

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Bakalářská práce**

**Návrh modulu pro redakční systém WordPress**

**Jan Jahoda**

© 2015 ČZU v Praze

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačního inženýrství

Provozně ekonomická fakulta

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jan Jahoda

Informatika

Název práce

**Návrh modulu pro redakční systém Wordpress**

Název anglicky

**Design of Blogging System Wordpress Plugin**

---

### Cíle práce

Cílem bakalářské práce je vytvoření modulu (pluginu) rezervačního systému pro tenisové kurty. Modul bude vytvořen jako rozšíření redakčního systému Wordpress pomocí technologií PHP a jquery.

### Metodika

1. Nastudování dostupných zdrojů o problematice redakčních systémů a jejich rozšíření formou pluginů

**Doporučený rozsah práce**

30 – 40 stran

**Klíčová slova**

plugin, wordpress, PHP, jquery,html,css,návrh databáze,webová aplikace, rezervační systém

---

**Doporučené zdroje informací**

Luboš Kudláček. WordPress: podrobný průvodce tvorbou a správou webů. 2010. Brno. ISBN 978-80-251-2734-6

Lucie Šestáková. WordPress: vlastní web bez programování. 2013. Brno. ISBN 978-80-251-3832-8

Michaela Horňáková. 333 tipů a triků pro WordPress: sbírka nejlepších řešení a postupů pro váš web. 2011. Brno. ISBN 978-80-251-3443-6

---

**Předběžný termín obhajoby**

2015/06 (červen)

**Vedoucí práce**

Ing. David Buchtela, Ph.D.

Elektronicky schváleno dne 16. 3. 2015

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 16. 03. 2015

### Čestné prohlášení

Prohlašuji, že svou bakalářskou práci Návrh modulu pro redakční systém WordPress jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 16.3.2015

---

## Poděkování

Rád bych touto cestou poděkoval vedoucímu bakalářské práce Ing. Davidu Buchtelovi, Ph.D. za odborné rady při psaní bakalářské práce.



# Návrh modulu pro redakční systém WordPress

---

## Design of blogging system WordPress plugin

### Souhrn

Tato bakalářská práce se zaměřuje na návrh pluginu v redakčním systému WordPress a na jeho funkce. První část teoretické práce se zabývá popisem a jednotlivými druhy webových aplikací. Zde jsou i představeny jednotlivé počítačové jazyky, ve kterých se dají webové aplikace vyvíjet. Další část se zabývá programy potřebnými pro pracování ve WordPressu. Konkrétně programem XAMPP, jenž vytvoří webový server, díky kterému se může poté používat jazyk PHP a MySQL, mimo jiné se práce věnuje vývojářskému prostředí NetBeans. V poslední teoretické části je nastíněna krátká historie WordPressu a po ní jsou vysvětleny šablony tohoto redakčního systému, funkce pro výpis příspěvků a samotné pluginy. V praktické části je popsán v jednotlivých bodech úplný postup pro vytvoření rezervačního systému. V této části je popisována instalace webového serveru a WordPressu, vytvoření šablony umožňující výpis příspěvků. Poslední část je věnována samostatnému naprogramování funkčního rezervačního pluginu.

### Summary

This bachelor thesis primary aims towards the plugin proposal in WordPress blogging tool and onto it's functions. The theoretical part describes individual web applications and introduces computer languages, in which those applications may be developed. The next segment deals with programs required for working in WordPress, specifically XAMPP, a program that creates a web server, thanks to which is possible to use PHP, MySQL and NetBeans developer's tool. In the last segment of the theoretical part, a short history of WordPress is outlined, followed by explanation of it's templates, insert-a-post function and plugins themselves. In the practical part, a complete walkthrough for full reservation system creation is being described in very detailed and easy-to-understand articles. In this segment, an

installation of a web server and WordPress itself is being described alongside with creation of the template, which allows articles listing, including description of the programming steps for the fully functional reservation plugin.

**Klíčová slova:** plugin, wordpress, PHP, jquery,html,css,návrh databáze,webová aplikace, rezervační systém

**Keywords:** plugin, wordpress, PHP, jquery, html, css, design of database, web application, content management system



## Obsah

Obsah .....	3
Úvod.....	6
1 Cíl práce a metodika .....	7
1.1 Cíl práce.....	7
1.2 Metodika .....	7
2 Webové aplikace.....	8
2.1 Typy webových aplikací .....	8
2.2 Webový prohlížeč .....	8
2.3 HTML .....	9
2.3.1 HTML5 .....	10
2.4 CSS .....	10
2.4.1 CSS3 .....	11
2.5 jQuery .....	12
2.5.1 jQuery UI.....	13
2.6 PHP.....	13
2.7 MySQL .....	14
2.7.1 Databázová struktura .....	15
2.7.2 Komunikace s MySQL serverem.....	15
2.7.3 SQL injection.....	16
2.8 Bootstrap.....	16
2.8.1 Responzivní web.....	17
3 Využité programy .....	17
3.1 NetBeans.....	17
3.1.1 Funkce NetBeans .....	17
3.2 XAMPP.....	18
4 Wordpress .....	18
4.1 Historie WordPress .....	19
4.2 Loop.....	20
4.3 Šablona.....	21
4.4 Wordpress pluginy.....	22

4.4.1	Správa a instalace pluginů .....	23
4.4.2	Instalace pluginů .....	23
4.4.3	Správa pluginů .....	24
4.4.4	Interakce pluginů s Wordpressem.....	24
5	Praktická část .....	29
5.1	Webový server .....	29
5.2	Připojení do phpMyAdmin .....	30
5.3	Instalace wordpresu .....	30
5.4	Vytvoření šablony .....	31
5.4.1	Projekt v NetBeans .....	31
5.4.2	Úvodní stránka .....	31
5.5	Vytvoření pluginu .....	33
5.5.1	Výběr dne.....	34
5.5.2	Výpis formuláře .....	35
5.5.3	Uložení rezervace .....	37
5.5.4	Administrativní rozhraní pluginu.....	38
6	Závěr .....	40
7	Seznam použitých zdrojů.....	41
	Seznam literatury .....	41
8	Přílohy.....	42
8.1	Obsah CD.....	42

## 1 Seznam tabulek

Tabulka 1. Verze WordPressu [21].....	20
Tabulka 2. Stavové kódy [24].....	27

## 2 Seznam obrázků

Obrázek 1. Nejpopulárnější webové prohlížeče za únor 2015 .....	9
Obrázek 2. HTTP požadavek.....	26
Obrázek 3. HTTP odpověď.....	27
Obrázek 4. XAMPP Control Panel .....	30
Obrázek 5. Registrace menu .....	31
Obrázek 6. Loop .....	32
Obrázek 7. Úvodní stránka .....	32
Obrázek 8. Vytvoření tabulky.....	33
Obrázek 9. Funkce pro výpis rezervace.....	34
Obrázek 10. Kalendář .....	34
Obrázek 11. Výpis datumů .....	34
Obrázek 12. jQuery - událost při kliknutí.....	35
Obrázek 13. Vybrání rezervovaných termínů.....	36
Obrázek 14. Inputy .....	36
Obrázek 15. jQuery - dialogové okno.....	36
Obrázek 16. Funkce pro ukládání rezervace.....	37
Obrázek 17. Rezervace - administrativní rozhraní .....	38
Obrázek 18. Smazání rezervace.....	38

## Úvod

V dnešní době je již zcela běžné, že webové stránky umějí kromě klasické prezentace informací i různé funkce jako jsou například rezervační systémy, přihlašování pomocí formulářů, chatovací boxy pro komunikaci přihlášených uživatelů a spoustu dalších. Tyto funkce se dají buď klasicky naprogramovat v příslušném programovacím jazyce jako je například skriptovací jazyk PHP (Hypertext Preprocessor) za užití jeho mnoha frameworků, či se dá využít framework od Microsoftu ASP.NET (Active Server Pages), možností je ale samozřejmě více.

Pokud webová aplikace funguje na redakčním systému, tak se pro ní dá vytvořit tzv. plugin. Jedná se o modul, který po naprogramování funguje v daném redakčním systému a je možné jej poté nabízet i ostatním uživatelům stejného systému, kteří jej pak pár klikama nainstalují na své stránky. Tato bakalářská práce se zaměřuje na v současnosti nejpoužívanější open source redakční systém na světě – WordPress. Tento systém se od svého vzniku v roce 2003 vyšplhal mezi jedničku ve svém oboru a v současnosti je na více než 15% všech stránek na webu s redakčním systémem. V práci jsou uvedeny základní informace o jeho vzniku, funkcích a fungování.

V praktické části je popsán vývoj naprogramovaného pluginu pro rezervaci tenisových kurtů. V České republice jsou redakční systémy již poměrně známé, ale většina lidí netuší jak jeho pluginy fungují, či jak se vytvářejí nové. Toto téma bylo zvoleno z důvodu, aby se s touto problematikou seznámili programátoři znalý základům vytváření webových aplikací.

# **1 Cíl práce a metodika**

## **1.1 Cíl práce**

Bakalářská práce je zaměřena na problematiku vývoje pluginů v redakčním systému WordPress a to jak z teoretické, tak i praktické části. Cílem práce je seznámit programátory, jenž znají základy vytváření webových aplikací s vytvářením aplikací pro systém WordPress. Podrobně vysvětlit fungování šablon a souborů, z nichž se skládá. Vysvětlit základní informace o použitých programovacích jazycích a o nástrojích potřebných k vyhotovení webové aplikace.

Cílem praktické části je zhotovit na základě informací získaných z teoretické části rezervační systém pro tenisové kurty. Detailně popsat postup od úplného začátku vytvořením webového serveru po úspěšné vyhotovení pluginu ve WordPressu.

## **1.2 Metodika**

Metodika řešené problematiky bakalářské práce je založena na nastudování a analýze odborných knížek a článků týkajících se programovacích jazyků potřebných pro vývoj webové aplikace a knih zabývajících se vytvářením šablon a pluginů pro WordPress. Kvůli malému množství kvalitních zdrojů souvisejích s bakalářskou prací jsem byl nucen spoléhat se na cizojazyčnou literaturu, svoje znalosti nabyté z vývoje webových aplikací a ze znalostí získaných při studiu na vysoké škole.

Celá aplikace s pluginem pro rezervaci kurtů je zhotovena pomocí programu pro vytvoření webového serveru XAMPP a vývojářského prostředí NetBeans. Za využití znalostí z teoretické části jsou postupně naprogramovány jednotlivé části webové aplikace.

## 2 Webové aplikace

Webové aplikace jsou všechny aplikace běžící ve webovém prohlížeči. Jedná se o software naprogramovaný v prohlížeči podporovaném jazyce, jako je například javascript, HTML a stylovací jazyk CSS. Díky všudypřítomnosti zařízení s prohlížeči, a schopnosti aktualizování a spravování bez nutnosti instalace softwaru, se stali velice populárními.

Na rozdíl od statických webových stránek, které pouze prezentují informace, webové aplikace umožňují i interakci s návštěvníkem serveru.

