

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**REDAKČNÍ SYSTÉMY WORDPRESS, DRUPAL
A JOOMLA Z POHLEDU TVORBY ŠABLON**
BAKALÁŘSKÁ PRÁCE

Miroslav Zíka

Informatika se zaměřením na vzdělávání

Vedoucí práce: PhDr. Tomáš Jakeš, Ph.D.

Plzeň, 2017

Prohlašuji, že bakalářskou práci jsem vypracoval samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 1. června 2017

.....
vlastnoruční podpis

Poděkování

Rád bych poděkoval PhDr. Tomáši Jakeši, Ph. D. za jeho odborné vedení, rady a názory, které vedly ke zdárnému vypracování bakalářské práce.

ZDE SE NACHÁZÍ ORIGINÁL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

SEZNAM POUŽITÝCH ZKRATEK.....	3
ÚVOD.....	4
1 CÍLE	6
2 HTML.....	7
2.1 HTML5 A JEHO VÝVOJ.....	7
2.2 NOVÉ PRVKY JAZYKA HTML5	8
2.2.1 Nav	8
2.2.2 Article, Section a Aside	9
2.2.3 Header a Footer	9
3 CSS3.....	10
3.1 PROPRIETÁRNÍ VLASTNOSTI.....	10
3.2 NOVÉ VLASTNOSTI CSS3	11
3.2.1 Border Radius.....	11
3.2.2 RGBA.....	11
3.2.3 Transition.....	11
3.2.4 Transform	11
3.2.5 @media	12
4 RESPONZIBILITA	13
4.1 BOOTSTRAP.....	14
4.1.1 Struktura gridů	14
4.2 W3.CSS.....	15
4.3 DOTAZY NA MÉDIA.....	17
4.3.1 Více CSS souborů.....	18
4.3.2 Jeden CSS soubor	18
4.4 VIEWPORT	18
5 DESIGN ŠABLONY	19
5.1 KRITÉRIA TVORBY NÁVRHU.....	19
5.2 REALIZACE NÁVRHU.....	21
5.2.1 Hlavička.....	21
5.2.2 Obsah.....	25
5.2.3 Patička	26
5.3 TESTY RESPONZIBILITY.....	26
6 WORDPRESS.....	28
6.1 ZÍSKÁVÁNÍ ŠABLON.....	28
6.2 MOŽNOSTI EDITACE A PŘIZPŮBENÍ	29
6.2.1 Přímá editace souborů šablony	29
6.2.2 Přizpůsobení šablony.....	30
6.3 TVORBA VLASTNÍ ŠABLONY.....	31
6.3.1 Struktura souborů šablony	31
6.3.2 Hlavní soubor index.php.....	31
6.3.3 Header.php	32
6.3.4 Footer.php	35
6.3.5 Functions.php	36
6.3.6 Search.php	37
6.3.7 Ostatní soubory šablony.....	37
7 JOOMLA.....	38

7.1	ZÍSKÁVÁNÍ ŠABLON.....	38
7.2	MOŽNOSTI EDITACE A PŘIZPŮBENÍ	38
7.3	TVORBA VLASTNÍ ŠABLONY.....	39
7.3.1	Definující soubor templateDetails.xml	40
7.3.2	Hlavní soubor index.php.....	40
7.3.3	Error.php a offline.php	41
7.3.4	Position.....	42
7.3.5	Ostatní soubory šablony.....	46
8	DRUPAL	47
8.1	ZÍSKÁVÁNÍ ŠABLON.....	47
8.2	MOŽNOSTI EDITACE A PŘIZPŮBENÍ	47
8.3	TVORBA VLASTNÍ ŠABLONY Z NÁVRHU	48
8.3.1	Definující soubor .info.yml.....	48
8.3.2	Regions	49
8.3.3	Libraries a soubor .libraries.yml.....	49
8.3.4	Templates	50
8.3.5	Jazykové verze	55
8.3.6	Ostatní soubory šablony.....	56
	ZÁVĚR	57
	RESUMÉ.....	59
	SEZNAM LITERATURY.....	60
	SEZNAM OBRÁZKŮ.....	62
	PŘÍLOHY.....	I

SEZNAM POUŽITÝCH ZKRATEK

CMS – **C**ontent **M**anagement **S**ystem, Systém pro správu obsahu.

PHP – Hypertext Preprocessor (**P**ersonal **H**ome **P**age), skriptovací programovací jazyk vykonávající se na straně serveru.

HTML – **H**ypertext **M**arkup **L**anguage, značkovací jazyk pro tvorbu webových stránek.

CSS – **C**ascading **S**tyle **S**heets, jazyk kaskádových stylů.

JS – **J**ava**S**cript, skriptovací programovací jazyk vykonávající se na straně klienta.

WYSIWYG – „**W**hat **y**ou **s**ee **i**s **w**hat **y**ou **g**et“, „Co vidíš, to dostaneš“. Editory sloužící k vytváření webových stránek. Uživatel pomocí nich vytváří pouze vizuální podobu webu, strukturu kódu zajišťuje program.

Úvod

V minulosti, v době vzniku prvních propojených počítačových sítí, byly webové stránky zpočátku doménou pouze nadnárodních celků a vládních organizací. Správu těchto internetových stránek zajišťovali školení experti. V době vývoje jazyka HTML nebylo z technického hlediska možné celosvětové sdílení informací, a tudíž bylo velice komplikované potřebné vědomostmi studovat. Jak pokračoval tento rozvoj, a rozrůstala se celosvětová síť, tak se principy rozšířily k více lidem. Bohužel i v tento okamžik bylo vytvoření a následná správa stránek pro běžného uživatele náročná. Internetové stránky totiž mohou kombinovat vědomosti z více technologií, např. HTML, CSS, JS, PHP a relační databáze, a proto jejich úplné nastudování zabere tedy mnoho času.

