorbu samotného modelu rozvrhu. Databázový diagram jsem vytvořila v programu Enterprise Architect.

10.1.1 Enterprise Architect

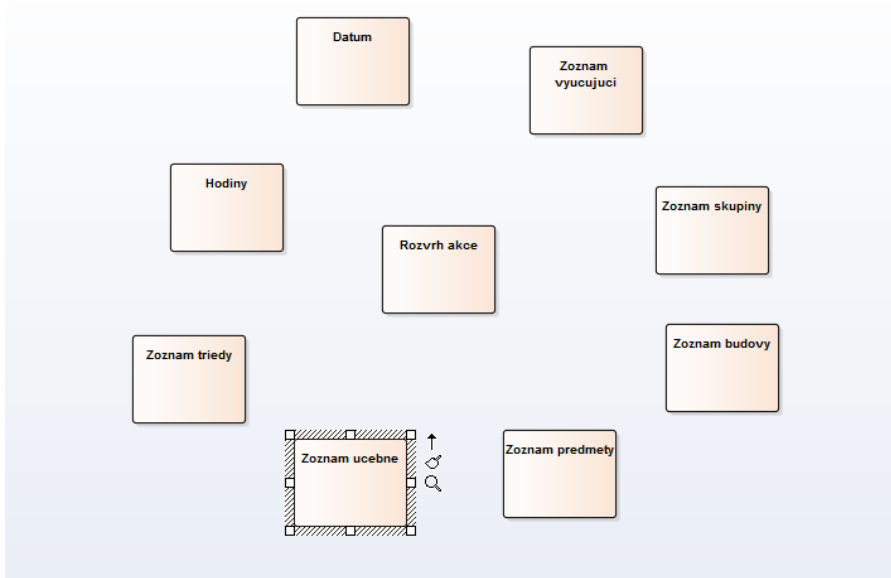
Enterprise Architect (EA) byl vyvinut a kolem r. 2000 i publikován australskou softwarovou společností Sparx Systems. Je to profesionální nástroj pro jednoduchou tvorbu vývojových diagramů a schémat potřebných při tvorbě aplikací. Při modelování využívá jazyka UML. [38]<https://www.slunecnice.cz/sw/enterprise-architect/>

Po instalaci nástroje Enterprise Architect jsem si vytvořila nový projekt a zadala požadavky k vytvoření diagramu a databáze (viz obr. 17).



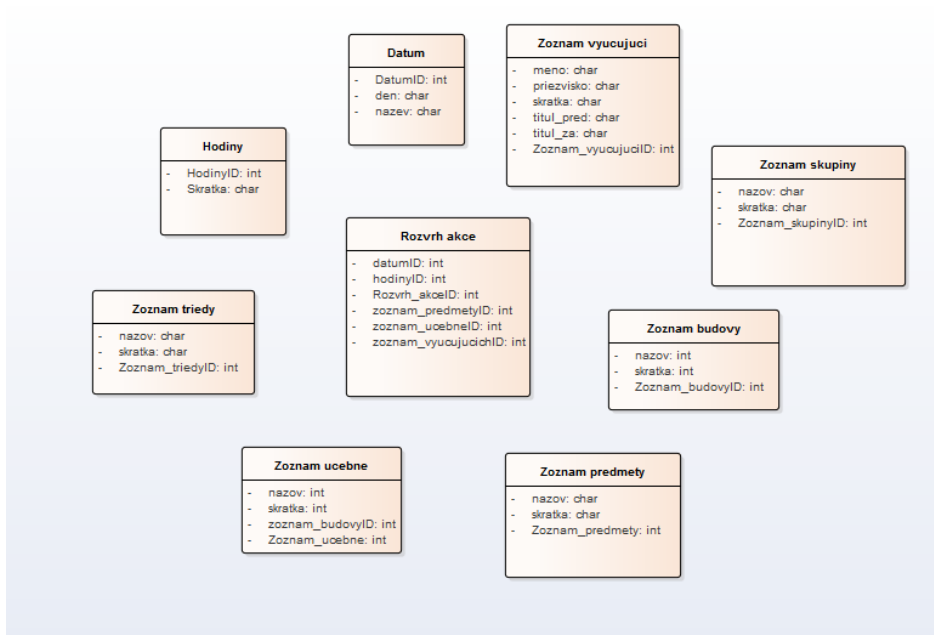
Obrázek 17: Požadavky na vytvoření diagramu a databáze [19]

Poté se pracuje s položkou „Class Model“ neboli s Modelem tříd, kde se prostřednictvím prvku „Class“, který slouží k vytvoření políček resp. tříd, vytvoří devět políček (viz obr. 18).



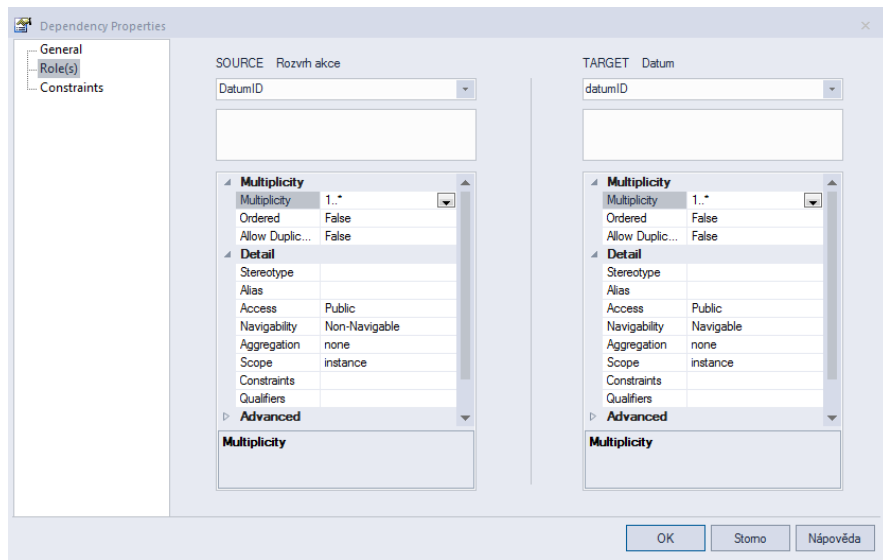
Obrázek 18: Vytvoření základních tříd v EA [19]

Po vytvoření tříd jsem do políček zadala jednotlivé atributy. Atributy se do tříd vkládají, pravým kliknutím myši na konkrétní pole, kde se následně objeví možnosti jeho úpravy a v záložce „Features & Properties“ → „Attributes“ vytvoříme nový atribut. Při zadávání atributů, požaduje program zadat datový typ (int, char), kvůli pozdějšímu vkládání dat. Jak lze vidět na obrázku č. 19, atributy v tomto případě simulují to, jaká data budou vkládána do samotné databáze

