



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE MODERNÍCH WEBOVÝCH STRÁNEK

DESIGN AND DEVELOPMENT OF MODERN WEBSITES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ondřej Zámečník

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Münster, Ph.D.

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Ondřej Zámečník

ID: 186463

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Návrh a realizace moderních webových stránek

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh a realizace webových stránek s interní sekci pro sdílení dokumentů či kalendáře. Součástí interní sekce bude rovněž implementace systému pro pracovní výkazy uživatelů. Návrh bude, kromě struktury webových stránek, řešit i grafickou podobu. Výstupem práce budou hotové webové stránky na bázi redakčního systému.

DOPORUČENÁ LITERATURA:

[1] SKLAR, Joel. Principles of web design. Sixth edition. Boston, MA: Cengage Learning, 2015. ISBN 978-1285852645..

[2] NIEDERST ROBBINS, Jennifer. Learning Web design: a beginner's guide to HTML, CSS, JavaScript, and web graphics. Fourth edition. Beijing: O'Reilly, 2012. ISBN 978-1449319274.

Termín zadání: 5.2.2018

Termín odevzdání: 29.5.2018

Vedoucí práce: Ing. Petr Münster, Ph.D.

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce je zaměřena na návrh a vyhotovení internetové prezentace pro laboratorní skupinu OptoLab. Stávající podoba stránek je kompletně rekonstruována s využitím moderních technologií, s ohledem na pravidla přístupnosti a dodržování webových standardů. Hlavním úkolem práce je mimo kompletního grafického návrhu také realizace interní sekce pro správu obsahu, komunikaci a sdílení obsahu mezi uživateli.

KLÍČOVÁ SLOVA

Internetové stránky, Webové stránky, návrh stránek, optimalizace, HTML, CSS, PHP, SEO, WWW, Bootstrap

ABSTRACT

This bachelor thesis is focused on the design and preparation of an Internet presentation for the laboratory group called Optolab. The current form of the site is completely reconstructed using modern technologies, with regard to accessibility rules and compliance with web standards. The main task of the thesis is besides the complete graphic design also the implementation of the internal section for content management, communication and sharing of content among users.

KEYWORDS

Internet pages, Web pages, page design, optimization, HTML, CSS, PHP, SEO, WWW, Bootstrap

ZÁMEČNÍK, Ondřej. *Návrh a realizace webových stránek*. Brno, 2018, 52 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Petr Münster, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh a realizace webových stránek“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	10
1 Historie a vyhlídky Internetu	11
2 Technologie pro tvorbu stránek	13
2.1 Apache	13
2.2 HTML	14
2.3 CSS	15
2.4 PHP	16
2.5 MySQL	16
2.6 JavaScript	19
3 Systémy pro správu obsahu	21
3.1 WordPress	21
3.2 Joomla!	21
3.3 Drupal	22
4 Bootstrap	23
5 Optimalizace internetových stránek	24
5.1 Web design	24
5.2 Optimalizace pro vyhledávače	25
5.2.1 Technické překážky indexace	25
5.2.2 Technické faktory	26
5.2.3 Analýza klíčových slov	26
5.3 Systém jednotného přihlašování	26
6 Analýza současného stavu	28
6.1 Základní rozložení	28
6.2 Hlavní sekce	28
6.3 Grafické zpracování	29
6.4 Navigační lišta	30
6.5 Nekonzistentnost článků	30
6.6 Zabezpečení komunikace	30
6.7 Bezpečnost WordPressu	31
7 Návrh a řešení webových stránek	32
7.1 Návrh struktury a grafiky	32
7.2 Správa webových stránek	34

7.3	Uživatelé a jejich rozdělení	35
7.4	Práva přístupu	36
7.5	Zabezpečení systému	36
7.6	Popis databáze	37
7.6.1	Rozpis tabulek a jejich relace	37
7.6.2	Programové spojení s databází	40
7.6.3	Komunikace s databází	41
7.7	Možnosti portálu	43
7.7.1	Interní sekce E-projekty	43
7.7.2	Správa textů	44
7.7.3	Správa členů	44
7.7.4	Správa souborů	44
7.7.5	Kalendář událostí	45
7.8	Možnosti rozšíření	45
8	Závěr	47
	Literatura	48
	Seznam symbolů, veličin a zkratk	52

SEZNAM OBRÁZKŮ

1.1	Grafické rozhraní prohlížeče Mosaic v prostředí MS Windows[4]. . . .	11
2.1	Ukázka relace 1-1.	18
2.2	Ukázka relace 1-N.	19
2.3	Ukázka relace M-N.	19
5.1	Zobrazení responzivního a adaptivního designu [29].	24
6.1	Grafické rozhraní internetové stránky laboratoře	29
7.1	Návrh první pro a) desktopovou, b) mobilní verzi	32
7.2	Návrh druhý pro a) desktopovou, b) mobilní verzi	33
7.3	Návrh třetí pro a) desktopovou, b) mobilní verzi	33
7.4	Diagram přístupu funkcí uživatelům stránek	35
7.5	Grafické rozvržení databáze v softwaru phpMyAdmin	38

SEZNAM TABULEK

2.1	Popis základních příkazů SQL	18
7.1	Rozdělení tříd Bootstrapu podle velikosti zařízení	34
7.2	Rozdělení pravomocí uživatelů	36

SEZNAM VÝPISŮ

2.1	Základní zápis html kódu.	14
2.2	Ukázka externího zápisu css kódu.	15
2.3	Implementace kaskádových stylů z externího souboru.	16
2.4	Vytvoření nové tabulky.	18
7.1	Připojení k místní databázi.	41
7.2	Výběr obsahu z tabulky projektů.	42
7.3	Přidání nového řádku do tabulky projektů.	42
7.4	Přepsání obsahu řádku z tabulky publikace.	43

ÚVOD

Ať už chceme objednat jídlo s donáškou až k našim dveřím, komunikovat či sdílet multimediální obsah s přáteli, nebo sledovat video v reálném čase, všechny tyto aktivity nám zprostředkovává věc, kterou nikdo nevlastní, ale je součástí každodenního života většiny populace. Vývoj celosvětově propojené sítě počítačů, neboli internetu, jde neustále kupředu a s ním i potřeba tvořit stránky pro různé organizace a firmy, ať už za účelem propagace, nebo za účelem komunikace s kolegy.

Z toho důvodu se tato práce věnuje návrhu webové prezentace pro laboratoř optických sítí ústavu Telekomunikací FEKT VUT v Brně, respektive pro skupinu OptoLab, která se zabývá testováním různých modulačních formátů a jejich vlivu na přenosu optického signálu. Důležitou součástí této skupiny je zejména vzájemná spolupráce, sdílení informací o nadcházejících událostí a sdílení multimediálního obsahu s ostatními členy.

Na základě řady zkušeností jsou na takové potřeby v rámci velké skupiny vhodné internetové stránky. Právě prostřednictvím webových prohlížečů lze přistupovat k informačním systémům provozovaných na webových serverech, které mnoha lidem a organizacím usnadňují každodenní práci.

Nynější webové stránky OptoLabu jsou již po grafické stránce zastaralé a jejich funkčnost nedostačuje požadavkům uživatelů. Proto se tato práce zabývá nejen grafickou modernizací stránek, ale také jejich vnitřní strukturou.

1 HISTORIE A VYHLÍDKY INTERNETU

Internet se oficiálně popisuje jako celosvětová síť propojující všechny kontinenty, státy a města tak, že jakýkoliv počítač dokáže komunikovat s jakýmkoliv jiným počítačem, ať už se nachází ve světě kdekoliv. Ve skutečnosti se ale nejedná pouze o jednu síť, nýbrž stovky tisíc sítí, ne-li víc, které spolu vzájemně komunikují.

Internet byl vytvořen na základě počítačové sítě Arpanet, jež fungoval bez centrálního prvku. Díky tomu bylo vyloučeno vyřazení sítě z provozu v případě výpadku jednoho z uzlů. K prvnímu sestavení decentralizované počítačové sítě došlo na amerických univerzitách v roce 1969. Nadále byla tato síť rozšiřována, až roku 1973 došlo k připojení počítačů mimo USA. Ukázalo se, že původně vojensky zaměřený projekt vlády Spojených států může sloužit i jako prostředek pro sdílení informací, a tak se stal plně mezinárodním [1].

V následujících letech začaly všechny počítače používat rodinu protokolů TCP/IP (Transmission Control Protocol/Internet Protocol) a ve výzkumných laboratořích CERN. V překladu znamená *Evropská organizace pro jaderný výzkum* zahájen projekt WWW (World Wide Web), díky kterému se stal Internet přístupný pro každého. S tím souvisel i rozvoj komerčních aktivit a prvnímu návrhu jazyka HTML, v roce 1991. Dva roky poté byl oznámen vývoj prvního grafického prohlížeče NCSA Mosaic a téže rok CERN deklaroval webový protokol a kód za volně přístupný uživatelům. Od té doby se Internetový svět začal exponenciálně rozvíjet [2] [3].



Obr. 1.1: Grafické rozhraní prohlížeče Mosaic v prostředí MS Windows[4].

V současné době došlo k digitalizaci jak rozhlasového či televizního vysílání, ale také síťového přenosu. Původní telefonní linky, přes které byl internet poskytován, byly nahrazeny síťovými kabely a nepřenáší se již samotný analogový signál, nýbrž jeho digitalizovaný tvar ve formě bitového kódu. Tento kód v sobě nese data, jako například obsah internetové stránky, kterou uživatel zadal do prohlížeče. Jejich

generování je dnes čím dál snazší, tím dochází k degradaci informací. Díky jednoduchosti se akumuluje vysoké množství dat, která ne vždy nesou přínosnou informaci. Navíc ne všechny informace se mohou označit za stejně kvalitní.

Do internetové sítě se nyní řadí i bezdrátový přenos dat, který se nadále rozvíjí. Budoucností Internetu tak bude trvalé a automatické bezdrátové připojení, a to nejen mobilních zařízení každé osoby, ale také chladniček, budíků a dalších domácích spotřebičů. Myšlen je Internet věcí, který nyní expanduje a v dalších několika desítek let se pravděpodobně rozšíří na vše, co by mohlo být připojené k Internetu. Významnou roli zde bude zastávat i virtuální a rozšířená realita, která nabízí přenesení Internetových informací a dat do reálného prostředí v audiovizuální podobě.

S konstantním připojením se však váže otázka bezpečnosti soukromí, která se již v dnešní době často řeší. V důsledku toho se očekává, že vznikne řada organizací a společností usilujících o udržení určité úrovně soukromí, bez ohledu na jejich úspěšnost.

2 TECHNOLOGIE PRO TVORBU STRÁNEK

Pokud chceme vytvořit webovou aplikaci, na počátku je potřeba se rozhodnout jak webové stránky vytvořit. Existují tyto základní možnosti pro tvorbu webových stránek:

- On-line webové editory
- Redakční systémy
- Wysiwyg editory
- Strukturní editory

Každá z uvedených možností má své výhody a nevýhody. Hlavní kritérium pro rozhodování je bezpochyby znalost, respektive neznalost jazyků pro psaní webu. Pro praktickou část této práce využívám strukturního editoru, proto v následujících kapitolách rozeberu jednotlivé technologie, potřebné pro vytvoření webové prezentace.

2.1 Apache

Pro přístup k internetovým stránkám je zapotřebí mít dané stránky přístupné z jakéhokoli místa, kde je přístup k internetu. Takovému účelu obecně slouží webové servery, které kromě uchovávání webových stránek také zpracovávají a odesílají data přístupujícím uživatelům. Nachází se na hardwaru, připojeném k internetu a zajišťuje obsluhu klientů tak, že přijímá dotazy ve tvaru HTTP, které zpracuje a následně odešle odpověď. Tou bývá obvykle HTML dokument, může se však jednat i o jiné formáty, ať už textové či multimediální.

Jedním z nejrozšířenějších webových serverů se nazývá Apache HTTP server[5], hovorově také nazýván Apache. Jedná se o bezplatný open-source software, sloužící jako webový server již od roku 1995, kdy byla vydána jeho první veřejná verze. Ta se však dočkala kompletního přepsání kódu a již od roku 1996 se jedná o celosvětově nejpopulárnější i přes to, že nepatří mezi nejvýkonnější servery. Nabízí však stabilitu, spolehlivost a podporu značného množství platforem. Jeho popularita ale pomalu klesá s narůstajícím počtem serverů Nginx, zaměřeným na vysoký výkon a nízké nároky na paměť [6].