Postupem času byly naprogramovány systémy umožňující jednodušší tvorbu a správu webových stránek. Tím pádem se svět webu otevřel pro širší veřejnost, která za pomoci těchto programů mohla jednodušeji zveřejňovat svůj obsah. A to i bez dříve zmíněných hlubších znalostí. Tyto programy, označované jako redakční systémy, či systémy pro správu obsahu, se dočkaly vysokého využití a jejich funkce byly dále rozšiřovány o další doplňky. Zpočátku se jednalo o graficky jednoduché webové stránky. Vývoj používaných technologií ovšem umožnil stále komplikovanější grafické designy, které mohl administrátor v případě potřeby upravovat či vytvářet svůj vlastní. Webové stránky se stávaly více dynamický a vzrůstal počet vzhledů, pomocí nichž bylo možno své stránky odlišit od jiných. Práce pojednává o postupu tvorby vlastní šablony pro nejvyužívanější redakční systémy Wordpress, Drupal a Joomla.

První kapitola práce bude obsahovat shrnuté cíle práce, neboť praktická část se skládá z více dílčích úloh. V následujících dvou kapitolách bude čtenář seznámen se základními technologiemi sloužícími ke tvorbě šablon – o HTML5 a CSS3. Ve stručnosti zde bude pojednáno o hlavních změnách oproti dřívějším verzím a také budou popsány některé nové prvky. Ve čtvrté kapitole budou představeny způsoby zajišťující responzivní chování stránek, aby jejich obsah byl dobře čitelný na co největší škále zařízení.

V páté kapitole budou shrnuty kritéria pro grafický návrh. Čtenáři bude představen vytvořený grafický design, který bude sloužit při tvorbě vzorové webové stránky. Části této stránky budou poté popsány, neboť poslouží při tvorbě šablon.

V posledních třech kapitolách budou představeny jednotlivé redakční systémy WordPress, Joomla a Drupal. Budou zde uvedeny některé zdroje šablon a možnosti jejich editace přes administrátorské rozhraní. Na závěr bude popsán postup tvorby vlastní šablony na základě vytvořené vzorové stránky.

1 CÍLE

Cílem práce je na příkladu vývoje vlastní šablony, s předem požadovaným designem, představit základní potřebné pojmy, východiska a vhodný postup tvorby validních a responzibilních šablon pro redakční systémy Wordpress, Drupal a Joomla.

Úkony spjaté s návrhem designu

- Provést analýzu současných stránek katedry Výpočetní a didaktické výchovy.
- Vytvořit grafické návrhy na základě analýzy.
- Prodiskutovat vhodnost jednotlivých grafických návrhů.
- Zpracovat finální grafický design na základě konzultací.

Tvorba webové stránky a šablon

- Nakódovat webovou stránku na základě designu.
- Prostudovat způsoby zajišťující responzibilitu stránek a jeden implementovat.
 - Přizpůsobit stránku pro full HD zařízení.
 - Přizpůsobit stránku pro tablety a mobilní zařízení.
- Zkontrolovat funkčnost webové stránky a popřípadě opravit chyby.
- Z webové stránky realizovat šablony pro jednotlivé redakční systémy.

2 HTML

Webové stránky lze vytvářet mnoha způsoby. Mezi neznámější možnosti patří tzv. WYSIWYG editory či přímý zápis struktury stránek pomocí speciálních formátovacích značek. Základním prvkem je ovšem kódovací jazyk HTML.

2.1 HTML5 A JEHO VÝVOJ

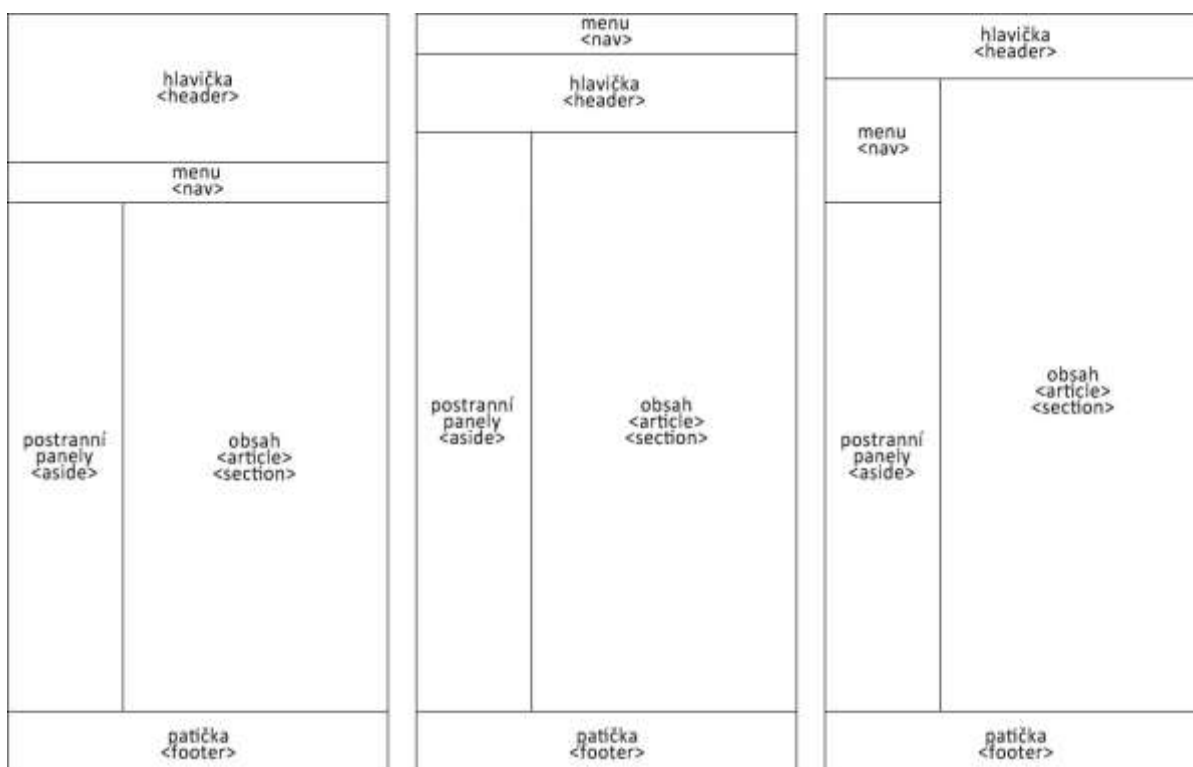
Jazyk HTML prodělal od roku 1991 řadu úprav. Jeho současná verze nese název HTML5 a navrácí se k původnímu významu kódovacího jazyka - měl by sloužit pouze k definici významu obsahu stránky a ne jejího vzhledu. Primárně by měl tedy sloužit k určení její sémantiky¹. (CASTRO et. al., 2012, s. 27) Právě na tomto je založena univerzálnost těchto textových dokumentů. Sémanticky validně kódované webové stránky jsou srozumitelnější pro indexovací roboty², jež sémantiku využívají pro vhodné zařazení a upřednostňování webových stránek pro výsledky vyhledávání uživatelů.