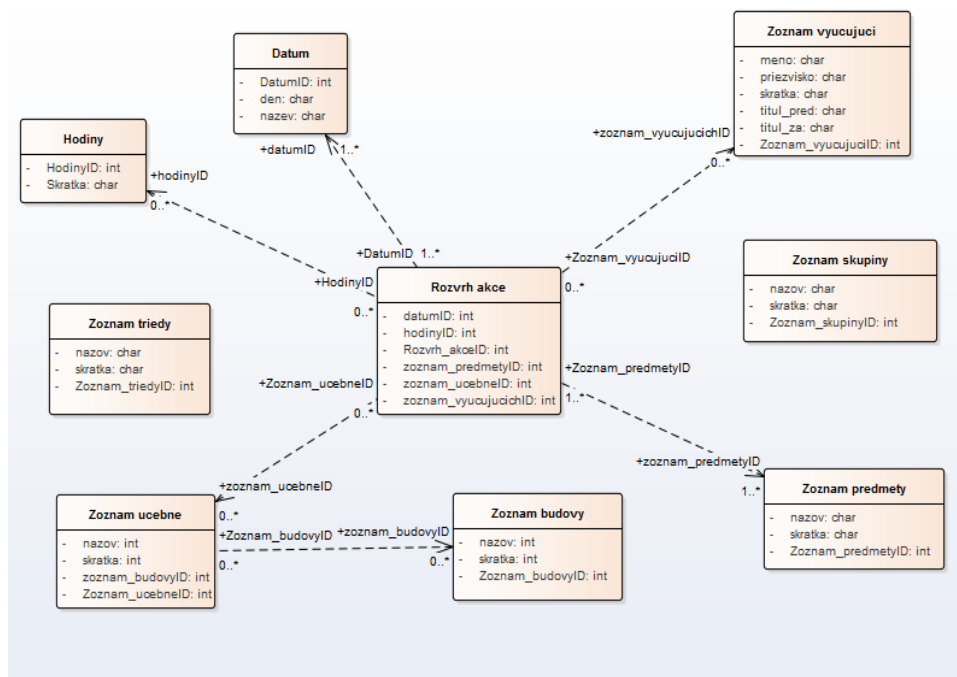


Obrázek 19: Třídy s vloženými atributy [19]

V tomto případě každá položka má svůj ID (identifikátor) kvůli vkládání dat, ale mimo jiné také kvůli vazbám mezi jednotlivými třídami. V tomto případě je použita vazba „Dependency“ neboli „Závislost“ (viz obr. 20). Tato vazba znamená, že jeden nebo více prvků jednoho modelu, vyžaduje další prvky modelu druhého, pro jejich implementaci nebo specifikaci.



Obrázek 20: Příklad zadávání parametrů pro zobrazení relace [19]



Obrázek 21: Vazby mezi jednotlivými třídami [19]

Na hotovém modelu (obr. 21) se vyskytují třídy, které mezi sebou mají vazby založené na vzájemné „závislosti“.

10.2 Tvorba databáze

Při tvorbě databáze pomůže opět již nainstalovaný program XAMPP, kdy se nám zadáním adresy **http://localhost/** úvodní stránka a menu, kde si vybereme možnost **phpMyAdmin** (viz obr. 22).



Welcome to XAMPP for Windows 5.6.30

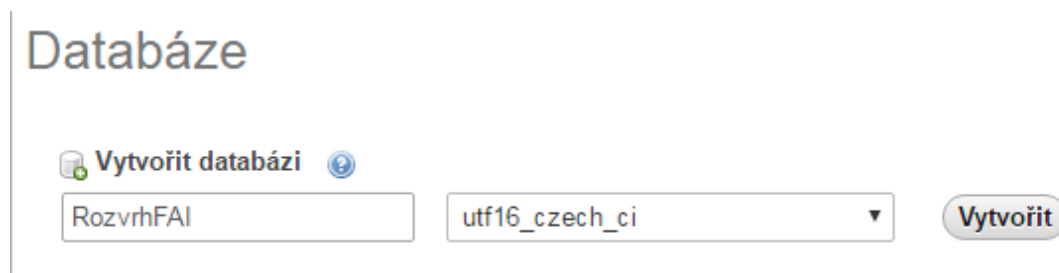
Obrázek 22: Ikona phpMyAdmin v menu programu XAMPP [19]

Po kliknutí na toto pole se nám zobrazí hlavní nabídka s obecnými nastaveními a možnými změnami vzhledu. Pro vytvoření databáze je v tomto menu horní záložka s jasným názvem „Databáze“ a na ni klikneme. Jednotlivé databáze lze buďto vytvořit způsobem, který je vyobrazen na obrázku č. 23.

Databáze

Obrázek 23: Vytvoření databáze [19]

Následně zadáme jméno databáze (viz obr. 24)



Obrázek 24: Zadání názvu databáze [19]

Po vytvoření nové databáze se nám automaticky objeví pole pro zadání názvu pro tabulku. V tomto ohledu byl velmi užitečný počáteční databázový diagram, na kterém byl jasně popsán název budoucí tabulky a její struktura. Každá tabulka by měla mít primární klíč, jinak nebudeme schopni jednoznačně určit konkrétní záznam.

Jiný způsob pro vytváření databáze je použití SQL příkazů, což bylo využito při této práci. Na obrázku č. 25, lze vidět část použitého kódu, ve kterém jsem vytvořila databázi „rozvrh“ a tabulku „rozvrh_akce“. Jako primární klíč jsem použila název „rozvrh_akceID“ a dále ID potřebných atributů, jako cizí klíče, které se odkazují na příslušné tabulky.

```
CREATE DATABASE rozvrh;
USE rozvrh;
CREATE TABLE rozvrh_akce(
  rozvrh_akceID int NOT NULL AUTO_INCREMENT,
  datumID int,
  hodinyID int,
  zoznam_vvucujuciID int,
  zoznam_predmetvID int,
  zoznam_ucebneID int,
  PRIMARY KEY(rozvrh_akceID),
  CONSTRAINT FK_ucebne FOREIGN KEY (zoznam_ucebneID)
  REFERENCES zoznamv.zoznam_ucebne(zoznam_ucebneID),
  CONSTRAINT FK_datum FOREIGN KEY (datumID)
  REFERENCES rozvrhova_tabula.datum(datumID),
  CONSTRAINT FK_hodiny FOREIGN KEY (hodinyID)
  REFERENCES rozvrhova_tabula.hodiny(hodinyID),
  CONSTRAINT FK_vvucujuci FOREIGN KEY (zoznam_vvucujuciID)
  REFERENCES zoznamv.zoznam_vvucujuci(zoznam_vvucujuciID),
  CONSTRAINT FK_predmetv FOREIGN KEY (zoznam_predmetvID)
  REFERENCES zoznamv.zoznam_predmetv(zoznam_predmetvID)
) DEFAULT CHARSET=utf8;
```

Obrázek 25: Část použitého kódu pro vytvoření databáze [19]

10.3 Struktura databází

Databáze pro optimalizaci zobrazení rozvrhů byly vytvořeny tři a to „rozvrh“, „rozvrhová tabuľa“ a „zoznamy“, každá z nich obsahuje tabulky a ty obsahují konkrétní data.