[1]

### 2.1 Typy webových aplikací

Webové aplikace byly vytvořeny k provedení prakticky všech užitečných funkcí, které by se daly implementovat online. Aplikace se neustále rozrůstají a stále vznikají nové. Mezi nejznámější typy spolu s příkladem patří například aplikace pro:

- **Nakupování** (Slevomat)
- **Sociální sítě** (Facebook)
- **Bankovní služby** (Komerční banka)
- **Webové vyhledavače** (Google)
- **Aukce** (Aukro)
- **Emailové služby** (Gmail)
- **Informační portály** (Wikipedia)

[1]

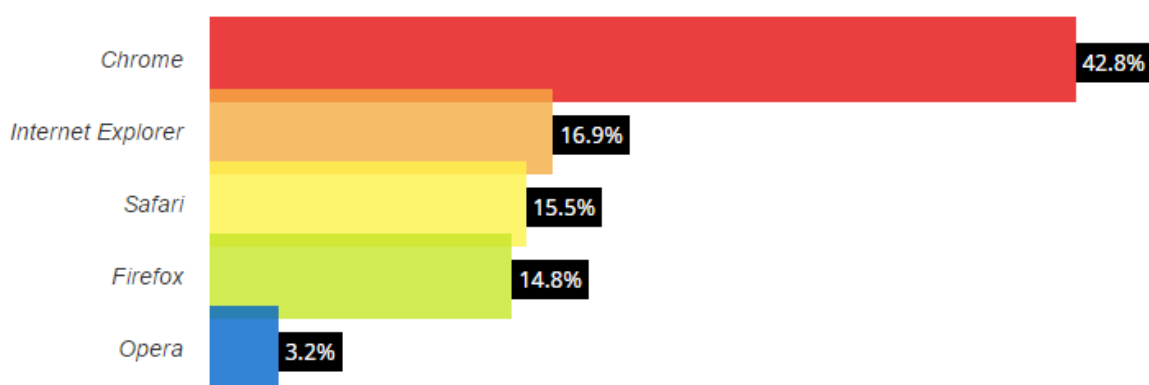
### 2.2 Webový prohlížeč

Webový prohlížeč, neboli zkráceně prohlížeč, je softwarová aplikace určená k vyhledávání a zobrazování obsahu na webu, včetně webových stránek, obrázků, videí a dalších souborů. Prohlížeč funguje dle síťové architektury klient/server, která odděluje klienta a server. V tomto případě je klient webový prohlížeč běžící na počítači, který kontaktuje webový server a vyšle požadavek na informace. Webový server odesílá informace zpět webovému prohlížeči, jenž zobrazí výsledky na počítači nebo jiném zařízení vybaveném prohlížečem. Dnešní prohlížeče jsou plně funkční

softwarové aplikace, které umějí interpretovat a zobrazovat HTML, webové stránky a aplikace, JavaScript, AJAX a další obsah hostovaný na webových serverech. Mnoho prohlížečů nabízí i různá rozšíření, které zvětšují možnosti softwaru, aby mohl zobrazovat multimediální informace, včetně zvuku a videa. Dále prohlížeč zvládne takové úkoly, jako jsou například videokonference, úprava webových stránek a mnohé další. Mezi nejznámější prohlížeče patří Mozilla Firefox, Internet Explorer, Google Chrome, Opera a Safari. V současnosti nejpoužívanějším webovým prohlížečem je Google Chrome.

[2,3]

Obrázek 1. Nejpoužívanější webové prohlížeče za únor 2015



Zdroj: w3counter.com

### 2.3 HTML

„HTML (*HyperText Markup Language*) je značkovací jazyk pro hypertext, který je určen k vytváření dokumentů, obsahujících hypertextové odkazy a pokročilejší formátování“ [4]. Je to jazyk umožňující vytvářet dokumenty, které na sebe mohou odkazovat. Skládá se z HTML elementů, které se skládají z tagů obalených v lomených závorkách jako například tag `<body>` - značící tělo dokumentu. Tyto tagy určují, jaké bude mít dokument po přečtení webovým prohlížečem rozložení, jak bude formátován text či kde přesně budou umístěny obrázky. Základní struktura HTML se skládá z pouhých tří tagů:

1. Tag `<html>` – předává prohlížeči informaci, že se jedná o HTML dokument.

2. Tag `<head>` – obsahuje všechny informace o stránce, dále se do tohoto tagu vkládají i externí skripty a stylopisy.
3. Tag `<body>` – obsahuje všechny obsah a strukturu viditelnou na stránce.

HTML tagy se dělí na:

- Párové tagy jako je například tag pro vytváření hlavních nadpisů – `<h1></h1>`, které se skládají z otevíracího a ukončovací tagu. Mezi tyto tagy se zpravidla vždy umístí text, či další tagy.
- Nepárové tagy – `<br>` skládající se pouze z jednoho tagu.

[5]

### 2.3.1 HTML5

HTML5 je nejnovějším nástupcem tohoto hypertextového jazyka, který nahradilo starší HTML 4.01 po více jak patnáctileté odmlce. Tato verze přidala vývojářům webových aplikací možnosti jak manipulovat a tvořit grafické objekty v prohlížeči bez užití pluginu jako je Flash, či jak vkládat videa a zvuky na webové stránky. Další novinkou, která do nové verze přibyla, jsou sémantické elementy, jako je například `<header>`, `<footer>`. Tyto elementy, jsou úplně stejné jako tag `<div>`, ale pouze zvyšují přehlednost kódu tím, že určují jejich význam.

[6]

## 2.4 CSS

Kaskádový stylovací jazyk CSS (Cascading style sheets) vznikl z důvodu očištění HTML a separace stylování dokumentu od struktury dokumentu. Vzhled dokumentu se dá tedy libovolně upravovat, aniž by se upravovalo HTML. CSS navíc umožňuje mnohem větší stylizaci, než HTML kdy umělo. Umožňuje měnit barvu jakéhokoliv fontu na stránce. Dále se s jeho pomocí dají nastavovat pozadí elementům a rámce. Stylopis se může použít pro změnu vizuálního vzhledu prvků, jako jsou seznamy (tag `<ul>`) a odkazy (tag `<a>`), či pro přizpůsobení rozvržení celé stránky. Styly platné pro celý dokument se zapisují buďto přímo do dokumentu v sekci `<head>` pomocí tagu `<style>`, nebo do externího souboru s koncovkou `.css` nepárovým tagem `<link/>`.

[7, 8]



Kaskádový stylovací jazyk CSS se skládá z bloků, jímž se říká pravidlo. Ty se poté skládají ze selektoru a deklaráce. Selektor identifikuje, jaký html element se chystáme stylovat. V případě stylizace více elementů pomocí jedné deklaráce, se selektory oddělují čárkou. Deklaraci tvoří vlastnost a hodnota.

[8]

### 2.4.1 CSS3

Specifikace CSS3 se stále mění a je stále pod aktivním vývojem. Některé části specifikace se považují za stabilní a jsou správně implementované v moderních, webových prohlížečích. Jiné části jsou považovány za experimentální a jsou zavedeny pouze částečně. Ještě existují tzv. teoretické návrhy a ty nejsou implementovány vůbec. Očekávané dokončení této technologie se předpokládá na rok 2015. Mezi nové vlastnosti CSS3 patří mimo jiné:

- **text-shadow** – Přidává možnost stínovat písmo. Pozice stínu se poté určuje podle x-ové a y-ové souřadnice, jenž se zapisuje společně s barvou do hodnoty této vlastnosti.
- **border-radius** – Od počátku webu přidávali web designeři zakulacené rohy elementům nepohodlně pomocí obrázků. Díky CSS3 stačí přidat pouze výše zmíněnou vlastnost. S každým rohem HTML elementu je zacházeno jako se čtvrtinou elipsy, která je definovaná křivkou, jež je mezi x-ovou a y-ovou souřadnicí, které jsou definovány v hodnotě vlastnosti. Border-radius umožňuje zakulatit celý element najednou nebo i jednotlivé rohy zvlášť.
- **transform** – Umožňuje různě otáčet a posouvat HTML elementy. Do její hodnoty se píše, o kolik stupňů se má objekt otočit. Zde se jedná o 2D otáčení. CSS3 však zvládá dokonce i 3D otáčení, opět pomocí stejné vlastnosti.
- **animations a transitions** – Webové stránky se skládají ze tří vrstev. Z obsahové (HTML), prezentační (CSS) a vrstvy ovlivňující chování (JavaScript). Tyto vrstvy by měly být od sebe absolutně separovány. CSS, ale ne vždy toto pravidlo dodržovalo a umí i věci ovlivňující chování, jako je například selektor hover, což je událost, která nastane po najetí myši na element. Splynutí mezi těmito dvěma vrstvami je ještě větší od doby co se do CSS3 přidaly vlastnosti animations a transitions, které umožňují plynule animovat a rozpohybovat elementy, dokonce i na stránky

s vypnutým JavaScriptem. Obě vlastnosti mají podobné zápisy hodnot, liší se pouze tím, že transitions se spouští pouze během nějaké události, jako je třeba událost po najetí myši, či při změně třídy. Animace žádné takové spouštění nepotřebuje a po její definici se hned sama spustí. Pro spuštění jednoduché animace stačí v ní definovat jako hodnotu vlastnost, která se má animovat a počet sekund, jež určí, jak dlouho bude animace probíhat.

[9]

## 2.5 jQuery

Příliš složitá API (rozhraní pro programování aplikací) na straně klienta u JavaScriptu je poznamenána velkými neslučitelnostmi mezi prohlížeči. Příchod Internet Explorer 9 většinu těchto neslučitelností eliminoval, ale i přesto je pro mnoho programátorů snazší psát webové aplikace pomocí JavaScriptového frameworku, který ruší rozdíly mezi prohlížeči a usnadňuje práci při psaní skriptů – jQuery.

jQuery najde snadno elementy v dokumentu (dokonce i pomocí CSS selektorů) a pak s těmito elementy manipuluje přidáváním obsahu, editováním jejich HTML atributů a CSS vlastností, prováděním animací. Má také ajaxové nástroje pro dynamické vytváření HTTP requestů, a spoustu univerzálních užitečných funkcí pro práci s objekty a poli.

Jak naznačuje jeho název, jQuery knihovna se soustředí na dotazy (queries). Typický dotaz využívá CSS selektoru pro identifikaci daných, dokumentových elementů, a poté vrací objekt, který reprezentuje tyto elementy. Tyto vracené objekty poskytují mnoho užitečných metod pro operování s odpovídajícími elementy jako se skupinou. Kdykoliv je to možné, tak se tyto metody vrací na objekt, na který byly uplatněny, což umožňuje tzv. řetězení.

jQuery se rychle stalo jednou z nejvíce populárních, dnes používaných, JavaScriptových knihoven, díky vysoké úrovni konfigurovatelnosti a výjimečně snadné implementaci. Knihovna je využívána a podporována některými velmi známými jmény jako jsou Microsoft, Wordpress, Adobe a Intel.