Ačkoliv je verze HTML5 označována jako finální i v současnosti dochází k jejímu vývoji v mnoha dalších ohledech. V budoucnu se plánuje od číslování odejít a směřovat k označení *HTML Living Standard*. (WHATWG, 2017)

Dříve se struktura stránky vytvářela pouze pomocí úrovní nadpisů a univerzálním rozdělovacím tagem **div**, jež dále bylo možno blíže specifikovat pomocí unikátního atributu **id** či **class** – např. `<div id="menu"> ... </div>`. Ačkoliv názvy těchto elementů mohly být jakékoliv, struktura designu stránek byla vždy velice podobná. Následující text týkající se HTML5 a nových prvků je reprezentován na Obr. 1, aby byl lépe patrný důvod jejich zavedení.

¹ Nauka o významu výrazů

² Google.com, Seznam.cz, Yahoo.com, aj.



Obr. 1: Příklady rozvržení webové stránky (zdroj: vlastní)

2.2 NOVÉ PRVKY JAZYKA HTML5

Mezi hlavní změny jazyka HTML5 patří nové sémantické prvky, které mnohem lépe rozčleňují obsah dokumentu, než jak toho bylo možno docílit v předchozích verzích³ jazyka HTML. Hlavním důvodem pro zavedení nových elementů byla snaha o přiřazení sémantického a logického významu jednotlivým částem stránek.

Nové elementy vytvářejí pomyslnou mapu dokumentu. Díky těmto změnám dochází rychlejšímu zpracování dokumentu prohlížečem a jinými programy, jako jsou třeba předčítače obrazovek pro zrakově postižené uživatele. (CASTRO et. al., 2012, s. 30)

Mimo to autoři specifikace HTML5 také zkrátili a zjednodušili zápis některých značek.

2.2.1 NAV

Značka `<nav>` definuje navigační oblast stránky. Standardně by se zde měly nacházet odkazy na ostatní dokumenty či na jinou část dané stránky, tzv. kotvy. Je to tedy ideální značka vymezující menu webových stránek.

³ HTML4.01 a XHTML

2.2.2 ARTICLE, SECTION A ASIDE

Spolu se značkou `<nav>` lze tyto elementy označit jako rozdělující. Označují totiž obsahovou část stránky, které by se měl uživatel všímat. Velice důležitou vlastností značek `<section>` a `<article>` je jejich závislost na použití. Záleží na kodérovi, jak se rozhodne je využít a zakombinovat do svých stránek – mohou být využity při rozčleňování článků, komentářů aj. (GASSTON, 2015, s. 36)

Element `<aside>` označuje obsahově nezávislou část dokumentu, která s hlavním obsahem souvisí jen okrajově nebo vůbec. Lze ho například využít pro postranní panely obsahující reklamu, panel s kalendářem, či odkazy na jiné stránky – spřátelené stránky či stránky s podobnou tématikou. (CASTRO et. al., 2012, s. 73)

2.2.3 HEADER A FOOTER

Tyto dvě značky definují obsah dokumentu vzhledem k rodičovskému prvku. Prvek `<header>` lze využít pro definování hlavičky dokumentu, zatímco `<footer>` určuje zápatí s důležitými odkazy či informacemi. Při jejich použití v hlavním elementu `<body>` lze snadno strukturovat dokument a následně ho také stylizovat pomocí kaskádových stylů. Protože definují obsah vzhledem ke svému rodičovskému prvku, lze je využít i uvnitř dříve zmiňovaného elementu `<article>` k doplnění informací - v `<header>` by se mohlo nacházet stručně shrnutí a v `<footer>` by mohl být podepsán autor, případně vysvětlivky k článku. (CASTRO et. al., 2012, s. 76)

3 CSS3

Jazyk CSS, znám též pod pojmem jazyk kaskádových⁴ stylů, slouží k popisu vzhledu elementů webových stránek. Soupis těchto pravidel se standardně nachází v externím textovém souboru s příponou *CSS*, nicméně lze tato pravidla psát také rovnou do webové stránky a to buď mezi značky `<style>` a `</style>`. Tato pravidla lze určovat také rovnou u daného elementu pomocí jeho atributu *style*. Pravidla ovlivňují širokou škálu vlastností.

Zjednodušeně řečeno pomocí pravidel CSS lze definovat vzhled jednotlivých elementů, a proto se s jejich pomocí nastavuje celkový design stránky.

V současnosti je CSS3 nejvíce podporovanou verzí. Důvodem je zavedení mnoha nových vlastností, které kodérům webových stránek usnadňují práci. (CASTRO et. al., 2012, s. 281) Vývojáři webových prohlížečů si zavádějí své nové testovací vlastnosti pro větší podporu jazyka CSS3 a jeho nových vlastností. Ty jim pomáhají při přechodu na nový standart CSS.