10.3.1 Struktura databáze „Rozvrhová tabuľa“

Databáze „Rozvrhová tabuľa“ se skládá z dvou základních tabulek (viz obrázek č. 26) datum a hodiny. Do tabulky „datum“ byly vloženy data s konkrétním datem a pro jistotu i se dnem, kdy bude výuka probíhat. Tabulka „hodiny“ obsahuje čas od 7 do 21 hodin v rozmezí výuky vždy po jedné hodině.

Tabulka	Operace	Řádků	Typ	Porovnávání	Velikost	Navíc
datum	Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	30	InnoDB	utf8_general_ci	16 KiB	-
hodiny	Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	15	InnoDB	utf8_general_ci	16 KiB	-
2 tabulky	Celkem	45	InnoDB	latin1_swedish_ci	32 KiB	0 B

Obrázek 26: Struktura databáze „Rozvrhová tabuľa“ [19]

Na obrázku č. 27 jsou hotové tabulky vyobrazené v SQL návrháři. V návrháři lze přehledně vidět i celkovou strukturu jednotlivých tabulek.

rozvrhova_tabula hodiny
hodinyID : int(11)
skratka : text

rozvrhova_tabula datum
datumID : int(11)
den : text
datum : date
nazev : text

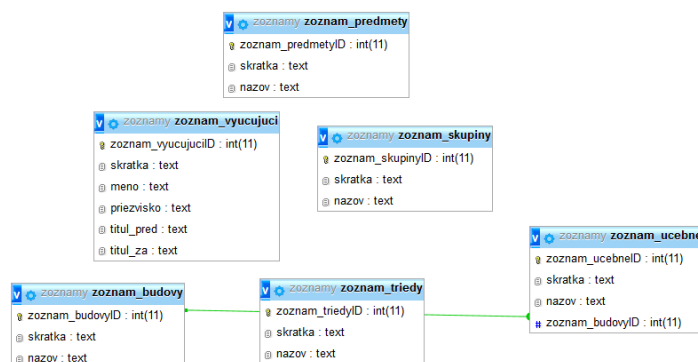
Obrázek 27: Zobrazení tabulek databáze „Rozvrhová tabuľa“ v návrháři SQL [19]

10.3.2 Struktura databáze „Zoznamy“

Databáze s příslušným názvem „Zoznamy“ obsahuje celkem šest tabulek jak lze vidět na obrázku č. 28. V jednotlivých tabulkách jsou vložena veškerá data se zkratkami, názvy a jmény jak lze pro představu vidět na obrázku č. 29, kde jsou vyobrazeny i relace mezi těmito prvky v jednotlivých tabulkách.

Tabulka	Operace	Řádků	Typ	Porovnávání	Velikost	Navíc
zoznam_budovy	Projít	15	InnoDB	utf8_general_ci	16 KiB	-
zoznam_predmety	Projít	101	InnoDB	utf8_general_ci	16 KiB	-
zoznam_skupiny	Projít	23	InnoDB	utf8_general_ci	16 KiB	-
zoznam_triedy	Projít	19	InnoDB	utf8_general_ci	16 KiB	-
zoznam_ucebne	Projít	28	InnoDB	utf8_general_ci	32 KiB	-
zoznam_vyucujuci	Projít	60	InnoDB	utf8_general_ci	16 KiB	-
6 tabulek	Celkem	246	InnoDB	latin1_swedish_ci	112 KiB	0 B

Obrázek 28: Struktura databáze „Zoznamy“ [19]



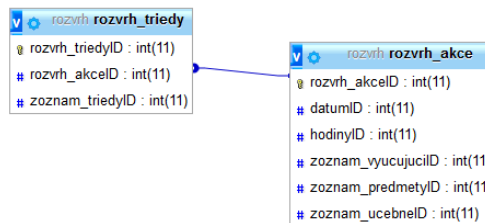
Obrázek 29: Zobrazení tabulek databáze „Zoznamy“ v návrháři SQL [19]

10.3.3 Struktura databáze „Rozvrh“

Databázi tvoří jednak tabulka „rozvrh_akce“ a také tabulka „rozvrh_triedy“ (viz obr. 30). Na obrázku č. 31 si lze všimnout, že obě tyto tabulky obsahují hlavně identifikátory tabulek z ostatních databází.

Tabulka	Operace	Řádků	Typ	Porovnávání	Velikost	Navíc
rozvrh_akce	Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	1 418	InnoDB	utf8_general_ci	336 KiB	-
rozvrh_triedy	Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	1 772	InnoDB	utf8_general_ci	176 KiB	-
2 tabulky	Celkem	3 190	InnoDB	latin1_swedish_ci	512 KiB	0 B

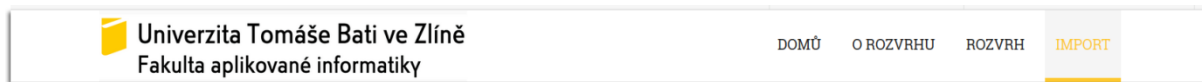
Obrázek 30: Struktura databáze „Rozvrh“ [19]



Obrázek 31: Zobrazení tabulek databáze „Rozvrh“ v návrháři SQL [19]

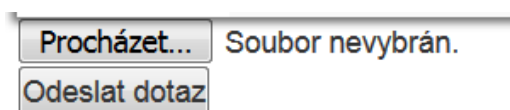
11 IMPORT DAT

Pro naplnění SQL databáze je připravena stránka, kterou si lze otevřít stisknutím možnosti “IMPORT” v horní nabídce (viz obr. č. 32).