[10]

### 2.5.1 jQuery UI

Uživatelské rozhraní moderních webových aplikací vyžaduje rychlý vývoj a prověřené výsledky. jQuery UI je důvěryhodná sada oficiálního pluginu pro javascriptovou knihovnu jQuery, která přidává pevný základ na jakém se dají vytvářet poutavé uživatelské rozhraní s maximální kompatibilitou, stabilitou a minimální časovou náročností. Díky tomu je množství kódu potřebného pro projekt výrazně sníženo. jQuery UI má řadu připravených widgetů, mezi které patří třeba:

- **Dialogové okna** – Na rozdíl od klasického JavaScriptového boxu volaného příkazem alert či prompt, mají dialogové okna modernější design, dají se stylovat pomocí CSS a jejich vnitřek mohou tvořit HTML tagy.
- **Tooltip** – Efektivní nápověda, jež se zobrazuje po najetí na určitý prvek v dokumentu.
- **Datepicker** – Spojený se standartním formulářovým inputem. Po kliku na input se zobrazí box s kalendářem, ve kterém si uživatel pohodlně vybere datum, podle specifikovaného formátu.

Mezi ostatní funkce jQuery patří tzv. interakce - umožňující označování, přesouvání a měnění velikosti většiny HTML elementů a dále ještě efekty, které přidávají animace na webovou aplikaci. Všechny tyto funkce se dají stahovat zvlášť, aby programátor využil jen ty, co na aplikaci potřebuje a zbytečně nestahoval celou jQuery UI knihovnu.

[11]

### 2.6 PHP

PHP (Hypertext preprocessor) je ve webovém vývoji nejpopulárnějším skriptovacím jazykem, jenž je zpracováván na straně serveru. PHP pohání odhadem 78,9% všech webových stránek. Tento jazyk byl vytvořen Rasmusem Lerdorfem v roce 1995. Nyní je jazyk řízen, kontrolován a dále vyvíjen skupinou vývojářů známých jako PHP Group, kteří pokračují v bezplatné distribuci skriptovacího jazyka skrze oficiální stránky PHP. Kód je většinou interpretován, zpracováván a renderován pomocí webového serveru s nainstalovaným PHP modulem, umožňujícím vnořování PHP mezi HTML elementy v souborech s koncovkou .php.

Hlavní předností PHP je multiplatformnost. Tento jazyk běží na Windows, Mac OS X, Linux, Solaris a i na dalších distribucích Unix. Při změně operačního systému tedy není nutné jakýmkoliv způsobem upravovat PHP soubory.

V současné době se vývoj soustředí hlavně na provádění skriptů na straně serveru, proto je obecně považován za konkurenty technologiím jako je ASP.NET od Microsoftu, či mod\_perl od Apache Software Foundation. PHP je hlavně využíváno ke komplexnímu zpracování dat, které umožňuje dynamické zobrazování dat na webových stránkách, jako jsou matematické výpočty, pracování s daty a interakce s databází. Umožňuje vývojářům proměnit běžný, statický HTML obsah a udělat z něj obsah responzivní uživatelským požadavkům.

PHP se soustředí hlavně na vývoj na webu, což z něj dělá jasnou volbu pro vývojáře, kteří vyvíjí webové aplikace a stránky. Jazyk není nikterak obtížný, a proto umožňuje vývojářům rychlý vývoj nových projektů v PHP. Zároveň díky podpoře mnoha funkcí umožňuje vývojářům rozšiřování a vylepšování aplikací bez nutnosti přecházení k jinému skriptovacímu jazyku.

[12, 13]

Díky PHP je vkládání dynamického obsahu na stránky velice jednoduchá záležitost. Po přidání koncovky .php souboru, ve kterém se má programovat, vznikne soubor, ve kterém je přístup ke skriptovacímu jazyku. Otvírací tag `<?php` sděluje webovému serveru, aby interpretoval všechny kód až do zavíracího tagu `?>`. Mimo tuto konstrukci se všechno posílá klientovi (prohlížeči) jako klasické HTML. Výsledný výstup tohoto kódu v závislosti na současném čase: Nyní je: 16 hodin.

[12]

## 2.7 MySQL

Mnoho dynamických webových stránek vyžaduje databázi. Databáze může obsahovat informace, které stránky zobrazují uživateli, či může ukládat informace poskytnutá uživatelem. V některých aplikacích databáze dokonce poskytuje jak zobrazování, tak ukládání. MySQL databáze, je nejpobulárnější databází pro vytváření webových aplikací. Byla vyvinuta za účelem být rychlá, a zároveň kapacitou malá a to je pro webové stránky velké plus. Obzvláště populární je hlavně mezi vývojáři, kteří

programují v jazyce PHP.

Software MySQL se skládá z MySQL serveru, několika pomocných programů, které asistují při správě MySQL databází a některé další podpůrné softwary, jenž server potřebuje. Server je správce celého databázového systému. Zabývá se instrukcemi vyslanými uživatelem databáze. Například při instrukci pro vytvoření nové databáze, server tuto instrukci nejprve zpracuje a vytvoří podadresář s námi zvoleným názvem ve svém datovém adresáři. Do nově vytvořeného podadresáře poté přidá soubory potřebné pro fungování databáze.

[14, 15]

### 2.7.1 Databázová struktura

Data v databázi jsou uchována v jedné nebo více tabulkách. Před přidáním nových dat, musí být nejdříve databáze a tabulky vytvořeny. Databázové tabulky jsou organizovány po řádcích a sloupcích. Každý řádek reprezentuje unikátní entitu v databázi (například zákazník, kniha nebo auto), v závislosti na druhu databáze. Každý sloupec obsahuje vlastnost entity. Místo, kde se konkrétní řádek protíná se sloupcem, se nazývá pole.

[14, 15]

### 2.7.2 Komunikace s MySQL serverem

Všechny interakce s databází jsou vytvářeny předáním instrukce MySQL serveru. Aby byla instrukce správně vykonána, musí jí MySQL server porozumět. Komunikace probíhá pomocí jazyka SQL (Structured Query Language), jenž je standardní počítačový jazyk, jemuž rozumí většina databázových systémů.

[14, 15]

SQL je tvořeno většinou z anglických slov, které dohromady tvoří řetězec připomínající anglickou větu. První slovo každého SQL dotazu značí akci, která říká MySQL, co se bude s databází provádět. Mezi nejznámější příkazy patří:

- **CREATE** – vytvoří novou tabulku.
- **DROP** – smaže tabulku.
- **ALTER** – upraví sloupec.

- **INSERT** – vloží nový záznam do tabulky.
- **SELECT** – vybere záznamy.
- **UPDATE** – upraví záznamy z tabulky.
- **DELETE** – smaže řádky z tabulky.

Poté následují slova a fráze (některá povinná a některá volitelná), jenž MySQL říká, jak akci provést.

[14, 15]

### 2.7.3 SQL injection

Při vytváření formuláře, si musí dát programátor pozor, že uživatel formuláře může kromě špatně zadaného vstupu (ty se dají ale většinou ošetřit), zadat i vstup, který může vážně ohrozit databázi webové aplikace. Takovéto technice se říká SQL injection. Útočník tak může například do inputu pro jméno zadat: „Jan; drop%20table%20uzivatele“. V případě, že formulář slouží k přidávání uživatelů, bude pak dotaz na přidávání vypadat takto: „INSERT INTO uzivatele VALUES (Jan; drop table uzivatele);“

V závislosti na konfiguraci serveru se může stát, že tento dotaz server vyhodnotí jako dva příkazy a druhým příkazem smaže celou tabulku uzivatele. Podobných typů existuje více. PHP, ale obsahuje spoustu řešení, jak tomu předejít

- Pomocí funkce `htmlspecialchars()` vyrušit všechny HTML tagy zadané do formuláře.
- Využití funkce `mysql_real_escape_string()`, která vyruší všechny speciální znaky v SQL příkazu, tím že je obalí do uvozovek, díky čemu jsou pak čteny pouze jako string.
- Použití parametrických dotazů pomocí PDO či `mysqli`.

[14, 16]

## 2.8 Bootstrap

Bootstrap je front-endový framework, který pomáhá webovým vývojářům při vývoji. Vývojářům soustředícím se hlavně na práci ve skriptovacích jazycích jako je PHP a Java pomáhá při práci s CSS a JavaScriptem, neboť jim umožňuje se během

vývoje soustředit pouze na HTML. Bootstrap obsahuje připravené CSS třídy a JavaScripty, které již mají nadefinované vlastnosti většiny objektů v HTML.

[17]

### 2.8.1 Responzivní web

Kromě dekorace odkazů a dalších HTML elementů se Bootstrap hodí i pro vytváření responzivních webů. Responzivní weby jsou weby, jejichž design funguje na jakékoli periférii, která umožňuje prohlížení webu (mobily, televize, počítače). Stránky takto designované se přizpůsobují velikosti jednotlivých HTML elementů podle velikosti obrazovky a jsou též funkční i na dotykových zařízeních. Při správném užití responzivního designu mohou vývojáři vytvořit webovou aplikaci, která může nahradit nativní aplikaci na mobilních platformách, jako je android, či iOS. V Bootstrapu se takovéto stránky dají vytvořit opět pomocí přidáním speciálních CSS tříd, určených k vytváření responzivity.

[17]

## 3 Využité programy

### 3.1 NetBeans

NetBeans je bezplatné, open source integrované vývojářské prostředí (IDE) umožňující programování ve vícero jazycích, jako jsou například: Java, CSS, HTML5, C/C++ a další. Dlouhou dobu to byl editor velkých vývojářských komunit. Díky čím dál větší oblíbenosti PHP, zahrnuje NetBeans vývojové funkce tohoto jazyka do verze 6.5 a nyní patří mezi nejpopulárnější vývojářské prostředí pro PHP komunitu.

[18]

#### 3.1.1 Funkce NetBeans

Mezi funkce NetBeans, které usnadňují každodenní práci při vývoji, patří následující:

- **Vytváření a spravování projektů:** IDE umožňuje vytváření projektů, což usnadňuje práci při jeho rozrůstání. Zvládá s tím i spojené nastavení a operace jako je vytváření dokumentací, testování projektu a další.

- **Funkce editující zdrojový kód:** Editor obsahuje spoustu vlastností zrychlující a zároveň usnadňující vývoj
  - **Zvýrazňování syntaxe PHP** v souborech projektu.
  - **Navigace** pomáhá prozkoumat metody i třídy.
  - **Formátování** poskytuje automatické formátování kódu, podle vlastního nastavení.
  - **Detekce chyb** ukazuje PHP chybu bezprostředně po dopsání kódu.
  - **Doplňování párových znamének** jako jsou uvozovky, či závorky.
- **Synchronizace projektu:** NetBeans umožňuje nahrávat i stahovat soubory projektu ze vzdáleného serveru.

[18]

### 3.2 XAMPP

XAMPP je bezplatný software, jehož cílem je co nejvíce usnadnit normálně obtížnou instalaci Apache serveru a s tím i PHP a MySQL. Po instalaci, jsou pro jednoduché fungování všechny jeho funkce automaticky zapnuté. Software, ale obsahuje i jednoduchý konfigurační nástroj, ve kterém se dají různé vlastnosti upravovat.