3.1 PROPRIETÁRNÍ VLASTNOSTI

Předpony proprietárních vlastností se pokoušejí vyhnout problému s podporou u standardních názvů. (GASSTON, 2015, s. 30) Z toho důvodu je tedy nutno pro jedinou vlastnost potřeba uvést více pravidel, což činí psaní vlastností poněkud složitější, ale tímto zápisem se docílí správného zobrazování vzhledu u starších verzí prohlížečů.

```
div .radius
{
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    border-radius: 10px;
}
```

Obr. 2: Ukázka proprietární vlastnosti zakulacení rohů (zdroj: vlastní)

⁴ Název je odvozen od kaskády, kterou pravidla vytvářejí.

3.2 NOVÉ VLASTNOSTI CSS3

Jazyk CSS3 obsahuje desítky nových vlastností. Mnou zde uvedené jsou dle mého názoru ty nejdůležitější, které jsou využívány při tvorbě moderního a responzivního⁵ webu.

3.2.1 BORDER RADIUS

Pomocí této vlastnosti se vytvářejí zaoblené rohy u elementů. Jedná se o zjednodušený způsob vytvoření vzhledově zajímavějšího designu, nejčastěji tlačítek. Zaoblené rohy dříve bylo možno udělat pouze pomocí obrázků podporující alfa kanál⁶.

3.2.2 RGBA

Představuje rozšířený barevný prostor RGB. Nejedná se o vlastnost, ale o způsob definice barvy např. pro barvu textu (*color*), barvu pozadí (*background-color*) aj. Klasická barva v RGB je definována pomocí trojice přiřazené každé barvě. RGBA je reprezentována čtveřicí čísel, přičemž první tři nabývají hodnotu 0-255 a jsou přiřazena červené, zelené a modré barvě. Poslední číslo představuje průhlednost barvy a může nabývat hodnoty od 0 do 1.

Tento způsob zápisu barvy odbourává potřebu využívat CSS vlastnost *opacity* a filtr *alpha*.

3.2.3 TRANSITION

Jedná se o zkrácený zápis čtyř vlastností, které udávají vlastnosti animace. Kombinuje se s pseudotřídami *:hover*, *:active* a *:focus*.

Vlastnost: Specifikuje, o kterou vlastnost se bude jednat. Např. *width*, *height*, *color* aj.

Trvání: Určuje časovou dobu přechodu.

Časová funkce: Nepovinná vlastnost. Např. *ease*, *linear*. Udává rychlost přechodu v čase.

Zpoždění: Časová doba začátku přechodu od najetí kurzorem myši.

3.2.4 TRANSFORM

Tato vlastnost aplikuje transformace na přidružený element. Jedná se o 2D i 3D transformace typu rotace, zkosení, změnu velikosti, posun aj. Stejně jako vlastnost *Transition* se aplikuje s pseudotřídou. To znamená, že přechod či transformace se aplikuje při kliknutí (*:focus*), přejetí (*:hover*) či výběru (*:active*).

⁵ Přizpůsobujícího se pro zařízení, na kterém je stránka zobrazena.

⁶ Formáty png či gif.

3.2.5 @MEDIA

V překladu uváděná vlastnost jako **dotazy na média** je využívána při tvorbě responzivního webu. Ve starších verzích jazyka CSS se *@media* využívalo již také, ale pouze pro úpravu vzhledu stránky pro tisk, viz Obr. 3. (GASSTON, 2015, s. 53)

```
@media print
{
p{color: Black; font-size: 12pt;}
}
```

Obr. 3: CSS pravidla využívaná pro tisk (zdroj: vlastní)

Ve verzi CSS3 se rozšiřuje syntaxe typů média z předchozí verze CSS2.1 a lze je chápat jako logický výrok složený z jedné či více podmínek. Jestliže je tento výrok pravdivý, aplikují se pravidla v něm obsažená. V opačném případě se pravidla ignorují.

V dotazech pro média se využívají podmínky, které specifikují důležité vlastnosti. Jedná se především o následující seznam, ale je jich mnohem více⁷:

- width, min-width, max-width
Šířka, minimální a maximální šířka zobrazovací plochy zařízení.
- height, min-height, max-height
Obdoba předchozího příkladu, tentokrát však spjato se šířkou.
- orientation
Nabývá hodnoty *landscape* či *portrait*. Jedná se o způsob natočení displeje chytrého telefonu či tabletu.
- device-width, min-device-width, max-device-width
Udává šířku, minimální a maximální šířku obrazovky výstupního zařízení. Stejná vlastnost je i pro výšku, ale již ji sem nebudu uvádět.

```
@media all and (max-width:850px)
{
/* vypínám pravý panel */
.right-aside{display:none;}
#content-in div{margin:0 0 0 160px;}
/*zmensuji pismo v menu*/
#nav ul li ul li {font-size:.85em;}
}
```

Obr. 4: Využití dotazu na média (zdroj: vlastní)

⁷ Kromě dotazů na velikosti existují také dotazy ohledně barev a rozlišení.

4 RESPONZIBILITA

S nástupem nejnovější verze CSS3 se vývoj responzibilního designu webových stránek v mnohém posunul dopředu. Podle provedení se webové designy dají dělit na dva základní typy – *adaptibilní* a *responzivní*. (GASSTON, 2015, s. 53)

„Rozdíl mezi adaptivní a responzivní metodikou spočívá v tom, jak se stránky mění mezi zlomovými body. Adaptivní web design používá sadu rozvržení s pevnou šířkou, kdežto responzivní web design se spoléhá na flexibilní rozměry, takže stránky se přizpůsobují i mezi zlomovými body.“ (GASSTON, 2015, s. 65)

Z výše zmíněného popisu dvou základních způsobu responzibility je jasné, že nutnou podmínkou pro tvorbu vlastního designu, který by byl schopen přizpůsobit se rozdílným zařízením, je nutná znalost kódovacího jazyka HTML a současně také CSS3, nejlépe již pokročilá.

Z toho důvodu vznikly balíky třetích stran, při jejichž správném aplikování se stránky přehledně zobrazují na displejích o různých rozlišeních. Mezi nejznámější patří balík knihoven Bootstrap, který byl vyvinut v rámci Twitteru. Velice podobnou cestou vývoje se také vydalo mezinárodní konsorcium W3C, pod jehož záštitou byl vyvinut balík W3.CSS.