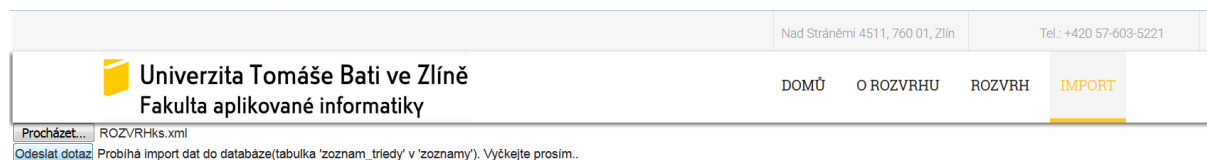
Obrázek 32: Možnost IMPORT na výběrové liště [19]

Na této stránce pomocí možnosti “Procházet...” (viz obr. č. 33) je možno vybrat z počítače soubor, který byl exportován programem Rozvrh 4.1© RNDr.Červený. Stiskem tlačítka „Odeslat dotaz“ se zahájí zpracování tohoto vybraného souboru. V tomto kroku se provádí základní kontrola, zda se jedná o validní soubor XML, následně obsah tohoto souboru je převeden na objekt XML v JavaScriptu. Dále skrze složitost exportovaného souboru XML z programu Rozvrh 4.1 © RNDr.Červený, je potřeba aby tento objekt byl dále zpracován a byli z něj zjištěny všechny požadované informace.



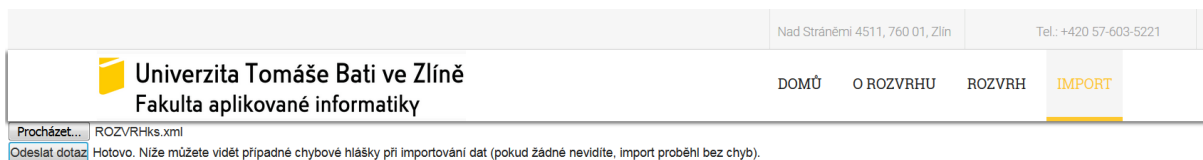
Obrázek 33: Výběr souboru pro importování [19]

Po získání všech potřebných dat, funkce v JS vyšle požadavek na server na funkci pro smazání všech stávajících záznamů v databázi. Tyto funkce, na které se JS odkazuje, jsou napsány v PHP. Po této akci následuje stejným způsobem import dat do databáze. O průběhu tohoto procesu je uživatel informován textem vedle tlačítka “Odeslat dotaz” viz obr. 34.



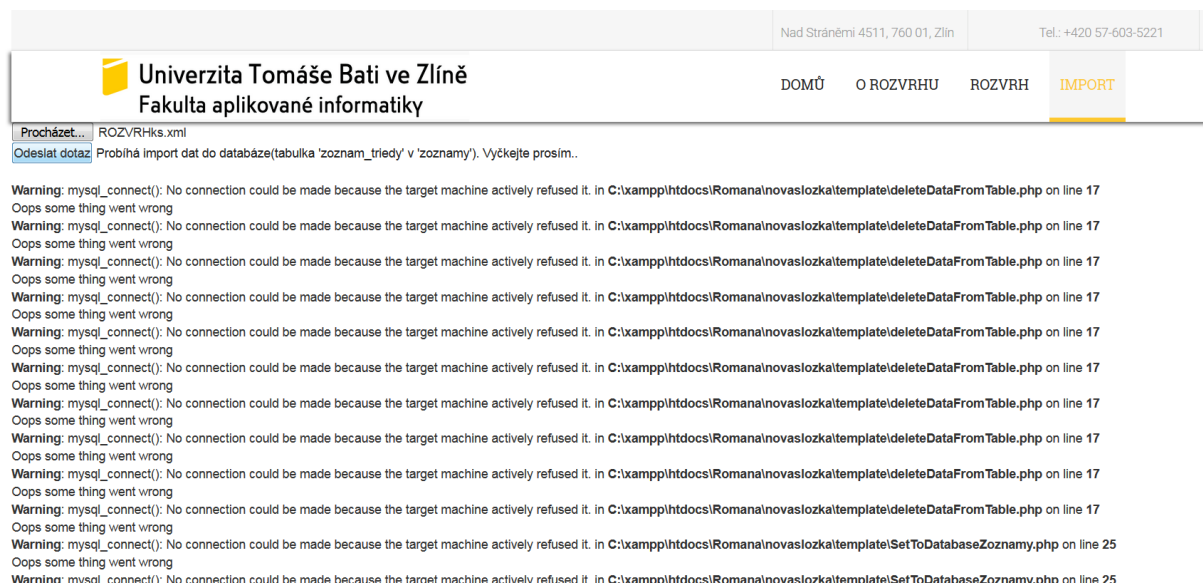
Obrázek 34: Probíhající import souboru [19]

Po dokončení importu dat se na tomto místě objeví zpráva “Hotovo“. Níže můžete vidět případné chybové hlášky při importování dat (pokud žádné nevidíte, import proběhl bez chyb).” viz obr. 35, na kterém žádná chyba nenastala.



Obrázek 35: Ukázka bezchybného importu [19]

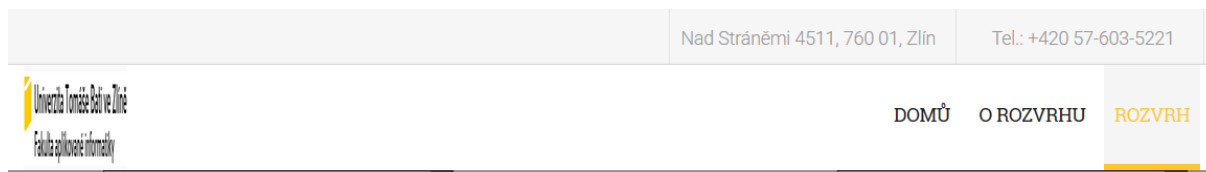
Pokud by nastala chyba, vypadalo by to podobně, jako na obrázku 36, kde pro názornou ukázkou byl odpojen MySQL server.



Obrázek 36: Ukázka chybného importu [19]

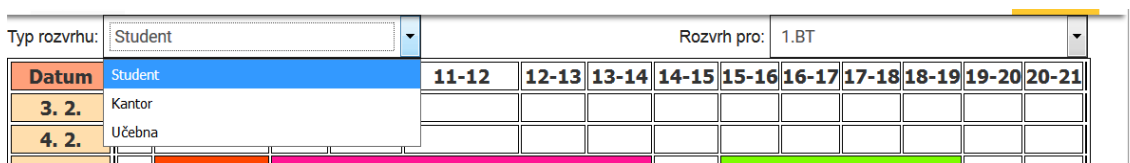
12 ZOBRAZENÍ ROZVRHU

Zobrazit rozvrh lze klepnutím na možnost “ROZVRH” v horní nabídce (viz. obr. č. 37).



