

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačních technologií**



**Bakalářská práce**

**Tvorba webu založeného na CMS "WordPress"**

**Oleksandr Kutsyn**

© 2020 ČZU v Praze

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Oleksandr Kutsyn

Systémové inženýrství a informatika  
Informatika

Název práce

**Tvorba webu založeného na CMS "WordPress"**

Název anglicky

**Creating a web-site based on CMS "WordPress"**

---

### Cíle práce

Bakalářská práce je tematicky zaměřena na problematiku vývoje webových stránek v CMS "WordPress". Hlavním cílem práce je tvorba webových stránek v komparaci s ověřením funkčnosti vytvořených webových stránek v CMS „WordPress“.

Díličními cíli jsou:

- tvorba jednotlivých variant vývoje webových stránek.
- metodický postup vývoje webových stránek a jejich možné způsoby umístění na vybraný webový server.

### Metodika

Teoretická část práce spočívá ve studiu informací z odborné literatury a primárních zdrojů. Na základě informací získaných z odborných pramenů bude sepsána řešeršní část práce.

Metodika řešené problematiky bakalářské práce je založena na studiu a analýze odborných informačních zdrojů. Vlastní práce spočívá v tvorbě vybrané webové stránky v CMS „WordPress“ pro dle stanovených kritérií a bude rozdělena do tří částí:

- vývojové prostředí – prostředí, ve kterém bude proveden veškerý vývoj a úprava webové stránky;
- testovací prostředí – prostředí, ve kterém bude provedeno testování před distribucí do produkčního prostředí;
- produkční prostředí – prostředí, které bude použito pouze pro distribuci webových stránek a jejich prezentaci uživatelům.

Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry bakalářské práce.

## Doporučený rozsah práce

30 – 40 stran

## Klíčová slova

CMS, WordPress, WP, vývoj, webová stránka

---

## Doporučené zdroje informací

BAŠE, Ondřej. PHP Okamžitě. 1.vyd. Brno: Computer Press, 2014. ISBN 978-80-251-4196-0

DOMES, Martin. Tvorba WWW stránek pro úplné začátečníky. 1. vydání, Brno: Computer Press, 2012. ISBN 978-80-251-2160-3

ŠPINAR, David. Tvoříme přístupné webové stránky : připraveno s ohledem na novelu Zákona č. 365/2000 Sb., o informačních systémech veřejné správy. Brno: Zoner Press, 2004. ISBN 80-86815-11-0.

WordPress dokumentace [online]. WordPress.org, 2017 [cit. 2017-12-27]. Dostupné z: <https://codex.wordpress.org/>

---

## Předběžný termín obhajoby

2019/20 LS – PEF

## Vedoucí práce

Ing. Eva Kánská

## Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 2. 10. 2018

**Ing. Jiří Vaněk, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 19. 10. 2018

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 23. 03. 2020

### **Čestné prohlášení**

Prohlašuji, že svou bakalářskou práci "Tvorba webu založeného na CMS WordPress" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 23. března 2019

---

### **Poděkování**

Rád(a) bych touto cestou poděkoval(a) Ing. Evě Kánský za odborné vedení při tvorbě této bakalářské práce.

# Tvorba webu založeného na CMS "WordPress"

## Abstrakt

Obsahem této bakalářské práce je shrnutí teoretickým poznatku v oblasti vývoje webu nutných k pochopení tvorby webových aplikací a také pro jejich implementaci v reálném životě. V této bakalářské práci jsou popsány všechny technologie, které byly použity v praktické části jak pro vývoj tak pro testování.

V praktické části bakalářské práce se autor zabývá vývojem webové stránky a její následným testováním také jednotlivými metodickými postupy, které jsou spojeny s nasazením webu do produkce. Na základě předložených požadavků autorovi bude vyvinuta webová stránka, která bude sloužit ke zviditelnění společnosti na trhu a také zefektivnění práce s rezervací služeb. Webová stránka bude vytvořena na CMS „WordPress“.

**Klíčová slova:** CMS, WordPress, WP, vývoj, webová stránka, MySQL, SQL.

# Creating a web-site based on CMS "WordPress"

## **Abstract**

The content of this thesis is a summary of theoretical knowledge in the area of web development that necessary to understand the creation of web applications and for their implementation in real life. In this thesis are described all the technologies that were used in the practical part for development and testing.

In the practical part of the thesis, the author develops web and after about its subsequent testing and individual methodical procedures that are associated with the deployment of the web to production server. Based on the submitted requirements the author will develop a website that will serve to promote the company on the market and to work more effectively on booking services in the company. The website will be created on CMS "WordPress".

**Keywords:** CMS, WordPress, WP, development, website, MySQL, SQL.

# Obsah

|  |           |
|--|-----------|
| <b>1 Úvod</b> .....                                | <b>9</b>  |
| <b>2 Cíl práce a metodika</b> .....                | <b>10</b> |
| 2.1 Cíl práce .....                                | 10        |
| 2.2 Metodika .....                                 | 10        |
| <b>3 Teoretická východiska</b> .....               | <b>11</b> |
| 3.1 WordPress .....                                | 11        |
| 3.2 Varianty vývoje webových stránek: .....        | 12        |
| 3.2.1 Strukturní editory .....                     | 12        |
| 3.2.2 Wysiwyg editory .....                        | 13        |
| 3.2.3 On-line editory .....                        | 13        |
| 3.2.4 Redakční systémy neboli tzv. „CMS“ .....     | 15        |
| 3.2.4.1 Princip fungování CMS .....                | 15        |
| 3.2.4.2 Praktická aplikace.....                    | 16        |
| 3.2.4.3 Výhody a nevýhody.....                     | 16        |
| 3.3 Technologie používané při vývoji webu .....    | 17        |
| 3.3.1 Apache .....                                 | 17        |
| 3.3.2 HTML .....                                   | 17        |
| 3.3.3 CSS .....                                    | 19        |
| 3.3.3.1 Jak funguje CSS .....                      | 19        |
| 3.3.3.2 Příklady: .....                            | 20        |
| 3.3.3.3 Varianty zápisu .....                      | 20        |
| 3.3.4 PHP .....                                    | 21        |
| 3.3.5 MySQL .....                                  | 22        |
| 3.3.5.1 SQL.....                                   | 23        |
| 3.3.5.2 Ukázka příkladů: .....                     | 24        |
| 3.3.6 JavaScript.....                              | 24        |
| 3.3.6.1 Výhody JavaScriptu.....                    | 25        |
| 3.3.6.2 Nevýhody .....                             | 25        |
| 3.3.6.3 Jak JavaScript funguje na vašem webu ..... | 26        |
| 3.3.7 JQuery .....                                 | 27        |
| 3.3.7.1 Způsoby připojení jQuery.....              | 27        |
| 3.3.7.2 Příklady jQuery: .....                     | 28        |
| 3.3.8 Bootstrap.....                               | 29        |



|          |   |           |
|----------|---|-----------|
| 3.3.8.1  | Výhody a nevýhody.....  | 30        |
| 3.3.9    | Vývojové prostředí JetBrains PhpStorm.....                                    | 30        |
| 3.3.9.1  | Podpora hlavních frameworků .....   | 30        |
| 3.3.10   | Xdebug.....   | 32        |
| 3.3.10.1 | Odstraňování potíží s pomocí Xdebugu .....                                    | 33        |
| 3.3.10.2 | Vývoj prostřednictvím testování .....   | 33        |
| <b>4</b> | <b>Vlastní práce .....</b>  | <b>34</b> |
| 4.1      | Úvodní analýza.....   | 34        |
| 4.1.1    | Důkladná analýza a určení požadovaných funkcí.....                            | 35        |
| 4.1.2    | Návrh řešení.....   | 36        |
| 4.1.3    | Sestavení základní kostry webu, návrh rozložení, vyhotovení makety webu<br>37 |           |
| 4.1.4    | Instalace WordPressu.....   | 38        |
| 4.1.4.1  | Instalace pluginu Salon booking system .....                                  | 40        |
| 4.1.4.2  | Instalace šablony .....   | 41        |
| 4.1.5    | Databázová struktura.....   | 44        |
| 4.1.6    | Relační schéma .....  | 48        |
| 4.1.7    | Umístění na webový server.....  | 49        |
| 4.1.8    | Úprava šablony .....  | 50        |
| 4.1.9    | SMTP/SMS Komunikace .....   | 52        |
| <b>5</b> | <b>Výsledky a diskuze.....</b>  | <b>53</b> |
| <b>6</b> | <b>Závěr.....</b>   | <b>54</b> |

## Seznam obrázků

|  |    |
|--|----|
| Obrázek 1 Struktura (Zdroj: vlastní).....                      | 18 |
| Obrázek 2 červeně a tučně (Zdroj: vlastní).....                | 20 |
| Obrázek 3 16px široký a šedý (Zdroj: vlastní) .....            | 20 |
| Obrázek 4 Vytvoření tabulky (Zdroj: vlastní) .....             | 24 |
| Obrázek 5 Vložení dat (Zdroj: vlastní).....                    | 24 |
| Obrázek 6 Změna záznamu (Zdroj: vlastní) .....                 | 24 |
| Obrázek 7 Script tag (Zdroj: vlastní) .....                    | 27 |
| Obrázek 8 Vložení lokální (Zdroj: vlastní).....                | 27 |
| Obrázek 9 Vložení CDN (Zdroj: vlastní) .....                   | 28 |
| Obrázek 10 SlideDown () (Zdroj: vlastní) .....                 | 28 |
| Obrázek 11 Skrytí prvky (Zdroj: vlastní) .....                 | 28 |
| Obrázek 12 Animace (Zdroj: vlastní) .....                      | 28 |
| Obrázek 13 Vývojový prostředí (Zdroj: vlastní) .....           | 32 |
| Obrázek 14 Wireframe (Zdroj: vlastní) .....                    | 37 |
| Obrázek 15 Instalace WordPress (Zdroj: vlastní).....           | 38 |
| Obrázek 16 Administrační rozhraní (Zdroj: vlastní) .....       | 39 |
| Obrázek 17 Domů (Zdroj: vlastní).....                          | 39 |
| Obrázek 18 Salon booking systém (Zdroj: vlastní) .....         | 40 |
| Obrázek 19 Změna vzhledu (Zdroj: vlastní).....                 | 41 |
| Obrázek 20 Změna vzhledu šablona (Zdroj: vlastní).....         | 42 |
| Obrázek 21 Po instalaci šablony (Zdroj: vlastní).....          | 43 |
| Obrázek 22 Relační schéma WP (Zdroj: codex.wordpress.org)..... | 48 |
| Obrázek 23 Endora (Zdroj: vlastní) .....                       | 49 |
| Obrázek 24 SMTP (Zdroj: vlastní) .....                         | 52 |
| Obrázek 25 SMS (Zdroj: vlastní).....                           | 52 |

## Seznam tabulek

|                                       |    |
|---------------------------------------|----|
| Tabulka 1 SQL příkazy.....            | 24 |
| Tabulka 2 wp_commentmeta.....         | 44 |
| Tabulka 3 wp_comments .....           | 44 |
| Tabulka 4 wp_links.....               | 45 |
| Tabulka 5 wp_options.....             | 45 |
| Tabulka 6 wp_postmeta .....           | 45 |
| Tabulka 7 wp_posts .....              | 46 |
| Tabulka 8 wp_termmeta .....           | 46 |
| Tabulka 9 wp_terms.....               | 47 |
| Tabulka 10 wp_term_relationships..... | 47 |
| Tabulka 11 wp_term_taxonomy .....     | 47 |
| Tabulka 12 wp_usermeta .....          | 47 |
| Tabulka 13 wp_users .....             | 47 |

# 1 Úvod

V dnešní době je jedním z nejaktuálnějších témat tvorba webových stránek. Důvodem je skutečnost, že téměř každá oblast lidské činnosti, bez ohledu na to, zda se týká nabídky zboží nebo služeb, potřebuje propagaci, na jejímž základě vejde daná společnost ve známost.

Hlavním účelem webu je informovat. Na internetu existují různé způsoby informování, ale zpravidla všechny vedou jedním směrem – na web.

Webová stránka představuje soubor elektronických dokumentů spojených jedinou adresou. S vývojem informačních technologií se webové stránky stávají nejen sbírkou dokumentů, ale také prezentací řady technologií, které se používají jak k upoutání pozornosti, tak k přidání různých funkcí, které uživateli umožní pohodlné prohlížení webu.

Vytváření webových stránek je čím dál tím více aktuální, a proto byla vytvořena celá řada speciálních systémů, které umožňují nejen tvorbu webů v poměrně krátké době, ale také správu obsahu těchto stránek pomocí speciálního rozhraní. Systém pro správu obsahu je webový nástroj, který umožňuje publikovat a upravovat informace na webu bez zapojení vývojářů.

S ohledem na výše uvedené se v této bakalářské práci budeme věnovat návrhu a tvorbě webové stránky založené na jednom ze systémů pro správu obsahu, s názvem „WordPress“, a to konkrétně prostřednictvím příkladu tvorby webu pro společnost Sweet Beauty, jejíž činnost se zaměřuje na kosmetické ošetření a depilaci.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Cílem této bakalářské práce je vytvořit funkční webovou stránku založenou na CMS "WordPress" a ověřit její funkčnost. Za pomoci které se společnost dokáže nejen zviditelnit na trhu ale i zefektivnit si práci s rezervací služeb.

### **2.2 Metodika**

Metodika zpracování požadavku byla založena na základních principech tvorby webu, především jde o rozdělení na jednotlivé části jako je analýza, vývoj, testování, a spuštění v provozu. Velký podíl vývoje bude obnášet precizní nastudování materiálu z různých zdrojů a následně jejich správné použití.

Jako první krok byla provedena analýza požadavku zadavatele, na její základě byl zpracován návrh který byl prezentován zadavateli, po schválení od zadavatele bude následovat vývoj a následně testování. Webová stránka by měla soužit k zviditelnění firmy na trhu a pomoci ve správě rezervaci uživatelů.

Aplikace bude vytvořena při pomoci již výše zmiňovaného CMS "WordPress", a databáze MySql, šablona od společnosti kterou vybere zadavatel. Celý vývoj proběhne ve vývojovém prostředí JetBrains PhpStorm

## 3 Teoretická východiska

### 3.1 WordPress

Historie vzniku této platformy začíná vývojem cms b2 (také známý jako „cafelog“). Na jeho počátku byl mladý programátor Matt Mullenweg (Matt Mullenweg). Pomocí této platformy pro blogování sdílel informace o své cestě do Washingtonu. Následně (na konci roku 2002) byl vývoj projektu b2 zastaven. V lednu 2003 Matt a Michael Little oznámili, že se chystají pokračovat ve vývoji b2, resp. spíše v jedné z jeho verzí. WordPress 0.70 byl vydán v květnu 2003.