Responzibilitu webových stránek lze zajistit také pomocí jazyku *JavaScript* a jeho knihovny *jQuery*. Nicméně přizpůsobovat stránky pouze tímto způsobem lze vzhledem k rozvoji CSS3, Bootstrapu či W3.CSS považovat za méně elegantní. I přes tento fakt je kombinace jazyka JS a jednoho z výše zmíněných způsobů na stránkách velice častá – při tvorbě šablony pro KVD je například pomocí skriptu upravována druhá úroveň menu.

4.1 BOOTSTRAP

Jedná se o open source framework, který byl v roce 2011 uveřejněný na GitHubu⁸. (BOOTSTRAP CORE TEAM, 2011) k zajištění responzibility jsou využívány externí CSS soubory a také soubory formátu *JS*.

Nejdůležitější podmínkou pro používání Bootstrapu je jeho přiložení v hlavičce webovým stránkám. Tyto dokumenty je možno si stáhnout na oficiálních stránkách vývojářů Bootstrapu a poté se na ně odkazovat na libovolném umístění na svém serveru. Druhou možností je odkazovat se přímo na oficiální soubory vývojářů. Tato možnost je doporučená, webový prohlížeč je totiž může mít umístěn již v mezipaměti a jejich načítání bude trvat kratší dobu⁹.

4.1.1 STRUKTURA GRIDŮ

Základním principem Bootstrapu je rozdělení webové stránky až na dvanáctiny, což svým způsobem může připomínat již zastaralé rozčleňování stránky pomocí HTML tabulek. Pro umístování prvků na stránce je používán systém tzv. *gridů*, což je vymezení obsahu o určité šířce, viz Obr. 5.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4				
span 4				span 8								
span 6						span 6						
span 12												

Obr. 5: Možnosti rozčlenění (Zdroj:W3C)

Obsah stránky je nutno umístit, tzv. *zabalit*, do elementu div třídy *container* či *container-fluid*. Tyto dvě třídy poskytují obalujícímu elementu pevnou či *fluidní*¹⁰ šířkou.

⁸ Webová služba sloužící k vývoji softwaru pomocí verzovacích nástrojů.

⁹ Jedná se ovšem pouze o nepatrný rozdíl.

¹⁰ procentuální

Následně je ještě potřeba vytvořit další obalující element s třídou *row*, neboli řádek. Až nyní lze umísťovat prvky zobrazující se na stránkách. Určení pozice se provádí pomocí identifikátoru *class*, který určuje viditelnost pro různá zařízení a má tvar *col-x-y*, kde *y* představuje šířku prvku na zařízení a *x* reprezentuje zařízení, pro které bude prvek viditelný. Jedná se o:

- *xs*: mobil
- *sm*: tablet
- *md*: monitor počítače
- *lg*: HD a full HD monitory

Pro správné zobrazování by součet všech šířek prvků v řádku neměl přesáhnout 12. Šířka se totiž bude automaticky přizpůsobovat podle velikosti zařízení.

Bootstrap umožňuje využívat i přednastavené bloky elementů. Například navigace, tabulky či oproti klasickému vzhledu modernější formulářové prvky, které jsou mnohem komplikovanější na provedení¹¹ v případě potřeby modifikace vzhledu. Na webových stránkách W3C je již vytvořeno několik ukázkových šablon rozvržení.

Pro více informací lze navštívit webovou stránku (w3schools, n.d.)¹².

4.2 W3.CSS

Oproti výše zmíněnému využívá tento open source framework, vyvinutý mezinárodním konsorciem, pouze jediný CSS soubor. Tím pádem se teoreticky jedná o rychlejší a jednodušší způsob zajištění responzibility, i když rozdíl v načítání souborů by byl prakticky nepostřehnutelný¹³. I přes tento rozdíl ve velikostech W3.CSS upravuje a definuje vzhled mnoha HTML elementů. Stejně jako v Bootstrapu, lze i zde využívat předdefinované vzhledy navigace, záložek, ikon a také varovných zpráv či modálních vyskakovacích oken. Protože se jedná o jediný soubor, lze vše jednoduše měnit pouze přepsáním příslušného pravidla. V případě využívání tohoto balíčku je nutno připojit ho v hlavičce dokument, viz Obr. 6. I zde je doporučovaná možnost odkazovat se na soubor z oficiálních stránek, a to z důvodu rychlejšího načítání.

¹¹ Umožňuje využívat i upravené modální vyskakovací okno.

¹² <https://www.w3schools.com/bootstrap/>

¹³ Rozdíl souborů činí pouze 208,8 kB.

```
<link rel="stylesheet" href="http://www.w3schools.com/lib/w3.css">
```

Obr. 6: Připojení souboru W3.CSS (zdroj: vlastní)

Nejdůležitější použitou třídou je *w3-container*. Jedná se o univerzální třídu a je ji možno aplikovat na elementy *div*, *article*, *section*, *header*, *footer*, *form* aj. Elementu poskytuje levý a pravý vnitřní okraj o velikosti 16 pixelů. Mnohem důležitější je však pravidlo *clear:both*; zalamující veškerý následující obsah pod tento element.

Pro rozvržení layoutu se zde také využívají gridy, velice podobným systémem jako v Bootstrapu. Obalující třídou je ale pouze *w3-row*. Lze využít i již dříve zmíněný systém dvanáctin. Výhodou oproti Bootstrapu je ovšem to, že W3.CSS poskytuje více možností jak vyčlenit elementy na celou šířku zobrazovacího zařízení. Využívá se zde také rozdělování na třetiny, čtvrtiny, poloviny či element s pevnou šířkou, přičemž zbývající elementy zaberou zbytek šířky.



Obr. 7: Možnosti struktury gridů (Zdroj:W3C)

K dispozici jsou ještě následující třídy.

- *w3-hide-small*: skryje obsah na malých displejích (menší než 601px).
- *w3-hide-medium*: skryje obsah na obvyklých displejích.
- *w3-hide-large*: skryje obsah na velkých displejích (větší než 992px).