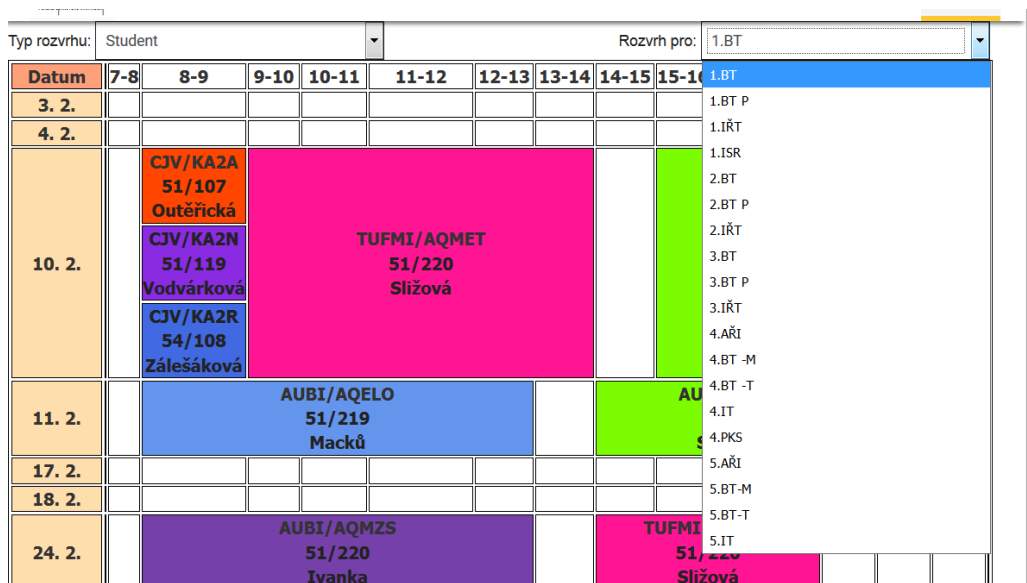
Obrázek 37: Výběrová lišta se záložkou “Rozvrh” [19]

Pomocí prvního ze selectorů si lze vybrat, požadovaný druh rozvrhu. V nabídce lze volit, zda požadujeme rozvrh pro studenta (dle skupiny), pro kantora (dle jména) a nebo pro učebnu (dle zkratky učebny) viz obr.č. 38.



Obrázek 38: Výběr z rolovací lišty [19]

Pomocí druhého ze selectorů je možné přepínat mezi požadovanými rozvrhy. Pokud je tedy například vybrán v prvním selectoru “Student”, můžeme vybírat, pro jakou studijní skupinu chceme zobrazit rozvrh. Rozkliknutou nabídku lze vidět na obrázku č. 39.



Obrázek 39: Rolovací lišta se seznamem studijních skupin [19]

Aby byly stránky pro uživatele (návštěvník, který si bude chtít rozvrh zobrazit) přívětivé na ovládání, rozvrh se aktualizuje ihned po výběru požadované hodnoty ze selectorů. Není tedy potřeba další potvrzení tlačítkem a podobně.

Dále, pro lepší časovou přehlednost se řádek s časem přesouvá podle polohy v prohlížeči. Pokud se tedy uživatel posune v prohlížeči níže, tak i přes to se bude tento řádek nacházet na viditelné pozici v horní části obrazovky, jak lze vidět na obrázku č. 40.

Univerzita Tomáše Bati ve Zlíně Fakulta aplikované informatiky		DOMŮ O ROZVRHU ROZVRH													
Datum	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21	
24. 2.	51/220 Ivanka						51/220 Sližová								
25. 2.	AUPKS/AQPRO 51/218 Košťálová						AUBI/AQELO 51/219 Macků								
3. 3.															
4. 3.															
10. 3.	TUFMI/AQMET 51/220 Sližová						AUBI/AQMZS 51/220 Ivanka			AUM/AQMAA 51/107 Sedláček					
11. 3.	AUBI/AQMZS 51/220 Ivanka						AUPKS/AQPRO 51/218 Košťálová								
17. 3.															
18. 3.															
24. 3.															
25. 3.															
31. 3.															
1. 4.															
7. 4.	AUBI/AQELO 51/219 Macků						AUBI/AQPR2 51/220 Štefka								
8. 4.	TUFMI/AQMET 51/220 Sližová						AUPKS/AQPRO 51/218 Košťálová								
14. 4.															

Obrázek 40: Zobrazení řádku s časy [19]

Pro snadnou orientaci v rozvrhu jsou jednotlivé předměty rozlišeny barevně. V kolonce s předmětem jsou pro přehlednost uvedeny jen zkratky předmětu, učebny a vyučujícího (pro rozvrh typu student - u ostatních obdobně). Po najetí kurzorem na tuto zkratku se u kurzoru objeví celý název (viz obr. č. 41). Dále si lze povšimnout, že pokud je kurzor na některém z řádků, je tento řádek barevně zvýrazněn, což zlepšuje orientaci, ve kterém dnu se předmět nachází.

17. 2.															
18. 2.															
24. 2.	AUBI/AQMZS 51/220 Ivanka						TUFMI/AQMET 51/220 Sližová								
	AUPKS/AQPRO						AUBI/AQELO								

Obrázek 41: Zobrazení konkrétnějších údajů o učebně [19]

Co se týká zobrazení více rozvrhů na jedné stránce, tak stačí aby si uživatel jako první zvolil hlavní rozvrh. Na obrázku č. 42 je jako hlavní rozvrh zvolen „Student“ tedy rozvrh studijní skupiny.

Datum	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21
3. 2.														
4. 2.														
10. 2.		CJV/KA2A 51/107 Outěřická	TUFMI/AQMET 51/220 Sližová						AUM/AQMAA 51/107 Sedláček					
		CJV/KA2N 51/119 Vodvářková												
		CJV/KA2R 54/108 Zálesáková												
11. 2.		AUBI/AQELO 51/219 Macků						AUM/AQMAA 51/107 Sedláček						
17. 2.														
18. 2.														
24. 2.		AUBI/AQMZS 51/220						TUFMI/AQMET 51/220						

Typ hlavního rozvrhu:
Student

Student

Třída:
1.BT

Kantor:
Bližňák Michal Ing. Ph.D.