Apache patří do tzv. triády spolu s PHP a MySQL, které společně slouží k tvorbě dynamického webu. Způsob spolupráce vychází z toho, že návštěvník stránek posílá PHP dotaz na server Apache, který zpracovává data, vyžádaná PHP dotazem z databáze MySQL. Tato trojice často bývá instalována lokálně za účelem programování webových stránek. Je sice možné využívat vzdálený webhosting, ale tato možnost se v praxi příliš nepoužívá, právě kvůli lepší odezvě lokálního serveru. Vlastního serveru lze docílit dvěma způsoby, a to sice nainstalovat si jednotlivé aplikace (Apache,

PHP, MySQL) zvlášť a nastavit, aby vzájemně spolupracovali, nebo využít nejrůznějších balíčků, např. WAMP¹, které již vše potřebné nainstalují samy a nastavení je pak dále dostupné přes grafické rozhraní daného programu [7].

2.2 HTML

První technologií, nezbytnou pro tvorbu webových stránek je HTML (Hypertext Markup Language), česky hypertextový značkovací jazyk. Je jedním z jazyků, využívaných ke tvorbě webových stránek v systému World Wide Web. Dokumenty HTML jsou ve skutečnosti soubory obsahující tzv. zdrojový kód, což je souhrn HTML značek, symbolů a textu, sloužící k přeložení do grafické podoby stránek. Tento kód si může uživatel zobrazit v jakémkoli textovém editoru, případně integrovaném editoru prohlížeče [8] [9].

Při zobrazení zdrojového kódu se nám naskytne pohled na samotný text, různé symboly a tagy neboli značky. Píší se v ostrých závorkách a dají se rozdělit do dvou kategorií - párové a nepárové. V případě párových příkazů se význam daného tagu projeví pouze v části, umístěné mezi počátečním a ukončovacím tagem. Příkazy slouží k vytváření, úpravě, případně jiné manipulaci s textem či jinými prvky webových stránek. Jednotlivé tagy se mohou vzájemně ovlivňovat a také nořit mezi sebe, tím se tvoří více hierarchický kód podobný zápisu 2.1.

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="windows-1250">
    <title>Jméno</title>
  </head>
  <body>
    Zde se vkládá hlavní část kódu.
  </body>
</html>
```

Výpis 2.1: Základní zápis html kódu.

Tento jazyk prošel mnohaletým vývojem, díky kterému se postupně rozšiřoval o různé prvky, např. formuláře, tabulky a podobně. Inspirací pro HTML syntaxi byl složitý jazyk SGML². Této podoby se HTML drželo až do verze HTML 4.01. Na konci devadesátých let minulého století se z několika důvodů začalo jednat o novém směru

¹zkratka WAMP značí balíček pro Windows, obsahující Apache, MySQL a PHP

²SGML je jazyk pro definování strukturovaných typů dokumentů a značkovacích jazyků, které představují instance tohoto typu dokumentu

vývoje HTML. Cestou pro řešení měl být Extensible Hypertext Markup Language, zkráceně XHTML, který sice ponechával veškeré elementy a atributy jazyka HTML beze změny, ale syntaxe musela splňovat přísnější pravidla jazyka XML, který je oproti jazyku SGML jednoduchý. Cíle jazyka XHTML se však kvůli mnoha chybám a téměř žádnému přínosu nových funkcí nepodařilo dosáhnout [10].

V posledních pár letech se nadále mluví o HTML5, která přináší nové možnosti pro práci s multimédií a novou javascriptovou API³ usnadňující tvorbu webových aplikací. Případně se také hovoří o nových grafických efektech, které však nemají přímo s HTML5 mnoho společného, neboť se většinou jedná o funkcionalitu CSS3.

2.3 CSS

CSS neboli Kaskádové styly je jazyk, používaný pro styl vizuální prezentace webových stránek. CSS sděluje prohlížečům, jakým způsobem mají různé části webové stránky vykreslit a tím nabývá samotný HTML kód nových možností [11].

Smyslem tohoto jazyka je formátování HTML dokumentů, které se dá sice provést i pomocí značek a atributů HTML jazyka, ale v dnešní době se nepoužívají, protože nám v praxi využití CSS může přinést i další možnosti. Kromě formátování HTML textů, včetně rozmístění prvků CSS nabízí i formátování XML⁴ dokumentů, snadnou tvorbu a údržbu konzistentního stylu dokumentů, oddělení struktury a stylu a s tím související přístupnost se zjednodušením HTML dokumentu [12].

V současné době se využívá verze CSS3, která je nejnovější evolucí Kaskádových stylů, rozšiřující CSS2.1. Přináší plno grafických doplňků od animací přes stínování textů po zaoblení hran bloků.

```
body {  
    font-family: Arial, Tahoma, sans-serif;  
    color: auto;  
    font-size: auto;  
    background-color: #ADD8E6;  
    background-image: url(images/pozadi.png);  
    background-repeat: repeat-x;  
}
```

Výpis 2.2: Ukázka externího zápisu css kódu.

CSS má dvě možnosti zápisu - interní a externí. Výsledná podoba stránky nemá na typu zápisu vliv a tak se dá říci, že je jedno, jaký postup zvolíme. Interním

³API označuje rozhraní pro programování aplikací, složena z anglického *application programming interface*

⁴je zjednodušenou podobou staršího jazyka SGML

zápisem se myslí, že styly jsou definovány přímo ve zdrojovém kódu dokumentu HTML, zatímco externí zápis 2.2 využívá externího dokumentu. Oba tyto zápisy se mohou vzájemně kombinovat.

V případě využití kaskádových stylů externím zápisem, je zapotřebí dát tuto skutečnost najevo prohlížeči odkazem z hlavičky příslušného souboru. To se udělá následujícím způsobem:

```
<link href="navezSouboru.css" rel="stylesheet"
      type="text/css">.
```

Výpis 2.3: Implementace kaskádových stylů z externího souboru.

2.4 PHP

PHP, neboli Hypertextový preprocesor, je skriptovací jazyk, určený především pro vývoj webových aplikací a internetových stránek. Dříve byly všechny stránky psány čistě statické, tzn. jak byla stránka naprogramována, tak byla také zobrazena. To pochopitelně přestalo postupem času stačit, a proto byla vyvinuta celá řada technologií, které měli rozšířit možnosti internetových stránek. Tyto technologie se dají rozdělit do dvou skupin - klientské a serverové [13].

Zatím co technologie klientské jsou na cílovém počítači ve vhodnou chvíli spuštěny, např. kliknutím na nějaké tlačítko, filozofie fungování serverové technologie je taková, že ke zpracování kódů se využívá serveru, který kombinuje kusy HTML kódu se zpracovanými skripty skriptovacího kódu a odesílá prohlížeči čistý HTML kód. Příkladem využití serverové technologie je právě PHP, oproti tomu klientskou technologii využívá např. JavaScript, popsáný v kapitole 2.6 [14] [15].

Nejčastěji se PHP kód vkládá přímo do HTML kódu, oddělen je pouze adekvátním párovým tagem `<?php a ?>`. Výhodou tohoto jazyka je jeho jednoduchost a podoba jazyku C a Java, proto je pro většinu programátorů velmi přívětivý. Nadále také jeho neustálý vývoj, který se již transformoval z dřívějšího procedurálního programování, do moderního objektově orientovaného prostředí.

2.5 MySQL

Obecně elektronické databáze slouží ke sběru dat nebo informací, které jsou organizovány pro rychlé vyhledávání a zpřístupnění počítačem. Databáze jsou logicky strukturovány podle určitých pravidel tak, aby usnadnily uchovávání, úpravy a mazání dat.

Z hlediska využívání a způsobu práce s daty a vazbami, se databáze dělí do pěti typů. Mezi nejpoužívanější typy databázových modelů v současnosti se řadí relační, objektové a objektově-relační databáze [16].

Relační databáze (RDBMS - Relational database management system), jsou databáze, ve kterých jsou data strukturovaná do tabulek, polí a záznamů. Každá databázová tabulka se skládá z jednoho nebo více databázových polí. Mezi relační databázové systémy se řadí MySQL, se kterým probíhá komunikace prostřednictvím dotazovacího jazyka SQL.

Mezi známé databázové systémy patří např. Oracle9i, Microsoft SQL Server nebo PostgreSQL. Pro programování dynamických webových stránek se však nejčastěji využívá zmíněné MySQL. To patří v tuto chvíli mezi nejrozšířenější databázový systém v této oblasti především díky podpoře webových hostingů, které ho nabízejí společně s PHP pro hostované weby jako součást portfolia svých služeb [17].

V rámci modernizace se MySQL dočkalo rozšíření známé pod zkratkou MySQLi, které bylo vyvinuto za účelem využití nových funkcí nalezených v MySQL systémech verze 4.1.3 a novější. Toto rozšíření je součástí PHP verze 5 a novější. Klíčová vylepšení oproti MySQL jsou:

- Objektově orientované rozhraní
- Podpora více příkazů
- Podpora transakcí
- Rozšířené možnosti ladění

Pro tvorbu tabulek může klient využít příkazů SQL nebo využít nástroj PH- PMyAdmin, což je nástroj na daném hostingu umožňující širokou škálu operací v MySQL v rámci webového rozhraní, naprogramovaného pomocí PHP. Toto rozhraní pak převádí vyplněné formuláře na SQL příkazy, které následně databáze vykoná [18].

Základní SQL příkazy, které se běžně používají při vytváření jsou uvedeny v tabulce:

Příkaz	Popis
CREATE	Vytvoření tabulek a instancí v databázi.
ALTER	Modifikace existující tabulky v databázi.
DROP	Mazání tabulek a instancí.
INSERT	Vytvoření instance.
SELECT	Načtení instance z tabulky.
UPDATE	Upravení instance.
DELETE	Mazání instance.
GRANT	Přiřazuje oprávnění uživateli.
REVOKE	Odebírá oprávnění.

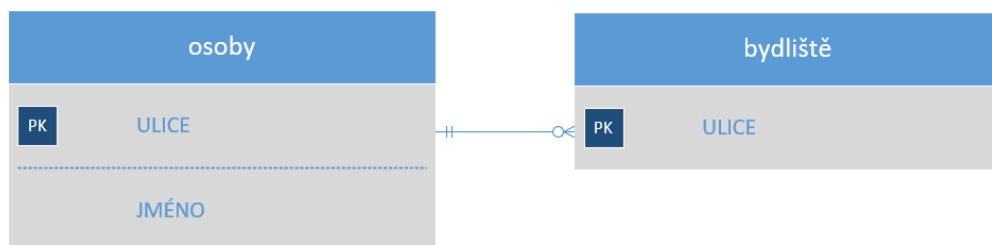
Tab. 2.1: Popis základních příkazů SQL

Příkladem takového příkazu může být vytvoření tabulky uživatelů 2.4 se sloupci ID, JMENO, PRIJMENI, PUBLIKACE, kde sloupce ID, JMENO A PRIJMENI nemohou být při zadávání dat prázdné a ID pracuje jako primární klíč, což značí entitu, která jednoznačně identifikuje určitou instanci (uživatele) z databázové tabulky.