Stalo se tak, že v roce 2003 začala velká společnost CNET pro své projekty používat WordPress. Matt i nadále používal platformu pro psaní blogu – tentokrát psal o cestě do San Francisca. Tohoto článku si všiml Mike Tatum, jeden z viceprezidentů CNET, a vyzval Matta, aby se setkali. Tato schůzka skončila pracovní nabídkou od CNET.

Od října 2004 do prosince 2005 vznikala bbPress; tehdy se také zrodila myšlenka na založení společnosti Automattic, která bude kombinovat tyto open source projekty, které jsou v zásadě neziskové, přesto mají za cíl generování zisku.

V roce 2005 Matt Mullenweg odešel ze společnosti CNET, založil Automattic a věnoval se vývoji projektů s WordPressem. V té době byl obchodní model následující: vytvářet služby, uvolňovat je jednotlivcům zdarma a pro organizace za poplatek. A podle Matta to fungovalo skvěle. [1]

První byl spuštěn Akismet. Po Akismetu nastal čas pro blog na platformě wordpress.com. Blog na této platformě je zdarma, dokud nejsou překročeny limity nebo není vyžadováno vázání domény.

Podle příslibu vývojářů bude WordPress v této platformě optimalizován poměrem funkčnosti a zatížení hostingu, zatížení se sníží. Funkce WordPressu se stanou více přizpůsobitelnými.

Plánuje se, že se stanou součástí standardní funkčnosti pluginů. Tento systém nestojí na místě, velmi aktivně se na něm pracuje a tato vylepšení jsou bezproblémově přenášena do projektu, který na něm běží.

## 3.2 Varianty vývoje webových stránek:

- strukturní editory,
- wysiwyg editory,
- on-line editory,
- redakční systémy.

Velký vliv na výběr konkrétní varianty má také rozpočet a naše časové možnosti. A právě o tomto se budeme v této kapitole bavit: rozebereme si jednotlivé možné varianty vývoje, ukážeme si jejich silné a slabé stránky a také si rozebereme jednotlivé technologie potřebné pro vývoj.

Nejprve se tedy podíváme na jednotlivé varianty vývoje webových stránek, následně na technologie používané při jejich vývoji.

### 3.2.1 Strukturní editory

Strukturní editor-program vypadá jako obyčejný textový editor, který všichni známe; dovoluje volný pohyb na obrazovce nebo tento pohyb může být spojen s logickou strukturou seznamu. V obou případech se editor může automaticky postarat o shodné závorky a aby odsazení logicky odpovídalo syntaxi jazyka. Inteligentní strukturální editory mohou obsahovat některé další vhodné automatické akce.

Strukturní editory, stejně jako většina všech aplikací, bývají rozděleny podle způsobu distribuce, a to na kategorii zdarma a na kategorii placené.

Příklady strukturních editorů:

- PSPad – zdarma,
- HomeSite – zdarma,
- Notepad ++ – zdarma,
- The HTML Editor – placený,
- Sublime Text – placený.

Každý z uvedených editorů má své výhody i nevýhody, které jsou ale v současné době většinou zanedbatelné, takže je lepší je vyzkoušet a vybrat si ten editor, který nám vyhovuje nejvíce.

### 3.2.2 Wysiwyg editory

Zkratka „WYSIWYG“ znamená „What You See Is What You Get“ (to, co vidíte, je to, co získáte), tzn. že význam wysiwyg editoru spočívá v tom, že při úpravách textu vypadá tento stejně jako na stránce webu: stejné písmo, řádkování, stejná barva odkazu atd. Přidání nových prvků na stránku (odkazy, obrázky, seznamy, tabulky atd.) se provádí pomocí jednoduchého uživatelského rozhraní, takže každý, kdo nezná HTML, může stránku upravit. [2]

Každý wysiwyg editor je založen na použití tzv. „rámce“, jehož vlastnost designMode je nastavena na polohu „On“. Tím se stránka přepne do režimu úprav; když je třeba změny uložit, HTML kód přijme z tohoto rámce vlastnosti innerHTML a odešle je na server k uložení.

Příklady wysiwyg editoru:

- Froala Editor – placený,
- TinyMCE – placený,
- Amaya – zdarma.

Většina z wysiwyg editorů je placená a je jenom na vás a vašich finančních možnostech, který si vyberete.

### 3.2.3 On-line editory

On-line editor je specializovaná on-line služba, která umožňuje rychle a snadno vytvořit web bez znalosti HTML a programování. On-line editory jsou webové služby v krabici – obsahují v sobě vše potřebné pro spuštění webu, hosting, CMS, hotové šablony atd. Úprava stránek, návrh šablony a obecná konfigurace webu se provádějí on-line prostřednictvím ovládacího panelu. Existuje velké množství editorů tohoto typu pro vytváření webů. [2]

Příklady on-line editorů:

- uCoz – placený,
- uKit – placený,
- Wix – placený,
- Webasyst – placený,
- Diafan – placený,
- Ucraft – placený,
- SITE123 – placený.

Mají jak nedostatky, tak výhody, které si zde rozebereme. Výhody on-line editorů webových stránek jsou následující:

- cena,
- rychlost tvorby,
- rychlost práce,
- málo programování,
- předem vytvořená struktura webu.

Také ale mají i své nevýhody které jsou uvedeny níže:

- design,
- velikost stránky,
- nemožnost upravovat kód ručně,
- struktura a obsazenost,
- příloha k webu,
- měsíční nebo roční placení,
- slabá možnost rozšíření,
- problémy s integrací externích modulů,
- slabá propagace.

On-line editory jednoduše zvládají úkol vytváření stránek vizitek, portfolií a tzv. „landing-page“ webu. S plnohodnotnými blogy a fóry je to trochu složitější, proto bychom tuto variantu realizace nedoporučovali.

Nejdůležitější věcí je vybrat správný on-line editor – takový, jehož schopnosti odpovídají náročnosti úkolu. Stejně jako si nepůjdeme do lékárny pro zmrzlinu, i zde platí stejný princip. Přečtěte si recenze, srovnání, informace na oficiálních webových stránkách systémů, testujte a dělejte správná rozhodnutí.



### 3.2.4 Redakční systémy neboli tzv. „CMS“

CMS je systém pro správu obsahu, sada skriptů pro vytváření, úpravy a správu obsahu webu. Slangově se CMS nazývá též „engine“.

- Příklady redakčních systémů:
- WordPress,
- phpBB,
- vBulletin,
- Magento,
- OpenCart,
- osCommerce,
- InstantCMS,
- Social Engine,
- Monstra,
- Joomla,
- DLE,
- Drupal.

Dříve byla většina webů statická a vyžadovala ruční změny obsahu, ale nyní vývojová dynamika projektů vyžaduje připravenost rychle reagovat na změny a implementovat je s maximální účinností. Zároveň ne všichni uživatelé chtějí nebo si mohou dovolit kontaktovat vývojáře, zejména pokud web vyžaduje neustálé úpravy a zásahy. [3]

Systémy pro správu obsahu umožňují uživatelům, kteří nemají dovednosti s vývojem webu ani znalost programovacích jazyků, pracovat nezávisle na tvorbě a změně stránek.

#### 3.2.4.1 Princip fungování CMS

Podstatou práce CMS je schéma oddělení obsahu webu a jeho design. Uživatel má možnost zvolit si prázdnou šablonu, která určuje design stránky. Zbývá pouze doplnit informace. Většina systémů pro správu obsahu je založena na použití vizuálního wysiwyg editoru-programu, který umožňuje přidávat nebo měnit obsah webu pomocí intuitivního rozhraní. [3]

Stojí také za zmínku, že web již nepředstavuje soubor stránek jako takový, ale je vytvářen dynamicky. Přidaný obsah je uložen v databázi, např. MySQL, a používá se při generování stránky po přijetí odpovídající žádosti od klienta.

#### 3.2.4.2 Praktická aplikace

Většina CMS je však flexibilní a může být použita prakticky pro jakýkoliv typ webu. Např. nejoblíbenější a nejuniverzálnější možností je WordPress, v kterém je možno vytvořit téměř jakýkoli projekt.

CMS se zpravidla používají pro tyto typy webu:

- blog, fórum,
- internetový obchod,
- osobní stránky,
- firemní stránky,
- portály,
- sociální sítě.

#### 3.2.4.3 Výhody a nevýhody

Výhody CMS:

- jednoduché a pohodlné použití,
- široká funkčnost je k dispozici díky dodatkům, tématům a rozšířením,
- web lze vytvořit v krátkém časovém období,
- dostupnost dokumentace.

Nevýhody CMS:

- není vhodný pro atypické úkoly,
- populární CMS jsou zranitelné,
- je nutné sledovat aktualizaci CMS a kompatibilitu verzí doplňků,
- zvýšená spotřeba zdrojů, zejména při použití pluginů.

Existují také CMS vyvíjené na zakázku a určené pro konkrétní projekt. Výsledkem je, že jejich funkčnost není tak široká jako funkčnost hromadných systémů, ale maximálně odpovídá sadě úkolů pro daný projekt a neobsahuje další nástroje, které pro určený projekt nejsou potřebné. V zakázkových CMS jsou vytvářeny komplexní a rozsáhlé weby se specifickými funkcemi. [2]

### 3.3 Technologie používané při vývoji webu

#### 3.3.1 Apache

Apache je webový server s otevřeným zdrojovým kódem, který provozuje zhruba 40 procentwebů po celém světě. Oficiální název tohoto webového serveru je „Apache HTTP Server“. Spravuje ho a vyvíjí Apache Software Foundation.

Webový server umožňuje majitelům webových stránek zobrazení jejich webu v rámci internetu, o čemž svědčí i jeho označení „webový server“. Apache je jedním z nejstarších a nejspolehlivějších webových serverů. Jeho první verze byla vydána v roce 1995.

Když chce někdo navštívit web, zadá název domény do adresního řádku svého prohlížeče. Webový server poté doručí požadované soubory jako virtuální posel. Role webového serveru spočívá v poskytnutí webu v rozhraní internetu, takže funguje jako zprostředkovatel mezi serverovým počítačem a klientským počítačem. Vezme obsah ze serveru na každý požadavek uživatele a doručí jej do sítě.

Hlavním úkolem webového serveru je současně obsluhovat různorodý požadavek mnoha uživatelů. Webový server zpracovává soubory různých programovacích jazyků, jako je PHP, Python, Java a další. Promění je ve statický soubor HTML a odešle je ke službě do prohlížeče uživatele. Když uslyšíte spojení „webový server“, přemýšlejte o něm jako o nástroji komunikace mezi serverem a klientem. [4]

#### 3.3.2 HTML

HTML je značkovací jazyk. Používá se po celém světě. Webový prohlížeč interpretuje HTML kód tak, aby se zobrazoval ve všech zařízeních, např. v počítači, tabletu nebo telefonu.

Jazyk HTML byl vyvinut Britem Timem Bernersem-Leem ve společnosti CERN. HTML byl určen primárně pro výměnu vědeckých dokumentů. K rozložení dokumentu dochází za pomoci tzv. „deskriptorů“ (ale častěji se nazývají „tagy“, toto označení je nejvíce rozšířeno mezi programátory). [5]

Kdybychom tedy chtěli jednoduše odpovědět na otázku „Co je to HTML?“, odpověď by zněla následovně: Je to jednoduchý text obsahující velký počet tagů, který tvoří webovou stránku.

Historicky vzniklo hodně verzí tohoto značkovacího jazyka. V současné době nejnovější verzí je HTML 5.2. První betaverze se objevila v roce 2007. Pro zjednodušení byl

představen pojem „hypertext“. Hypertextový odkaz (nebo jen „odkaz“) je součástí hypertextu a odkazuje na jiný dokument HTML.

HTML byl vytvořen tak, aby se stránky zobrazovaly na všech zařízeních stejně; později byl přidán grafický design (CSS).

Struktura dokumentu HTML se sklada z deskriptorů (tagů). Každá webová stránka má začátek a konec; jsou označeny dvojicí `<html>` `</html>`. [2]

Příkladem (na obrázku) je jednoduchý dokument HTML:

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Název</title>
</head>
<body>
  <p>Hlavní část kódu </p>
</body>
</html>
```

**Obrázek 1 Struktura (Zdroj: vlastní)**

Jak můžeme vidět, HTML dokument začíná `<!DOCTYPE HTML>` (každá webová stránka by měla začínat právě tímto). Dříve existovalo velké množství variant značky DOCTYPE, ale podle nové verze HTML potřebujeme pouze tuto verzi !DOCTYPE, což je v příkladu.

Co je DOCTYPE? Je to tag, který sděluje prohlížeči verzi HTML. Tento tag by měl být zapsán na prvním řádku dokumentu HTML.

Pak přichází párový tag `<html>` `</html>`. Tento tag je kontejnerem webové stránky, tzn. že před a za ním by nemělo nic být.

V párovém tagu `<head>` `</head>` je zapsán povinný tag `<title>` `</title>`. V tomto tagu je zapsán název stránky, který bude zobrazen ve vyhledávači a na kartě prohlížeče. V kontejneru `<head>` `</head>` se obvykle nacházejí různé metatagy, styly, skripty a další. Pro nic jiného se nepoužívá.

Veškerý text, grafika a další kód musí být uvnitř tagu `<body>` `</body>`.

### 3.3.3 CSS

CSS neboli Cascading Style Sheets je faktický jazyk stylu, který definuje, jak bude zobrazen HTML dokument. CSS pracuje s písmy, se znaky, barvami pozadí, s poli, s řetězci, s výškou a šířkou zobrazovacích prvků, s obrázky na pozadí, s umístěním prvků apod. Pokud tedy je pro strukturování obsahu stránky potřebný HTML, tak pro formátování tohoto strukturovaného obsahu je potřeba CSS.

CSS byl vyvinut W3C (World Wide Web Consortium) v roce 1996 z poměrně prostého důvodu – v HTML nebyly vyvinuty žádné značky, s jejichž pomocí by bylo možné formátovat stránku. Všechno, co bylo potřeba, bylo napsat označení pro web.

Značky jako `<font>` byly zavedeny v HTML od verze 3.2, což vývojářům způsobovalo mnoho problémů. Jelikož webové stránky měly různá písma, barevné pozadí a styly, přepisování kódu bylo dlouhým a finančně náročným procesem. CSS byl tedy vytvořen W3C k vyřešení tohoto problému.

Vztah mezi HTML a CSS je vysoce důležitý. Protože HTML je značkovací jazyk (samotný základ webu) a CSS zdůrazňuje styl (celá estetika webu), jdou společně ruku v ruce.

Z technického hlediska není CSS nutný, ale pravděpodobně se nebudete chtít podívat na web, který obsahuje pouze HTML, protože bude vypadat jinak, než jsme zvyklí. [2]

#### 3.3.3.1 Jak funguje CSS

CSS používá prostou syntaxi v anglickém jazyce s určitou sadou pravidel, kterými je řízen. Struktura syntaxe je celkem jednoduchá. Obsahuje CSS selektory – používají se k výběru obsahu, který chcete stylovat. Selektory jsou součástí sady pravidel CSS. Blok oznámení obsahuje jednu nebo více deklarací oddělených středníky. Každá deklarace obsahuje název vlastnosti CSS a hodnotu oddělenou dvojtečkou. Deklarace CSS vždy končí středníkem a bloky deklarace jsou uzavřeny v závorkách. [2]

### 3.3.3.2 Příklady:

Prvky `<p>` budou zvýrazněny červeně a tučně.