[19]

## 4 Wordpress

WordPress je jedním z mnoha redakčních systémů, využívající pro fungování jazyky PHP a MySQL, který umožňuje uživatelům systému užívat webové rozhraní pro správu stránky, namísto nepraktického upravování a následné nahrávání HTML souborů na server. Původně začínal jako blogovací nástroj, ale později přidali vývojáři možnost přidávat kromě klasických příspěvků (posts) i tzv. stránky (pages). To otevřelo dveře pro lidi, kteří nechtěli jen pouhý blogovací systém, ale i jednoduchý webový rozhraní pro vytváření a spravování webového obsahu.

Mezi základní funkce WordPress patří:

- **Příspěvky** – Tvoří blogovou část stránky. Většinou bývají řazeny podle data, ale mohou se řadit i podle kategorie, ve které se nachází. Jednoduchý příspěvek se skládá z jeho názvu a jeho obsahu. Kromě toho přidává WordPress ke každému



článku i důležitá data, jako je unikátní identifikační číslo článku, jméno autora, datum publikování, kategorii a viditelnost článku.

- **Stránky**- Týkají se všeho na stránkách, co není součástí blogu. Stránky jsou velice podobné příspěvkům a skládají se stejných věcí, až na kategorie. Většinou se v nich zobrazují stránky, jako jsou kontakty, či o nás.
- **Multimediální soubory** – Příspěvky a stránky umožňují vložit do obsahu článku i obrázky, zvuky, videa, PDF soubory a další.
- **Uživatelské role a profily** – Uživatelé WordPress mohou mít jednu z pěti rolí se stupňujícími pravomocemi (návštěvník, spolupracovník, redaktor, šéfredaktor a administrátor). Až na administrátora, který má přístup do všech sekcí WordPress, má každá role nějaké omezení. Například nikdo kromě administrátora nemá ve webovém rozhraní přístup ke zdrojovému kódu. Uživatelské profily obsahují mimo jiné popis uživatele, možnost změny hesla, či úpravu barevného schématu administrátorského rozhraní.
- **Automatický update** – Wordpress při zjištění nové verze systému, šablony nebo pluginu zobrazí na hlavní stránce upozornění pro administrátora.

[20]

#### 4.1 Historie WordPress

V roce 2001 byl vydána nová open source, blogovací platforma b2/cafelog, která fungovala až do roku 2003, kdy byl její vývoj zastaven. Mladý programátor, bloger a student Houstonské univerzity, Matt Mullenweg, který tuto platformu využíval, se v roce 2003 rozhodl s jejím dalším aktivním uživatelem a vývojářem, Mikem Littlem, vytvořit nový projekt za užití kódu b2/cafelog, a tím vzniknul WordPress 0.70. O rok později v roce 2004 vydal WordPress verzi 1.2 obsahující velké změny, jako je například podpora oblíbených pluginů a aplikační programové rozhraní, které z WordPress vytváří flexibilní platformu. V roce 2005 byla vyvinuta šablonová architektura a funkce pro správu stránek. Díky těmito dvěma přídatkům nemusí WordPress fungovat pouze jako blogovací platforma, ale i jako systém pro správu stránek umožňující vytvořit i náročnější projekty.

Díky snadnému používání a bezpečnosti se stal WordPress nejpopulárnějším blogovacím systémem na internetu a je používán na více než 60 miliónech stránek.

Verze WordPress jsou kromě číselného pojmenování pojmenovány i podle jmen slavných jazzových muzikantů, díky tradici započaté zakladatelem Mattem Mullenwegem, který kromě programování hraje na saxofon.

Tabulka 1. Verze WordPressu [21]

Datum	Název verze	Nové funkce
03.01. 2004	Miles Davis (1.0)	<ul style="list-style-type: none"> <li>- jednodušší instalace</li> <li>- možnost upravit URL pro lepší optimalizaci vyhledávačům</li> </ul>
22.05. 2004	Charles Mingus (1.2)	<ul style="list-style-type: none"> <li>- pluginová architektura</li> <li>- vytváření náhledových obrázků k příspěvkům</li> </ul>
26.12. 2005	Duke Ellington (2.0)	<ul style="list-style-type: none"> <li>- nahrávání obrázků a souborů do příspěvků</li> <li>- uživatelské role</li> </ul>
24.09. 2007	Dexter Gordon (2.3)	<ul style="list-style-type: none"> <li>- každý příspěvek může mít tzv. štítky, značící klíčová slova článku, která pomáhají při vyhledávání,</li> <li>- notifikace při vydání nové verze platformy</li> </ul>
17.06. 2010	Thelonius Monk (3.0)	<ul style="list-style-type: none"> <li>- vydání šablony twenty ten, ukazující většinu možností nové verze a započala tradici, ve které WordPress pokračuje dodnes</li> </ul>
17.12. 2014	Dinah Washington (4.1)	<ul style="list-style-type: none"> <li>- instalace jazykové mutace z administračního rozhraní</li> </ul>

[20, 21]

## 4.2 Loop

Loop je hlavní částí šablony ve WordPress a slouží k výpisu příspěvků a stránek do HTML podoby. Jednoduše řečeno se jedná o cyklus v PHP, který je uvnitř podmínek if a else. První podmínka pomocí funkce `have_posts()` zkontroluje, jestli v databázi existují články pro aktuální stránku (WordPress rozpozná, jaký články nebo stránky se budou vypisovat), a v případě že existují, tak je vypíše. Při každé iteraci cyklu se zavolá funkce `the_post()`, která nastaví globální proměnnou `wp_query` na další příspěvek. Pro výpis jednotlivých částí aktuálního příspěvku se pak využívají funkce zvané šablonové tagy, mezi které patří:

- **the\_title()** - Tato funkce vypíše autora příspěvku, či stránky. Do závorek se pak dají napsat nepovinné parametry pro text před názvem autora a text za názvem. Když se tedy do těchto parametrů dosadí například `the_title(<h3>, </h3>)`; zobrazí se na stránce jméno autora v nadpisu třetí úrovně.
- **the\_content()** – Vypíše celý obsah článku. V případě, že byl v editoru při psaní nastaven náhled článku, tak se zobrazí jen náhled. Podobná funkce `the_excerpt()` vypisuje automaticky zkrácenou verzi článku, bez nutnosti používat náhled.
- **the\_date()** – Funkce zobrazí datum příspěvku, dle časového formátu, jaký si zvolíme v parametru.

V případě, že loop žádný článek nenajde, tak vypíše chybovou zprávu.

[20, 21, 22]

### 4.3 Šablona

Šablony jsou kolekcí souborů, CSS stylopisů a JavaScriptu, které jsou společně provázané a vytváří grafické rozhraní sjednocující vzhled stránky. Skládá se z více php souborů, pojmenovaných přesně podle specifických pravidel, které tvoří tzv. šablonovou hierarchii. Pro fungování jednoduché šablony, stačí mít pouze soubor `index.php`. `Index.php` je také poslední možný soubor, kam WordPress automaticky přesměruje, pokud nenajde vhodnější. Některé soubory jako je například `header.php`, který se stará o výpis HTML hlavičky a počátku těla dokumentu, jsou použity na všech stránkách webu, zatímco ostatní pouze za specifických podmínek. Mezi další soubory v šablonové hierarchii patří mimo jiné soubory pro:

- **Stylopis** – Pro vznik jednoduché šablony musí být přítomný kromě `index.php` i soubor obsahující kaskádové styly – `style.css`. Do doby, než jsou obě tyto podmínky splněny, tak WordPress šablonu nerozpozná.
- **Výpis příspěvku** – Stránka zobrazená po kliku na příspěvek, obvykle složená z loopu a komentářů. Soubor se jmenuje buď `single.php`, či `single-ID.php`, kdy ID je identifikační číslo příspěvku, který chceme vypsát.
- **Výpis stránky** – Pro zobrazení statické stránky používá WordPress soubory `page.php`, `page-ID.php`. Pokud se v editoru při vytváření stránky zaškrtnla specifická šablona pro stránku, tak použije ji.

- **Výpis kategorie příspěvků** – Na této stránce jsou vypsané příspěvky dané kategorie od nejnovějšího po nejstarší. Pro jejich zobrazování musí být dostupný PHP soubor `category.php`, či `category-ID.php`.
- **Zobrazení patičky** – Vkládá se obvykle na konec každé šablony. Pro její zobrazení se volá funkce PHP `get_footer()`;
- **Funkce** – Poskytují místo pro všechny funkce, logiku a PHP třídy, které jsou potřebné pro konkrétní šablonu. Pro jejich využití je nutné vytvořit soubor `functions.php`, ten pak může udělat prakticky vše, co umí pluginy. Rozdíl mezi nimi, ale je že `function.php` je specifický pro každou šablonu a soběstačný, tudíž není potřeba jej aktivovat.

[20, 21, 23]

#### 4.4 Wordpress pluginy

Pluginy ve Wordpress jsou PHP skripty, které rozšiřují či upravují funkcionalitu Wordpress. Jednoduše řečeno, pluginy jsou soubory nainstalované do Wordpress přidávající rezervačnímu systému vlastnosti nebo sadu vlastností. Pomocí pluginů se dá vytvořit prakticky cokoliv. Ať už se jedná o jednoduchou sociální síť, či propracovaný eshopový systém. Mezi jeho hlavní výhody patří:

- **Úprava vlastností WordPress bez zásahu do jeho fungování** – Úpravou souborů potřebných pro jeho fungování totiž může vést k nestabilitě webových stránek, neboť WordPress funguje na principu, kdy jeho specifické oblasti jsou potřebné pro fungování jiných oblastí.
- **Využívání široké škály funkcí** – Další výhodou při vývoji pluginů je možnost používat funkce, které jsou ve WordPress již zabudované, jako jsou třeba uživatelské role. Díky tomu není nutné spoustu věcí programovat od začátku, což může velice snížit čas potřebný pro vývoj pluginu. Navíc používání zabudovaných funkcí zvětšuje celkovou kompatibilitu pluginu.
- **Oddělení pluginů od šablony** – Smyslem pluginů je přidávat nové funkcionality WordPress nezávisle na použité šabloně. To vytváří oddělení mezi designem stránky a jeho funkcionalitou. Důvodem této separace je, aby nebyla šablona spjata s funkcionalitou, poněvadž WordPress umožňuje kdykoliv jednoduše změnit

z administrátorského nastavení design, či šablonu webové stránky. Kdyby byl kód pluginů vložený přímo do šablony, tak by po její změně nefungoval.