```
SQL>CREATE TABLE UZIVATELE(
    ID INT NOT NULL,
    JMENO VARCHAR (20) NOT NULL,
    PRIJMENI VARCHAR (20) NOT NULL,
    PUBLIKACE VARCHAR (80),
    PRIMARY KEY (ID) );
```

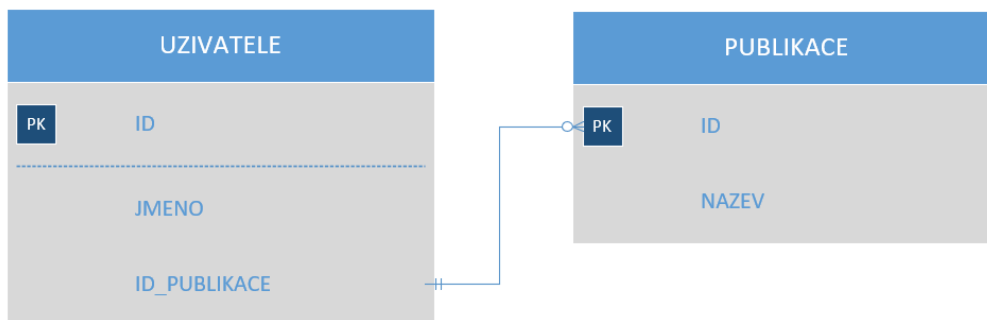
Výpis 2.4: Vytvoření nové tabulky.

Při vytvoření druhé tabulky publikací, mohou být jednotlivé instance vzájemně provázány. Vazby se tvoří mezi tabulkami, které obsahují stejný parametr, nebo se řídí parametrem tabulky jiné. Tyto vazby se rozdělují do tří základních typů: 1:1, 1:N a M:N.



Obr. 2.1: Ukázka relace 1-1.

V relaci 1:1 odpovídá jednomu záznamu v první tabulce maximálně jeden záznam v druhé tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce. Typ relace 1:1 se až na výjimky nepoužívá, a to proto, že takové provázání může být nahrazeno jednou tabulkou. Využití této vazby by bylo vhodné v případě potřeby doplňujících informací pouze k určitým instancím.



Obr. 2.2: Ukázka relace 1-N.

Typ relace 1:N znamená, že pro každého uživatele uvedeného v tabulce UZIVATELE, může existovat celá řada publikací zaznamenaných v tabulce PUBLIKACE.



Obr. 2.3: Ukázka relace M-N.

Relace M:N pracuje se třemi tabulkami, neboť jedna rozděluje relaci M:N na dvě relace 1:N. Příkladem může být relace mezi tabulkami AUTORI a PUBLIKACE. Pro každý záznam v tabulce AUTORI může existovat mnoho záznamů v tabulce PUBLIKACE, respektive pro každý záznam v tabulce PUBLIKACE může existovat řada záznamů v tabulce AUTORI.

2.6 JavaScript

JavaScript, často zkracován na JS, patří do skupiny skriptovacích, objektově orientovaných jazyků, může však pracovat i ve formě procedurálního programování. Laik by mohl z názvu usoudit, že se jedná o technologii odvozenou z jazyku Java, ale není tomu tak. I přes to však mají tyto jazyky pár společných rysů.

Původně se jazyk vyvíjel pod názvem LiveScript, avšak z důvodu že z části využívá Java syntaxi a její jmenné konvence, změnil se název na JavaScript. Tuto podobnost má nejen s jazykem Java, ale i s C++, a to kvůli minimalizování neustálého učení se nových interpretací toho samého. Příkladem mohou být konstrukce příkazu if, cyklů for a while, nebo switch a try...catch bloků, které jsou stejné jako ve zmíněných dvou jazycích.

JavaScriptový kód je možné umístit, stejně jako v případě CSS či PHP, do externího souboru a odvolávat se na něj. Příkladem, jak se na daný soubor odkazovat, může být kód:

```
<script src="externi_skript.js"></script>.
```

Často se však Javascriptový kód vkládá přímo do HTML souboru.

Na rozdíl od PHP se celá akce provádí na straně klienta, což má své výhody především pro server, který nemusí být zbytečně zatěžován. Používá se také pro různé interaktivní prvky jako je skrývání obsahu stránek a další. [19].

3 SYSTÉMY PRO SPRÁVU OBSAHU

Content management system neboli CMS je softwarová aplikace, případně balíček provázaných programů, které se používají pro vytváření a správu webového obsahu. Z pravidla tyto systémy umožňují intuitivně a pohodlně přidávat a spravovat obsah, ať už se jedná o články, fotografie nebo videa. V dnešní době se však využívají ke tvorbě celé webové prezentace. Pro CMS se také používají i termíny redakční či publikační systém.

Výhodou takového systému je fakt, že tvůrce webu nemusí být příliš znalý jazyků HTML a CSS a i bez znalosti jazyka PHP dokáže vytvořit dynamické stránky. To je dáno konstrukcí webu, která ve většině redakčních systémů funguje formou WYSIWYG¹ editoru. Ten pracuje na způsobu editace dokumentů, při kterém je výsledná podoba totožná s verzí zobrazenou při editaci. Z pohledu vývojáře je CMS výhodný také v tom, že samostatný vývoj je s použitím šablon znatelně rychlejší a jednodušší. Navíc pro celkovou správu webových stránek stačí jen klasický internetový prohlížeč [20].

Redakční systémy jsou většinou nabízeny jako *open source* software, to znamená, že jeho zdrojový kód je dostupný každému, kdo o něj má zájem. Mezi tyto systémy patří nejznámější trojice systémů - WordPress, Joomla! a Drupal.

3.1 WordPress

Původně se o WordPressu vedla diskuze, zda by se vůbec měl považovat za redakční systém, neboť z počátku cílil na blogovací platformy. V dnešní době ale na WordPressu dokáží pracovat i plnohodnotné sociální sítě a to především díky pluginům, na kterých je práce s WordPressem převážně založena. To má za následek výhodu v poměrně snadné rozšiřitelnosti funkcionalit na úkor vysoké ztráty rychlosti. Jeho dalšími problémy pak bývá bezpečnost systému a nulová oficiální podpora, pokud nepočítáme uživatelská fóra. Vzhledem k využívání WordPressu na původních stránkách jsou dále v kapitole 6 rozepsány jeho možnosti a případná rizika [21].

3.2 Joomla!

Redakční systém Joomla! se staví někam mezi její dva nejznámější konkurenty, neboť je pokročilejší než WordPress, ale ne tak komplikovaný jako Drupal. Podobně jako

¹WYSIWYG je zkratka z anglické věty „What you see is what you get.“, přeloženo do češtiny „Co vidíš, to dostaneš.“.

u WordPressu má Joomla! možnost rozšíření o několik tisíc pluginů, avšak již v základu má více možností, než již zmíněný konkurent. Její výhodou je také technická podpora. V případě potřeby nalezneme obrovské množství tutoriálů a nástrojů a také má velmi rozsáhlé diskuzní fórum. Problémem bývají její požadavky na server. Systém oproti ostatním CMS ubírá serveru znatelné množství výkonu serveru a tak se pokročilejší web může načítat pomaleji [22].

3.3 Drupal

Jedná se o další velmi populární redakční systém, který obsahuje spoustu funkcí, ať už pro vytváření interních a externích webů, tak pro organizaci obsahu. Jeho uživatelské prostředí je značně složitější, než u jmenovaných konkurentů, za to nabízí pokročilejší možnosti. Jelikož se jedná o relativně nový systém, v jeho řešení je to často poznat. Co se týče výkonu, zaostává Drupal díky jeho vysokému zatížení za WordPressem. Nabízí se však možnost pluginů, které zajišťují generování webových prezentací do vyrovnávací paměti serveru, což má za následek snížení jeho zatížení [23].

4 BOOTSTRAP

Bootstrap je světově nejpoblárnější framework¹, určený pro budování webových stránek v první řadě pro mobilní zařízení. V jeho zdrojovém kódu se nachází vysoce kvalitní HTML, CSS a JavaScriptový kód, který zjednodušuje počátek jakéhokoli projektu [24].

Výhodou této sady nástrojů je snadná implementace a uživatelsky přívětivý vývoj webových stránek, bez potřeby vysoké znalosti programování v jazycích HTML a CSS. Bootstrap je kompatibilní se všemi hlavními prohlížeči na jejich nejnovějších verzích, a tak se přizpůsobuje zobrazení na každém z nich. Stejně tak je přizpůsobený pro fungování na starších či méně známějších prohlížečích.

Bootstrap se sestává z již kompilovaných CSS a JS souborů, které v případě použití stačí načíst v hlavičce webového projektu. Pro jeho správnou funkčnost je však zapotřebí i JavaScriptové knihovny jQuery.

Vedle obvyklých HTML prvků, nabízí Bootstrap pokročilejší možnosti jako například seskupování tlačítek, vytváření navigace, záložek, štítků a dalších funkcí. Stejně tak obsahuje i několik JavaScriptových komponent ve formě jQuery pluginů, poskytujících rozšíření uživatelského rozhraní o rozbalovací políčka s nabídkou, neboli Dropdown, dále také Alerts, Tooltips atp [25].

¹Framework slouží jako podpora při programování a organizaci jiných softwarových projektů a může obsahovat podpůrné programy, knihovny API, atp.

5 OPTIMALIZACE INTERNETOVÝCH STRÁNEK

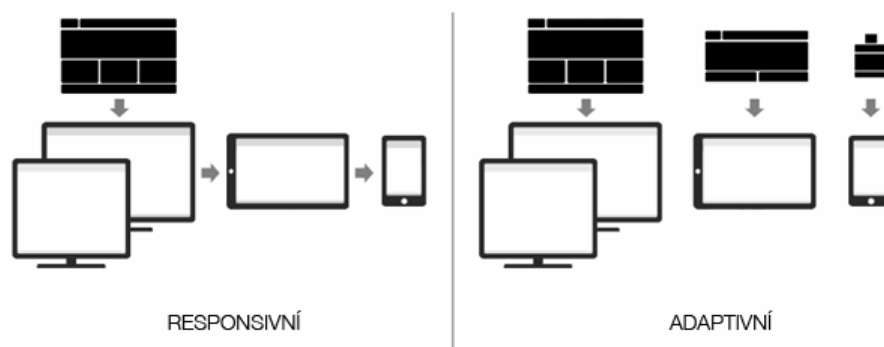
5.1 Web design

Slovní spojení responzivní design známe již několik let. Přesněji řečeno od doby, kdy ho poprvé použil americký programátor Ethan Marcotte v článku na blogu A List Apart [26]. Od té doby se dostal do podvědomí všech programátorů a postupně se rozšiřuje počet stránek, které tento typ designu využívá.

Původní a také jedinou možností dřívější tvorby webových stránek byl fixní návrh, kdy byl web postaven pouze na jedno rozlišení (nejčastěji 980px). Problém nastává při zobrazení tohoto webu na mobilních zařízeních, kde takový web přestává být čitelným. Řešením takové situace mohlo být využití dvou domén, kde jedna z nich byla určena pro mobilní zařízení.

Z toho důvodu se rozšířil tzv. responzivní design, díky kterému máme možnost upravovat jednotlivé bloky, případně obrázky, pro jakékoliv rozlišení. Responzivní design tedy využívá detekci rozlišení na straně klienta, aby následně upravil tomuto rozlišení rozvržení bloků na stránce. Výhodou je tak přizpůsobení webu libovolnému rozlišení v rámci jedné HTML šablony. Toho využívá mimo jiné i zmiňovaný framework Bootstrap a to od verze 2.0.

Poslední možností, často zaměňovanou s responzivním designem, je adaptivní design. Ten v první řadě zjistí, na jakém zařízení uživatel pracuje a server odešle pouze potřebná data (HTML, CSS, JS), která opravdu v zařízení chceme zobrazit [27] [28].



Obr. 5.1: Zobrazení responzivního a adaptivního designu [29].

5.2 Optimalizace pro vyhledávače

Účelem optimalizace pro vyhledávače je usnadnit vyhledávačům procházení, indexování i pochopení vašeho obsahu. Cílem je pak získání lepší pozice odkazu stránky ve výsledcích vyhledávače. Kromě zlepšení postavení webových stránek také zahrnuje odstranění technických nedostatků bránících v prohlížení webové prezentace vyhledávacím robotům a uživatelům internetu. Vylepšení pozice ve vyhledávačích však můžeme dosáhnout dvěma způsoby:

- placením vyhledávače za zviditelnění stránek.
- optimalizací stránek pro lepší indexaci.

Vyhledávačem je software, který stahuje dokumenty (webové stránky) a indexuje (zpracovává a ukládá) do své databáze. V této databázi pak umožňuje uživatelům vyhledávat pomocí jednoduchých i složitých dotazů. Mezi nejznámější patří vyhledávač Google [30].

V případě snahy o zviditelnění stránek se setkáme s termínem SEM – Search Engine Marketing, jehož podskupina Search engine Optimization – SEO se zabývá tím, jak optimalizovat webovou stránku pro vyhledávače. Tuto optimalizaci provádí programátor při samotné tvorbě webu. Součástí optimalizace můžeme rozřadit na dva faktory:

- On-page
 - Technické překážky indexace
 - Technické faktory
 - Obsah, klíčová slova, struktura
- Off-page
 - Zpětné odkazy a linkbuilding

5.2.1 Technické překážky indexace

Problémy s indexováním stránek mohou nastat v mnoha případech. Jeden z nich nastává při využívání Javascriptu či Flashe. Vyhledávače tyto jazyky velmi často neindexují. Takový případ může nastat, pokud je například navigace webu provedena pomocí Javascriptu. Vyhledávač nedokáže přechít Javascriptový kód, tudíž zaindexuje pouze úvodní stránku. Ideálním řešením za účelem přehlednosti kódu pro roboty je umístění Javascriptu do externího souboru, který nebude daný vyhledávač číst. Mezi další výhody při využití scriptu na více místech z externího souboru, že ho prohlížeč stahuje pouze jednou.

Dalšími překážkami bývá často duplicita, neplatnost odkazů, případně klientské přesměrování a závislost na typu a nastavení prohlížeče.

5.2.2 Technické faktory

Technické faktory na rozdíl od technických překážek nezabraňují vyhledávačům stránku indexovat, ale účelem je strukturování a sémantika kódu, které bude vyhledávač rozumět.

I když technickým faktorům není přidělena taková váha jako ostatním faktorům, jeden aspekt zde hraje velkou roli. A to sice značka `<Title>`, která je určena pro popis dané webové stránky. Klíčovým slovům, které jsou v této značce, přiřazují vyhledávače velmi vysokou váhu. Problém však nastává, pokud je titulek příliš dlouhý. Váha klíčových slov v titulku se přiřazuje všem slovům rovnoměrně, tedy při dlouhém titulku budou mít všechna klíčová slova velmi malou váhu. Proto se doporučuje mít titulek krátký. Dále vyhledávače přihlížejí i k tagům `<h1>`, `<h2>`, `<h3>`, ``, `<i>` a klíčovým slovům v URL [31].

5.2.3 Analýza klíčových slov

Klíčovými slovy se myslí souhrn hledanosti jednotlivých slov, která souvisejí s danou webovou stránkou. Díky promyšlené analýze můžeme najít desítky slovních spojení, pokud se jedná o menší stránku nebo pouze o konkrétní podstránku či článek. Existují však i rozsáhlé analýzy obsahující tisíce klíčových slov, přes které si potenciální uživatelé mohou danou stránku zobrazit.

Při řešení analýzy se často využívá tabulky, kterou zaplníme co nejvíce klíčovými slovy, podle kterých by návštěvník mohl najít dané webové stránky. K tomu je možné využít nástroje, které nabízí samotné vyhledávače. Příkladem může být služba Adwords od společnosti Google, případně Sklik společnosti Seznam. Tento seznam klíčových slov se následně aplikuje do hlaviček jednotlivých stránek, přičemž platí pravidlo čím méně slov, tím větší váha se jim přiřazuje [32].

5.3 Systém jednotného přihlašování

Technologie jednotného přihlašování, ve zkratce SSO, funguje na principu, kdy má uživatel své přihlašovací údaje uložené na jednom centrálním místě a používá je k přihlášení u různých internetových služeb. Tyto internetové služby tak nemají k přihlašovacím údajům uživatele přístup. Aby nebylo nutné při registraci k nové internetové službě znovu zadávat své osobní údaje jako například jméno, e-mail nebo adresu, nabízí SSO možnost tyto údaje dané službě předat.

Mezi hlavní výhody SSO se řadí snížení rizika zneužití hesla a to především díky jednomu místu uložení. Statistiky často poukazují na fakt, že uživatelé mnohdy využívají stejné heslo na více službách, což má za následek větší riziko uniknutí

hesla. Navíc služby poskytující možnost jednotného přihlášení často nabízejí silnější metody přihlašování jako například přihlášení osobním certifikátem, které však není součástí řešení této práce.

Mezi největší riziko patří zcizení uživatelského klíče, neboť tím útočník získá přístup nejen k jedné, nýbrž ke všem službám sdruženým pod tímto klíčem. Dané výhody, které univerzální klíče a jednotné přihlašování přinášejí, tedy musí být adekvátně vyváženy zvýšenou ochranou.