```
HTML
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>Název</title>
5 </head>
6 <body>
7 <p>Příklad obsahu.</p>
8 </body>
9 </html>

CSS
1 p {
2   color: red;
3   text-weight: bold;
4 }
```

Příklad obsahu.

Obrázek 2 červeně a tučně (Zdroj: vlastní)

Prvky `<p>` vycentrovány, 16px široké a šedé:



```
HTML
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>Název</title>
5 </head>
6 <body>
7 <p>Příklad obsahu.</p>
8 </body>
9 </html>

CSS
1 p {
2   text-align: center;
3   font-size: 16px;
4   color: grey;
5 }
```

Příklad obsahu.

Obrázek 3 16px široký a šedý (Zdroj: vlastní)

### 3.3.3.3 Varianty zápisu

Existuje několik variant zápisu: interní, externí a tzv. „inline“. Nejdříve se budeme věnovat internímu způsobu zápisu. Styly zapsané tímto způsobem se načtou při každé aktualizaci webu, což může způsobit delší dobu načítání webové stránky; kromě toho, že se stránky mohou načítat pomaleji, nebudeme moci použít již definovaný styl na jiné stránce webu. To však má i své výhody. Díky tomu, že styl je definován na jedné stránce, je snadné sdílet šablonu náhledu.

Externí metoda zápisu může být nejvíce pohodlná pro práci s CSS; všechny styly se zapisují do externího souboru s příponou „.css“. Tzn. že tyto styly můžeme použít na jakékoliv stránce, kde je to potřeba. Zápis do externího souboru může také zlepšit dobu načítání.

Jako poslední styl zápisu zůstal tzv. „inline“ neboli integrovaný styl CSS. Je spojený se specifickými prvky, které obsahují tag `<style>`. Každý element musí být stylizovaný zvlášť, takže tato varianta nemusí být nejlepším ani nejrychlejším způsobem zpracování.

### 3.3.4 PHP

PHP (Hypertext Preprocessor) - je skriptovací jazyk speciálně navržený pro psaní webových aplikací (skripty, scénáře), které jsou spouštěny na webovém serveru. Vytvořil ho Rasmus Lerdorf v roce 1995. PHP byla původně zkratka pro „Personal Home Page (Tools)“ (což bychom mohli přeložit jako „Nástroje pro osobní domovské stránky“), ale nyní je spíše rekurzivní zkratkou pro „PHP: Hypertext Preprocessor“ (v překladu „Hypertextový procesor PHP“). [6]

Syntaxe jazyka je do značné míry založena na syntaxi C, Java a Perl. Je velmi podobný jazyku C a Perlu, takže pro profesionálního programátora není těžké tento jazyk se naučit. Na druhou stranu PHP je jednodušší než C a zvládne se ho naučit i člověk, který žádný programovací jazyk neumí.

Obrovské plus PHP, na rozdíl od např. JavaScriptu, je, že PHP skripty jsou prováděny na straně serveru. PHP nezávisí na rychlosti počítače uživatele ani jeho prohlížeče; úplně vše je prováděno na serveru. Uživatel nemusí ani vědět, zda dostává normální soubor HTML nebo výsledek skriptu.

Scénáře v jazyce PHP mohou být interpretovány na serveru jako separátní soubory anebo mohou být integrovány přímo do HTML stránky. [6]

PHP dokáže generovat nejenom HTML dokumenty, ale také obrázky různých formátů – JPEG, GIF, PNG, PDF a FLASH. PHP dokáže generovat data v jakémkoli textovém formátu, včetně XHTML a XML.

Také se jedná o mezipatformovou technologii, distributiv PHP je dostupný pro většinu operačních systémů. PHP je podporován na většině webových serverů, jako je Apache, Microsoft Internet Information Server (IIS), Microsoft Personal Web Server a další.

Jazyk PHP je neustále zdokonalován a je mu zajisté zajištěna dlouhá podpora v oblasti webových technologií.

### 3.3.5 MySQL

Úloha dlouhodobého ukládání a zpracování informací se objevila téměř okamžitě s příchodem prvních počítačů. K vyřešení tohoto problému byly na konci 60. let vyvinuty specializované programy nazvané „systémy správy databází“ (DBMS). DBMS prošly dlouhým vývojem. Na konci 80. let se stal dominantním systémem správy relačních databází (RDBMS). Od té doby se DBMS staly de facto standardem a za účelem sjednocení práce s nimi byl vyvinut strukturovaný dotazovací jazyk (SQL), což je jazyk pro správu relačních databází.

MySQL je jedním z nejpobulárnějších a nejrozšířenějších DBMS v kontextu internetu. Není určen k práci s velkým objemem informací, jeho použití je ideální pro malé a střední internetové stránky.

MySQL se vyznačuje dobrou rychlostí, spolehlivostí a flexibilitou. Práce s ním zpravidla nezpůsobuje velké potíže. Podpora serveru MySQL je automatickou součástí balíčku PHP.

Důležitým faktorem je jeho distribuce, která odpovídá obecné licenci GNU (GPL, GNU Public License).

Dříve jsme pro dlouhodobé ukládání informací pracovali se soubory: do nich jsme vložili množství řádků a poté jsme je extrahovali pro další práci. Úkol dlouhodobého ukládání informací je při programování webových aplikací velmi běžný: počítání návštěvníků v přepážce, ukládání zpráv ve fóru, dálková správa obsahu informací na webu atd. [7]

Mezitím profesionální metody práce se soubory jsou časově velmi náročné: musíte se starat o umístování informací do nich, o jejich třídění a extrahování a neměli byste zapomenout, že všechny tyto akce se objeví na serveru hostitelského poskytovatele, kde je velmi pravděpodobné, že je instalována jedna z variant Unixu – proto se musíme také starat o oprávnění souborů a jejich správné umístění. Zároveň se výrazně zvyšuje množství kódu a je velmi snadné udělat v kódu chybu.

Všechny tyto problémy se ale dají vyřešit pomocí databáze. Samotné databáze se starají o zabezpečení informací a jejich třídění; umožňují načíst a zveřejnit informace na jednom řádku. Kód používající databázi je kompaktnější a ladění je mnohem snazší. Kromě toho je rychlejší – informace se načítají z databáze mnohem rychleji než ze souborů.

Aplikace psaná v jazyce PHP, která používá databázi k ukládání informací (zejména MySQL), vždy pracuje rychleji než aplikace, která k ukládání dat používá soubory. Faktem



je, že databáze MYSQL je psána v C++, a už toto napovídá, že práce s databází bude rychlejší. [7]

Také aplikace v PHP, která by pracovala s pevným diskem efektivněji než s databází, je skoro nerealizovatelným úkolem, protože programy PHP ve své podstatě pracují pomaleji než programy C++, protože PHP je interpretovaný a C++ je kompilovaný.

Hlavní výhodou databáze je tedy to, že se stará o veškerou práci s pevným diskem a dělá ji velmi efektivně.

Existují následující typy databází:

- hierarchická,
- relační,
- objektově orientovaná,
- hybridní.

Hierarchická databáze je založena na stromové struktuře ukládání informací. V tomto smyslu jsou hierarchické databáze velmi podobné souborovému systému počítače.

V relačních databázích jsou data shromažďována v tabulkách, které se skládají ze sloupců a řádků, na jejichž průsečíku jsou umístěny buňky. Dotazy do těchto databází jsou vráceny tabulkou, která může být znovu účastníkem dalšího dotazu. Data v tabulkách jsou zpravidla spojena s daty jiných tabulek (odtud název „relační“). [7]

V objektově orientovaných databázích jsou data ukládána jako objekty. Je vhodné pracovat s objektově orientovanými databázemi pomocí objektově orientovaného programování. Dosud však takové databáze nedosáhly popularity relačních databází, protože doposud jsou z hlediska výkonu výrazně podřadné.

Hybridní DBMS kombinují schopnosti relačních a objektově orientovaných databází. Webové aplikace obvykle používají relační databáze.

### 3.3.5.1 SQL

Pro práci s databází se používá dotazovací jazyk SQL. Jazyk SQL je strukturovaný dotazovací jazyk; umožňuje provádět různé operace s databázemi, např. vytvářet tabulky, aktualizovat a mazat z nich data, provádět dotazy apod. Níže v tabulce jsou uvedeny základní SQL příkazy:

|                  |                             |
|------------------|-----------------------------|
| SHOW DATABASES;  | Zobrazit dostupné databáze. |
| CREATE DATABASE; | Vytvoření nové databáze.    |

|                              |   |
|------------------------------|---|
| USE <nazev_db>;              | Výběr databáze k použití.                     |
| SOURCE <slozka_soubor_.sql>; | Import příkazu SQL ze souboru .sql.           |
| DROP DATABASE <nazev_db>;    | Odstranění databáze.                          |
| SHOW TABLES;                 | Zobrazit tabulky dostupné v databázi.         |
| CREATE TABLE;                | Vytvoření nové tabulky.                       |
| DESCRIBE <nazev_db>;         | Podrobnosti tabulky.                          |
| INSERT;                      | Přidávání dat do tabulky.                     |
| UPDATE;                      | Aktualizace dat tabulky.                      |
| DELETE FROM <nazev_tabulky>; | Vymazat všechna data z tabulky.               |
| DROP TABLE <nazev_tabulky>;  | Smazat tabulku.                               |
| SELECT;                      | Načtení instance z tabulky.                   |
| SELECT DISTINCT;             | Pouze neopakující se data.                    |
| WHERE;                       | Uvedení podmínek příkazu.                     |
| GROUP BY;                    | Seskupení výstupních hodnot.                  |
| HAVING;                      | Pro práci s agregovanými funkcemi (WHERE).    |
| ORDER BY;                    | Třídění výsledků dotazu.                      |
| BETWEEN;                     | Výběr datových hodnot z určitého intervalu.   |
| LIKE;                        | Vyhledávací vzorec podobné hodnoty.           |
| IN;                          | Pro klauzuli WHERE zadat několik hodnot.      |
| JOIN;                        | Propojení tabulek pomocí společných atributů. |

Tabulka 1 SQL příkazy

### 3.3.5.2 Ukázka příkladů:

`CREATE TABLE ucitel ( ID CHAR(5), JMENO VARCHAR (20) NOT NULL , PRIJMENI VARCHAR (20) NOT NULL, PLAT NUMERIC(8,2), PRIMARY KEY (ID) );`

| #                          | Název    | Typ          | Porovnávání        | Vlastnosti | Nulový | Výchozí | Komentáře | Další | Operace               |
|----------------------------|----------|--------------|--------------------|------------|--------|---------|-----------|-------|-----------------------|
| <input type="checkbox"/> 1 | ID       | char(5)      | utf8mb4_unicode_ci |            | Ne     | Žádná   |           |       | Změnit Odstranit Více |
| <input type="checkbox"/> 2 | JMENO    | varchar(20)  | utf8mb4_unicode_ci |            | Ne     | Žádná   |           |       | Změnit Odstranit Více |
| <input type="checkbox"/> 3 | PRIJMENI | varchar(20)  | utf8mb4_unicode_ci |            | Ne     | Žádná   |           |       | Změnit Odstranit Více |
| <input type="checkbox"/> 4 | PLAT     | decimal(8,2) |                    |            | Ano    | NULL    |           |       | Změnit Odstranit Více |

Obrázek 4 Vytvoření tabulky (Zdroj: vlastní)

`INSERT INTO ucitel VALUES (1,"Jan","Janovič", 12302);`

| ID | JMENO | PRIJMENI | PLAT     |
|----|-------|----------|----------|
| 1  | Jan   | Janovič  | 12302.00 |

Obrázek 5 Vložení dat (Zdroj: vlastní)

`UPDATE ucitel SET PLAT = 30333 WHERE ID=1;`

| ID | JMENO | PRIJMENI | PLAT     |
|----|-------|----------|----------|
| 1  | Jan   | Janovič  | 30333.00 |

Obrázek 6 Změna záznamu (Zdroj: vlastní)

### 3.3.6 JavaScript

JavaScript je jedním z nejpopulárnějších programovacích jazyků na světě, s více než 20letou historií. JavaScriptu se lze velmi rychle naučit rozumět a snadno ho použít v celé

řadě aplikací, od jednoduchých vylepšení funkčnosti webu až po psaní her a webových aplikací.

JavaScript vytvořil za 10 dnů Brendan Eich, který pracoval v Netscape v roce 1995. Původně se jmenoval Mocha, dále byl přejmenován na Mona a poté na LiveScript, až se nakonec stal JavaScriptem. Počáteční verze jazyka byla omezena pouze na prohlížeč Netscape a nabízela úzkou funkčnost. Postupem času se rozvíjel, částečně díky komunitě vývojářů, kteří na něm dále pracovali.

V roce 1996 byl JavaScript standardizován, s oficiálním názvem „ECMAScript“, přičemž ECMAScript 2 byl vydán v roce 1998 a ECMAScript 3 v roce 1999. Pak se změnil na dnešní JavaScript, který nyní funguje nejen v různých prohlížečích, ale také na různých zařízeních, včetně mobilních a stolních počítačů. [8]

Používání JavaScriptu od té doby nadále roste a v roce 2019 jej používalo 95 % webů. Za 20 let se vyvinul z primitivního programovacího jazyka do jednoho z nejpoužívanějších nástrojů v arzenálu webového vývojáře.

#### 3.3.6.1 Výhody JavaScriptu

JavaScript má velkou řadu výhod, díky kterým je nejlepší volbou mezi podobnými jazyky, zejména v některých aplikacích.

Uvedme několik hlavních výhod používání JavaScriptu:

- Není potřeba používat kompilátor, protože jej webový prohlížeč interpretuje.
- Je snazší se ho učit než jiné programovací jazyky.
- Může být připojen ke zvláštním prvkům stránek nebo událostí, jako je kliknutí (click) nebo přejetí myši (mouseover).
- Pracuje v různých prohlížečích a na různých platformách.
- JavaScript může být použit k ověření vstupu a snížit potřebu ručního ověření dat a tím také snižuje počet dotazů pro ověření v backendu.
- Díky JavaScriptu je web interaktivnější a atraktivnější pro návštěvníky.
- Je asynchronní.
- Snadnější než jiné programovací jazyky.

#### 3.3.6.2 Nevýhody

Na druhou stranu má ale i své slabiny. Každý programovací jazyk má své nevýhody a slabiny. Jednou z příčin problémů je popularita jazyka. Když se programovací jazyk stane

populárním jako JavaScript, stane se předmětem zvýšeného zájmu hackerů, podvodníků a jiných škodlivých projevů třetích stran, které se snaží najít slabiny a chyby v zabezpečení.

Některé nevýhody a slabiny:

- Zranitelný pro zneužití (škodlivý kód, který využívá zranitelnosti v softwarovém produktu).
- Může být použit ke spuštění škodlivého kódu v počítači uživatele.
- Některé prohlížeče nebo zařízení nejsou plně podporovány.
- Některé části kódu JavaScriptu mohou být velmi velké.
- Může být interpretován odlišně na různých zařízeních, což vede k nedostatečné integritě.

Důvodem, proč je JavaScript jedním z nejpobulárnějších programovacích jazyků, je jeho univerzálnost. Mimochodem, mnoho vývojářů používá JavaScript jako hlavní programovací jazyk a jiné programovací jazyky používá, pouze pokud potřebují něco specifického.

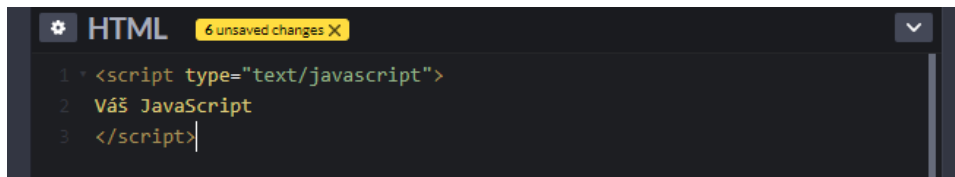
### 3.3.6.3 Jak JavaScript funguje na vašem webu

Kód JavaScript je obvykle buď vložen do vaší stránky, nebo odkazuje na samostatný soubor s příponou “.js”. Jedná se o jazyk, který je interpretován na straně klienta, což znamená, že skript se stáhne do počítače návštěvníka a tam se zpracuje – to je hlavní rozdíl v tom, jak fungují jazyky na straně serveru (jsou spuštěny na vašem serveru před odesláním souboru návštěvníkům).

Je ale dobré si uvědomit, že většina prohlížečů také nabízí uživatelům možnost vypnout JavaScript. Proto je vždy dobré zvážít, co se stane, pokud není kód interpretován v některých počítačích.

## Přidání JavaScriptu na web

Řádek kódu JavaScript můžete na web přidat pomocí značky `<script>` (viz následující příklad).

A screenshot of a code editor window titled 'HTML' with a yellow notification '6 unsaved changes X'. The editor shows three lines of code: 1. `<script type="text/javascript">`, 2. `Váš JavaScript`, and 3. `</script>`. The cursor is at the end of the third line.

**Obrázek 7 Script tag (Zdroj: vlastní)**

Obecným pravidlem je vložení kódu JavaScript do značek `<header>`, pokud jej nebudete muset spustit v určitém čase nebo na určitých prvcích stránky. Kód JavaScript můžete také uložit do samostatného souboru a v případě potřeby jej na webu vyvolat.

A konečně důvod, proč je JavaScript tak populární, je ten, že je velmi dobrý v tom, k čemu je určen.

### 3.3.7 JQuery

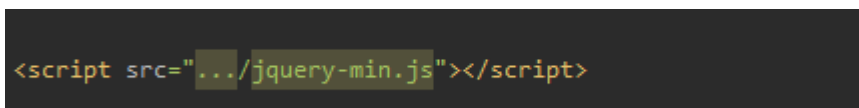
JQuery je velmi populární knihovna JavaScriptu. Byla vytvořena Johnem Rezigem v roce 2006 s cílem usnadnit vývojářům používání JavaScriptu na webových stránkách. Není to samostatný programovací jazyk a funguje ve spojení s JavaScriptem. S jQuery budete dělat mnohem víc a budete do toho investovat mnohem méně času. [9]

Psaní kódu může být zdlouhavé, zejména pokud obsahuje mnoho řádků. JQuery komprimuje více řádků kódu do jediné funkce, takže k dokončení jedné úlohy není třeba přepisovat celé bloky kódu.

#### 3.3.7.1 Způsoby připojení jQuery

Aby bylo možné používat funkce knihovny jQuery, musí být připojena k webové stránce jedním z následujících způsobů:

1. Stáhnutím z oficiálního webu a jejím připojením pomocí tagu `<skript>`:

A screenshot of a code editor showing a single line of code: `<script src="../../../jquery-min.js"></script>`. The text is highlighted in yellow.

**Obrázek 8 Vložení lokální (Zdroj: vlastní)**

2. Obvykle je umístěn na konci dokumentu HTML, před zavírající značku `<body>`, ale před jinými knihovnami, které jej používají.

3. Použitím Google CDN nebo jiných serverů CDN což má své výhody při návštěvě webů, kde odkaz URL ke stažení knihovny jQuery je stejný, se knihovna jQuery nenačte ze serveru, ale z mezipaměti prohlížeče uživatele.