- **Jednoduché updatování** – Wordpress velmi usnadňuje práci při updatování pluginů na novější verzi. U pluginů nahrávané na oficiální stránku Wordpress, systém sám rozpozná příchod nové verze a zobrazí adminovy oznámení. Pluginy nahrané na jiné stránky mohou definovat adresu, kde se plugin nachází.
- **Snadné sdílení a opětovné použití** – Sdílení pluginů je mnohem jednodušší, než modifikace kódu na jiné stránce využívající WordPress. Oblíbené pluginy se dají opětovně využít na jakémkoliv stránce s tímto redakčním systémem.
- **Plugin Sandbox** – Aktivování nefunkčního pluginu neshodí stránky, neboť WordPress automaticky plugin, který chybu spustí deaktivuje. Tato funkce velice usnadňuje práci při testování nových pluginů.

Pluginy se při nahrávání stránky s WordPressem spouštějí v rané fázi nahrávacího procesu. Jako úplně první se načte konfigurační soubor wp-config.php, ve kterém jsou uložena důležitá data jako jsou například přístupy k databázi a další. Poté se načtou WordPress funkce na kterých běží celý redakční systém, a po nich i všechny aktivní pluginy. Na závěr nahrávacího procesu se nahrávají jazykové mutace, šablona a její obsah.

[24]

#### 4.4.1 Správa a instalace pluginů

Celá správa pluginů se nachází v levém menu v administrativní části Wordpress. Tuto položku v menu uvidí pouze uživatelé s rolí administrátor.

[24]

#### 4.4.2 Instalace pluginů

Pro instalaci pluginů existují tři možnosti. První možností je instalace přímo z administrační stránky WordPress. Tato možnost umožňuje vyhledání pluginů z oficiálního úložiště pluginů. Druhou možností je stáhnutí pluginu ve formátu .zip, neboť WordPress umí takto zazipované soubory nahrát na server a extrahovat je. Třetí možností je manuální nahrání pluginu do pomoci FTP do složky wp-content/plugins.

#### 4.4.3 Správa pluginů

Správa probíhá na stránce nainstalované pluginy. Zde je vypsan seznam všech aktivních i neaktivních pluginů, které jsou ve WordPress nahrány. Pluginy se zde mohou rovnou aktivovat, či deaktivovat. Aktivovaný plugin se automaticky spustí, zatímco neaktivní se vypne, ale jeho data zůstanou. Další z možných akcí je možnost editace pluginů pomocí zabudovaného, webového editoru. Editovat lze pouze soubory, které mají nastavené oprávnění pro zapisování, jinak se zobrazí pouze kód.

[24]

#### 4.4.4 Interakce pluginů s Wordpressem

Wordpress obsahuje mnoho rozdílných API, která se dají využít při vývoji pluginu. Každé rozhraní pro programování aplikací pomáhá odlišným způsobem v interakci s Wordpressem.

[24]

##### 4.4.4.1 Plugin

Poskytuje sadu hooků (hooks), které umožňují pluginu přístup do specifických částí. Wordpress obsahuje dva odlišné typy hooků: Akční a Filtrové (z anglického action hooks a filter hooks). **Action hooks** umožňují spustit daný kód v pluginu na specifických místech stránky a za specifických událostí. Například se dá spustit vlastní naprogramovaná funkce, která proběhne pokaždé poté, co se na stránkách zaregistruje nový uživatel. Pro definování action hooku se používá funkce `add_action()`. Parametry této funkce jsou:

- **Název action hooku.** Zde může být náš libovolně pojmenovaný název pro hook, či již definovaný Wordpress hook, mezi které patří:
- **wp\_footer:** Tento hook se spustí těsně před tagem `</body>`. Je tak díky němu například možné vložit libovolný text na konec stránky.
- **get\_footer:** Hook se spustí při každém zavolání funkce `get_footer()`, který do kódu vkládá obsah patičky stránky ze souboru pojmenovaného `footer.php`.

- **template\_redirect**: Díky tomuto hooku, se funkce zavolá po každém přesměrování, které proběhne na webu. V případě, že chceme, aby se nepřihlášený uživatel nedostal na jinou stránku bez toho, aby se přihlásil, můžeme použít tento action hook.
- **plugins\_loaded**: Funkce se spustí po načtení pluginů.
- A spoustu dalších hooku, které se dají využít za libovolné události.
- **Název funkce**, která se provede zavoláním action hooku.
- **Priorita** určující pořadí action hooku se stejnou akcí
- **Počet argumentů**, které přijímá volaná funkce. Tento parametr není společně s prioritou povinný a nemusí se zadávat.

Po definování action hooku se dá poté hook zavolat, kdekoliv v pluginu, či přímo v kódu Wordpress, díky funkci `do_action()`. Její parametry jsou:

- **Název action hooku**, který chceme provádět
- **Seznam argumentů**, které se předají volané funkci

**Filtres hooks** jsou velice podobné action hooks. Na rozdíl od action ale umí vracet hodnoty, či nějaká data z volané funkce. Pro definování filtru se používá funkce `add_filter()`. Ta obsahuje úplně stejné parametry, jako action hook, jen místo názvu action, se píše název filtru, který chceme provádět. Opět si může vybrat z mnoha již Wordpressem definovaných filtrů, či nějaké nové vytvořit. Do názvu funkce se zadává funkce, která vrací nějakou hodnotu. Pro použití filtru se používá funkce `apply_filter()`. Jejimi parametry jsou:

- **Název filtru**, který chceme použít.
- **Hodnota proměnné**, jenž nám vrátí funkce volaná při definici filtru.

[21, 24]

#### 4.4.4.2 HTTP

Z anglického Hypertext transfer protocol. Jedná se o síťový protokol tvořící základ datové komunikace na internetu. Tato komunikace probíhá na bázi síťové architektury klient/server, kdy klient (zařízení s prohlížečem) posílá požadavek serveru (například Apache server), a ten poté vrací odpověď. Při nahrávání stránky `http://seznam.cz` z výsledkové stránky Googlu se provede následující požadavek:

Obrázek 2. HTTP požadavek

---

▼ **General**

**Remote Address:** [2a02:598:2::3]:443  
**Request URL:** https://www.seznam.cz/  
**Request Method:** GET  
**Status Code:** 🟢 200 OK

---

▶ **Response Headers (13)**

---

▼ **Request Headers** [view source](#)

**Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
**Accept-Encoding:** gzip, deflate, sdch  
**Accept-Language:** cs-CZ,cs;q=0.8  
**Cache-Control:** max-age=0  
**Connection:** keep-alive  
**Cookie:** \_\_gfp\_64b=vZcQ\_zRjYavykE.SyHGoYVzYiaKb3T4YqzRUR.RY3WD.o7; sznmaildomain=seznam.cz; dp=columnjs; ftxt\_geo=2,14.0809290879,49.9684813226,5756.0,weather Cookie; ftxt\_pref=OICXkGkBDaEu4jI9XZt1QcPz; hint=MTQyNDc3MzExOSwwOw==; hw=86400; hz\_amChecked=1; WeatherLocality=CESKA\_REPUBLIKA  
**Host:** www.seznam.cz  
**Referer:** https://www.google.cz/  
**User-Agent:** Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.76 Safari/537.36

Zdroj: vlastní zpracování

HTTP definuje několik metod, které se mají provést nad uvedeným dokumentem. Mezi nejznámější patří metody:

- GET – Požadavek na uvedený objekt, ke které se dají připojit pomocné proměnné, jenž se zobrazí v URL.
- POST – Tato metoda odesílá uživatelská data na server. Její nejčastější využití je při odesílání formuláře na webu, neboť se její hodnoty nezobrazují v URL, tudíž nejsou většinou citlivá data z formuláře vidět. S předaným objektem se pak zachází podobně jako při metodě GET. Data může odesílat i metoda GET, metoda POST se ale používá pro větší objem dat.



Server zpracuje daný požadavek a pošle klientovi odpověď:

Obrázek 3. HTTP odpověď

```
Status Code: 200 OK
Response Headers view source
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=UTF-8
Date: Tue, 10 Mar 2015 19:49:43 GMT
Expires: Tue, 10 Mar 2015 19:49:34 GMT
Keep-Alive: timeout=60
Pragma: no-cache
Server: nginx/1.6.0
Transfer-Encoding: chunked
Vary: Accept-Encoding
X-Frame-Options: sameorigin
X-XRDS-Location: http://id.seznam.cz/yadis
```

Zdroj: vlastní zpracování

U odpovědi je důležitou informací stavový kód, jenž by měl uveden každým serverem. Stavový kód značí, jak byla zpracována transakce serverem.

Tabulka 2. Stavové kódy [24]

Statusový kód	Význam
2xx	Požadavek byl úspěšný
3xx	Požadavek byl přesměrován jinému zdroji
4xx	Požadavek byl neúspěšný – chyba na straně klienta
5xx	Požadavek byl neúspěšný – chyba na straně serveru

V odpovědi serveru jsou také zveřejněny informace o softwaru, na kterém běží server a typu dokumentu, jenž je klientovi poslán. Pro zasílání HTTP požadavku se dají použít některé PHP funkce, mezi které patří:

- **HTTP rozšíření**
- **Funkce fopen()**
- **Standartní funkce fopen()**
- **Funkce fsockopen()**
- **CURL rozšíření**

Všechny tyto funkce mají svoje výhody a nevýhody. Některé podporují více parametrů pro lepší kontrolu, jiné zase umožňují výběr z více požadavkových metod, či se provádí rychleji. Problémem, ale je, že ne všechny tyto funkce jsou kvůli konfiguraci serveru nebo bezpečnostnímu nastavení povolené či dostupné. WordPress obsahuje třídu WP\_Http, která testuje všechny tyto funkce. Z nich poté zvolí nejlepší na aktuálním počítači. Pro zaslání HTTP požadavku ve WordPress jsou na výběr tři funkce, jejichž parametrem je url stránky:

- **wp\_remote\_get()**
- **wp\_remote\_post()**
- **wp\_remote\_head()**

Tyto funkce vracejí pole s HTTP hlavičkami, liší se jen dotazovací metodou.

[21, 24]

#### 4.4.4.3 Widgets

Toto API vytváří a spravuje widgety v pluginu. Widgety jsou obsah, který je většinou zobrazován v postranním menu, jenž se nazývá sidebar. WordPress dokonce nabízí vlastní widgety jako je například výpis kategorií s odkazy, výpis štítků k příspěvkům, navigace, či vyhledávání na stránce. Pro vytvoření vlastního widgetu je nutné jej tzv. zaregistrovat. To se provede zavoláním action hooku s parametry widgets\_init, díky kterému se tento hook spustí po registraci defaultních WordPress widgetů, a s parametrem volajícím vlastní funkci, ve které se provede registrace. Ta se provádí pomocí funkce register\_widget() a jejím parametrem je název widgetu. Poslední fází je vytvoření widgetu vytvořením nové třídy, jenž se musí jmenovat stejně, jako registrovaný widget. Tato třída pak se musí rozšířit o funkce a proměnné třídy WP\_Widget. Dovnitř této třídy se poté kromě základních informací o widgetu, programuje i samostatná funkce widgetu, tedy co bude dělat. S vytvořenými widgety, pak můžeme díky pluginům různě manipulovat. Mezi funkce tohoto API patří:

- **is\_active\_widget()** – Tato podmínková funkce zkontroluje, zda je widget aktivní a zobrazuje se na stránkách. Když je hledaný widget aktivní, tak funkce vrátí TRUE, v opačném případě FALSE.