Mezi nejrozšířenější služby, které nabízejí možnost uchovávat jednotné přihlašovací údaje, patří Facebook, Google nebo Seznam. Pro českou akademickou sféru existuje projekt EduID, za kterým stojí sdružení CESNET [33] [34].

6 ANALÝZA SOUČASNÉHO STAVU

Skupina OptoLab již delší dobu provozuje webové stránky [35], vytvořené pomocí redakčního systému WordPress. Při jejich analýze se však narazilo na nedostatky, díky kterým se z mnoha ohledů jeví jako nedostatečné. V první řadě upoutá případného návštěvníka, kromě samotného obsahu, především design stránek a jejich přehlednost. Přestože se jedná o poměrně jednoduchou webovou prezentaci, naskytá se zde několik zastaralých či nelogických metod řešení. V následujících podkapitolách se budu zabývat podrobněji jednotlivými argumenty, proč jsem se rozhodl naprogramovat stránky zcela znovu.

6.1 Základní rozložení

Nynější stránky jsou rozděleny do čtyř základních bloků – hlavička, postranní panel, obsahová část, zápatí. Tyto části sdružuje tzv. wrapper o pevné velikosti 940 px. Tento způsob řešení se používal velmi často, avšak v posledních letech se s nástupem responzivního designu tato konvence přestala dodržovat a stránky tak využívají celý potenciál prostoru webových prohlížečů a díky využití automatické šíře bloků a jejich vzájemného rozložení na stránkách se stávají webové stránky atraktivní i pro mobilní telefony.

6.2 Hlavní sekce

Základním prvkem stránek je navigační lišta, která již od pohledu skýtá příliš mnoho odkazů. Navigace by měla být především přehledná a to se s prvky, které po rozbalení nabízí desítku odkazů, příliš neslučuje. Dále by se webová navigace měla držet konvence zobrazení nejvýznamnějších odkazů po ty méně významné, a to v pořadí v jakém se čte – tedy zleva doprava nebo shora dolů, což není zcela splněno.

Kromě navigační lišty zůstává na každé podstránce neměnný postranní panel, nabízející téměř stejné odkazy, jako navigační lišta. Dále se v něm nachází kalendář, který v tuto chvíli pouze zvýrazňuje aktuální den a nenese tedy žádnou užitečnou informaci. Celkově se dá říci, že se jedná o přebytečný panel, který pouze vyplňuje místo stránek.

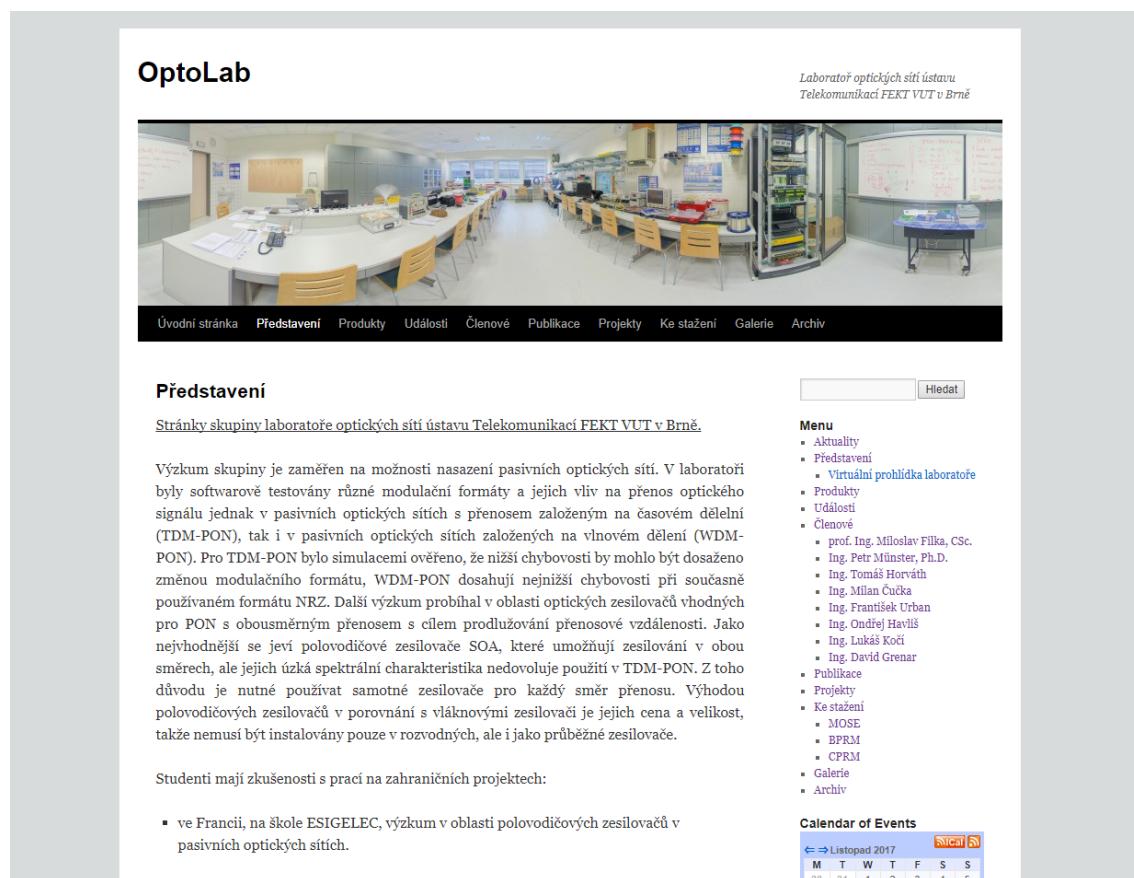
Posledním prvkem, mimo obsahovou část, je tzv. footer, tedy zápatí stránek, které zůstalo bez využití. Odkazuje se zde pouze na stránky WordPressu, na kterém je webová prezentace vytvořena.

6.3 Grafické zpracování

Mezi nejčastěji skloňované vlastnosti webového designu patří originalita a celková kreativita řešení. Souvisí to s hlavními úkoly designu webu, tj. vytvořit příznivý první dojem a i když se jedná o výzkumnou laboratoř, kde je cílem soustředit se především na obsahovou část, neměl by být opomíjen design. Ten je v tuto chvíli velmi střídmy a omezuje se pouze na základní grafické zpracování hlavičky, kde se nachází fotografie dané laboratoře.

Stránky využívají pouze odstín šedé pro nevyužitou část prohlížeče, zbylou část pak tvoří text na bílém podkladě se základním formátováním a zanecháním výchozího nastavení vyobrazení odkazů, viz obr. 6.1.

Nepomáhá tomu také fakt, že skupina nemá žádné logo, kterým by se mohla prezentovat. Ve vrchní části je tak vyobrazen pouze název skupiny OptoLab a její krátký popis.



Obr. 6.1: Grafické rozhraní internetové stránky laboratoře

6.4 Navigační lišta

Navigační lišta v sobě ukrývá mnoho dropdown prvků, přičemž některé zůstávají téměř nevyužité. Příkladem může být položka *Události*, která kromě jednotlivých odkazů ke článkům také sama odkazuje na stránku událostí, kde však nejsou rozepsány jednotlivé články, nýbrž pouze text popisující tuto podstránku: „Akce pořádané skupinou OPTOLAB.“ Tato podstránka tak postrádá účel. Podobně je na tom podstránka *Archiv*.

6.5 Nekonzistentnost článků

Dalším problémem je také rozčlenění článků do kategorií *Události* a *Archiv*, kde archiv měl sloužit pro uchování starších článků, naopak v událostech se měly nacházet novější články. Na úvodní stránce se měl zobrazit přehled nejnovějších článků v podobě aktualit. Členové však na takové pravidlo pravděpodobně nehleděli, tudíž jsou články nahrány do jedné ze tří zmíněných kategorií dle uvážení autora. To má za následek zbytečné rozčlenění celkem do tří kategorií tedy i špatnou přehlednost pro uživatele.

Nastává zde také estetická chyba v podobě různého formátování textu, ať už se jedná o zarovnání textu, použití jiné velikosti písma, apod.

6.6 Zabezpečení komunikace

Nedílnou součástí dynamických stránek bývá administrace. Ta se však musí nacházet v interní sekci, která je přístupná pouze po autorizaci určitým uživatelům. Autorizace na stránkách Optolab probíhá zadáním jména a hesla do formuláře, jehož obsah se posílá serveru a kontroluje se s údaji v databázi uživatelů. Ke komunikaci s tímto webem se však využívá zastaralého HTTP protokolu, kde dochází k nešifrovanému přenosu dat, která může odposlouchávat potenciální útočník.

I přes to, že se webové stránky zabezpečí pomocí kryptografického protokolu TLS ¹ nebo jeho předchůdce SSL, může potenciální útočník stále získat přístup k určitým datům a získat přístup k interní sekci webových stránek. Nejznámějším útokem za účelem získání citlivých dat bývá tzv. Man in the middle. V praxi není vůbec nutné, aby se útočník nacházel na fyzické cestě mezi účastníky, a tak útok probíhá zachycením veřejných klíčů účastníků, které pozmění tak, aby se útočník tvářil pro ostatní jako účastník. Cokoliv si pak účastníci pošlou, útočník zachytí a dešifruje.

¹jedná se o protokol *Transport Layer Security*, který poskytuje zabezpečenou komunikaci na Internetu pro služby WWW

6.7 Bezpečnost WordPressu

Samotný WordPress, na kterém tato webová stránka staví je vzhledem ke své rozšířenosti velmi častým cílem nejrůznějších útoků. Jedná se totiž o opensource, tedy jeho kód je volně přístupný, čehož útočníci využívají pro hledání chyb za účelem útoku. Proto je velmi důležité pravidelně instalovat aktualizace. Ty se od verze 3.7 instalují automaticky, pokud je uživatel nezakáže. Aktuální verze WordPressu na stránkách Optolabu je 4.1, která je však zastaralá a obsahuje určité kritické bezpečnostní chyby [36]. Nejzávažnější problém se objevil v podobě XSS neboli Cross-site scripting², kdy za určitých okolností mohl na web proniknout nepřihlášený uživatel a provést nežádoucí změny. Dále se ve WordPressu 4.1.2 našlo několik významných chyb [37]:

- Vývojáři se pokoušeli opravit problém s chybnou kontrolou některých souborů ve funkci `wp_check_filetype()`, tím ale umožnili nahrávat i soubory s neplatným nebo nebezpečným názvem.
- Od WordPressu 3.9 byla aktivní další XSS chyba, která mohla být zneužita pro skrytou instalaci škodlivých pluginů
- Některé pluginy mohly být potenciálně napadeny prostřednictvím SQL injection.

Vzhledem k jednoduchosti útoku tzv. hrubou silou zkouší útočník často hádat hesla do administrace. Wordpress sám o sobě neomezuje počet pokusů o přihlášení, útočník tak může zkoušet velké množství kombinací. Navíc lze často zjistit přihlašovací jméno uživatele, takže stačí hádat pouze heslo.

Jednoduchou cestou pro posílení bezpečnosti při práci s WordPressem může být použití některého z jeho bezpečnostních pluginů. Tyto pluginy dokáží odfiltrovat nejčastější formy útoků, jako například zmíněný útok hrubou silou, a také skrýt některá citlivá data. Naopak odstranění nepotřebných pluginů a šablon, které v sobě mohou obsahovat bezpečnostní chybu, také dopomůže webovým stránkám k lepšímu zabezpečení.

Stejně jako je zapotřebí omezit počet pluginů, omezuje se i přístup uživatelských účtů k citlivým informacím. V rámci laboratorní skupiny mají všichni její členové přístup ke kompletnímu obsahu webových stránek i jejím citlivým informacím. Díky tomu se zvyšuje riziko zásahu do funkčnosti stránek, ať už úmyslného, či nikoliv.

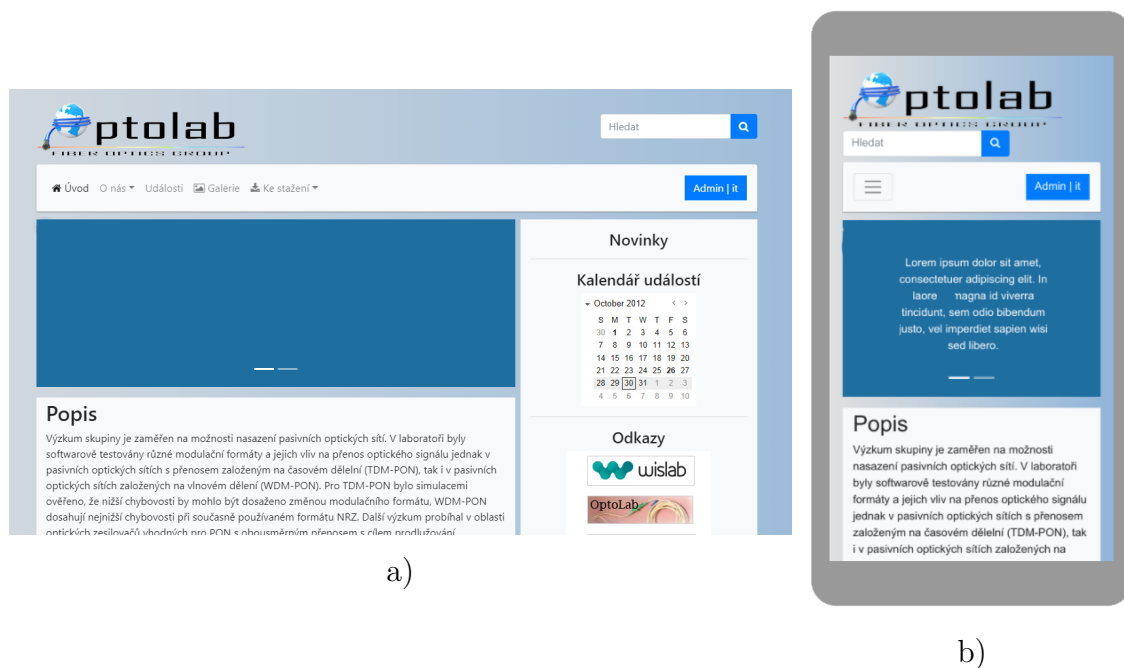
²je metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech

7 NÁVRH A ŘEŠENÍ WEBOVÝCH STRÁNEK

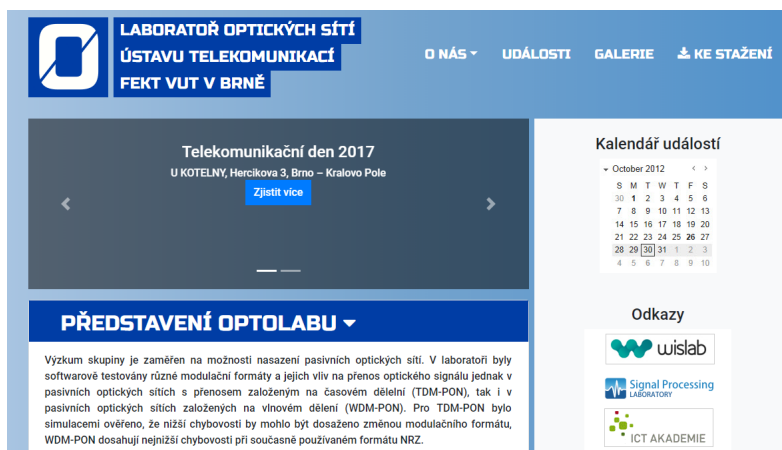
V této kapitole se věnuji konkrétním přínosům, kterých jsem během této práce dosáhl. Při návrhu vycházím z technologií popsanych v kapitole 2. Dále z kapitoly 6, kde jsem zhodnotil stav aktuálně používaných stránek. Výsledný návrh webových stránek je dostupný přes webhosting společnosti Endora na adrese [38], který byl vytvořen pro testovací účely.

7.1 Návrh struktury a grafiky

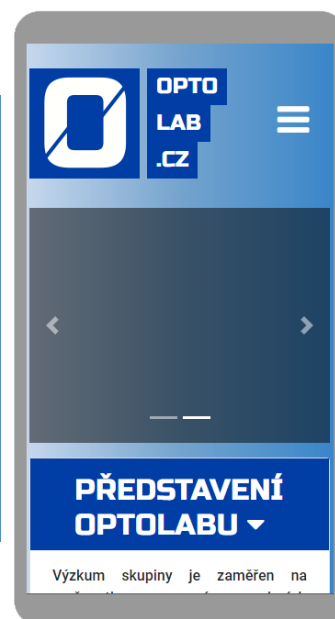
Při návrhu rozložení webu vycházím z vlastních zkušeností. Bylo by například zcela nevhodné umísťovat hlavní menu stránky do spodní části webu, stejně tak ho nedefinuji nejmenším písmem a nevýraznými barvami. Vzhledem k nejasné představě členů Optolabu jsem tak zpracoval několik grafických návrhů, ze kterých jsem následně vybral nejvíce vyhovující návrh podle doporučení členů skupiny.



Obr. 7.1: Návrh první pro a) desktopovou, b) mobilní verzi



a)

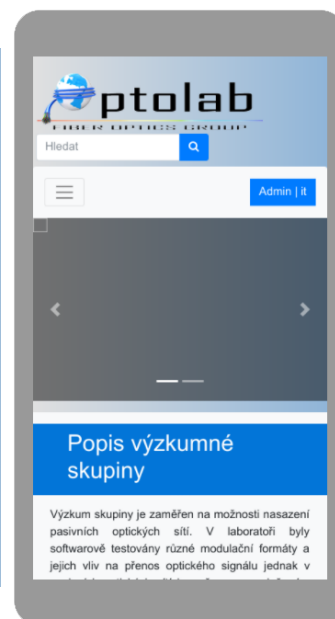


b)

Obr. 7.2: Návrh druhý pro a) desktopovou, b) mobilní verzi



a)



b)

Obr. 7.3: Návrh třetí pro a) desktopovou, b) mobilní verzi

Jednotlivé návrhy již byly vytvořeny pomocí HTML a CSS kódu s podporou knihovny Bootstrap, díky které se přizpůsobovaly velikosti zobrazovacího zařízení.

Veškeré návrhy si zakládají na společných prvcích a barevné kombinaci, avšak liší se v různých detailech. Prvním grafickým návrhem z obr. 7.1 jsem se snažil vnést co největší jednoduchost, druhý návrh, vyobrazený na obr. 7.2 byl především více kontrastní a tmavší. Třetí a poslední obr. 7.3 spojuje kombinaci obojího a právě tento návrh byl zvolen nejvhodnějším.

Stránky jsou tvořeny pomocí předem nadefinovaných tříd objektů z knihovny Bootstrap, které jsou následně upraveny dle potřeby. Díky využití těchto tříd jsou veškeré komponenty nadefinovány na několik různých šíří daného prohlížeče, viz tabulka.

Šířka prohlížeče	Popis
méně než 576 px	Extra malá zařízení
576 px a více	Malá zařízení (telefony)
768 px a více	Střední zařízení (tablety)
992 px a více	Velká zařízení (desktohy, notebooky)
1200 px a více	Extra velká zařízení

Tab. 7.1: Rozdělení tříd Bootstrapu podle velikosti zařízení

Jednotlivé prvky stránek se pak upravují podle nastavených tříd pro danou šíři. Hlavním prvkem obsahujícím celé stránky je kontejner, který má maximální šíři 1140 px. Ten je pak dále rozdělen do tří horizontálních částí – záhlaví, obsahové části a zápatí.

V rámci záhlaví je řešeno hlavní menu a logo skupiny Optolab. Z toho důvodu využívám konceptu loga, které dodá specializovaný grafik.

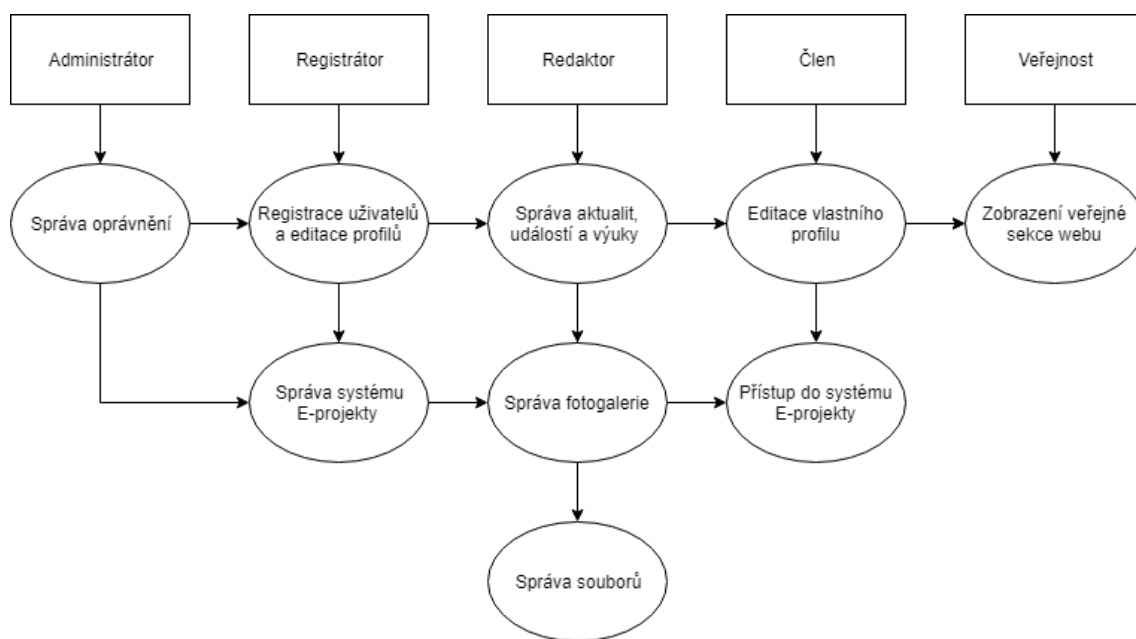
Obsahová část je rozdělena na dva sloupce, z nichž jeden nese informace o chystaných událostech, které budou zobrazeny v kalendáři. Dále obsahuje seznam aktualit, odkazy na spolupracující společnosti apod. Zbýlá část obsahové části vyplní hlavní text dané podstránky. V případě úvodní stránky webu se zobrazí popis samotné organizace.

Zápatí je využíváno pouze pro zobrazení kontaktních informací, odkazů na sociální síť skupiny, případně stránky školy, pod kterou organizace patří.

7.2 Správa webových stránek

V rámci řešení webových stránek je také její interní sekce, díky které bude možné stránky udržovat. Interní sekce bude řešit správu uživatelů, souborů, fotogalerie a především správu aktualit a událostí. Uživatelé budou mít přístup k webové prezentaci pro zobrazení aktualit, případně informací o skupině a členové skupiny budou moci upravovat webový kontext.

Na obrázku níže je tak vyobrazen vyobrazen diagram možností webových stránek.



Obr. 7.4: Diagram přístupu funkcí uživatelům stránek