```
<!-- Verze 1.11.3 -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<!-- Verze 2.1.4 -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
```

**Obrázek 9 Vložení CDN (Zdroj: vlastní)**

Tento řádek je obvykle umístěn na konci dokumentu HTML, před zavírající značkou <body>, ale před jinými knihovnami, které jej používají.

### 3.3.7.2 Příklady jQuery:

- Efekty posouvání můžeme vytvořit pomocí několika řádků kódu. Použitím příkazu SlideDown (), SlideUp () a SlideToggle ().

```
$("#flip").click(function(){
    $("#panel").slideDown();
});
```

**Obrázek 10 SlideDown () (Zdroj: vlastní)**

- Pomocí jQuery můžeme také skrýt prvky HTML pomocí příkazů Hide () a Show ().

```
$("#hide").click(function(){
    $("#p").hide();
});
$("#show").click(function(){
    $("#p").show();
});
```

**Obrázek 11 Skrýti prvky (Zdroj: vlastní)**

- Také můžeme animovat prvky.

```
$("#button").click(function(){
    $("#div").animate({
        left: '250px',
        height: '+=150px',
        width: '+=150px'
    });
});
```

**Obrázek 12 Animace (Zdroj: vlastní)**

### 3.3.8 Bootstrap

Bootstrap je open source a bezplatný HTML, CSS a JS, framework, který je používán webovými vývojáři k rychlému vytváření responzivních návrhů webových stránek a webových aplikací. Je používán po celém světě nejen nezávislými vývojáři, ale někdy i celými společnostmi. [10]

Hlavní oblastí jeho aplikace je vývoj frontendových částí webů a administračních rozhraní. Mezi podobnými frameworky (Foundation, UIKit, Semantic UI, InK atd.) je rámec Bootstrap nejoblíbenější.

Důvodem jeho popularity je skutečnost, že umožňuje vyvíjet weby několikrát rychleji, než je tomu u „čistých“ CSS a JavaScript webů. A v našem světě je čas nejdražší komoditou. S popularitou je spojena také dostupnost. Ta spočívá v tom, že i začínající vývojář může s jeho pomocí vytvořit poměrně kvalitní rozvržení, které by bylo obtížné dokončit bez důkladné znalosti webových technologií a bez dostatečné praxe.

Framework Bootstrap je kolekce souborů CSS a JavaScript. Pokud jej vybereme k použití na stránce, je nutné k němu připojit tyto soubory. Po připojení se nám nástroje tohoto frameworku stanou dostupnými.

Bootstrap se skládá z:

- sítí (Bootstrap grid);
- tříd pro stylizaci textu, obrázků, tabulek a jiného obsahu;
- součástí určených k vytváření:
  - tlačítek,
  - různých formulářů,
  - horizontálních a vertikálních navigačních nabídek,
  - posuvníků,
  - rozevíracích seznamů,
  - akordeonů, modálních oken,
  - popisků a dalších prvků rozhraní;

Tříd pro řešení pomocných problémů, které se nejčastěji vyskytují před webovými vývojáři (zarovnání textu, skrytí nebo zobrazení prvku, nastavení barvy a pozadí prvku, nastavení okraje a odsazení výplně atd.).

#### 3.3.8.1 Výhody a nevýhody

Výhody frameworku při vývoji frontendové části webů a administračního rozhraní na jeho základě: vysoká rychlost vytváření webových stránek, přítomnost velkého počtu hotových dobře navržených komponentů, schopnost přizpůsobit se vašemu projektu, nízký vstupní práh; pro práci s frameworkem není nutné mít podrobné znalosti HTML, CSS, JavaScriptu ani jQuery (stačí znát pouze základy výše uvedených technologií), přítomnost obrovské komunity, velké množství článků, receptů a videí.

Bootstrap, stejně jako většina takových rámců, má i nedostatky. Jedním z nich je, že nemá smysl používat jej pro rozložení webových stránek. Když bude vyžadováno přepsání velké části kódu CSS a jednoduše např. nastavení proměnných Bootstrap, nejde to udělat.

Další nevýhodou je, že kód CSS a JavaScript vašeho projektu bude mnohem větší, než kdybychom ho napsali sami pro tento projekt. A v neposlední řadě – i tento framework, jako všechny ostatní, je třeba aktualizovat.

### 3.3.9 Vývojové prostředí JetBrains PhpStorm

PhpStorm je integrované vývojové prostředí PHP s inteligentním editorem, který hluboce rozumí kódu, podporuje PHP od verze 5.3 pro moderní a klasické projekty, poskytuje špičkové dokončování kódu, refactoring, on-the-fly prevenci chyb a podporuje míchání různých jazyků v jednom projektu.

#### 3.3.9.1 Podpora hlavních frameworků

PhpStorm je ideální pro práci se Symfony, Laravelem, Drupalem, WordPressem, Zend Frameworkem, Magentem, Joomla, CakePHP, Yii a dalšími frameworky.

Vývojové prostředí JetBrains PhpStorm je specializovaný nástroj pro vývoj webových aplikací, které jsou vytvářeny pomocí jazyka PHP a pomocí HTML, JavaScriptu a CSS. Řešení PhpStorm implementuje nasazení a synchronizaci projektů přes FTP. Prostředí PhpStorm nabízí funkce automatického dokončení jazykových konstrukcí PHP v kódu, inspekcí kódu, různé algoritmy refaktoringu a rychlou navigaci v kódu.



Stovky inspekci se postarají o ověření správnosti kódu a také o analýzu celého projektu během vývoje. Podpora pro PHPDoc, (re)arranger kód, formátovací program s konfigurací stylu kódu a další funkce, které pomáhají vývojářům psát elegantní a snadno udržovatelný kód.

Podporovány jsou pokročilé technologie pro vývoj webových aplikací, včetně HTML5, CSS, Sass, SCSS, LESS, Stylusu, Kompasu, CoffeeScriptu, TypeScriptu, ECMAScriptu Harmony, Jade šablon, Zen Codingu, Emmetu a samozřejmě JavaScriptu.

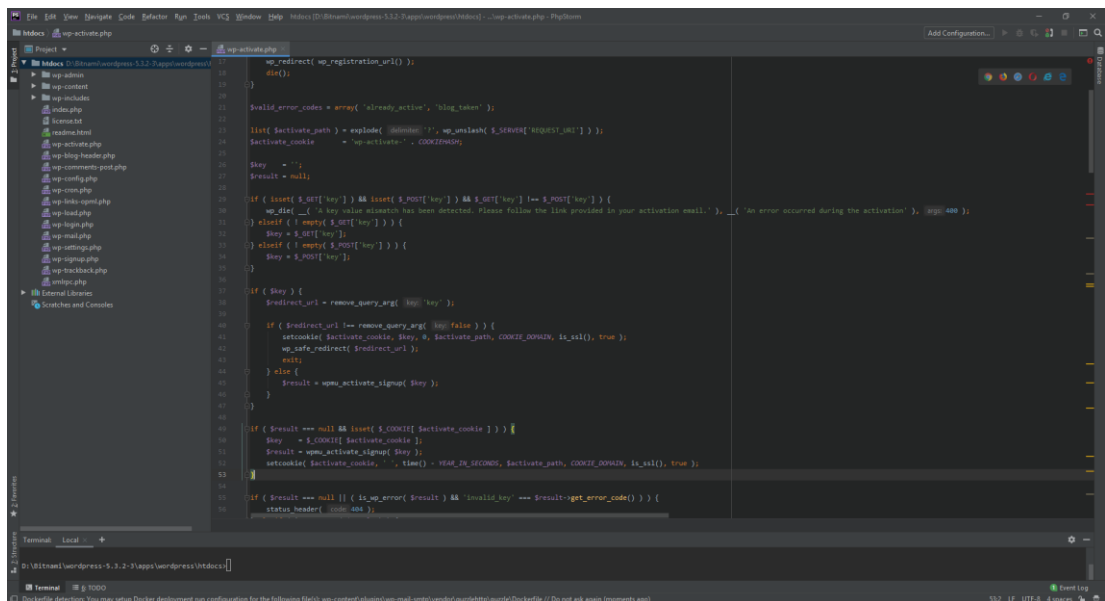
PhpStorm zahrnuje všechny funkce WebStorm (editor HTML/CSS, editor JavaScriptu) a přidává plnohodnotnou podporu pro PHP a databázi/SQL.

Klíčové vlastnosti:

- inteligentní editor kódu PHP se zvýrazňováním syntaxe, dokončováním kódu;
- pokročilými možnostmi formátování kódu a předcházením chybám za běhu;
- podpora Vagrantu, Composer, vestavěný REST klient, nástroje příkazového řádku, konzole SSH;
- podpora frameworků;
- vizuální debugger pro PHP aplikace;
- kompletní sada nástrojů pro front-end vývoj;
- podpora stylů kódu, vestavěných stylů PSR1/PSR2, Symfony2, Zendu, Drupalu a dalších;
- integrace se systémy pro správu verzí, včetně jednotného rozhraní;
- vzdálené nasazení aplikací a automatická synchronizace pomocí FTP, SFTP, FTPS atd;
- živé úpravy: změny kódu lze okamžitě prohlížet v prohlížeči bez opětovného načtení stránky;
- PHP UML;
- integrace s debug nástroji;
- databázové nástroje, editor SQL;
- meziplatformní (Windows, Mac OS X, Linux).

PhpStorm je placené vývojové prostředí, ale před zakoupením je možno si stáhnout zkušební verzi na 30 dnů, která není nijak omezena ve své funkčnosti.

Vizuální stránka tohoto vývojového prostředí je taky velmi atraktivní a propracovaná, ukázka je na obrázku níže.



Obrázek 13 Vývojové prostředí (Zdroj: vlastní)

### 3.3.10 Xdebug

Xdebug je rozšíření pro PHP, které pomáhá s laděním a vývojem. Jeho Klíčové vlastnosti jsou:

- obsahuje tzv. „jednokrokové debugování“ pro použití s IDE;
- vylepšuje funkci `var_dump()` v PHP;
- přidává stopy zásobníku pro oznámení, varování, chyby a výjimky;
- vybaven funkcemi pro nahrávání každého volání funkcí a nahrávání proměnných na disk;
- obsahuje profiler;
- poskytuje funkce pokrytí kódu pro použití s PHPUnit. [11]

Samozřejmě můžeme ladit PHP kód pomocí jednoduchých funkcí, jako je `error_log`, `print_r` a `var_dump`, ale tyto funkce neumožňují sledovat proces zpracování kódu. A k tomu používají vývojáři PHP Xdebug.

Při ladění pomocí `error_log` jsou požadované proměnné ne vždy zobrazeny. Např. v procesu odstraňování chyb přidáte funkci pro ladění proměnné, aktualizaci stránky, kontrolu protokolu chyb. Pokud není možné zjistit, o jakou chybu jde, musíte tento proces opakovat znovu a znovu.

Xdebug umožňuje zastavit proces během provádění a kontrolovat proměnné. Tzn. že během jedné iterace této smyčky máte vše, co potřebujete k řešení problému. Tato funkce pomáhá zefektivnit proces vývoje.

#### 3.3.10.1 Odstraňování potíží s pomocí Xdebugu

Body přerušení neboli tzv. „breakpoints“ v Xdebugu umožňují kdykoli zastavit provádění kódu a dále použít krokování. Tzn. že můžeme přesouvat body přerušení, dokud nenajdeme oblast, odkud problém vychází. Také v neposlední řadě můžeme použít podmínovaný „breakpoint“.

#### 3.3.10.2 Vývoj prostřednictvím testování

Další oblastí aplikace pro Xdebug je Debug Driven Development, který umožňuje rychle řešit problémy. Například kontrolu existujícího pole, objektu a instance třídy a najde kousky kódu, které lze použít vícekrát.

Není nutné jej používat často, ale skutečnost, že taková funkce existuje, je velice příjemná

## 4 Vlastní práce

Praktická část této bakalářské práce je zaměřena na vývoj a testování vyvinuté aplikace a následné umístění produkční verze na webový server.

### 4.1 Úvodní analýza

V rámci této části analýzy jsme byli seznámeni se současnou situací podniku, kterou je potřeba změnit dle požadavku zadavatele.

Tento podnik aktuálně nedisponuje žádnou webovou stránkou ani rezervačním systémem. Všechny požadavky na rezervaci jsou přijímány pomocí telefonu nebo e-mailu a následně pro přehlednost zapsány do tabulky v MS Excelu.

Tato varianta vyhovovala jen zpočátku, ale se zvyšujícím se počtem klientů se stala nevhodnou kvůli časové náročnosti. K nutnosti změny přispěl též zvyšující se počet zaměstnanců, jimž bylo potřeba distribuovat informace ohledně rezervací.

Na rozdíl od manuální varianty, kdy provozní personál musel informace ohledně klienta vytěžit ze schránky nebo telefonního rozhovoru, je varianta webu s rezervačním systémem rychlejší, přehlednější a méně pracná.

V neposlední řadě se jednou z potřeb stalo zviditelnění podniku na internetu, protože v dnešní době nebýt on-line znamená nebýt vůbec.

Z těchto důvodů se zadavatel rozhodl pro vytvoření webové stránky, která by umožňovala sdílení informací, zefektivnila práci s rezervacemi a přispěla k celkovému přehledu napříč celým podnikem. Kromě těchto požadavků vznikl i jeden požadavek, který byl orientovaný na klienty podniku, a to zasílání rekapitulace a připomínky rezervace prostřednictvím SMS.