- **the\_widget()** – Díky této funkci se dá widget zobrazit kdekoliv v šabloně, a ne pouze v sidebaru. Tuto funkci stačí jen vložit na specifické místo, kde se má nacházet. Parametrem této funkce je název třídy, ve kterém je widget registrovaný.

[21, 24]

#### 4.4.4.4 Shortcode

Shortcode je jednoduchý hook přidávající možnost používat v pluginu zkrácený kód a pomocí něj volat PHP funkce. Ten se poté může vkládat do šablony stránek a příspěvků ve tvaru [shortcode]. Výhodou shortcodu, je jejich snadné použití. Pro vytvoření takového zkráceného kódu je potřeba vytvořit libovolnou funkci například funkci, která bude vracet aktuální datum. Poté stačí zavolat action hook `add_shortcode` a do něj vložit jako první parametr název shortcodu, jenž poté vyvoláme pouhým obalením do hranatých závorek, a jako druhý parametr použijeme název funkce vracující datum.

[21, 24]

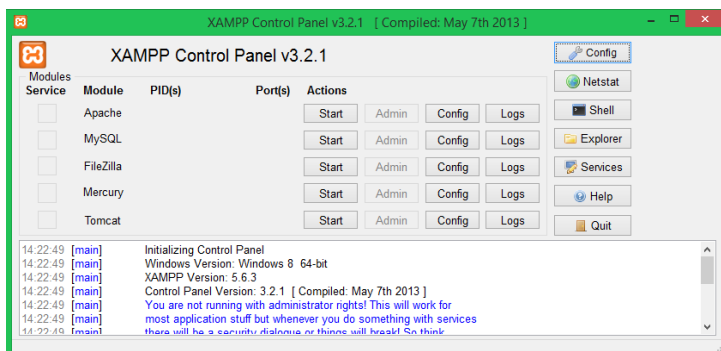
## 5 Praktická část

V této části bakalářské práce bude navržen modul pro redakční systém WordPress od úplného začátku. Modul vytvoří formulář pro rezervaci kurtů na tenisovém hřišti.

### 5.1 Webový server

Pro fungování webové aplikace, je nutné mít fungující webový server, aby bylo možné užívat PHP jazyk a MySQL databáze. Mezi ověřený a oblíbený server pro lokální vývojové prostředí patří program XAMPP. Po klasické instalaci se na obrazovce počítače objeví XAMPP Control Panel neboli software, ze kterého se dá webový server jednoduše ovládat. Pro zapnutí serveru a MySQL modulu, stačí pouze kliknout na tlačítko Start v příslušném řádku. O výsledku se dozvíme v dolní konzoli Control panelu nebo po kliknutí na tlačítko admin v modulu Apache, jenž spustí ve výchozím prohlížeči stránku localhost. Při zobrazení administrační stránky XAMPPu proběhlo vše v pořádku. Na této stránce dají kromě jiných věcí nastavit i údaje pro přístup k MySQL, které je doporučeno vyplnit, aby byly databáze co nejvíce chráněné.

Obrázek 4. XAMPP Control Panel



Zdroj: vlastní zpracování

## 5.2 Připojení do phpMyAdmin

Po kliknutí na tlačítko Admin v XAMPP Control Panelu v řádku MySQL se otevře prohlížeč na přihlašovací stránce do aplikace phpMyAdmin umožňující jednoduchou správu obsahu databáze MySQL. Pro přihlášení do systému je potřeba vyplnit přihlašovací údaje nastavené v XAMPP administraci. V této aplikaci se musí pro chod WordPressu vytvořit libovolně pojmenovaná nová databáze, ve které redakční systém při instalaci vytvoří tabulky uchovávající data pro příspěvky, uživatele a další.

## 5.3 Instalace wordpressu

Pro vytvoření nové stránky na adrese localhost, je nutné vytvořit ve výchozí složce C:\xampp\htdocs nový adresář, ve kterém bude projekt uložen. WordPress se na svých stránkách chlubí, tím že jeho instalace trvá maximálně pět minut. A je to pravda, neboť jeho instalace je velice snadná. Po stažení tohoto redakčního systému jej stačí celý vložit do nově vytvořeného adresáře. Ve WordPress adresáři se nachází textový soubor wp-config-sample.php. Jedná se o důležitý, konfigurační soubor tohoto systému. V něm je potřeba zadat pro chod WordPressu:

- Název nově vytvořené databáze, do které se nahrají nové tabulky.
- Uživatelské jméno a heslo pro přístup do MySQL.
- Hostname, jenž bude v tomto případě localhost.

Konfigurační soubor je pak nutné ještě přejmenovat na wp-config.php. Nyní stačí do prohlížeče zadat adresu ve tvaru „http://localhost/jmenoSlozky“, tím se otevře

instalační stránka WordPressu, kde se zadává pouze titulek stránky a heslo pro přihlášení do administrace. Po potvrzení formuláře se instalace WordPressu dokončí.

## 5.4 Vytvoření šablony

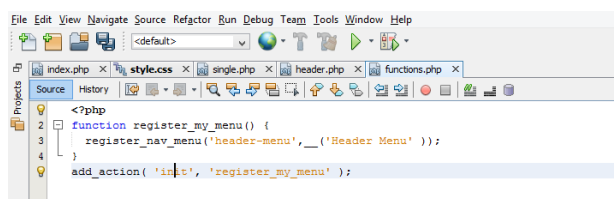
### 5.4.1 Projekt v NetBeans

Ve vývojářském prostředí NetBeans se musí vytvořit nový projekt a nastavit mu cestu do složky s projektem. V novém projektu ve složce themes se vytvoří složka s názvem nové šablony.

### 5.4.2 Úvodní stránka

Před vytvořením úvodní stránky je potřeba vytvořit soubor style.css, ve kterém budou kromě kaskádových stylů uloženy i informace o šabloně. Ty se píšou do poznámky a obsahují informace jako je název šablony i autora, popis a licenci. Po vytvoření stylů vytvoříme soubory index.php, header.php a footer.php. V souboru header se bude nacházet počátek HTML i s hlavičkou. Jedná se o soubor, který se bude vyskytovat na každé stránce šablony (spolu s patičkou), takže se něj vkládá i navigační menu. To se může naprogramovat buďto tak, že se vytvoří pomocí HTML seznam s napevno danými odkazy se stránkami, které bychom v něm chtěli mít. Tímto způsobem, ale hrozí, že při vzniku nových stránek budeme muset stále do hlavičky přidávat nové odkazy a to není moc efektivní. Lepším způsobem je vytvořit soubor function.php a v něm si pomocí krátkého kódu vytvořit navigaci, jenž se dá snadno spravovat přes administrační rozhraní WordPressu v záložce Vzhled / Menu. Menu se musí nejprve zaregistrovat funkcí register\_nav\_menu(), jejímiž parametry je název menu v kódu a název, jaký bude menu mít v administraci. Action hook s parametrem init se postará, aby se funkce, obsahující registraci navigace, spustila pokaždé po načtení redakčního systému.

Obrázek 5. Registrace menu



```
<?php
function register_my_menu() {
    register_nav_menu('header-menu', __('Header Menu' ));
}
add_action( 'init', 'register_my_menu' );
```

Zdroj: vlastní zpracování

Po registraci stačí zavolat funkci `wp_nav_menu()`, jenž se postará o zobrazení navigace. Jejím parametrem je název navigace.

Na hlavní stránce bývá obvykle úvodní článek popisující nejdůležitější detaily a výpis novinek. Pro úvodní článek stačí v administraci vytvořit novou stránku. Novinkám se zase vytvoří specifická kategorie, která bude obsahovat příspěvky s novinkami. Článek i novinky se poté vypíší pomocí loopů. Na začátku obou vypisovacích cyklů, se ale musí určit pomocí instance třídy `WP_Query()`, jaké příspěvky, či články se budou vypisovat.

Obrázek 6. Loop

```
// Úvodní článek (Loop pro stránky)
$uvodniStrankaId = new WP_Query("page_id=13");
while($uvodniStrankaId->have_posts())
{
    $uvodniStrankaId->the_post();
    the_title( '<h1>', '</h1>' );
    the_content();
}
?>
```

Zdroj: vlastní zpracování

Stránka poté může po na stylování, užití Bootstrapu a různých skriptu vypadat takto:

Obrázek 7. Úvodní stránka



Zdroj: vlastní zpracování

## 5.5 Vytvoření pluginu

Nejprve je nutné vytvořit v adresáři plugins novou složku, ve které budou soubory a složky pluginu. Aby WordPress nový plugin rozpoznal musejí být v hlavním PHP souboru pluginu podobné informace, jako u souboru style.css při vytváření šablony, čili například název pluginu, jeho popis a autor. Po přidání informací se plugin objeví v administraci WordPressu v záložce Pluginy, ze které se dá již aktivovat.

Jednotlivé rezervace je potřeba někam ukládat, proto se musí nastavit, aby se po aktivaci pluginu vytvořila v databázi nová tabulka pro rezervace. To se provede pomocí `register_activation_hook()`, jenž po aktivaci spustí funkci uvedenou jako parametr. V uvedené funkci je použit SQL příkaz `CREATE TABLE`, který vytvoří tabulku se zadanými sloupci. Aby příkaz proběhnul, musí se vložit do jako parametr funkci `dbDelta()`, jenž se stará o vytváření a updatování tabulek.

Obrázek 8. Vytvoření tabulky

```
function instalaceTabulky() {
    global $wpdb;
    global $verzeRezervacniTabulky;

    $table_name = $wpdb->prefix . 'rezervace';

    $charset_collate = $wpdb->get_charset_collate(); // kódování

    $sql = "CREATE TABLE $table_name (
        id mediumint(9) NOT NULL AUTO_INCREMENT,
        jmeno varchar(55) DEFAULT '' NOT NULL,
        cisloKurtu mediumint(9) NOT NULL,
        datumRezervace datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
        datum datetime DEFAULT '0000-00-00 00:00:00' NOT NULL,
        UNIQUE KEY id (id)
    ) $charset_collate:";

    require_once( ABSPATH . 'wp-admin/includes/upgrade.php' ); // voláme tento soubor pro funkci dbDelta
    dbDelta($sql);
}

register_activation_hook( __FILE__, 'instalaceTabulky' ); // action hook, co se spusti při aktivaci pluginu
```

Zdroj: vlastní zpracování

Výpis rezervačního systému obstará funkce `vypis RezervacniSystem()`. V ní se nacházejí funkce `wp_enqueue_style` a `wp_enqueue_script`, které přidají stylopisy a skripty pro úpravu designu a funkcionality. Soubory se, ale nejprve musí zaregistrovat, aby je WordPress rozpoznal a zjistil jejich umístění. Toto se provede pomocí funkce `wp_register_style()` a `wp_register_script()`. Kód pro vypsání kalendáře na výběr dne a rezervačního formuláře je umístěn v souborech `casovaNavigace.php` a `rezervačniSystem.php`.