Pro řešení tak komplexního systému se často využívá redakčních systémů, které usnadňují tvorbu stránek a mimo jiné umožňují přiřazovat práva jednotlivým uživatelům a přihlašovat se k systému dle těchto práv. Nynější stránky pracují na systému WordPress, který však obsahuje určitá bezpečnostní rizika popsána v kapitole 6.7. Dalším aspektem pro výběr redakčního systému byla složitost, respektive jednoduchost administrace, kterou měla stránka Optolabu zastávat.

Vzhledem k využití pouze jednodušších funkcí, potřebných v rámci administrace stránek, jsem se rozhodl tyto redakční systémy vynechat a využít vlastní tvorby. Administrace tak bude zastávat pouze potřebné funkce. Díky tomu nebudou stránky zatěžovány nadbytečným kódem a tak nebude zbytečně zpomalován jejich provoz.

7.3 Uživatelé a jejich rozdělení

Uživatelem je fyzicky existující osoba, užívající webový portál. Může se jednat o uživatele bez přístupu do interní sekce neboli návštěvníka, případně o přihlášeného uživatele, tedy člena výzkumné skupiny. V případě registrovaného člena se nejprve musí ověřit jeho identita přes přihlašovací formulář, zadáním jeho emailové adresy a hesla. V případě úspěšného ověření uživatele dostává skript na serveru přímo od interpreta PHP pomocí superglobální proměnné `$_SESSION` uživatelské ID, jeho práva přístupu, které jsou popsány v kapitole 7.4 a čas poslední aktivity uživatele.

Registraci nového člena může provést pouze jiný člen s dostatečnými právy. Každý registrovaný uživatel má možnost editace svých osobních údajů a zobrazení sekce E-projekty. Z osobních údajů jsou povinné pouze jméno, emailová adresa a heslo. Zbylé údaje jsou volitelné a uživatel je může kdykoli změnit. V případě nastavení profilu jako viditelný, se profil stává veřejným a některé údaje profilu tak budou viditelné i pro veřejnost. Díky této možnosti se tak může vytvořit správce webových stránek, aniž by se nutně musel stát členem této skupiny.

7.4 Práva přístupu

Možnosti editace stránek se pro každého uživatele liší, podle proměnné uložené v databázi pod každým profilem. Proměnná může nabývat čtyř různých hodnot, přičemž se jedná o násobky dvou. Každá hodnota reprezentuje výši pravomocí daného člena a plní, že čím vyšší je hodnota proměnné, tím více pravomocí uživatel získá. V následující tabulce jsou zobrazeny hodnoty, jaké funkce reprezentují a jaké možnosti jsou pro zvolenou funkci přístupné. V případě vyšších hodnot získává uživatel povolené funkce od nižších.

Hodnota	Název	Povolené funkce
1	Člen	Správa vlastního profilu
2	Redaktor	Správa aktualit, událostí, výuky a galerie
4	Registrátor	Správa e-projektů a registrace členů
8	Administrátor	K dispozici funkce celého webu

Tab. 7.2: Rozdělení pravomocí uživatelů

Každý člen dostane od registrátora případně administrátora přiřazenou hodnotu přístupových práv, kterou také mohou později měnit. Nastavená hodnota práv editovaného člena však nemůže být vyšší než hodnota práv editora.

7.5 Zabezpečení systému

Hlavním důvodem zabezpečení interní sekce je znepřístupnění manipulace s obsahem webových stránek a utajení dat na serveru. Nepřihlášeného uživatele systém pozná díky nevytvořené superglobální proměnné *session*. V případě, že se návštěvník bude snažit spustit některou ze stránek interní sekce, bude přesměrován na úvodní stránku.

Při zpracování zadaných údajů do formulářů, zabezpečuje systém proti útoku typu SQL injection PHP funkce `mysqli::real_escape_string`, která nahrazuje

speciální znaky, používané v SQL dotazech, jako jsou např. uvozovky, apostrofy nebo zpětná lomítka za bezpečnou podobu těchto znaků.

Hesla k uživatelským účtům jsou pro větší bezpečnost do databáze ukládána v šifrované podobě prostřednictvím hashovací funkce `hash()` s využitím algoritmu SHA-256. Navíc se hashované heslo tzv. solí, tedy k samotnému heslu se přidává další řetězec znaků. V tomto případě se přidává uživatelský email, přes který se členové přihlašují. Díky tomu má každý uživatel svou specifickou sůl, což zesiluje zadané heslo a výrazně ztěžuje slovníkový útok a útok přes duhové tabulky¹. Útočník místo jedné informace (hash) potřebuje znát tři nezávislé informace: hash, sůl a algoritmus, jakým se sůl kombinuje se vstupním heslem. Při snaze o prolomení hashovací funkce tak útočník nestačí najít pouze libovolný hash, který souhlasí s uloženým v databázi, ale musí najít ten, který obsahuje konkrétní sůl, pak z ní sůl odstranit a výsledek teprve může zadat jako heslo.

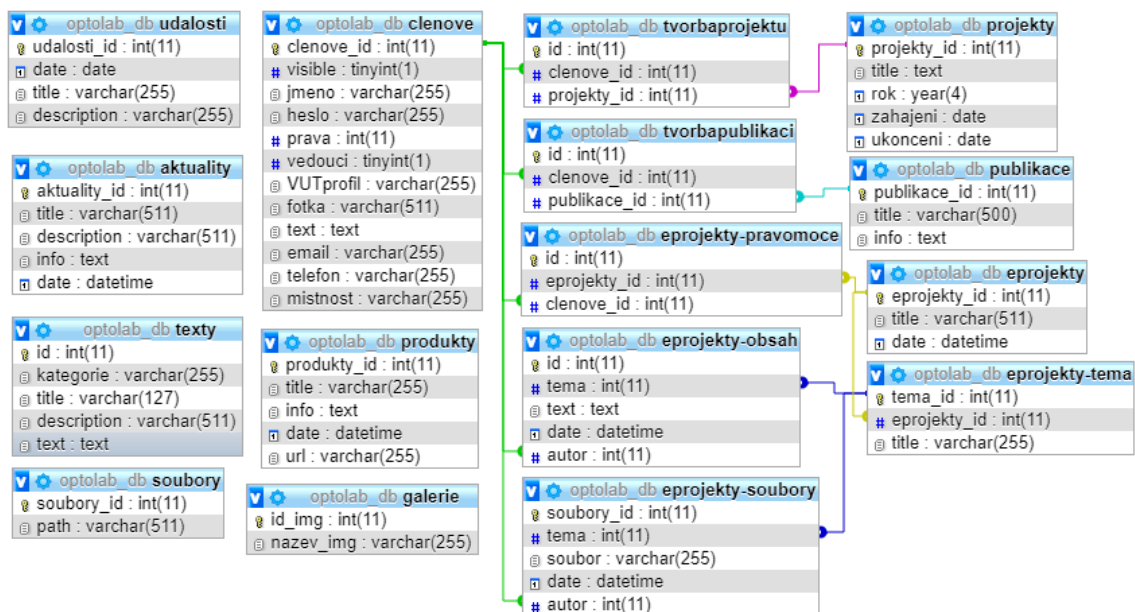
Ochrana proti odposlechu spočívá v použití šifrovaného protokolu https, na který je uživatel při vstupu na stránky vždy přesměrován. Tento protokol využívá SSL certifikát, který je potřeba zřídit. Díky tomu jsou všechny citlivé informace, které návštěvník s portálem sdílí šifrované a nikdo tak nemůže odposlouchávat komunikaci mezi serverem a klientem a zjistit tak především přihlašovací údaje uživatelského účtu. Dodání certifikátu si však musí zařídit zřizovatel ze strany serveru.

7.6 Popis databáze

7.6.1 Rozpis tabulek a jejich relace

Návrh nové databáze čítá 16 tabulek, které jsou mezi sebou propojeny jedenácti relacemi. Všechny tabulky popíšu níže a u každé z nich zhodnotím jejich atributy a jaká data lze zadat do každé hodnoty atributu. V návrhu jsou si některé tabulky velmi podobné, proto vždy popíšu jen jednu z nich. Pro lepší čitelnost budu tabulky jmenovat s diakritikou, i když jsou v samotném systému zadávány bez diakritiky. Celá databáze se všemi relacemi na obr. 7.5.

¹Označení pro tabulku s vygenerovanými páry heslo/hash pro všechny možné kombinace z požadované množiny znaků se zadanou délkou hesel.



Obr. 7.5: Grafické rozvržení databáze v softwaru phpMyAdmin

Tabulka členové

Tabulka se sestává z primárního unikátního klíče s názvem *členové_id*. Jedná se o typ integer, který může obsahovat pouze číselné entity. Při vkládání nového uživatele se vygeneruje hodnota o jedna vyšší, než byla hodnota předchozí. Při registraci nového uživatele je nutné vložit hodnoty *jméno*, *email* a *heslo*, které zastávají typ varchar, obsahující textové řetězce o maximální délce 255 znaků. Dále tabulka obsahuje atributy typu integer o maximální velikosti jednoho znaku s názvy *visible* a *vedoucí*, dopomáhající ke správnému zobrazení/skrytí členů na webové stránce a jejich pořadí. Důležitým atributem typu integer je také *práva*, který obsahuje hodnotu pravomocí daného uživatele 7.2. Zbývající atributy *VUTprofil*, *telefon* a *místopost* o maximálním počtu 255 znaků zastávají textovou formou url adresu VUT profilu uživatele, telefonního čísla a označení kanceláře. Posledním atribut tabulky s názvem *text* zastává krátký popis uživatele.

Tabulka události

Stejně jako všechny ostatní tabulky, obsahuje primární klíč unikátní ID s názvem *udalosti_id*. Z toho důvodu tento klíč u ostatní tabulek již nebudu zmiňovat. Hlavními atributy zde jsou *date* s datem konání a *title* s názvem události.

Tabulka aktuality

Naskýtá informace o aktualitách, které uživatelé vytváří. Mimo primárního identifikačního klíče obsahuje *title* neboli název aktuality a její krátký popis *description*, které se zobrazují na úvodní stránce webu. V případě otevření aktuality se zobrazí obsah atributu *info*, který obsahuje textový řetězec s html značkami pro stylizování řetězce. Značky jsou do řetězce vloženy automaticky, pomocí textového WYSIWYG editoru, který je dále popsán v kapitole 7.7.2

Tabulka texty

Podobně jako v tabulce s aktualitami se zde nachází atributy jako *title* a *description*, zde však zastávají důležitou roli pouze pro entitu s atributem *kategorie* obsahující řetězec „download“. Ten označuje, že se jedná o obsah spojený s výukou předmětů. Dále se zde nachází kategorie „index“ a „footer“, obsahující texty pro úvodní stránku a zápatí webových stránek.

Tabulka soubory a galerie

V rámci této tabulky se ukládá pouze relativní umístění souboru, případně složky, díky které mohou být tyto soubory načítány z různých adresářů. Podobným způsobem pracuje i tabulka galerie, kde jsou uloženy názvy nahraných souborů.

Tabulka produkty, projekty a publikace

Tabulky produktů, projektů a publikací jsou si velmi podobné. Každá z nich obsahuje atribut *title*, obsahujícím název dané položky a nadále několik atributů popisujících tuto položku. Ať už se jedná o datum zahájení, respektive ukončení projektu, uložených v atributu typu *date*, nebo o textový řetězec s podrobnějšími informacemi o dané položce v atributu *info*.

Tabulka tvorba-projektů a tvorba-publikací

Ke správnému zobrazení projektů a publikací, na kterých se člen podílí, je zapotřebí mezi tabulkou *členové* a tabulkami *projekty* a *publikace* vytvořit relaci M-N, popsanou v kapitole 2.5. *Tvorba-projektů* a *tvorba-publikací* tak zastávají prostředníka, ve kterém se ukládají identifikační klíče členů a projektů, respektive publikací.

Tabulka eprojekty

Jedná se o stěžejní tabulku interního systému E-projekty. Ukládají se zde názvy projektů – atribut *title* ve formátu *varchar* a data poslední změny ve formátu *datetime*

do atributu *date*. Jakákoliv změna v jednotlivých projektech tak upraví datum a čas jejich poslední změny. Nechybí ani primární klíč, a tedy ID projektu, které se dále uplatňuje u relacích s ostatními tabulkami.

Tabulka eprojekty-pravomoce

Touto tabulkou se docílí přístupu do projektů pouze přidáním členům daného projektu. Funguje na podobném principu, jako tabulky *tvorba-projektů* a *tvorba-publikací*, tedy jedná se o prostředníka pro vytvoření M:N relace mezi *eprojekty* a *členové*. Každý řádek tak musí uchovávat jedinečný klíč člena a jedinečný klíč projektu, do kterého je člen přidán.

Tabulka eprojekty-téma

Pod každý projekt může spadat určitě téma, ve kterých členové komunikují či sdílí soubory. Z toho důvodu je zapotřebí této tabulky, v níž se nachází název tématu v podobě varchar atributu *title* a především unikátní klíč projektu, pod který dané téma spadá. Vytvořil jsem zde tedy relaci 1-N, popsané v kapitole 2.5, kdy ke každému projektu může spadat několik témat.

Tabulka eprojekty-obsah a eprojekty-soubory

Obsahem těchto tabulek jsou veškeré příspěvky a soubory, přidané do všech témat. Vzniká zde relace 1-N s tabulkou *eprojekty-téma* s pomocí atributu *téma*, obsahujícím identifikační klíč tématu a stejným způsobem je vytvořena relace 1-N s tabulkou *členové*, pro uložení unikátního identifikátoru autora příspěvku. Zbývající atributy obsahují samotný příspěvek, tedy textový řetězec s html značkami, pokud se jedná o příspěvek, nebo název, pokud o soubor. Do zbývajících atributů se ukládá datum a čas vytvoření příspěvku, respektive nahrání souboru.

7.6.2 Programové spojení s databází

Spojení s databází probíhá pomocí skriptovacího jazyku PHP, popsaného v kapitole 2.4. Tento jazyk nabízí dvě různá rozšíření pro spojení s MySQL databází. První možností je rozšíření *mysql*, které je k dispozici pro PHP verze 4 i 5 a pracuje pouze s MySQL serverem do verze 4.1. Toto rozšíření neumožňuje vylepšený autentizační protokol, ani nepodporuje volání předpřipravených příkazů.

Druhou možností je vylepšené rozšíření *mysqli*, které je k dispozici jen pro PHP verze 5 a vyšší a podporuje MySQL server verze 4.1.1 a vyšší. Toto rozšíření již umožňuje vylepšený autentizační protokol použitý v MySQL serveru 5.0, stejně jako předpřipravené příkazy. Navíc toto rozšíření nabízí pokročilé rozhraní pro objektově

orientované programování. V této práci je tak využívána komunikace s MySQL přes rozhraní *mysqli* v rámci objektově orientovaného prostředí.

Následující ukázka kódu 7.2 vyobrazuje, jak probíhá připojení k databázi v jazyce PHP:

```
$host = 'localhost'; //připojení na lokální server
$username = 'root'; //název uživatele
$password = ''; //heslo uživatele
$database = 'optolab_db'; //název databáze

$mysqli = new mysqli($host,$username,$password,$database);

if ($mysqli->connect_errno) {
    printf('Connect failed: %s\n', $mysqli->connect_error);
    exit();
}
```

Výpis 7.1: Připojení k místní databázi.

Připojení pracuje na základě vytvoření nové instance třídy *mysqli*, kde se jejímu konstruktoru předají informace potřebné k připojení. Mezi ně patří adresa hostitele, název databáze a uživatelské jméno a heslo. Pokud se nepodaří připojit k databázovému serveru, vypíše se chybové hlášení.

7.6.3 Komunikace s databází

Na základě prostředí webové stránky a vyvolání určitých situací se data z databáze načítají a zobrazují uživatelům, případně se ukládají do tabulek v databázi. K této komunikaci slouží takzvané dotazy, které se vyvolají metodou *mysqli::query*.

Veškerá komunikace při výběru dat z databáze probíhá podobným způsobem, proto zde popíšu pouze několik ukázkových kódů, použitých při programování stránek.

Výběr a zobrazení obsahu z tabulek

Základním příkazem v SQL dotazech bývá příkaz **SELECT**. Ten obsahuje části, kterými určujeme funkci příkazu. Při jejich zápisu záleží na pořadí, avšak jsou nepovinné a používají se pouze ty, které jsou potřebné.

V rámci kódu 7.2, vybereme příkazem **SELECT** všechny sloupce z tabulky *projekty* a zároveň všechny řádky z tabulky *tvorbaprojektů*, ve které se atribut *projekty_id* rovná stejnojmennému atributu z tabulky *projekty*. Tím dochází k vnitřnímu spojení

dvou zmíněných tabulek. Následně se vyberou pouze ty řádky, které obsahují atribut *clenove_id* rovný globální proměnné *profil*. Výsledek se seřadí sestupně podle atributu *rok*. V případě, že v tabulce existuje alespoň jeden řádek vyhovující daným kritériím, vypíšu se tyto řádky pomocí cyklu **while**. Pro lepší přehlednost v dotazu také využívám zkrácené podoby názvu tabulek, pomocí příkazu **AS**.

```
$query = 'SELECT 'projekty'.*
          FROM 'projekty' AS 'p'
          INNER JOIN 'tvorbaprojektu' AS 't'
            ON 'p'.'projekty_id' = 't'.'projekty_id'
          WHERE 't'.'clenove_id' = '$_GET['profil']'.'
          ORDER BY 'rok' desc';
$result = $mysqli->query($query);
$row_cnt = $result->num_rows;
if($row_cnt != 0) {
    while($row = $result->fetch_assoc()) {...}
}
```

Výpis 7.2: Výběr obsahu z tabulky projektů.

Vložení nového obsahu

Pro vkládání obsahu se používá SQL příkaz **INSERT** a patří mezi jeden z nejjednodušších příkazů. V rámci webových stránek se tento příkaz používá výhradně v interní sekci, nepřihlášený uživatel tak nemá možnost zasahovat do chodu databáze. Může pouze zobrazit výsledný obsah.

Ukázka kódu 7.3 se týká přidání jednoho řádku do tabulky *publikace*, ve které se nachází atributy vypsané v ukázce. Tyto atributy nejsou jediné, které jsou obsažené v tabulce, avšak zbylé se doplňují automaticky podle určité funkce nebo výchozí hodnoty, případně se ponechají prázdné. Do atributů se ukládají stejnojmenné PHP proměnné, ve kterých jsou uloženy hodnoty načtené z formuláře. Funkce s názvem **real_escape_string** opatří všechny znaky, které by mohly mít určité bezpečnostní potíže, zpětným lomítkem. Toto opatření zabrání, aby upravený vstup mohl způsobit nějakou škodu, nebo vedlejší efekt, který původně programátor nezamýšlel.

```
$title = $_POST['TITLE'];
$title = $mysqli->real_escape_string($title);
$info = $_POST['INFO'];
$info = $mysqli->real_escape_string($info);

$query="INSERT INTO 'publikace' ('title', 'info')
```

```
VALUES ( '$title', '$info' );
```

```
$sql=$mysqli->query($query) or die(mysql_error());
```

Výpis 7.3: Přidání nového řádku do tabulky projektů.

Podobným způsobem se vkládá obsah i do ostatních tabulek, proto není nutné vypisovat jednotlivé příkazy.

Úprava stávajícího obsahu

Podobně jako u příkazu `INSERT` se při úpravě již vytvořených dat využívá `UPDATE`. V případě užití tohoto příkazu bez podmínky, nastavili bychom dané hodnoty ve všech řádcích tabulky. Tento případ se však v praxi příliš nevyužívá. V případě jejího využití se však hodnoty nastaví pouze u řádků, splňujících danou podmínku. Příkaz v ukázce 7.4 navazuje na předešlý kód 7.3, jediná úprava se týká proměnné `$query`, ve které se nyní využije příkaz `UPDATE` s využitím příkazu `WHERE`, zastávající podmínku, tudíž příkaz platí pouze pro řádek, obsahující hodnotu atributu `publikace_id` rovné proměnné `id`.