Cílem této bakalářské práce je vytvořit funkční webovou stránku s pluginem rezervačního systému, který bude evidovat všechny rezervace zákazníků na určité služby podniku, informovat personál o nových rezervacích a odesílat SMS s rekapitulací rezervace zákazníkům.

#### 4.1.1 Důkladná analýza a určení požadovaných funkcí

V první řadě je třeba se sejít se zadavatelem a ujasnit si v diskuzi jeho přání a požadavky. Je možné, že zadavatel následně přehodnotí některou z požadovaných funkcionalit a změní tak část požadavků vzešlých při úvodní analýze.

Po naší schůzce se požadavky zadavatele nezměnily, ale naopak se rozšířily. Jsou následující:

- web bude běžet v CMS WordPress;
- web s možností registrace uživatelů;
- možnost zveřejňování článků;
- web neboli její „landing page“ bude rozdělený na sekce;
- web má obsahovat animovaný číselník spokojených zákazníků;
- možnost přidávání komentářů ke zveřejněným článkům;
- web má umožnit rezervaci služeb;
- při rezervaci si zákazník může vybrat, kdo bude ošetření provádět;
- systém bude ověřovat dostupnost služby a zobrazí kalendář;
- zákazník bude mít možnost vybrat určitý den v kalendáři a určitý čas;
- registrace dostupná jenom v případě rezervace služby;
- zákazníkovi a ošetřovateli bude odeslána rekapitulace v podobě e-mailu a SMS;
- každý ošetřovatel bude moci prohlédnout svůj kalendář;
- export měsíčního reportu do MS Excelu;
- firemní identita a šablona stránky budou vytvořeny externí agenturou.

#### 4.1.2 Návrh řešení

Na základě analýzy výše uvedených požadavků od zadavatele bylo rozhodnuto, které technologie budou použity pro tvorbu webu a jaký plugin rezervace bude nejvhodnější pro splnění požadavku zadavatele.

Jelikož zadavatel přesně specifikoval, jaký CMS má být použit, postupujeme dle požadavku, a tedy vybereme WordPress, aby byl zadavatel maximálně spokojený.

Má to ale i své složitosti, jelikož zadavatel chce, aby web obsahoval i rezervační systém, proto se musíme obrátit na adresář pluginů pro WordPress. Po dlouhé úvaze a analýze pluginů jsme našli pouze jeden, který absolutně odpovídal našim požadavkům, a tedy bylo rozhodnuto, že využijeme Salon booking system, jelikož má nejlepší hodnocení, velkou komunitu i důkladnou dokumentaci a je často aktualizován, tzn. že oprava chyb a převod na novou verzi jsou zajištěny; také má bezplatnou podporu.

Jelikož se pohybujeme ve webových technologiích a CMS, s nímž budeme pracovat, je psán v PHP, zvolíme jako hlavní programovací jazyk PHP.

Web budeme tvořit ve vývojovém prostředí PHPStorm a jako pomocník pro testování bude využit Xdebug.

Pro celý projekt budou využity tři typy prostředí:

- DEV – vývojové prostředí, ve kterém bude proveden veškerý vývoj a úprava webové stránky;
- TST – testovací prostředí, ve kterém bude provedeno testování před distribucí do produkčního prostředí;
- STD – produkční prostředí, které bude použito pouze pro distribuci webových stránek a jejich prezentaci uživatelům.

### 4.1.3 Sestavení základní kostry webu, návrh rozložení, vyhotovení makety webu

Na základě požadavku zadavatele na web a jeho počet stránek se zhotoví hrubý návrh webu a rozložení jeho prvků.



Obrázek 14 Wireframe (Zdroj: vlastní)

Tento návrh rozložení byl po schválení zaslán externí agentuře k přípravě grafických podkladů.

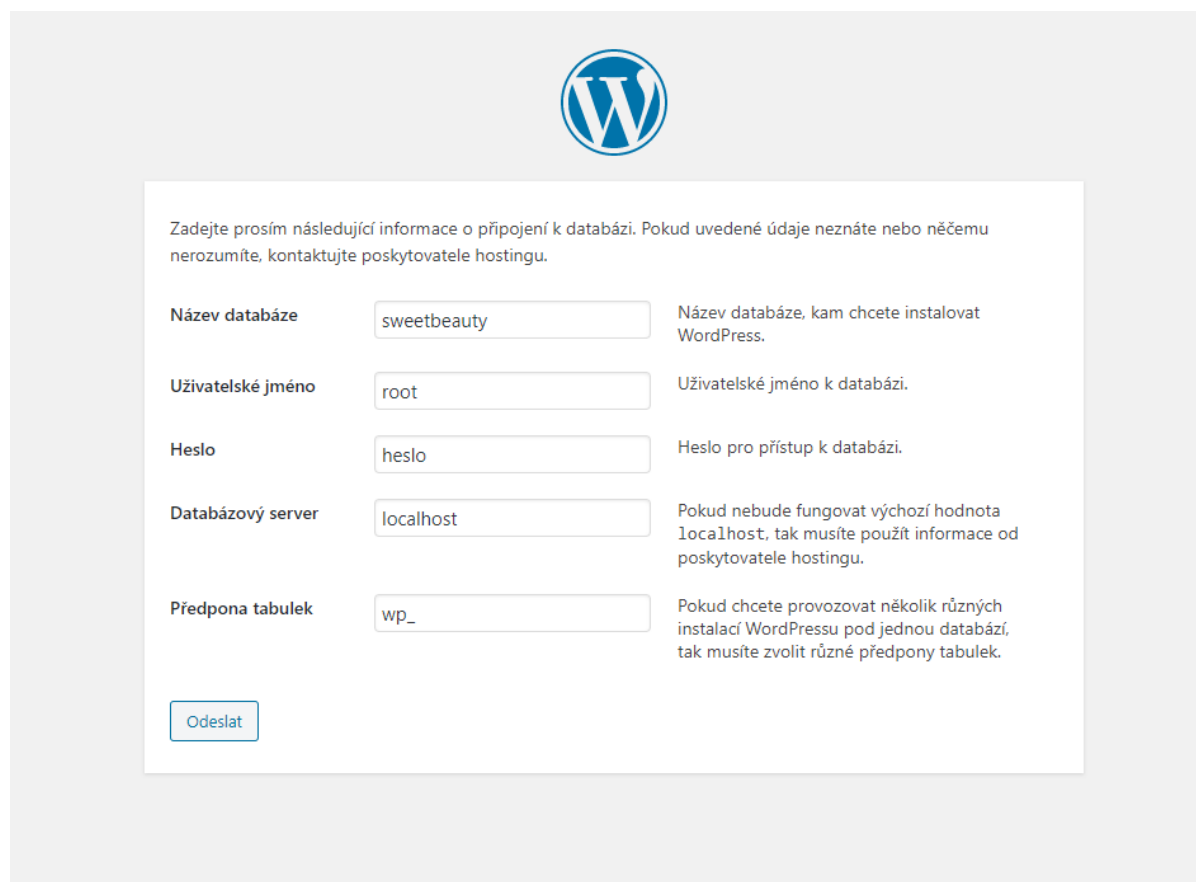
V čase, kdy agentura bude připravovat grafiku pro web, byla instalována nová instalace WordPressu na lokální server a byly připravovány moduly pro testování funkcionality.

#### 4.1.4 Instalace WordPressu

Krok 1: Pro úspěšné zahájení instalace CMS WordPress je třeba vytvořit prázdnou databázi a disponovat následujícími údaji o ní:

1. název databáze,
2. uživatelské jméno k databázi,
3. heslo pro přístup k databázi,
4. databázový server,
5. předpona tabulek

Tyto údaje se vyplní do příslušného okna:



The image shows the WordPress installation database configuration screen. At the top center is the WordPress logo. Below it is a white box containing the following text and form fields:

Zadejte prosím následující informace o připojení k databázi. Pokud uvedené údaje neznáte nebo něčemu nerozumíte, kontaktujte poskytovatele hostingu.

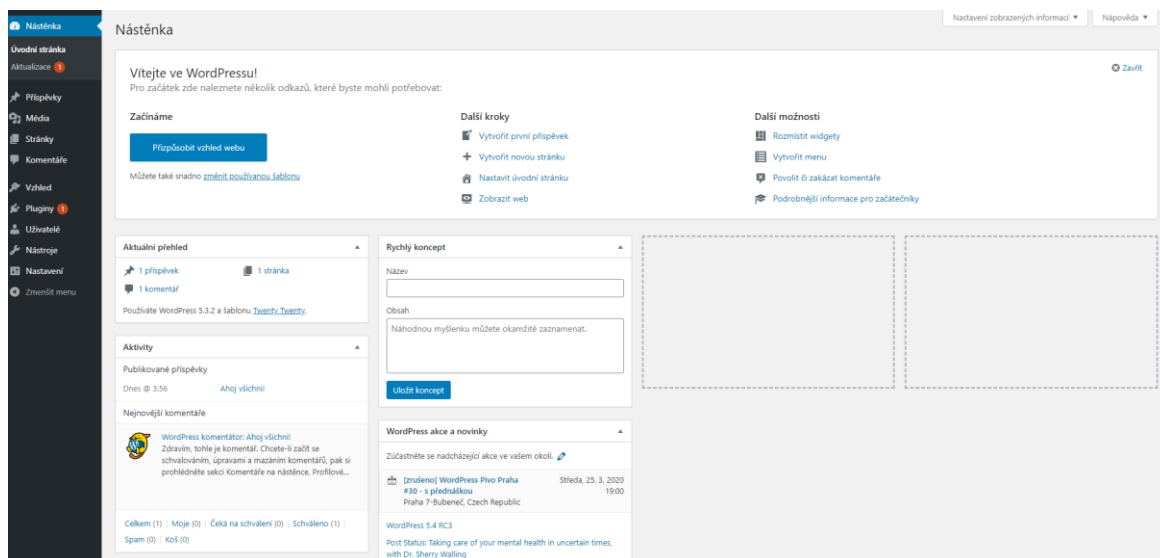
|                   |  |   |
|-------------------|--|---|
| Název databáze    | <input type="text" value="sweetbeauty"/> | Název databáze, kam chcete instalovat WordPress.  |
| Uživatelské jméno | <input type="text" value="root"/>        | Uživatelské jméno k databázi.   |
| Heslo             | <input type="text" value="heslo"/>       | Heslo pro přístup k databázi.   |
| Databázový server | <input type="text" value="localhost"/>   | Pokud nebude fungovat výchozí hodnota localhost, tak musíte použít informace od poskytovatele hostingu.                     |
| Předpona tabulek  | <input type="text" value="wp_"/>         | Pokud chcete provozovat několik různých instalací WordPressu pod jednou databází, tak musíte zvolit různé předpony tabulek. |

At the bottom left of the form is a blue button labeled "Odeslat".

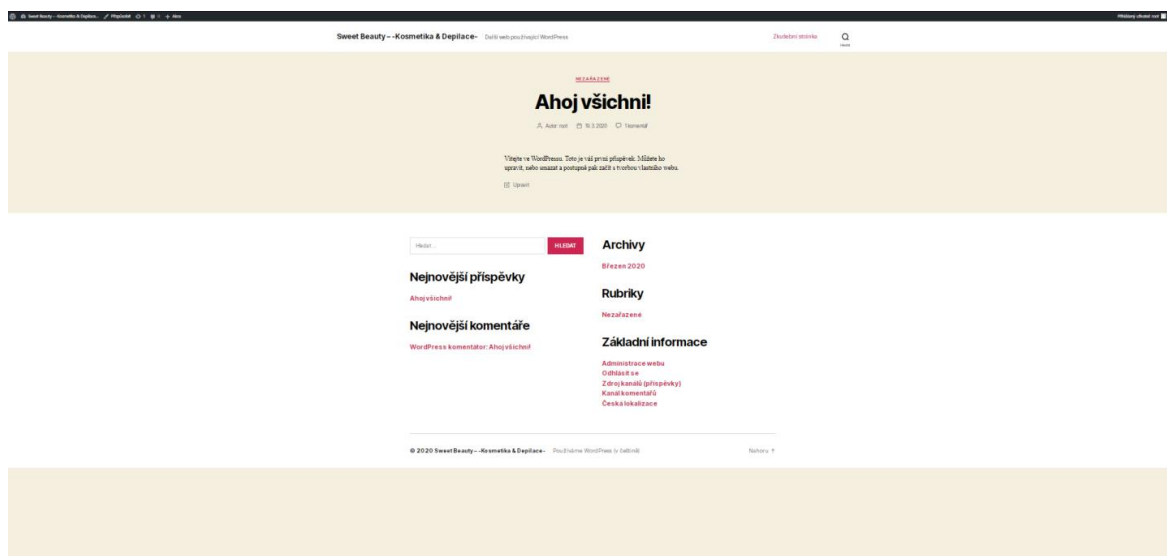
Obrázek 15 Instalace WordPress (Zdroj: vlastní)



Následně bude během několika minut provedena automatická instalace; do procesu se nijak nezasahuje. Po úspěšné instalaci se zobrazí stránka, kde je vyžadováno přihlášení za pomoci údajů, které byly zadány při instalaci. Administrační rozhraní, na které jsme přeměrováni po dokončení instalace, je zobrazeno na obrázku níže.



**Obrázek 16 Administrační rozhraní (Zdroj: vlastní)**  
Automaticky vygenerovaná stránka této instalace vypadá následovně:

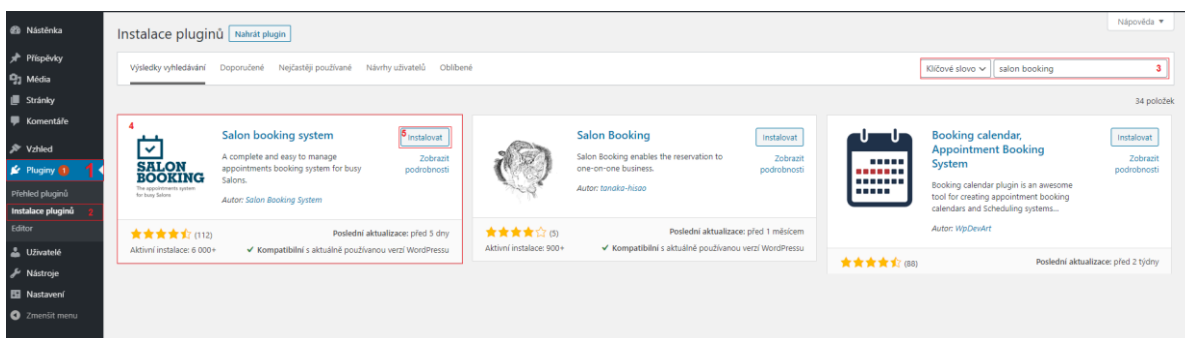


**Obrázek 17 Domů (Zdroj: vlastní)**  
První krok, tj. instalaci, jsme zvládli úspěšně. Jelikož externí agentura, která připravuje šablonu pro web, zatím nedodala grafické podklady, budeme se v následujících krocích zabývat instalací pluginu rezervačního systému.

#### 4.1.4.1 Instalace pluginu Salon booking system

Instalace pluginu ve WordPressu je velmi jednoduchá. Jedná se o pět základních kroků, ale i přesto si je ukážeme:

1. Otevřeme kartu Pluginy.
2. Klikněte na Instalace pluginů.
3. Do vyhledávacího pole zadáme název pluginu, např. Salon booking system.
4. V seznamu výsledků vyhledávání nalezneme potřebný plugin (Salon booking system).
5. Klikněte na tlačítko Instalovat.



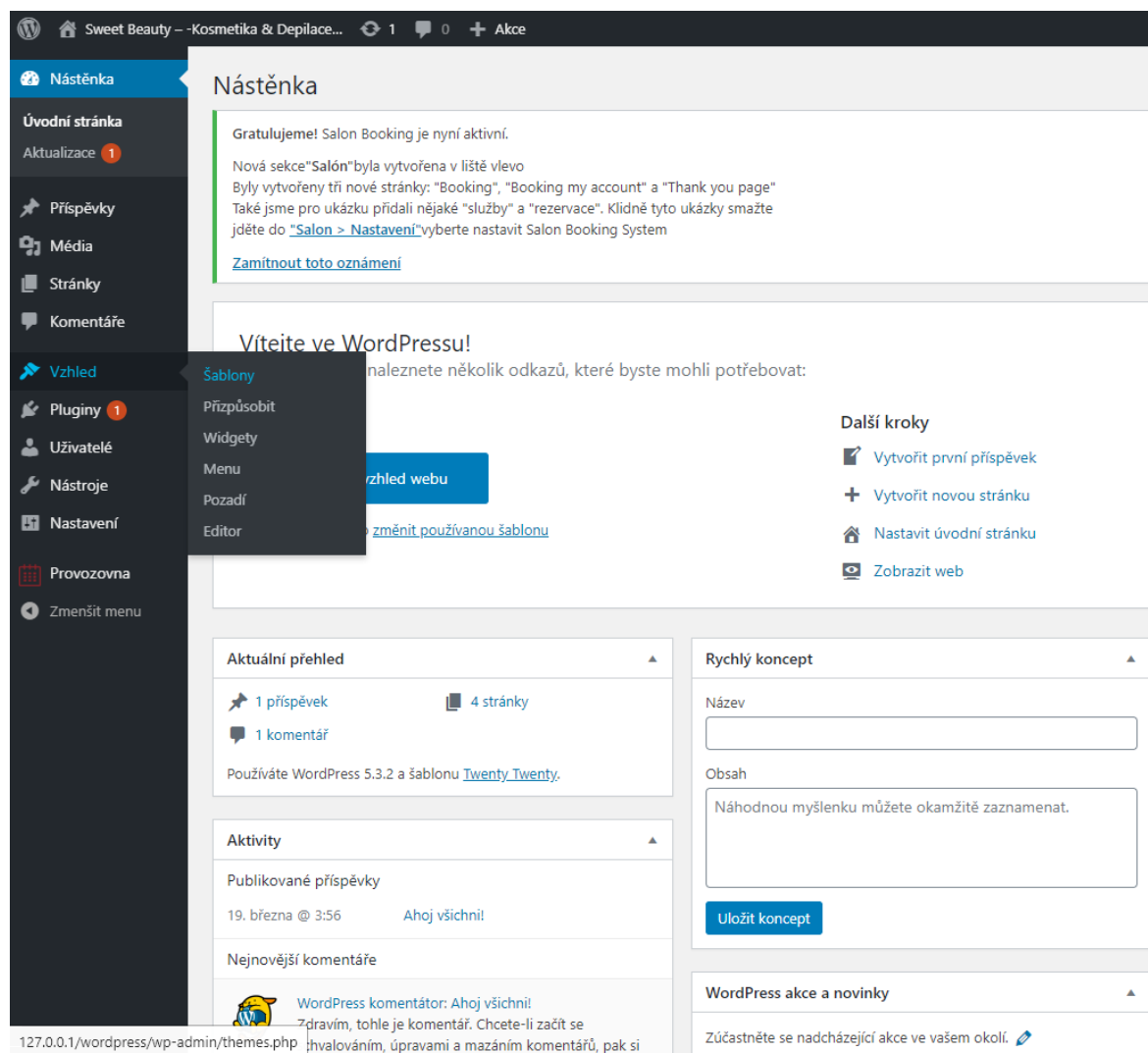
**Obrázek 18 Salon booking systém (Zdroj: vlastní)**

Po úspěšné instalaci nového pluginu klikneme na tlačítko Aktivovat, které se zobrazí místo tlačítka Instalovat. Nový plugin byl úspěšně nainstalován a aktivován.

#### 4.1.4.2 Instalace šablony

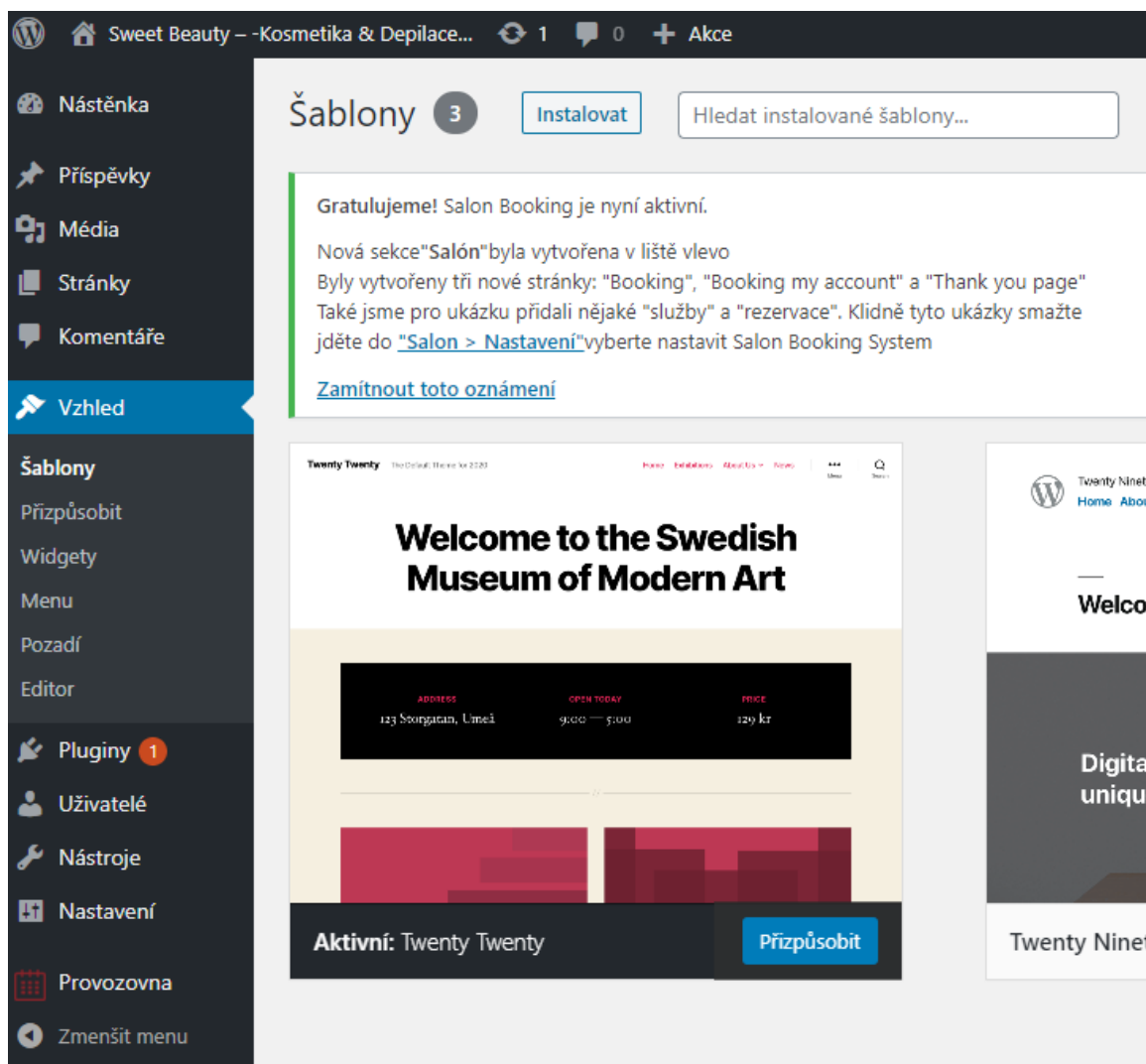
Všechny instalační kroky lze provést v rozhraní WordPress. Díky tomu je proces pohodlný a rychlý. Jednotlivé kroky jsou uvedeny níže:

1. Přejdeme do nabídky Vzhled a následně klikneme na Šablony.



Obrázek 19 Změna vzhledu (Zdroj: vlastní)

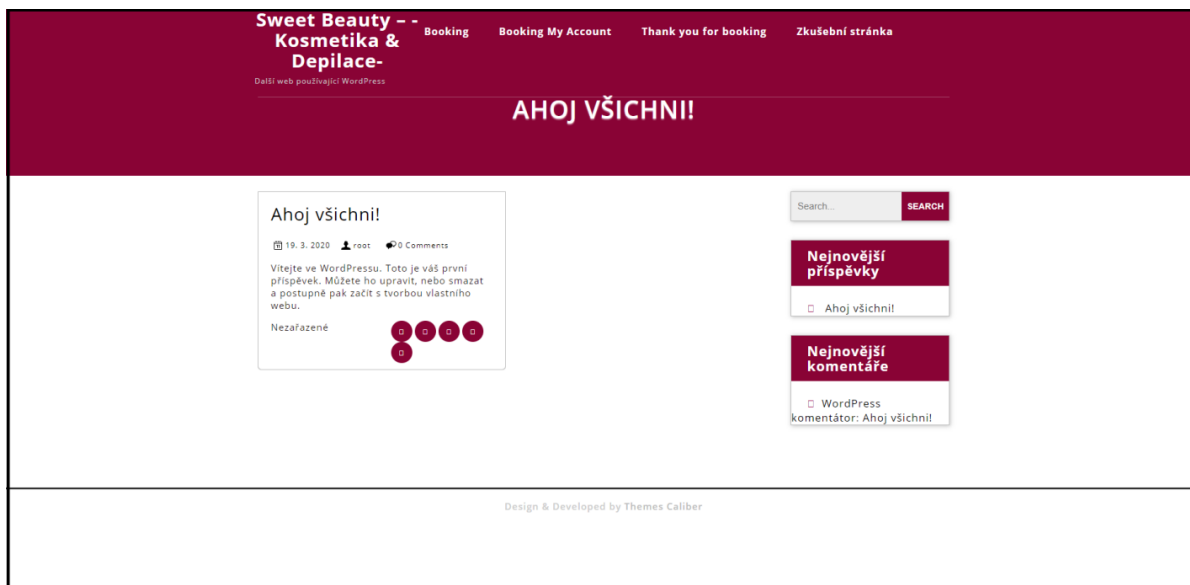
2. Jakmile jste na stránce Šablony, klikneme na tlačítko Instalovat v horní části.



Obrázek 20 Změna vzhledu šablona (Zdroj: vlastní)

3. Následně vybereme naši šablonu a instalace bude provedena.

Po úspěšné instalaci naší šablony web vypadá následovně:



Obrázek 21 Po instalaci šablony (Zdroj: vlastní)

#### 4.1.5 Databázová struktura

Přehled databázové struktury vypadá takto:

| Pole       | Typ          | Vlastnosti | Nulový | Výchozí | Další |
|------------|--------------|------------|--------|---------|-------|
| comment_id | bigint(20)   | UNSIGNED   | Ne     | 0       |       |
| meta_key   | varchar(255) |            | Ano    | NULL    |       |
| meta_value | longtext     |            | Ano    | NULL    |       |

**Tabulka 2 wp\_commentmeta**

| Pole                 | Typ          | Vlastnosti | Nulový | Výchozí             | Další          |
|----------------------|--------------|------------|--------|---------------------|----------------|
| comment_ID           | bigint(20)   | UNSIGNED   | Ne     |                     | auto_increment |
| comment_post_ID      | bigint(20)   | UNSIGNED   | Ne     | 0                   |                |
| comment_author       | tinytext     |            | Ne     |                     |                |
| comment_author_email | varchar(100) |            | Ne     |                     |                |
| comment_author_url   | varchar(200) |            | Ne     |                     |                |
| comment_author_IP    | varchar(100) |            | Ne     |                     |                |
| comment_date         | datetime     |            | Ne     | 0000-00-00 00:00:00 |                |
| comment_date_gmt     | datetime     |            | Ne     | 0000-00-00 00:00:00 |                |
| comment_content      | text         |            | Ne     |                     |                |
| comment_karma        | int(11)      |            | Ne     | 0                   |                |
| comment_approved     | varchar(20)  |            | Ne     | 1                   |                |
| comment_agent        | varchar(255) |            | Ne     |                     |                |
| comment_type         | varchar(20)  |            | Ne     |                     |                |
| comment_parent       | bigint(20)   | UNSIGNED   | Ne     | 0                   |                |
| user_id              | bigint(20)   | UNSIGNED   | Ne     | 0                   |                |

**Tabulka 3 wp\_comments**

| Pole             | Typ          | Vlastnosti | Nulový | Výchozí                | Další          |
|------------------|--------------|------------|--------|------------------------|----------------|
| link_id          | bigint(20)   | UNSIGNED   | Ne     |                        | auto_increment |
| link_url         | varchar(255) |            | Ne     |                        |                |
| link_name        | varchar(255) |            | Ne     |                        |                |
| link_image       | varchar(255) |            | Ne     |                        |                |