Obrázek 9. Funkce pro výpis rezervace

```
function vypisRezervacniSystem() {
    wp_enqueue_style("rezervacniSystemCSS"); // css pluginu
    wp_enqueue_style("jqueryUICSS"); // jquery ui css

    wp_enqueue_script("jqueryUIJS"); // jquery ui js - pro dialog
    wp_enqueue_script("rezervacniSystemJS"); // js pluginu

    include('casovaNavigace.php');
    include('rezervacniSystem.php');
}

wp_register_style("rezervacniSystemCSS", plugins_url("rezervacniSystem/rs.css"));
wp_register_style("jqueryUICSS", plugins_url("rezervacniSystem/jquery-ui.min.css"));
wp_register_script("jqueryUIJS", plugins_url("rezervacniSystem/jquery-ui.js"));
wp_register_script("rezervacniSystemJS", plugins_url("rezervacniSystem/rs.js"));
```

Zdroj: vlastní zpracování

### 5.5.1 Výběr dne

Obrázek 10. Kalendář



Zdroj: vlastní zpracování

Kalendář bude tvořen HTML seznamem, ve kterém budou odkazy na jednotlivé dny, boxem zobrazujícím datum vybraného dne a legendou pro formulář. Pro vytvoření takového kalendáře se musí v PHP vytvořit pole s jednotlivými názvy dnů. Dále je potřeba, aby aplikace věděla co je zrovna za den a jaký den je zrovna vybraný. Aktuální datum se dá zjistit pomocí funkce `date(Y-m-d)`. Při vložení funkce `date()` do funkce `date_create()` vytvoří PHP objekt třídy `DateTime`, jenž umožní s datem různě manipulovat (přičítat k němu dny, vypisovat v libovolném tvaru). Vybrané datum se zjistí z URL parametru, uloženým v poli s názvem `$_GET`. Pokud pole neexistuje, znamená to, že žádný den nebyl vybrán, takže se jako vybrané datum nastaví dnešní datum.

Obrázek 11. Výpis datumů

```
if(date_format($vybranyDen,"Y-m-d") == date_format($datum,"Y-m-d"))
    echo '<a class="dnesek vybranyDen" href="http://jan-jahoda-bakalarka.8u.cz/?page_id=6&datum='.date_format($datum,"Y-m-d").'>';
else
    echo '<a class="dnesek" href="http://jan-jahoda-bakalarka.8u.cz/?page_id=6&datum='.date_format($datum,"Y-m-d").'>';
```

Zdroj: vlastní zpracování

Kalendář se vytvoří pomocí cyklu `while`, kdy v každém cyklu se vypíše jeden box s datumem. Odkazu se zrovna vybraným datum se přidá třída `vybranyDen`, aby se poté



pomocí CSS zvýraznil. Datum v cyklu se na jeho na konci pokaždé zvětší o jeden den pomocí funkce `date_add()`.

### 5.5.2 Výpis formuláře

Formulář pro výběr hodiny se dá naprogramovat mnoha způsoby. Jedním ze způsobů je vytvoření HTML tabulky, ve které bude každá buňka představovat určitou hodinu a název kurtu. Tyto data budou uvedena ve skrytém inputu uvnitř buňky v jeho atributu `name`. Aby se poznalo, která buňka byla zaškrtnuta po odeslání formuláře, musí se nastavit skrytým inputům do atributu `value` jejich stav určující, jestli je daný termín volný, plný či zrovna zaškrtnutý. Pro nastavení hodnoty `value` na stav zaškrtnuto při kliknutí na buňku je nutno vytvořit jQuery skript.

Obrázek 12. jQuery - událost při kliknutí

```
jQuery('.zaskrtavacíBunky').click(function() {  
  if (!jQuery(this).hasClass("cervena")) {  
    $$(".zaskrtavacíBunky").each(function() {  
      if(jQuery("input", this).val() === "zaskrtnuto") {  
        // Nastavíme hodnotu jako volny  
        jQuery("input", this).val("volny");  
        jQuery(this).removeClass("svetleZelena");  
        jQuery(this).addClass("bila");  
        jQuery(this).prevAll(".cas").css("background-color", "transparent");  
      }  
    });  
    jQuery(this).removeClass("bila");  
    jQuery(this).addClass("svetleZelena");  
    // Nastavíme hodnotu jako vyplněnou  
    jQuery("input", this).val("zaskrtnuto");  
  }  
});
```

Zdroj: vlastní zpracování

Funkce proběhne při kliknutí na buňku tabulky. Nejprve je nutné zkontrolovat, jestli se jedná o volný termín, aby nešlo rezervovat, již rezervované termíny. K tomu pomůže jQuery funkce `hasClass()`, jenž zkontroluje HTML element jestli nemá třídu, kterou mají rezervované termíny. Rezervační systém bude umožňovat výběr pouze jednoho termínu za jednu objednávku, toho se docílí pomocí funkce `each()`. Ta po každém kliknutí projede všechny buňky v tabulce a v případě, že se najde buňka jejíž skrytý input bude mít hodnotu `zaskrtnuto`, tak se nastaví funkcí `val()` na `volny`.

U zaškrtnuté buňky se musí změnit i její třída, aby se změnila její barva a bylo poznat, že již není označená. Nejdříve se odstraní třída jakou mají zaškrtnuté buňky funkcí `removeClass()`, a poté se funkcí `addClass()` přidá třída, kterou mají buňky nezabraných termínů. Po skončení tohoto cyklu se nastaví nově zaškrtnuté buňce třída a jejímu skrytému inputu hodnota s odpovídajícími hodnotami.

Obrázek 13. Vybrání rezervovaných termínů

```
global $wpdb;
$poleRezervaci = array();
$i = 0;

$vyberRezervace = $wpdb->get_results("
    . "SELECT jmeno, DATE_FORMAT(datumRezervace, '%Y-%m-%d %H:%i') as datum, "
    . "cisloKurtu FROM wp_rezervace "
    . "WHERE DATE_FORMAT(datumRezervace, '%Y-%m-%d') = '".date_format($vybranyDen, "Y-m-d")."'");
foreach($vyberRezervace as $rezervace)
{
    $poleRezervaci[$i]["jmeno"] = $rezervace->jmeno;
    $poleRezervaci[$i]["datum"] = $rezervace->datum;
    $poleRezervaci[$i]["cisloKurtu"] = $rezervace->cisloKurtu;
    $i++;
}
```

Zdroj: vlastní zpracování

Aby se zjistili rezervované termíny, musí se vytvořit globální proměnná `$wpdb`, jenž je objektem třídy starající se o práci s databází ve WordPressu. Příkazem `$wpdb->get_results()` se z databáze vyberou všechny řádky z tabulky `wp_rezervace`, které mají jako datum rezervace uvedeno vybrané datum ve formuláři.

Obrázek 14. Inputy

```
<div id="input">
  <div class="form-group">
    <label for="jmeno">Jméno: (povinný údaj)</label>
    <input type="text" class="form-control" id="jmeno" name="jmeno" required="">
  </div>
  <div class="form-group">
    <label for="email">Email: </label>
    <input type="email" class="form-control" id="email" name="email">
  </div>
  <button type="submit" class="btn btn-default">Rezervovat</button>
</div>
```

Zdroj: vlastní zpracování

Za tabulkou s termíny budou ještě inputy pro jméno rezervujícího, email a tlačítko pro odeslání formuláře. První input se musí vyplnit, aby se vědělo, kdo si danou hodinu rezervoval. To se nastaví přidáním atributu `required` do požadovaného inputu.

Obrázek 15. jQuery - dialogové okno

```
// Zkontrolujeme jestli, byla zaškrtnuta hodina
jQuery("#form").submit(function (e) {
    var zaskrtnuto = 0;

    jQuery("#zaskrtavaciBunky").each(function() {
        if(jQuery("input", this).val() === "zaskrtnuto") {
            zaskrtnuto = 1;
        }
    });

    if(zaskrtnuto === 1) {
        jQuery("#form").submit();
    }
    else {
        e.preventDefault(); //prevent default form submit
        jQuery("#dialog2").dialog({
            modal: true,
            buttons: {
                Ok: function() {
                    jQuery( this ).dialog( "close" );
                }
            }
        });
    }
});
});
```

Zdroj: vlastní zpracování

V souboru rezervační System.php budou také ještě dvě dialogová okna o jejichž chod se stará jQuery UI. První se objeví pokaždé co proběhne rezervace. Druhé dialogové okno se zobrazí, když se uživatel pokusí odeslat formulář bez nezaškrtnutého datumu. To se naprogramovat pomocí jQuery funkcí submit(), která se provádí při odesílání formuláře. Uvnitř této funkce se provádí cyklus each(), jenž prohledává skryté inputy a hledá v nich input s hodnotou zaskrtnuto a při jeho nalezení se formulář odešle. V opačném případě se odesílání přeruší, díky funkci e.preventDefault() a zobrazí se dialogové okno s chybovou hláškou.

### 5.5.3 Uložení rezervace

Plugin se o odeslání formuláře dozví díky action hooku s parametrem init a názvem funkce, která se má zavolat.

Obrázek 16. Funkce pro ukládání rezervace

```
function ulozRezervaci() {
    if(isset($_POST["jmeno"])) {
        $jmeno = mysql_real_escape_string($_POST["jmeno"]);
        if(isset($_POST["email"]))
            $email = mysql_real_escape_string($_POST["email"]);

        unset($_POST["jmeno"]);
        unset($_POST["email"]);

        // Najdeme zaškrtnuté políčko
        $nazev = array_search('zaskrtnuto', $_POST);

        $cut = explode("_", $nazev);
        $cisloKurtu = $cut[0];
        $datumRezervace = $cut[1] . " " . $cut[2];

        // Uložíme rezervaci do databáze
        vlozRadekDoTabulky($jmeno, $cisloKurtu, $datumRezervace);

        // V případě vyplnění emailu, odešleme zprávu
        if(isset($email))
            odesliEmail($email, $datumRezervace, $cisloKurtu);

        wp_redirect("http://jan-jahoda-bakalarka.eu.cz/?page_id=6&datum=$cut[1]&hotovovano", 301); exit;
    }
}
```

Zdroj: vlastní zpracování

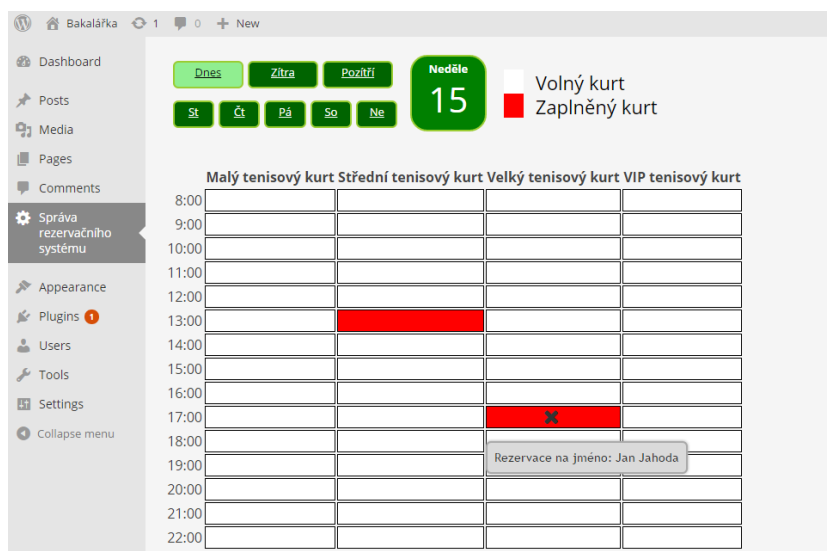
V zavolané funkci se za užití funkce mysql\_real\_escape\_string() bezpečně uloží data z formuláře. Další informací, kterou zbývá zjistit je zaškrtnutý termín. Spolu s ním se totiž v poli \$\_POST odeslaly i ostatní skryté inputy, jen jeden z nich ale má hodnotu zaskrtnuto. Tento input se najde díky funkci array\_serach(), jenž prohledá celé pole \$\_POST a vrátí hodnotu name, ve které je uložený datum a číslo kurtu.