V případě využití dvanáctin se využívají třídy jako v Bootstrapu.

- *l*: třída pro zařízení s velkým displejem.
- *m*: třída pro obvyklá zařízení.
- *s*: třída pro zařízení s malým displejem.

Je možno zdarma využít desítky rozvržení webových stránek, tzv. templates. Tyto, můžeme říci šablony, jsou cíleny do mnoha odvětví. Například blog, webový životopis, fotogalerie, internetové obchody či portfolia, aj. Pro jejich využití je ovšem potřeba využívat další CSS soubory.

Pro více informací lze navštívit webovou stránku (w3schools, 2017)¹⁴.

4.3 DOTAZY NA MÉDIA

V případě, že bychom nechtěli na stránky umístit žádný volně licencovaný kód a spoléhat se výhradně na něj, můžeme responzibilitu zajistit také. Jak již bylo dříve zmíněno, dotazy na média jsou novým prvkem, který nám umožňuje používat CSS3. Jedná se o nejčastější způsob úpravy webových stránek podle rozlišení zařízení. Nicméně frameworky Bootstrap a W3C.CSS je využívají ve svých kódech a jsou na nich založeny.

Výhoda je více než zřejmá - je odbourána závislost stránky na cizím kódu. Samozřejmě také narůstá obsáhlost a tím pádem velikost souboru CSS. Jeho výsledná velikost se ale odvíjí od komplikovanosti stránek.

Jsou dva způsoby, jak dotazy na média aplikovat. Buď nahrávat různé CSS soubory, nebo vytvořit jediný CSS soubor, který bude obsahovat pravidla pro všechna zařízení. Jedná se o nepatrný rozdíl ve způsobu využití dotazů na média.

Postupů pro tvorbu responzibilního webu s aplikováním dotazů na média je mnoho. Jedním z nich je prvotní vytvoření stránek pro klasické stolní počítače, přičemž se následně pouze přepíšou pravidla zajišťující změnu pro větší a menší rozlišení. Opačný postup, tedy prvotní přizpůsobování stránek mobilům, tzv. *mobile-first*, je v současné době také hojně využíván. Níže dvě zmíněné kapitoly 4.3.1 a 4.3.2 pojednávají pouze o způsobu aplikování pravidel kaskádových stylů pro různá zařízení.

¹⁴ <https://www.w3schools.com/w3css/>

4.3.1 VÍCE CSS SOUBORŮ

Dotazy na media je možné aplikovat v HTML vrstvě webových stránek, a to pomocí atributu *media* u připojování pravidel v hlavičce dokumentu. Jak je možné vidět na Obr. 8, lze si vytvořit více CSS souborů, přičemž každý se bude aplikovat pro jiné šířky zobrazovacího zařízení. (GASSTON, 2015, s. 55) Výhodou tohoto provedení je větší přehlednost v pravidlech a jejich jednodušší úprava. Nicméně prohlížeč je nucen stahovat více než jeden soubor.

```
< link rel="stylesheet" media="only screen and (min-width: 480px)" href="styly_min_480.css" />  
< link rel="stylesheet" media="only screen and (min-width: 600px)" href="styly_min_600.css" />  
< link rel="stylesheet" media="only screen and (min-width: 780px)" href="styly_min_780.css" />
```

Obr. 8: Příklad načítání souborů CSS pro různé šířky (zdroj: vlastní)

4.3.2 JEDEN CSS SOUBOR

V tomto případě se veškerá pravidla zapisují do jediného souboru CSS. Pro tvorbu šablony byla v první řadě zapsána pravidla aplikována na běžné velikosti stolních počítačů. Až po nich byly použity dotazy na média s pravidly aplikovanými pro daná rozlišení. Tento způsob zápisu je dán povahou CSS – dědičností a přepisováním pravidel.

4.4 VIEWPORT

K responzibilnímu zobrazení stránek přispívá meta tag *viewport*, který se podobně jako jiné meta informace umísťuje do hlavičky. Tento údaj zabraňuje automatickému zmenšování stránky v případě menšího rozlišení displeje. Nastavuje šířku a procento přiblížení dle informací zadané v atributu *content*.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Obr. 9: Ukázka zapsání meta tagu viewport (zdroj: vlastní)

5 DESIGN ŠABLONY

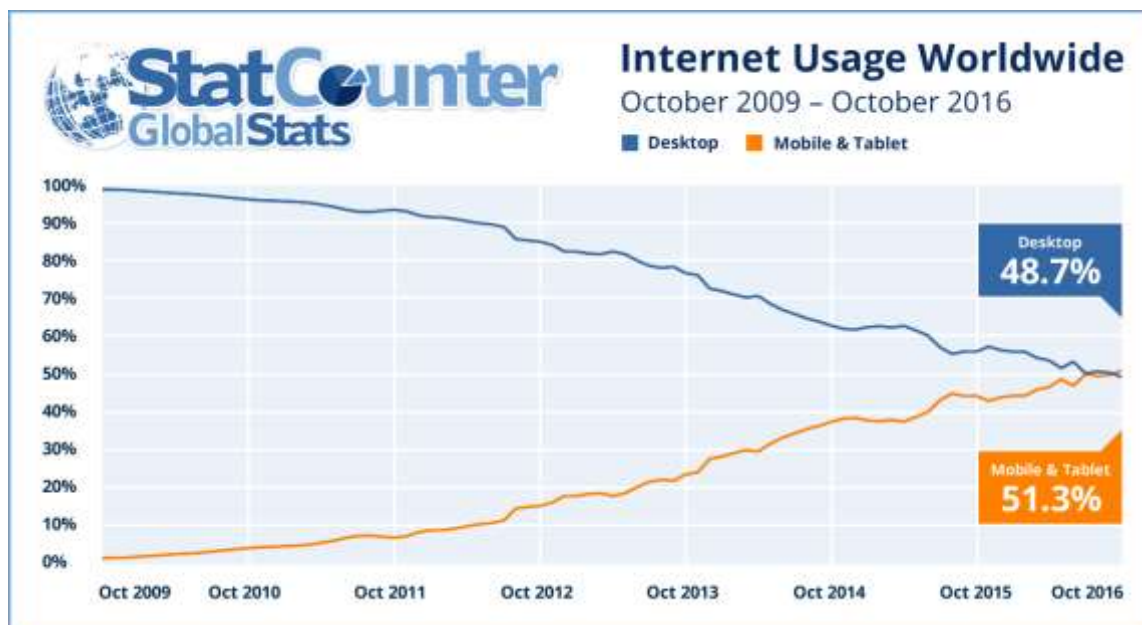
V následujícím textu bych rád rozvedl aspekty a postup tvorby webové stránky, z které poté byla vytvořena šablona pro jednotlivé redakční systémy.