| link_target      | varchar(25)  |            | Ne     |                        |                |
| link_description | varchar(255) |            | Ne     |                        |                |
| link_visible     | varchar(20)  |            | Ne     | Y                      |                |
| link_owner       | bigint(20)   | UNSIGNED   | Ne     | 1                      |                |
| link_rating      | int(11)      |            | Ne     | 0                      |                |
| link_updated     | datetime     |            | Ne     | 0000-00-00<br>00:00:00 |                |
| link_rel         | varchar(255) |            | Ne     |                        |                |
| link_notes       | mediumtext   |            | Ne     |                        |                |
| link_rss         | varchar(255) |            | Ne     |                        |                |

**Tabulka 4 wp\_links**

| Pole         | Typ          | Vlastnosti | Nulový | Výchozí | Další          |
|--------------|--------------|------------|--------|---------|----------------|
| option_id    | bigint(20)   | UNSIGNED   | Ne     |         | auto_increment |
| option_name  | varchar(191) |            | Ne     |         |                |
| option_value | longtext     |            | Ne     |         |                |
| autoload     | varchar(20)  |            | Ne     | yes     |                |

**Tabulka 5 wp\_options**

| Pole       | Typ          | Vlastnosti | Nulový | Výchozí | Další          |
|------------|--------------|------------|--------|---------|----------------|
| meta_id    | bigint(20)   | UNSIGNED   | Ne     |         | auto_increment |
| post_id    | bigint(20)   | UNSIGNED   | Ne     | 0       |                |
| meta_key   | varchar(255) |            | Ano    | NULL    |                |
| meta_value | longtext     |            | Ano    | NULL    |                |

**Tabulka 6 wp\_postmeta**

| Pole                  | Typ          | Vlastnosti | Nulový | Výchozí             | Další          |
|-----------------------|--------------|------------|--------|---------------------|----------------|
| ID                    | bigint(20)   | UNSIGNED   | Ne     |                     | auto_increment |
| post_author           | bigint(20)   | UNSIGNED   | Ne     | 0                   |                |
| post_date             | datetime     |            | Ne     | 0000-00-00 00:00:00 |                |
| post_date_gmt         | datetime     |            | Ne     | 0000-00-00 00:00:00 |                |
| post_content          | longtext     |            | Ne     |                     |                |
| post_title            | text         |            | Ne     |                     |                |
| post_excerpt          | text         |            | Ne     |                     |                |
| post_status           | varchar(20)  |            | Ne     | publish             |                |
| comment_status        | varchar(20)  |            | Ne     | open                |                |
| ping_status           | varchar(20)  |            | Ne     | open                |                |
| post_password         | varchar(255) |            | Ne     |                     |                |
| post_name             | varchar(200) |            | Ne     |                     |                |
| to_ping               | text         |            | Ne     |                     |                |
| pinged                | text         |            | Ne     |                     |                |
| post_modified         | datetime     |            | Ne     | 0000-00-00 00:00:00 |                |
| post_modified_gmt     | datetime     |            | Ne     | 0000-00-00 00:00:00 |                |
| post_content_filtered | longtext     |            | Ne     |                     |                |
| post_parent           | bigint(20)   | UNSIGNED   | Ne     | 0                   |                |
| guid                  | varchar(255) |            | Ne     |                     |                |
| menu_order            | int(11)      |            | Ne     | 0                   |                |
| post_type             | varchar(20)  |            | Ne     | post                |                |
| post_mime_type        | varchar(100) |            | Ne     |                     |                |
| comment_count         | bigint(20)   |            | Ne     | 0                   |                |

**Tabulka 7 wp\_posts**

| Pole       | Typ          | Vlastnosti | Nulový | Výchozí | Další          |
|------------|--------------|------------|--------|---------|----------------|
| meta_id    | bigint(20)   | UNSIGNED   | Ne     |         | auto_increment |
| term_id    | bigint(20)   | UNSIGNED   | Ne     | 0       |                |
| meta_key   | varchar(255) |            | Ano    | NULL    |                |
| meta_value | longtext     |            | Ano    | NULL    |                |

**Tabulka 8 wp\_termmeta**



| Pole       | Typ          | Vlastnosti | Nulový | Výchozí | Další          |
|------------|--------------|------------|--------|---------|----------------|
| term_id    | bigint(20)   | UNSIGNED   | Ne     |         | auto_increment |
| name       | varchar(200) |            | Ne     |         |                |
| slug       | varchar(200) |            | Ne     |         |                |
| term_group | bigint(10)   |            | Ne     | 0       |                |

**Tabulka 9 wp\_terms**

| Pole             | Typ        | Vlastnosti | Nulový | Výchozí | Další |
|------------------|------------|------------|--------|---------|-------|
| object_id        | bigint(20) | UNSIGNED   | Ne     | 0       |       |
| term_taxonomy_id | bigint(20) | UNSIGNED   | Ne     | 0       |       |
| term_order       | int(11)    |            | Ne     | 0       |       |

**Tabulka 10 wp\_term\_relationships**

| Pole             | Typ         | Vlastnosti | Nulový | Výchozí | Další          |
|------------------|-------------|------------|--------|---------|----------------|
| term_taxonomy_id | bigint(20)  | UNSIGNED   | Ne     |         | auto_increment |
| term_id          | bigint(20)  | UNSIGNED   | Ne     | 0       |                |
| taxonomy         | varchar(32) |            | Ne     |         |                |
| description      | longtext    |            | Ne     |         |                |
| parent           | bigint(20)  | UNSIGNED   | Ne     | 0       |                |
| count            | bigint(20)  |            | Ne     | 0       |                |

**Tabulka 11 wp\_term\_taxonomy**

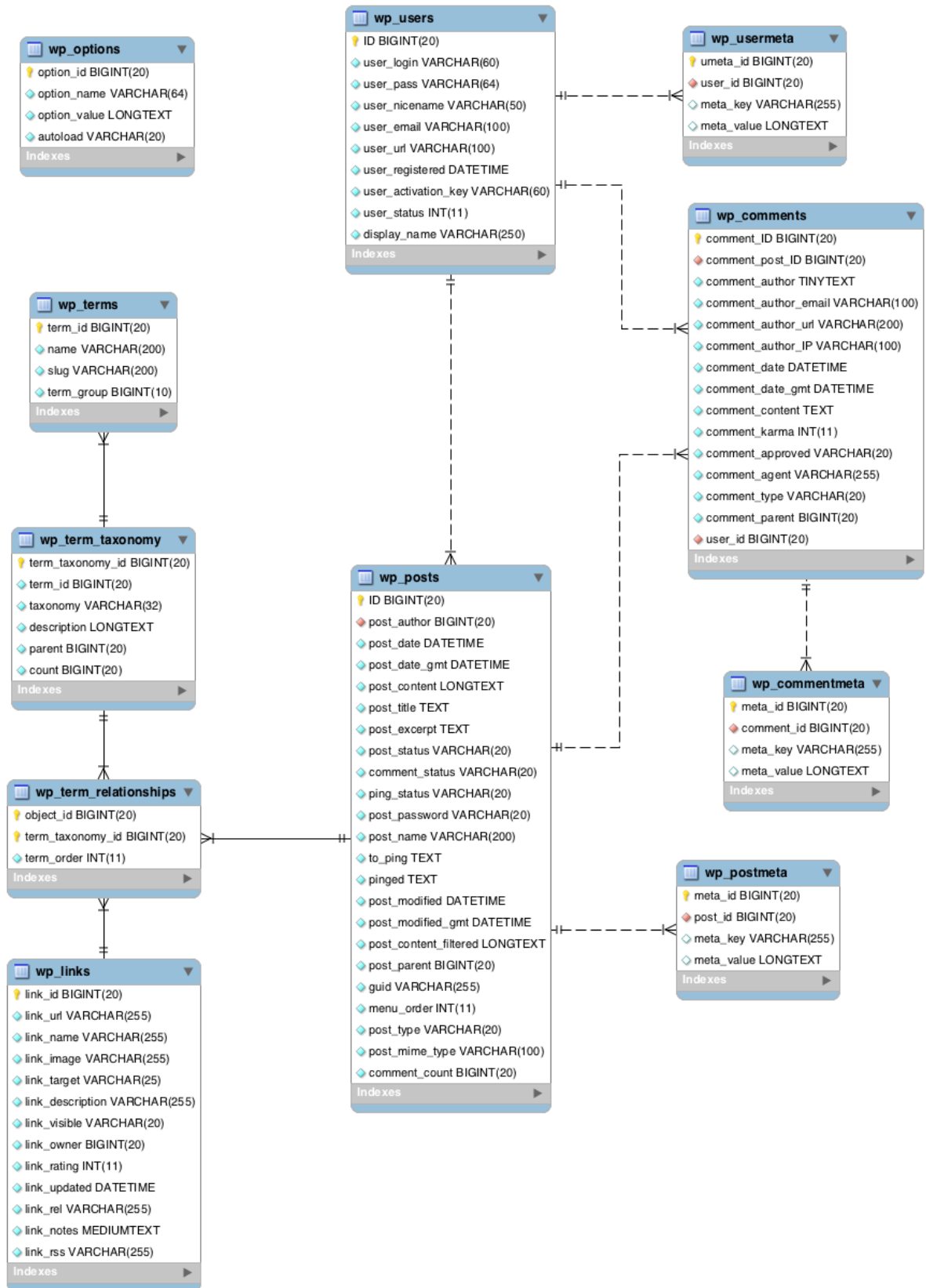
| Pole       | Typ          | Vlastnosti | Nulový | Výchozí | Další          |
|------------|--------------|------------|--------|---------|----------------|
| umeta_id   | bigint(20)   | UNSIGNED   | Ne     |         | auto_increment |
| user_id    | bigint(20)   | UNSIGNED   | Ne     | 0       |                |
| meta_key   | varchar(255) |            | Ano    | NULL    |                |
| meta_value | longtext     |            | Ano    | NULL    |                |

**Tabulka 12 wp\_usermeta**

| Pole                | Typ          | Vlastnosti | Nulový | Výchozí    | Další |
|---------------------|--------------|------------|--------|------------|-------|
| ID                  | bigint(20)   | UNSIGNED   | Ne     |            |       |
| user_login          | varchar(60)  |            | Ne     |            |       |
| user_pass           | varchar(255) |            | Ne     |            |       |
| user_nicena         | varchar(50)  |            | Ne     |            |       |
| user_email          | varchar(100) |            | Ne     |            |       |
| user_url            | varchar(100) |            | Ne     |            |       |
| user_register       | datetime     |            | Ne     | 0000-00-00 |       |
| user_activation_key | varchar(255) |            | Ne     |            |       |
| user_status         | int(11)      |            | Ne     | 0          |       |
| display_name        | varchar(250) |            | Ne     |            |       |

**Tabulka 13 wp\_users**

#### 4.1.6 Relační schéma



Obrázek 22 Relační schéma WP (Zdroj: codex.wordpress.org)

#### 4.1.7 Umístění na webový server

Zadavatelem byl zvolen dodavatel webhostingu Endora. Cílem je tedy otestovat funkčnost našeho webu na jeho hostingu. Jelikož tento dodavatel nabízí i bezplatné tarify pro hostování webu, můžeme testování provádět na testovacím serveru.

Tzn. že náš web pomocí FTP údaje, který obdržíme ihned po registraci, můžeme nahrát na webový server. Veškeré úpravy dále budeme provádět pomocí vývojového prostředí PhpStorm od firmy JetBrains. Toto vývojové prostředí umí naše změny promítat na webový server pomocí FTP, takže nebudeme muset náš web pokaždé přenášet ručně.

Administrační rozhraní webhostingu Endora je velmi přehledné (viz ukázka níže).

|                              | Aktuálně využito | Free   | Free neověřeny | Plus   | Mega    |
|------------------------------|------------------|--------|----------------|--------|---------|
| Počet souborů                | 7.280            | 25.000 | 15.000         | 50.000 | 100.000 |
| Místo na disku               | 398MB            | 2GB    | 1GB            | 10GB   | 30GB    |
| Přenos dat                   | 0B               | 30GB   | 1GB            | 100GB  | ∞       |
| Počet cron úloh              | 0                | 3      | 1              | ∞      | ∞       |
| Max. velikost jedné databáze | ∞                | 30MB   | 30MB           | 60MB   | 150MB   |

Obrázek 23 Endora (Zdroj: vlastní)

Po umístění byla změněna konfigurace souboru wp-config.php ve kterém je hlavní konfigurace. Následně byly učiněné vyžadované změny v databázi, aby web fungoval korektně.

Především se jedna o tyto změny v souboru:

```
<?php
define( 'DB_NAME', 'database_name_here' );
/** Jmeno uživatele pro MySQL databazi*/
define( 'DB_USER', 'username_here' );
/** Heslo uživatele pro MySQL databazi */
define( 'DB_PASSWORD', 'password_here' );
/** Hostname MySQL databaze a port */
define( 'DB_HOST', 'localhost' );
/** Znaková sada. */
define( 'DB_CHARSET', 'utf8' );
define( 'DB_COLLATE', '' );
?>
```

#### 4.1.8 Úprava šablony

Úprava šablony dle grafického návrhu a požadavku zadavatele byly provedeny již na serveru přes vývojové prostředí PhpStorm od firmy JetBrains. Některé ze změn vzhledu jsou uvedeny níže:

```
/*-----Uprava menu dle navrh-----*/
.sf-arrows .sf-with-ul{
  padding-right: 15px;
}
.sf-arrows .sf-with-ul:after{
  content: '';
  position: absolute;
  right: 0;
  margin-top: -2px;
  height: 0;
  width: 0;
  border-top-color: #e1e1e1;
}
.sf-arrows ul .sf-with-ul:after{
  margin-top: -5px;
  margin-right: -3px;
  border-color: transparent;
  border-left-color: #888;
}
.sf-arrows ul li > .sf-with-ul:focus:after,
.sf-arrows ul li:hover > .sf-with-ul:after,
.sf-arrows ul .sfHover > .sf-with-ul:after{
  border-left-color: #333;
}
.toggle-nav, a.closebtn{
  display: none;
}
```

V této šabloně neexistovala stránka 404 tak byla doplněna, kdyby uživatel na ni byl z nějakého důvodu přesměrován. Ukázka stránky 404 kód stránky je vložen níže .

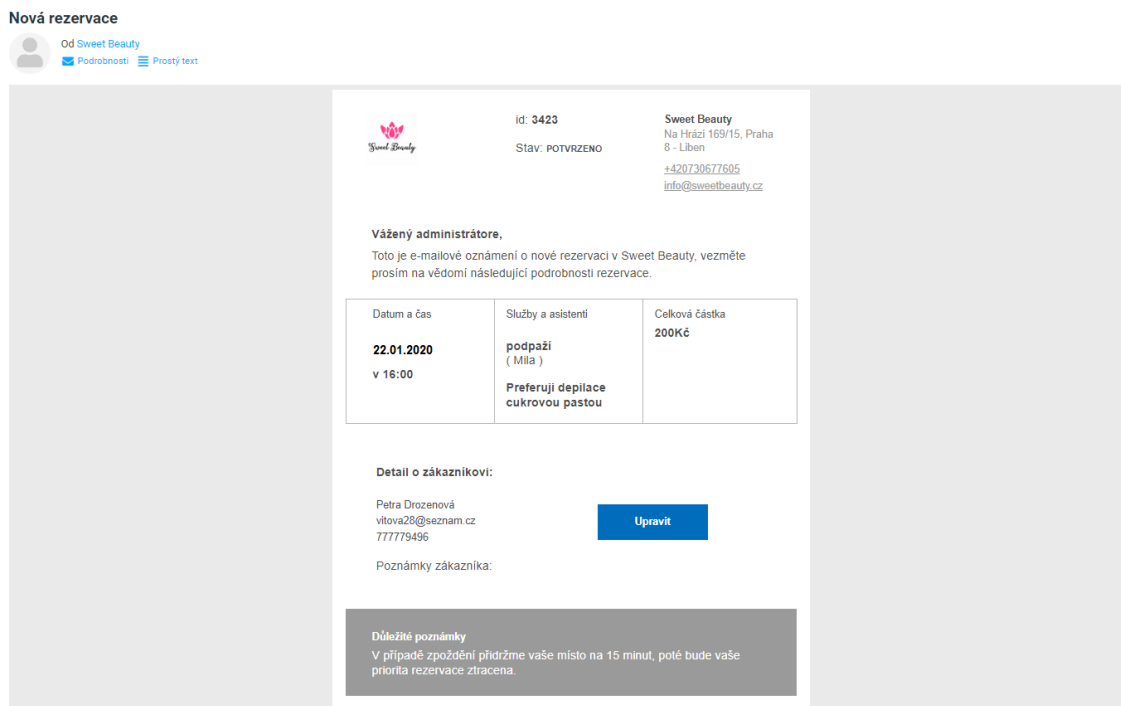
```
<?php
get_header();
?>
<div class="title-box">
  <div class="container">
    <h1><?php the_title();?></h1>
  </div>
</div>