Učebna:
51/107

Obrázek 42: Příklad zobrazení rozvrhu „Student“ [19]

Na tomto rozvrhu si vybereme předmět se zkratkou „TUFMI/AQMET“ a klikneme na číslo třídy což je v tomto případě 51/220 nebo na příjmení vyučujícího a následně se nám objeví dva rozvrhy vyučujícího daného předmětu, což je v tomto případě RNDr. Marta Sližová, CSc. a číslo třídy s číslem 51/220 (viz obrázek č. 43)

Datum	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21
3. 2.														
4. 2.														
10. 2.		CJV/KA2A 51/107 Outěřická	TUFMI/AQMET 51/220 Sližová						AUM/AQMAA 51/107 Sedláček					
		CJV/KA2N 51/119 Vodvářková												
		CJV/KA2R 54/108 Zálesáková												
11. 2.		AUBI/AQELO 51/219 Macků						AUM/AQMAA 51/107 Sedláček						
17. 2.														
18. 2.														
24. 2.		AUBI/AQMZS 51/220						TUFMI/AQMET 51/220						

Typ hlavního rozvrhu:
Student

Student

Třída:
1.BT

Kantor:
Bližňák Michal Ing. Ph.D.

Učebna:
51/107

Datum	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20	20-21
3. 2.														
4. 2.														
10. 2.		TUFMI/AQMET 51/220 1.BT												
11. 2.														
17. 2.														
18. 2.														
24. 2.		TUFMI/AQMET 51/220 1.BT												

Typ hlavního rozvrhu:
Student

Student

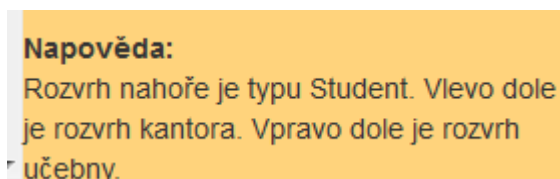
Třída:
1.BT

Kantor:
Bližňák Michal Ing. Ph.D.

Učebna:
51/107

Obrázek 43: Zobrazení dalších dvou rozvrhů na stránce [19]

Pro větší přehlednost se pod lištou pro výběr hlavního rozvrh nachází malá nápověda (viz obr. 44), aby se uživatel lépe orientoval v rozvrzích. Vždy zobrazuje název hlavního rozvrhu a následně i umístění dalších dvou. Pokud je otevřen pouze jeden rozvrh, nápověda se nezobrazuje.



Nápověda:
Rozvrh nahoře je typu Student. Vlevo dole je rozvrh kantora. Vpravo dole je rozvrh učebny.

Obrázek 44: Nápověda [19]

ZÁVĚR

Cílem této bakalářské práce bylo vytvořit optimalizaci zobrazení na webových stránkách fakulty. Pro tento účel jsem vytvořila prostřednictvím multiplatformního balíčku XAMPP, databázi v MySQL a simulované webové stránky Fakulty aplikované informatiky pro import a následné zobrazení rozvrhu. Optimalizace zobrazení se týkala především spojeného zobrazení rozvrhu skupiny, vyučujícího a místnosti a také pak možnosti přepínání mezi jednotlivými rozvrhy. Rozvrh je vybaven pohyblivou lištou s časy a malou nápovědou pro lepší orientaci uživatele. Účel této optimalizace zobrazení je jednoznačně zjednodušení, kdy uživatel místo vyhledávání jednotlivých rozvrhů skupin, vyučujících a učeben má možnost je všechny zobrazit na jedné stránce.

SEZNAM POUŽITÉ LITERATURY

- [1] LUBBERS, Peter, ALBERS a Frank SALIM. *HTML5: programujeme moderní webové aplikace*. Brno: Computer Press, 2011. ISBN 978-80-251-3539-6.
- [2] PÍSEK, Slavoj. *HTML: začínáme programovat*. 4., aktualiz. vyd. Praha: Grada, 2014. Průvodce (Grada). ISBN 978-80-247-5059-0.
- [3] HTML 5.1. *W3C* [online]. 2016 [cit. 2017-05-22]. Dostupné z: <https://www.w3.org/TR/html51/>
- [4] ODELL, Den. *JavaScript: průvodce programováním ajaxových aplikací*. Brno: Computer Press, 2010. ISBN 978-80-251-2733-9.
- [5] HTML5 - Základní informace. *Výukový materiál HTML5 a CSS3* [online]. 2013 [cit. 2017-05-22]. Dostupné z: <http://html5css3.4fan.cz/html-uvod.php>
- [6] Úvod do HTML. *Seminarka* [online]. [cit. 2017-05-22]. Dostupné z: <http://seminarka-web.wz.cz/main.php?page=html&specif=o>
- [7] Jak se zrodilo HTML5 a v čem spočívají jeho výhody. *Dostupny internet* [online]. [cit. 2017-05-22]. Dostupné z: <http://www.dostupnyinternet.cz/blog/html-5/>
- [8] Kaskádové styly. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-22]. Dostupné z: https://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styly
- [9] Co je CSS. *Pěstujeme web* [online]. [cit. 2017-05-22]. Dostupné z: <http://www.pestujemeweb.cz/obsah/css/co-je-css.php>

[10] CSS styly - úvod. *Jak psát web* [online]. [cit. 2017-05-22]. Dostupné z: <https://www.jakpsatweb.cz/css/css-uvod.html>

[11] Připojení CSS k HTML. *Jak psát web* [online]. [cit. 2017-05-22]. Dostupné z: <http://www.pestujemeweb.cz/obsah/css/pripojeni-css-k-html.php>

[12] Vztah HTML a CSS. In: *Pěstujeme web* [online]. [cit. 2017-05-29]. Dostupné z: <http://www.pestujemeweb.cz/obsah/css/co-je-css.php>

[13] Výhody CSS a srovnání z HTML. *Garth* [online]. [cit. 2017-05-22]. Dostupné z: <http://www.garth.cz/uvod-do-css/vyhody-css-a-srovnani-z-html/>

[14] SUEHRING, Steve. *JavaScript: krok za krokem*. Brno: Computer Press, 2008. Krok za krokem (Computer Press). ISBN 978-80-251-2241-9.

[15] PHP – základní informace. *Interval* [online]. [cit. 2017-05-25]. Dostupné z: <https://www.interval.cz/clanky/php-zakladni-informace/>

[16] LECKY-THOMPSON, Ed a Steven D. NOWICKI. *PHP 6: programujeme profesionálně*. Brno: Computer Press, 2010. Programujeme profesionálně. ISBN 978-80-251-3127-5.