5.1 KRITÉRIA TVORBY NÁVRHU

Při grafickém zpracování šablony pro webové stránky katedry KVD bylo nutno brát v potaz jejich reprezentativnost. Samotný vzhled stránky by měl úzce souviset s tematickým zaměřením, tedy s informatikou. Současně by měl budit dojem modernosti a také celistvosti, poněvadž do vzhledu bylo potřeba zahrnout i stávající logo katedry.

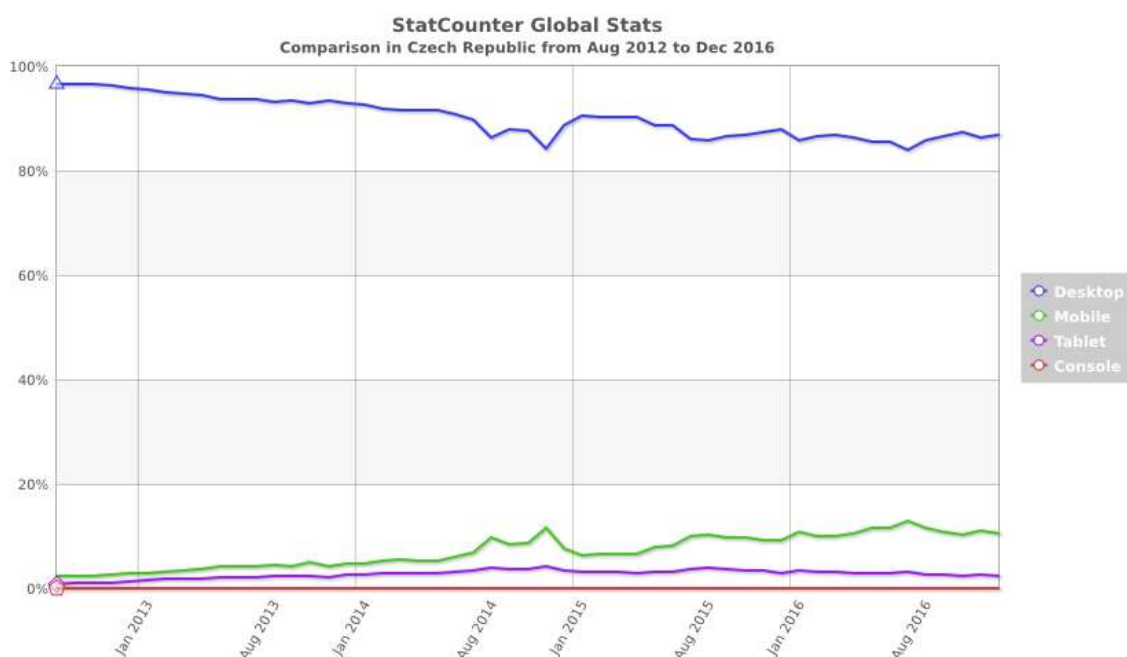
Pojem moderní web lze chápat mnoha významy a při jeho tvorbě je potřeba se držet určitých kritérií. Za nejdůležitější podmínku moderního webu lze pravděpodobně považovat jeho grafickou jednoduchost. Ta totiž přímo souvisí s orientací uživatelů na stránkách. Přesplnění grafického obsahu může působit rušivě na návštěvníky. Ti se tím pádem budou na stránkách mnohem hůře orientovat a to je může odrazovat od opětovného navštívení stránky. (KRUG, 2006)

Dalším důležitým aspektem stránek je jejich responzibilita. Přehledné zobrazování webových stránek na co největší škále zařízeních je v dnešní době klíčovým prvkem. Celosvětový počet uživatelů využívající přenosná zařízení pro prohlížení internetových stránek stále roste. To je více než patrné z údajů tří milionů webových stránek, které byly poskytnuty společnosti StatCounter, viz Obr. 10. Tím pádem také roste potřeba vývojářů webových stránek a aplikací, aby jejich rozhraní byla plně responzibilní. V případě špatného zobrazování na určitém zařízení lze totiž předpokládat, že uživatelé s tímto zařízením již webovou stránku nenavštíví či aplikaci nepoužijí, ačkoliv by chtěli.



Obr. 10: Graf zobrazující nárůst prohlížení webu pomocí mobilů a tabletů.
 (zdroj: (STATCOUNTER GLOBAL STATS, 2016))

Oproti tomu situace v České republice je dle údajů již dříve zmiňované společnosti naprosto jiná. Stolní počítače stále patří k preferovanému zařízení pro prohlížení webových stránek a mobily se kupodivu využívají o mnoho méně (10,59%). Nicméně tento údaj nemůže být vzhledem k neurčitosti českých zdrojů pro StatCounter zcela přesný.



Obr. 11: Srovnání využití zařízení pro přístup k internetu v České republice.
 (zdroj: (STATCOUNTER GLOBAL STATS, 2016))

5.2 REALIZACE NÁVRHU

Při realizaci grafických návrhů byl brán zřetel na plánovanou strukturu stránek. Ta je tvořena co nejjednodušeji, takže na první pohled je zřejmé umístění hlavičky, hlavního obsahu i patičky.

Po analýze všech kritérií popsaných v kapitole *Kritéria tvorby návrhu* byly vypracovány čtyři grafické návrhy. Po projednání s vedoucím práce byl jeden z nich vybrán a následně v průběhu práce upravován do jeho konečné podoby. Viz přílohy.