Při vyplnění nepovinného inputu pro email, se zavolá funkce odesliEmail. V ní se uživateli odešle email, pomocí wp\_mail() s parametry určujícími příjemce, odesílatele, obsah emailu a předmět.

Zaškrtnutý termín se vloží do databáze pomocí funkce `$wpdb->insert`. Jejími parametry jsou název tabulky a pole obsahující v indexu název sloupce a jeho hodnotu. Na konci funkce ukládající rezervaci se provede přesměrování zpátky na rezervační systém.

## 5.5.4 Administrativní rozhraní pluginu

Obrázek 17. Rezervace - administrativní rozhraní



Zdroj: vlastní zpracování

V administrativní rozhraní pluginu je zobrazen stejný formulář, jako na stránce rezervace. Správce stránek si zde může pohodlně zjistit kdo a v kterou dobu zrovna hraje na daném kurtu, či smazat smazat zrušenou rezervaci. Pro výpis datumů a formuláře je použit stejný postup jako u rezervační stránky, jen jsou zde přidány jiné jQuery funkce. Po najetí na jednotlivé rezervované termíny se objeví tzv. tooltip, který zobrazí hodnotu, jenž má buňka v atributu title. Pro jejich zapnutí stačí v souboru s JavaScriptem přidat řádek `jQuery(document).tooltip()`;

Obrázek 18. Smazání rezervace

```
function smazRezervaci ($cisloKurtu, $datumRezervace) {
    global $wpdb;
    $wpdb->show_errors();
    $table_name = $wpdb->prefix . 'rezervace';

    $wpdb->delete($table_name, array('
        . 'datumRezervace' => $datumRezervace, '
        . 'cisloKurtu' => $cisloKurtu));
}
```

Zdroj: vlastní zpracování

Při kliknutí na rezervovaný termín vyskočí dialogové okno s potvrzením, že se má rezervace opravdu smazat. Při volbě Ano se skrytému inputu ve formuláři s názvem smazMe nastaví hodnota, kterou má skrytý input v buňce na kterou se kliklo. Poté se odešle formulář. Plugin se o odeslání formuláře dozví opět díky action hooku s parametrem init a funkcí ve které proběhne smazání řádku. V ní se provede funkce \$wpdb->delete, která smaže řádek jehož hodnoty sloupců se rovnají hodnotám uvedeným v parametru.

## 6 Závěr

Tato bakalářská práce se zaměřuje na návrh pluginu v redakčním systému WordPress.

V první části práce jsou popisovány webové aplikace, mezi které patří rezervační systémy. Jsou zde uvedeny jejich druhy a prohlížeče na kterých běží. Představeny jsou jednotlivé počítačové jazyky jako je například jazyk HTML, starající se o rozvržení stránky pomocí tagů, dále skriptovací jazyk PHP, jenž dokáže z jednoduchých statických stránek vytvořit dynamickou webovou aplikaci či JavaScriptový jazyk jQuery a jeho rozšíření jQuery UI, starající se o chování a vzhled stránky.

Další část teoretické práce pojednává o programech, potřebných pro naprogramování pluginu v redakčním systému. Jedná se o program XAMPP, jenž nainstaluje na počítač webový server, umožňující vytvářet databáze a programovat v PHP jazyce. Dále je představeno vývojářské prostředí NetBeans, které svými funkcemi usnadňuje práci při vývoji webových aplikací.

Poslední část teoretické práce se věnuje redakčnímu systému WordPress. Nejdříve je nastíněna krátká historie tohoto systému, poté následuje přehled základních věcí, týkající se šablon a souborů, z kterých se skládá. Dále je zde popsána funkce Loop, jejímž úkolem je vypisovat příspěvky a články. Zbytek části se zabývá hlavním tématem bakalářské práce – pluginy. Jsou zde vysvětleny jejich základy, správa a instalace. Na závěr jsou vysvětleny jednotlivá aplikační rozhraní pluginů, společně s jejich funkcemi.

Praktická část se zaměřuje na návrh pluginu pro redakční systém WordPress. Konkrétně se jedná o rezervační systém pro tenisový klub. V této části je popsán celý postup pro vytvoření takového pluginu. Nejprve se práce zabývá instalací webového serveru a následně instalací WordPressu. V další části praktické práce je ukázán postup pro vytvoření stránky s navigací a její šablony. Na závěr je ukázán detailní postup pro vytvoření pluginu, jenž je rozdělen do čtyř sekcí, kdy každá popisuje vytváření jednotlivých částí pluginu.

Cíl bakalářské práce byl úspěšně splněn a podařilo se vytvořit WordPress plugin pro rezervaci tenisových kurtů. V bakalářské práci byly uvedeny jednotlivé postupy pro zhotovení pluginu, díky kterým by měl programátor se základní znalostí programování webových aplikací zvládnout vytvořit vlastní plugin sám.

## 7 Seznam použitých zdrojů

### 7.1 Seznam literatury

1. STUTTARD, Dafydd a Marcus PINTO. *The web application hacker's handbook: discovering and exploiting security flaws*. Indianapolis, IN: Wiley Pub., c2008, xxxii, 736 p. ISBN 9780470170779.
2. NIXON, Robin. *Learning PHP, MySQL, JavaScript, CSS*. Third edition. xxv, 700 pages. ISBN 14-919-4946-5
3. Web browser (browser). BEAL, Vangie. *Webopedia* [online]. [cit. 2015-03-11]. Dostupné z: <http://www.webopedia.com/TERM/B/browser.html>
4. PÍSEK, Slavoj. *HTML: začínáme programovat*. 3., aktualiz. vyd. [i.e.] 1. vyd. Praha: Grada, 2010, 190 s. ISBN 978-80-247-3117-9.
5. ROBSON, Elisabeth a Eric FREEMAN. *Head first HTML with CSS*. 1st ed. Sebastopol, CA: O'Reilly, c2006, xxxv, 658 p. ISBN 05-961-0197-X.
6. GUSTAFSON, J. *HTML5 web application development by example: beginner's guide*. Birmingham: Packt Pub., 2013, vi, 260 p. ISBN 978-1-84969-594-7.
7. PÍSEK, Slavoj a Jasmin SCHMIDT. *HTML: začínáme programovat*. 4., aktualiz. vyd. Praha: Grada, 2014, 181 s. Průvodce (Grada). ISBN 978-80-247-5059-0.
8. HARRIS, Andy. *Html5 and css3 all-in-one for dummies, 3rd edition*. 3rd edition. pages cm. ISBN 11-182-8938-2
9. GASSTON, Peter. *The book of CSS3: a developer's guide to the future of web design*. Second edition. San Francisco: No Starch Press, [2015], 1 online zdroj (306 pages). ISBN 978-1-59327-643-0.
10. FLANAGAN, David. O'REILLY MEDIA. *JQuery Pocket Reference: Read Less, Learn More*. 2010. vyd. ISBN 978-1-4493-9722-7
11. LIBBY, Alex a Dan WELLMAN. *JQuery UI 1.10: the user Interface library for jQuery*. Second edition. Birmingham: Packt Publishing, 2013, 1 online zdroj (502 pages). ISBN 978-1-78216-221-6.
12. NIXON, Robin. *Learning PHP, MySQL, JavaScript, CSS*. Third edition. xxv, 700 pages. ISBN 14-919-4946-5
13. HOPKINS, By Callum a Dan WELLMAN. *Jump start PHP: the user Interface library for jQuery*. Second edition. Collingwood, Australia: SitePoint, 2013, 1 online zdroj (502 pages). ISBN 978-098-7467-409.

14. SUEHRING, Steve a Janet VALADE. *PHP, MySQL, JavaScript: the user Interface library for jQuery*. Second edition. Collingwood, Australia: SitePoint, 2013, xxiv, 692 pages. --For dummies. ISBN 978-1-118-21370-4.
15. NIXON, Robin a Janet VALADE. *Learning PHP, MySQL, and JavaScript: the user Interface library for jQuery*. 1st ed. Sebastopol, CA: O'Reilly, c2009, xvii, 505 p. --For dummies. ISBN 978-059-6157-135.
16. PHP: mysql\_real\_escape\_string. In: [online]. [cit. 2015-03-12]. Dostupné z: <http://php.net/manual/en/function.mysql-real-escape-string.php>
17. *Jump start bootstrap*. S.l.: O'Reilly Media, 2014. ISBN 978-099-2279-431.
18. HOSSAIN TONU, M. *PHP application development with NetBeans beginner's guide*. Birmingham: Packt Publishing, 2012, vi, 283 p. ISBN 978-1-8495-1580-1.
19. About. [online]. [cit. 2015-03-12]. Dostupné z: <https://www.apachefriends.org/about.html>
20. BRAZELL, Aaron. *WordPress bible*. Indianapolis, IN: Wiley Pub., c2010, xxvii, 644 p. ISBN 04-705-6813-5.
21. *Codex* [online]. [cit. 2015-03-12]. Dostupné z: <http://codex.wordpress.org/>
22. MESSENLEHNER, Brian a Jason COLEMAN. *Building web apps with WordPress*. First edition. xxi, 435 pages. ISBN 978-144-9364-076.
23. SILVER, Tessa Blakeley a Jason COLEMAN. *WordPress 3.2 theme design: beginner's guide*. First edition. Birmingham: Packt Publishing, 2013, 545 p. ISBN 9781849514224.
24. WILLIAMS, Brad, Ozh RICHARD a Justin TADLOCK. *Professional WordPress plugin development: beginner's guide*. First edition. Indianapolis, IN: Wiley Pub., c2011, xxvii, 524 p. Wrox professional guides. ISBN 04-709-1622-2.

## 8 Přílohy

### 8.1 Obsah CD

- /wp-content/themes/bakalarka – Kompletní šablona pro stránky tenisového hřiště
- /wp-content/plugins/rezervacniSystem – Plugin pro rezervaci hřišť