```
$query="UPDATE 'publikace' SET  
    'title' = '$title',  
    'info' = '$info'  
WHERE 'publikace_id' = '". $_GET['id'] . "' ;"
```

Výpis 7.4: Přepsání obsahu řádku z tabulky publikace.

7.7 Možnosti portálu

V této kapitole se věnuji popisu hlavních funkčních prvcích, které se na stránkách vyskytují, případně jak s nimi uživatel může pracovat. Nebudu vysvětlovat zdrojový kód, nýbrž za jakým účelem jsem daný výsledek realizoval a jakým způsobem to ovlivní práci uživatele.

7.7.1 Interní sekce E-projekty

Členský interní systém kromě správy webového obsahu dále nabízí možnosti vytvoření projektů, do kterých se mohou již registrovaní uživatelé vzájemně přidávat, v případě že toto právo mají. Dále zde sdílet své výsledky výzkumů, své názory, případně i soubory k jednotlivým projektům.

7.7.2 Správa textů

Interní sekce nabízí vytváření, editaci a mazání několika různých prvků, rozdělených podle již zmíněných tabulek v kapitole 7.6.1. Tyto tabulky v sobě ukrývají mimo jiné i textové atributy, do kterých se může ukládat textový řetězec, včetně html značek, které následně budou pozorovatelné v podobě vizuální změny textu. Aby editoři nemuseli nutně užívat pro editaci textu HTML tagy, využívá se tzv. textových wysiwyg editorů.

Tyto editory splňují roli prostředníka mezi běžným uživatelem, který nezná HTML kód, a strojem, pracujícím na základě příkazů, respektive značek v jazyce značkovacím. Z toho důvodu využívám v této práci volně dostupného editoru TinyMCE [39], který při použití textového pole využívá vlastní API, jež pracuje jako běžný textový editor.

7.7.3 Správa členů

Jak již bylo zmíněno v kapitole 7.4, jednotlivý člen má určitá přístupová práva. Každý člen může spravovat svůj profil, který může i nemusí být viditelný veřejnosti. Každý si do profilu může přidat své kontaktní údaje, odkaz na VUT profil, krátký popis, případně publikace a projekty, na kterých se podílel. Jednotlivé projekty a publikace se přidávají pomocí vyskakovacího okna, ve kterém se nachází kompletní seznam všech přidávaných publikací či projektů na stránky. V případě, že se v seznamu nenachází a týká se bezprostředně laboratorní skupiny, musí být nejprve přidány do seznamu projektů, respektive publikací. Teprve poté mohou být přiřazeny k jednotlivým členům.

7.7.4 Správa souborů

V případě nutnosti sdílet na webových stránkách dokumenty, je tento postup možný následujícím způsobem. Uživatel s dostatečnými právy pro nahrávání souborů může soubory nahrávat v interní sekci *Správa souborů*. Zde je možné vytvořit adresáře, které mohou tvořit stromovou strukturu. Díky tomu se mohou jednotlivé soubory ukládat přehledněji. Jedna z nevýhod tohoto systému je ta, že v případě nahrání souborů jiným způsobem, např. přes FTP klienta, nebude soubor dostupný z webu. Což je způsobeno ukládáním názvu, respektive adresy souborů do databáze, při nahrávání souborů přes web.

Mimo adresář nahraných souborů se na webu nachází také složka s e-projekty, která slouží také pro sdílení souborů mezi členy, avšak pouze těmi, kteří se na daném projektu podílejí.

7.7.5 Kalendář událostí

K zobrazení událostí aktuálního měsíce v kalendáři využívám vytvořených PHP a JS funkcí, které pracují společně s tabulkou událostí. V té je uložen název události a datum konání. Tyto události může člen v interní sekci editovat, popřípadě vytvářet nové. Výsledek je následně zobrazen v boční liště veřejně přístupné části webu v podobě kalendáře.

Kromě zvýrazněného aktuálního data se u dnů s plánovanou událostí zobrazí barevný bod. Po přejetí myši přes daný den, se zobrazí název plánované události.

Jednou v velkých nevýhod tohoto přístupu, je nemožnost propojení s kalendáři typu kalendář Google, nebo Microsoft Calendar.

7.8 Možnosti rozšíření

Velkou nevýhodou většiny webových aplikací je nutnost přístupu na samotné stránky, pokud chce být uživatel dostatečně informován, co se na stránkách událo nového. To se týká především interní sekce E-projekty, která má pracovat na principu podobném sociálním sítím, avšak pouze pro uzavřenou skupinu lidí. Značným rozšířením pro tento systém by tak mohlo být zasílání emailových zpráv, které by registrované uživatele informovaly o přidání nového příspěvku, či nahrání souborů v určitém projektu.

S podobným problémem se potýká i upozornění na události, které bylo popsáno v předchozí kapitole 7.7.5. Možné rozšíření by mohlo být propojení s účtem Google, přes který by se synchronizoval kalendář Optolabu. V případě registrace nového člena by se tak kalendář mohl sdílet mezi uživateli i pomocí mobilních aplikací.

Mimo samotné sdílení a upozornění by se stránky mohly v budoucnu dočkat více uživatelských funkcí, jako například změna pozadí stránek, bez nutnosti editace zdrojového kódu, vkládání podkladového obrázku pro aktuality na úvodní stránce apod. Z větší části se tak jedná spíše o kosmetické úpravy webu.

V rámci zabezpečení stránek se místo využití starší ochrany proti SQL injection metodou `real_escape_string`, ošetřující vkládané proměnné, může využít nového přístupu a to pomocí parametrizovaných dotazů, díky kterým se proměnné do SQL dotazu nevkládají přímo. Dotazy se nejprve připraví tak, že místo hodnot se napíší zástupné znaky. Hodnoty a dotaz se předá databázi odděleně a ta je automaticky vloží zabezpečeným způsobem.

Vzhledem ke sdílení dat mezi uživateli by také bylo vhodné implementovat zálohovací funkci, která bude data z databáze a adresářů se sdílenými soubory, zálohovat na adresu mimo webový portál. Čímž by se v případě ztráty dat, mohl obsah obnovit.

Jedním z plánovaných doplňků pro webový portál je také technická realizace připojení do federace EduID, popsána v kapitole 5.3, díky které by se uživatelé mohli přihlašovat do interní sekce účtem z tohoto portálu.

8 ZÁVĚR

Cílem bakalářské práce bylo vytvoření webových stránek pro laboratoř optických sítí Optolab, včetně sekce pro administraci webového obsahu a interní sekce pro komunikaci a sdílení souborů mezi členy. Tohoto cíle se dle mého názoru podařilo dosáhnout s požadavky vedoucího práce i některých členů Optolabu.

Vzhledem k původnímu stavu stránek, vytvořených pomocí WordPressu, bylo zapotřebí vytvořit webovou prezentaci na míru, která nebude zbytečně zatěžována složitým redakčním systémem, nabízejícím služby, které se nevyužijí. Design návrhu se snaží působit moderně a jednoduše, zároveň web řeší i přizpůsobení obsahové části pro různé velikosti zobrazovacího zařízení. Následně bylo řešení analyzováno testem použitelnosti v mobilních zařízeních od společnosti Google, ve kterém stránky uspěly.

Všechny aplikované technologie podléhající doporučeným standardům byly podstoupeny kontrole v podobě validace zdrojových kódů. Tato analýza vyhodnotila pouze několik drobných nedostatků, nicméně kód je po optimalizační stránce na dobré úrovni. Správnou funkci tak splňuje většina on-page faktorů SEO optimalizace.

LITERATURA

- [1] A Very Short History Of The Internet And The Web. *Forbes* [online]. 2015, **2015** [cit. 2017-12-12]. Dostupné z: <https://www.forbes.com/sites/gilpress/2015/01/02/a-very-short-history-of-the-internet-and-the-web-2/10/#44f2b48a2690>
- [2] Co je to Internet a jak vznikl. *Orbis Pictus* [online]. Plzeň: TOP-CZ, ©2017 [cit. 2017-12-12]. Dostupné z: <https://www.orbis-pictus.topcz.net/internet.html>
- [3] Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. STEJSKAL, Jan. *Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu*. Brno: Computer Press, 2004, s. 11-12. ISBN 80-251-0167-3.
- [4] Grafické rozhraní prohlížeče Mosaic v prostředí MS Windows. In: *Internet Society / Twitter* [online]. Internet Society, 2015 [cit. 2017-12-14]. Dostupné z: <https://twitter.com/internetsociety/status/590863343444975616>
- [5] *The Apache HTTP Server Project* [online]. The Apache Software Foundation, ©1997-2017 [cit. 2017-12-12]. Dostupné z: <https://httpd.apache.org/>
- [6] The Secret To 10 Million Concurrent Connections: The Kernel Is The Problem, Not The Solution. *High Scalability* [online]. TODD HOFF, 2013 [cit. 2017-12-12]. Dostupné z: <http://highscalability.com/blog/2013/5/13/the-secret-to-10-million-concurrent-connections-the-kernel-i.html>
- [7] Péhápkó: Učebnice PHP. *Péhápkó.cz* [online]. Michal Mencl [cit. 2017-12-12]. Dostupné z: <http://pehapko.cz/sprava-serveru/instalace>
- [8] STEJSKAL, Jan. *Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu*. Brno: Computer Press, 2004. ISBN 80-251-0167-3.
- [9] Základy HTML, jak na HTML. *Jak psát web, návod na html stránky* [online]. Praha: Dušan Janovský [cit. 2017-12-12]. Dostupné z: <https://www.jakpsatweb.cz/zaklady-html.html>
- [10] <https://www.zdrojak.cz/clanky/xhtml-je-mrtve-at-zije-html5-nebo-ne/>. *Zdroják - o tvorbě webových stránek a aplikací* [online]. Praha: Devel.cz Lab, 2017 [cit. 2017-12-12]. Dostupné z: <https://www.zdrojak.cz/clanky/xhtml-je-mrtve-at-zije-html5-nebo-ne/>
- [11] PROKOP, Marek. *CSS pro webdesignery*. Vyd. 2. Brno: CP Books, 2005. ISBN 80-251-0487-7.

- [12] CSS for Those Just Learning To Code. *HTML.com: Learn HTML The Easy Way* [online]. London: Quality Nonsense, ©2015-2017 [cit. 2017-12-12]. Dostupné z: <https://html.com/css/>
- [13] PHP (1) - Historie a budoucnost. *Linux Software* [online]. Praha: Pavel Kysilka, 2004 [cit. 2017-12-12]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=171
- [14] What is PHP? *PHP: Hypertext Preprocessor* [online]. The PHP Group, ©2001-2017 [cit. 2017-12-12]. Dostupné z: <http://php.net/manual/en/intro-what-is.php>
- [15] Understanding PHP basics. *Packt Publishing / Technology Books, eBooks & Videos* / [online]. Birmingham: Packt Publishing Limited, 2004 [cit. 2017-12-12]. Dostupné z: <https://www.packtpub.com/books/content/understanding-php-basics-0>
- [16] Databáze. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2017 [cit. 2017-12-14]. Dostupné z: <https://cs.wikipedia.org/wiki/Datab%C3%A1ze>
- [17] Tvorba databází v MySQL - I. *AbcLinuxu.cz - Linux na stříbrném podnose* [online]. Praha: Nitemedia, 2003 [cit. 2017-12-12]. Dostupné z: <http://www.abclinuxu.cz/clanky/navody/tvorba-databazi-v-mysql-i>
- [18] Overview. *PHP: Hypertext Preprocessor* [online]. The PHP Group, ©2001-2017 [cit. 2017-12-12]. Dostupné z: <http://php.net/manual/en/mysqli.overview.php>
- [19] Introduction: JavaScript | MDN. *MDN Web Docs* [online]. Mozilla, ©2005-2017 [cit. 2017-12-12]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- [20] What is content management system (CMS)? *Enterprise content management strategy, news and information* [online]. Newton: TechTarget, ©2011-2017 [cit. 2017-12-12]. Dostupné z: <http://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS>
- [21] About WordPress. *Blog Tool, Publishing Platform, and CMS WordPress* [online]. WordPress Foundation, 2003 [cit. 2017-12-12]. Dostupné z: <https://www.joomla.org/about-joomla.html>
- [22] About Joomla!. *Joomla!: Content Management System to build websites & apps* [online]. New York: Open Source Matters, ©2005-2017 [cit. 2017-12-12]. Dostupné z: <https://www.joomla.org/about-joomla.html>

- [23] About Drupal. *Drupal - Open Source CMS* [online]. Dries Buytaert, ©1999-2017 [cit. 2017-12-12]. Dostupné z: <https://www.drupal.org/about>
- [24] *Bootstrap: The most popular HTML, CSS, and JS library in the world* [online]. [cit. 2017-12-13]. Dostupné z: <http://getbootstrap.com/>
- [25] Bootstrap Get Started. *W3Schools Online Web Tutorials* [online]. Refsnes Data, ©1999-2017 [cit. 2017-12-13]. Dostupné z: https://www.w3schools.com/bootstrap/bootstrap_get_started.asp
- [26] MARCOTTE, Ethan. Responsive Web Design. *A List Apart* [online]. 2010 [cit. 2017-12-13]. Dostupné z: <https://alistapart.com/article/responsive-web-design>
- [27] *Is adaptive better than responsive design?* [online]. [cit. 2017-12-13]. Dostupné z: <https://thenextweb.com/dd/2015/09/01/is-adaptive-better-than-responsive-design/>
- [28] COUSINS, Carrie. Is adaptive better than responsive design? *The Next Web* [online]. Amsterdam, 2015 [cit. 2017-12-13]. Dostupné z: <https://thenextweb.com/dd/2015/09/01/is-adaptive-better-than-responsive-design/>
- [29] VOJÁK, Michal. Responsivní a adaptivní design. In: *AW-dev blog* [online]. 7.10.2015, 2015 [cit. 2017-12-13]. Dostupné z: <https://blog.aw-dev.cz/wp-content/uploads/responsivni-vs-adaptivni-design1.png>
- [30] *Začínáme s optimalizací pro vyhledávače* [online]. Google, ©2010 [cit. 2017-12-13]. Dostupné z: <http://static.googleusercontent.com/media/www.google.cz/cs/cz/intl/cs/webmasters/docs/search-engine-optimization-starter-guide-cs.pdf>
- [31] DARIE, Cristian. *SEO v PHP: programujeme profesionálně* [online]. Brno: Computer Press, 2008 [cit. 2017-12-13]. Programmer to programmer. ISBN 978-80-251-2083-5.
- [32] WALL, Aaron Matthew. *Search Engine Optimization* [online]. Caldecott, ©2003-2017 [cit. 2017-12-13]. Dostupné z: <http://www.seobook.com/seobook53.pdf>
- [33] Jednotné přihlašování. *Jak na Internet* [online]. Praha: CZ.NIC, ©2012-2014 [cit. 2017-12-14]. Dostupné z: <https://www.jaknainternet.cz/page/1189/jednotne-prihlasovani/>

- [34] NOVOTNÝ, Pavel. *Systémy jednotného přihlášení — Single Sign On (SSO)*. Praha, 2009. Bakalářská práce. Bankovní institut vysoká škola.
- [35] *OptoLab / Laboratoř optických sítí ústavu Telekomunikací FEKT VUT v Brně* [online]. Brno: OptoLab [cit. 2017-12-14]. Dostupné z: <http://optolab.utko.feec.vutbr.cz/>
- [36] Bezpečnost WP — úvod — díl 1. *Lynt services* [online]. Praha: Lynt services, ©2013 [cit. 2017-12-14]. Dostupné z: <https://lynt.cz/blog/bezpecnost-wp-uvod-dil-1>
- [37] WordPress < 4.1.2 Stored XSS vulnerability. *Cedric's Cruft* [online]. Cedric Van Bockhaven, ©2013-2017 [cit. 2017-12-14]. Dostupné z: <https://cedricvb.be/post/wordpress-stored-xss-vulnerability-4-1-2/>
- [38] *Optolab: Návrh webových stránek* [online]. Brno: Ondřej Zámečník, 2018 [cit. 2018-05-17]. Dostupné z: <https://zamec-webhost.cz/optolab/>
- [39] *TinyMCE / The Most Advanced WYSIWYG HTML Editor* [online]. Palo Alto: Ephox Corporation, 2018 [cit. 2018-05-21]. Dostupné z: <https://www.tinymce.com/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

TCP/IP	Transmission Control Protocol/Internet Protocol
CERN	European Organization for Nuclear Research
WWW	World Wide Web
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
PHP	Hypertext Preprocessor
SQL	Structured Query Language
SGML	Standard Generalized Markup Language
API	Application Programming Interface
CSS	Cascading Style Sheets
JS	JavaScript
CMS	Content Management System
WYSIWYG	What you see is what you get
URL	Uniform Resource Locator
SSO	Single Sign-On