<div id="content-vw">
  <div class="container">
    <div class="page-content">
      <div class="col-md-12">
        <h3><?php esc_html_e( '404 Nenalezeno', 'tc-sauna-pro' );
?></h3>
        <p class="text-404"><?php esc_html_e( 'Je nám líto, ale
požadovaná stránka nebyla nalezena.', 'tc-sauna-pro' ); ?></p>
        <p class="text-404"><?php esc_html_e( 'Nebojte se ... stane
se to i nejlepším z nás.', 'tc-sauna-pro' ); ?></p>
        <div class="read-moresec">
          <div><a href="<?php echo esc_url( home_url() );
?>"><?php esc_html_e( 'Zpět na domovskou stránku', 'tc-sauna-pro' );
?></a></div>
        </div>
      </div>
      <div class="clearfix"></div>
    </div>
    <div class="clearfix"></div>
  </div>
</div>
<?php get_footer(); ?>
```

Zaznamenat veškeré části kódů v bakalářské práci není reálný, takže pro příklady použitý náhodný kód z různých částí webu.

#### 4.1.9 SMTP/SMS Komunikace

Jelikož WordPress ve své základní funkcionalitě obsahuje možnost odesílání mailu při určitých událostech tak jde jenom o správnou konfiguraci, poté provedené testování ukázalo že SMTP komunikace funguje správně.



Obrázek 24 SMTP (Zdroj: vlastní)

SMS potvrzení je realizováno pomocí cloudové služby od firmy Twilio, webová stránka má nastavenou událost, při které se odešle požadavek s daty přes API, následně je automaticky vygenerovaná a odeslaná SMS zpráva která neobsahuje telefonní číslo ale alfanumerický ID odesílatele. Kontrola funkčnosti provedena a výsledky jsou zaznamenány níže



Obrázek 25 SMS (Zdroj: vlastní)

## 5 Výsledky a diskuze

Zvoleným tématem této bakalářské práce byla tvorba webu založeného na CMS WordPress. Základním předpokladem bylo důkladné prostudování problematiky vývoje webu založeného na systému pro správu obsahu a dále nastudování materiálů s tím spojených. Nastudované materiály byly následně použity pro tvorbu teoretické části bakalářské práce.

Výsledkem této bakalářské práce je funkční podniková webová stránka s pluginem rezervačního systému. Webová stránka prezentuje společnost zadavatele na internetu a pomocí pluginu rezervace umožňuje provést rezervaci služeb pro jednotlivé uživatele. Systém odesílá aktuální informaci s pomocí e-mailových notifikací a také notifikuje pomocí SMS uživatele, který úspěšně dokončil rezervaci.

Během tvorby praktické části bakalářské práce zhotovitel dospěl k tomu, že samotný vývoj webové stránky za pomoci CMS WordPress není tak jednoduchý, jak se může na první pohled zdát, ale to může být dáno tím, že tento způsob vývoje nebyl pro zhotovitele doposud známý.

Úspěšnost vývoje závisela na několika faktorech, jako je kvalitní návrh šablony a správné rozvržení myšlenek po získání veškerých informací, a v neposlední řadě také na podpoře vedoucího této bakalářské práce. Velkou část úspěchu nese i správně zvolený plugin a moduly, které by plně vyhovovaly zadaným kritériím anebo do nichž by se muselo zasahovat jen minimálně.

Práce na tomto projektu byla pro zhotovitele zcela něčím novým, jelikož zkušenosti s vývojem za pomoci tohoto CMS žádné neměl, ale i přes veškeré nejasnosti a potíže, ke kterým během vývoje došlo, je zhotovitel rád za zkušenosti s touto technologií.

## 6 Závěr

Bakalářská práce byla tematicky zaměřena na problematiku vývoje webových stránek v CMS WordPress. Plně funkční verze systému byla vytvořena dle zadání za použití CMS WordPress, funkčnost webových stránek byla ověřena velkým počtem uživatelů, jelikož web již běží v provozu po dobu šesti měsíců.

Při tvorbě webové stránky autor dospěl k poznatku, že samotný vývoj není tak složitý jako dohledání správných zdrojů a následné testování jím vytvořeného webu. Autor též zjistil, že flexibilita WordPressu a jeho bezplatnost mu hodně přidávají na popularitě u většiny vývojářů webů, stejně jako u běžných uživatelů.

K popularitě tohoto systému přispívá také intuitivní rozhraní, lokalizace pro většinu jazyků, rozšiřitelnost funkčnosti prostřednictvím pluginů a schopnost změnit design pomocí šablon, které jsou v některých případech dostupné i zcela zdarma.

Autor je rád, že měl šanci seznámit se s něčím novým a vyzkoušet si to v praxi. Zkušenosti získané v této oblasti plánuje dál rozšiřovat a implementovat je v praxi.



## 7 Seznam použitých zdrojů

- [1] *WPkniha* [online]. b.r. [cit. 2020-03-02]. Dostupné z: <http://www.wpkniha.cz/wordpress/historie>
- [2] *Jak psát web: o tvorbě internetových stránek* [online]. b.r. [cit. 2020-03-02]. Dostupné z: <https://www.jakpsatweb.cz/>
- [3] *Kinsta: KNOWLEDGE BASE* [online]. b.r. [cit. 2020-03-02]. Dostupné z: <https://kinsta.com/knowledgebase/content-management-system/>
- [4] *Apache HTTP Server Documentation* [online]. b.r. [cit. 2020-03-02]. Dostupné z: <https://httpd.apache.org/docs/>
- [5] DOMES, Martin. *Tvorba WWW stránek pro úplné začátečníky*. 1. Brno: Computer Press, 2012. ISBN 978-80-251-2160-3.
- [6] HOPKINS, Callum. *PHP okamžitě*. 1. Brno: Computer Press, 2014. ISBN 978-80-251-4196-0.
- [7] WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL*. 1. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1.
- [8] ASTON, Ben. *Medium* [online]. b.r. [cit. 2020-03-02]. Dostupné z: <https://medium.com/@benastontweet/lesson-1-a-the-history-of-javascript-8c1ce3bffb17>
- [9] *JQuery history* [online]. b.r. [cit. 2020-03-02]. Dostupné z: <https://jquery.org/history/>
- [10] *Bootstrap 4.1 Introduction* [online]. 2018 [cit. 2020-03-02]. Dostupné z: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- [11] RETHANS, Derick. *Xdebug dokumentace*. [online]. 2020 [cit. 2020-03-02]. Dostupné z: <https://xdebug.org>