[17] Schéma spolupráce PHP s databází. In: *Jak psát web* [online]. [cit. 2017-05-29]. Dostupné z: <https://www.jakpsatweb.cz/php/moznosti-php.html>

[18] PhpMyAdmin. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-25]. Dostupné z: <https://cs.wikipedia.org/wiki/PhpMyAdmin>

[19] Vlastní zdroj

[20] Co je to databáze MySQL? *Arctic Studio* [online]. 2011 [cit. 2017-05-25]. Dostupné z: <https://www.artic-studio.net/slovnicek-pojmu/databaze-mysql>

[21] MySQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-25]. Dostupné z: <https://cs.wikipedia.org/wiki/MySQL>

[22] MySQL. *Adaptic* [online]. [cit. 2017-05-25]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/mysql/>

[23] Dynamická webová stránka. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-25]. Dostupné z: https://cs.wikipedia.org/wiki/Dynamick%C3%A1_webov%C3%A1_str%C3%A1nka

[24] Dynamické webové stránky. *SHOPCentrik* [online]. [cit. 2017-05-25]. Dostupné z: <http://www.shopcentrik.cz/slovník/dynamicke-webove-stranky.aspx>

[25] Co jsou to dynamické webové stránky. *HelpMark* [online]. [cit. 2017-05-25]. Dostupné z: <http://www.helpmark.cz/slovníkpojmu/32-dynamicke-webove-stranky>

[26] Statické webové stránky. *Help Mark* [online]. [cit. 2017-05-29]. Dostupné z: <http://www.helpmark.cz/slovníkpojmu/46-staticke-webove-stranky>

[27] Význam a tvorba dynamických webových stránek. *Olomoucký deník* [online]. [cit. 2017-05-25]. Dostupné z: <http://olomoucky.denik.cz/podnikani/vyznam-a-tvorba-dynamickych-webovych-stranek.html>

[28] Dynamická webová stránka. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-25]. Dostupné z: https://cs.wikipedia.org/wiki/Dynamick%C3%A1_webov%C3%A1_str%C3%A1nka

[29] RNDr. Lubomír Červený. *Žilina* [online]. [cit. 2017-05-29]. Dostupné z: <https://www.zilina.sk/firmy/rndr-lubomir-cerveny>

[30] Rozvrh v4.1. *Ťahaj.sme* [online]. [cit. 2017-05-29]. Dostupné z: <http://tahaj.sme.sk/software/338#>

- [31] Rozvrh kombinovaného studia pro letní semestr 2016/2017. *Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky* [online]. [cit. 2017-05-29]. Dostupné z: <http://nod32.fai.utb.cz/rozvrh/rozvrhK.htm>
- [32] Prohlížení IS/STAG. *Portál UTB* [online]. [cit. 2017-05-29]. Dostupné z: <https://stag.utb.cz/portal/studium/prohlizeni.html>
- [33] Co je to XAMPP. *ITBIZ* [online]. [cit. 2017-05-29]. Dostupné z: <http://www.itbiz.cz/slovník/informacni-technologie-it/xampp>
- [34] XAMPP. *Instaluj* [online]. [cit. 2017-05-29]. Dostupné z: <https://www.instaluj.cz/xampp>
- [35] XAMPP. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-29]. Dostupné z: <https://en.wikipedia.org/wiki/XAMPP>
- [36] Textový editor PSPad. *PSPad* [online]. [cit. 2017-05-29]. Dostupné z: <http://www.pspad.com/cz/>
- [37] Brackets. *Brackets* [online]. [cit. 2017-05-29]. Dostupné z: <http://brackets.io/>
- [38] Enterprise Architect. *Slunečnice* [online]. [cit. 2017-05-29]. Dostupné z: <https://www.slunecnice.cz/sw/enterprise-architect/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML	HyperText Markup Language (značkovací jazyk pro hypertext)
XHTML	Extensible HyperText Markup Language (HTML jazyk přeformulovaný podle XML syntaxe)
CSS	Cascading Style Sheets (Kaskádové styly)
PHP	Hypertext Preprocessor (skriptovací jazyk na straně serveru)
SQL	Structured Query Language (strukturovaný dotazovací jazyk)
MySQL	My Structured Query Language (multiplatformní databázový systém, který pracuje s jazykem SQL)
RNDr.	Rerum naturalium doctor (doktor přírodních věd)
FAI	Fakulta aplikované informatiky
DOM	Document Object Model
W3C	World Wide Web Consortium (hlavní mezinárodní standardizační organizaci pro World Wide Web)
ASP	Association of Shareware Professionals (Asociace sharewarových profesionálů)
AJAX	Asynchronous JavaScript and XML (technologie pro vývoj webových aplikací)
PDF	Portable Document Format (přenosný formát dokumentů)
ITA	Informační technologie v administrativě
BTSM	Bezpečnostní technologie, systémy a management
IŘT	Informační a řídicí technologie
Kb	kombinované studium
CSc.	Candidatus Scientiarum (kandidát věd)
ID	identifikátor
EA	Enterprise Architect

URL	Uniform Resource Locators (jednotný popis umístění zdroje)
UML	Unified Modeling Language (modelovací jazyk a souhrn metod pro popis procesů)

SEZNAM OBRÁZKŮ

Obrázek 1: Připojení CSS k prvku HTML [12].....	17
Obrázek 2: Schéma spolupráce PHP s databází [17].....	20
Obrázek 3: Hlavní stránka phpMyAdmin[19].....	20
Obrázek 4: Rozvrh kombinovaného studia[19].....	34
Obrázek 5: Konkrétní pole rozvrhu [19].....	34
Obrázek 6: Návrh zobrazení rozvrhů [19].....	35
Obrázek 7: Kontrolní panel XAMPPu [19].....	37
Obrázek 8: Menu programu XAMPP [19].....	37
Obrázek 9: Šablona pro tvorbu webové stránky [19].....	38
Obrázek 10: Zmínka o tvůrci šablony na webových stránkách [19].....	38
Obrázek 11: Prostřední v programu PSPad [19].....	39
Obrázek 12: Prostředí v programu Brackets [19].....	40
Obrázek 13: Příklad funkce vytvořené v PHP [19].....	41
Obrázek 14: Kód souboru „getRozvrhUcebna.php“ [19].....	42
Obrázek 15: Příklad využití ajaxu [19].....	43
Obrázek 16: Webové stránky fakulty [19].....	43
Obrázek 17: Požadavky na vytvoření diagramu a databáze [19].....	44
Obrázek 18: Vytvoření základních tříd v EA [19].....	45
Obrázek 19: Třídy s vloženými atributy [19].....	45
Obrázek 20: Příklad zadávání parametrů pro zobrazení relace [19].....	46
Obrázek 21: Vazby mezi jednotlivými třídami [19].....	46
Obrázek 22: Ikona phpMyAdmin v menu programu XAMPP [19].....	47
Obrázek 23: Vytvoření databáze [19].....	47
Obrázek 24: Zadání názvu databáze [19].....	48
Obrázek 25: Část použitého kódu pro vytvoření databáze [19].....	48
Obrázek 26: Struktura databáze „Rozvrhová tabuľa“ [19].....	49
Obrázek 27: Zobrazení tabulek databáze „Rozvrhová tabuľa“ v návrháři SQL [19].....	49
Obrázek 28: Struktura databáze „Zoznamy“ [19].....	50
Obrázek 29: Zobrazení tabulek databáze „Zoznamy“ v návrháři SQL [19].....	50
Obrázek 30: Struktura databáze „Rozvrh“ [19].....	51
Obrázek 31: Zobrazení tabulek databáze „Rozvrh“ v návrháři SQL [19].....	51
Obrázek 32: Možnost IMPORT na výběrové liště [19].....	52

Obrázek 33: Výběr souboru pro importování [19]	52
Obrázek 34: Probíhající import souboru [19]	52
Obrázek 35: Ukázka bezchybného importu [19]	53
Obrázek 36: Ukázka chybného importu [19].....	53
Obrázek 37: Výběrová lišta se záložkou “Rozvrh” [19].....	54
Obrázek 38: Výběr z rolovací lišty [19]	54
Obrázek 39: Rolovací lišta se seznamem studijních skupin [19]	54
Obrázek 40: Zobrazení řádku s časy [19]	55
Obrázek 41: Zobrazení konkrétnějších údajů o učebně [19]	55
Obrázek 42: Příklad zobrazení rozvrhu „Student“ [19].....	56
Obrázek 43: Zobrazení dalších dvou rozvrhů na stránce [19]	56
Obrázek 44: Nápověda [19].....	57

SEZNAM TABULEK

Nenalezena položka seznamu obrázků.

SEZNAM PŘÍLOH

Příloha PI: Tvorba databází a tabulek

PŘÍLOHA P I: TVORBA DATABÁZÍ A TABULEK

Následující příkazy byly vytvořeny v SQL a použity pro tvorbu databází a jejich tabulek.

```
CREATE DATABASE rozvrhova_tabula;
```

```
CREATE DATABASE rozvrh;
```

```
CREATE DATABASE zoznamy;
```

```
USE rozvrhova_tabula;
```

```
CREATE TABLE hodiny(  
    hodinyID int NOT NULL AUTO_INCREMENT,  
    skratka text,  
    PRIMARY KEY(hodinyID)  
) DEFAULT CHARSET=utf8;
```

```
CREATE TABLE datum(  
    datumID int NOT NULL AUTO_INCREMENT,  
    den text,  
    datum date,  
    nazev text,  
    PRIMARY KEY(datumID)  
) DEFAULT CHARSET=utf8;
```

```
USE zoznamy;
```

```
CREATE TABLE zoznam_triedy(  
    zoznam_triedyID int NOT NULL AUTO_INCREMENT,  
    skratka text,
```

```
nazov text,  
  
PRIMARY KEY(zoznam_triedyID)  
  
) DEFAULT CHARSET=utf8;  
CREATE TABLE zoznam_predmety(  
  
zoznam_predmetyID int NOT NULL AUTO_INCREMENT,  
  
skratka text,  
  
nazov text,  
  
PRIMARY KEY(zoznam_predmetyID)  
  
) DEFAULT CHARSET=utf8;  
CREATE TABLE zoznam_vyucujuci(  
  
zoznam_vyucujuciID int NOT NULL AUTO_INCREMENT,  
  
skratka text,  
  
meno text,  
  
priezvisko text,  
  
titul_pred text,  
  
titul_za text,  
  
PRIMARY KEY(zoznam_vyucujuciID)  
  
) DEFAULT CHARSET=utf8;  
  
CREATE TABLE zoznam_skupiny(  
  
zoznam_skupinyID int NOT NULL AUTO_INCREMENT,  
  
skratka text,  
  
nazov text,  
  
PRIMARY KEY(zoznam_skupinyID)  
  
) DEFAULT CHARSET=utf8;  
CREATE TABLE zoznam_budovy(  
  
zoznam_budovyID int NOT NULL AUTO_INCREMENT,
```

```

    skratka text,
    nazov text,
    PRIMARY KEY(zoznam_budovyID)
) DEFAULT CHARSET=utf8;

CREATE TABLE zoznam_ucebne(
    zoznam_ucebneID int NOT NULL AUTO_INCREMENT,
    skratka text,
    nazov text,
    zoznam_budovyID int,
    PRIMARY KEY(zoznam_ucebneID),
    CONSTRAINT FK_budova FOREIGN KEY (zoznam_budovyID)
    REFERENCES zoznam_budovy(zoznam_budovyID)
) DEFAULT CHARSET=utf8;

USE rozvrh;

CREATE TABLE rozvrh_akce(
    rozvrh_akceID int NOT NULL AUTO_INCREMENT,
    datumID int,
    hodinyID int,
    zoznam_vyucujuciID int,
    zoznam_predmetyID int,
    zoznam_ucebneID int,
    PRIMARY KEY(rozvrh_akceID),
    CONSTRAINT FK_ucebne FOREIGN KEY (zoznam_ucebneID)
    REFERENCES zoznamy.zoznam_ucebne(zoznam_ucebneID),
    CONSTRAINT FK_datum FOREIGN KEY (datumID)
    REFERENCES rozvrhova_tabula.datum(datumID),

```

```

CONSTRAINT FK_hodiny FOREIGN KEY (hodinyID)
REFERENCES rozvrhova_tabula.hodiny(hodinyID),
CONSTRAINT FK_vyucujuci FOREIGN KEY (zoznam_vyucujuciID)
REFERENCES zoznamy.zoznam_vyucujuci(zoznam_vyucujuciID),
CONSTRAINT FK_predmety FOREIGN KEY (zoznam_predmetyID)
REFERENCES zoznamy.zoznam_predmety(zoznam_predmetyID)
) DEFAULT CHARSET=utf8;

CREATE TABLE rozvrh_triedy(
    rozvrh_triedyID int NOT NULL AUTO_INCREMENT,
    rozvrh_akceID int NOT NULL,
    zoznam_triedyID int NOT NULL,
    PRIMARY KEY(rozvrh_triedyID),
    CONSTRAINT FK_akce FOREIGN KEY (rozvrh_akceID)
REFERENCES rozvrh_akce(rozvrh_akceID),
CONSTRAINT FK_triedy FOREIGN KEY (zoznam_triedyID)
REFERENCES zoznamy.zoznam_triedy(zoznam_triedyID)
) DEFAULT CHARSET=utf8;

```