5.2.1 HLAVIČKA

Rozvržení hlavičky je z celých stránek nejvíce komplikovanější částí, protože obsahuje mnoho dílčích elementů. Byla realizována jejím rozčleněním na čtyři oddíly. Viz Obr. 12.



Obr. 12: Návrh hlavičky (zdroj: vlastní)

První část obsahuje většinu aktivních prvků, těch je dohromady pět. Logo, více informací níže, je umístěno na levou stranu obrazovky s určitým odsazením¹⁵. To i přes vygenerování patřičné velikosti působí rozmazaně, se zhoršenou kvalitou. Pro pozdější využití by bylo pravděpodobně lepší využít formát typu *svg*, pro zajištění lepšího vzhledu loga. Následně se zde nachází seznam odkazů, který je umístěný na středu obrazovky. Asi graficky nejpestřejší je vyhledávací formulář, který se nachází na pozadí plošných spojů, přičemž každý z nich má jiný odstín zelené. Samotný vzhled elementů *input* je navržen tak, aby zapadaly do zeleného designu stránek – tedy s bílým pozadím a zeleným zakulaceným rámečkem. Posledním prvkem v této části je odkaz vypadající jako tlačítko, sloužící k přepínání jazyku stránky.

¹⁵ Tzv. bezpečná zóna loga.

Druhá část je ryze navigační a obsahuje horizontální menu vytvořené pomocí seznamu a je animováno pomocí pravidel CSS. Menu je pouze dvouúrovňové a jeho obsah byl vytvořen na základě položek menu na původních stránkách katedry. Zčásti překrývá následující část, čímž se vytváří vzhledově hezčí jeho druhá rozbalující se úroveň. Problémovou částí menu je nutné vytvoření *podmenu* pro všechny položky v menu obsazené.

Poslední části slouží k oddělení hlavičky od obsahu stránky. K tomuto účelu bylo vyfoceno několik fotek, jejichž zobrazování je závislé ryze na způsobu implementace – fotografie by bylo možné měnit pomocí JS skriptu či náhodně určovat fotografii při načítání stránky, např. pomocí PHP. Také se zde nachází aktuální pozice návštěvníka, tzv. drobečková navigace.

Logo

V případě loga byl největší problém jeho nesymetričnost, protože ta způsobovala v mnoha případech omezení při tvorbě návrhu. Rozvržení loga na čtvercovou obrázkovou a textovou část zahrnující název katedry umožňuje jediné vhodné umístění, a to na levé polovině stránky. Umístění loga na střed stránky nepůsobí z grafického hlediska dobře, aspoň pro zařízení s vyšším rozlišením obrazovky. Stránka poté budí dojem zastaralosti.

Tvorba designu stránek z největší části závisela na vzhledu loga. Jeho manuální vizuální styl povoluje pouze čtyři varianty loga, přičemž se jedná pouze o bílou či černou verzi. Ostatní barevné kombinace, či jakákoliv jiná úprava loga je zakázaná. Tím pádem se vzhled stránky vždycky směřoval do zeleného barevného schématu, neboť černobílé stránky nepůsobí správným dojmem pro vzdělávací instituci.

Při prohlížení stránek na mobilním zařízení byla použita bílá verze loga s černými jedničkami v levém poli. Toto logo koresponduje s ostatním bílým textem. V případě ponechání původní verze loga by docházelo ke splývání, až mizení, světle zeleného zaobleného čtverce, z kterého by byly viditelné pouze bílé jedničky. Bohužel toto provedení plně nekoresponduje s podmínky využití loga, neboť povolená barva pozadí je pouze bílá či černá.

Vyhledávání

Jak bylo již řečeno, vyhledávací formulář je na popředí obrázku plošných spojů. Jeho vzhled, bílý zaoblený obdélník se zeleným rámem a ikonou lupy, je pro většinu zařízení nezměněný.

Změna nastává až pro šířku 600 pixelů a méně, kdy dochází k překreslení celé hlavičky. Pozice zmenšeného vyhledávání je zde mezi tlačítkem vyvolávající menu a zkratkou alternativního jazyka, v kterém lze stránky zobrazit.

Jedná se o jednoduché nastavení vzhledu, avšak komplikace by mohly nastat u starších verzí prohlížečů. V počítačových učebnách KVD dochází k mírnému posunutí tlačítka na starších verzích Firefoxu.

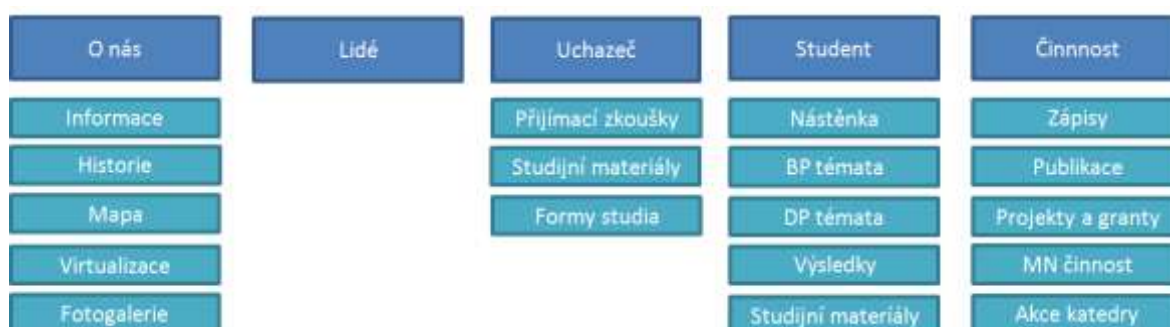
Odkazy

Odkazy nacházející se v hlavičce stránek patří mezi odkazy s menším informačním rázem. Jedná se například o zpětné vazby na Západočeskou univerzitu, Pedagogickou fakultu a několik dalších vedlejších informací. Po celou dobu jsou vycentrovány na střed a pro rozlišení o šířce 1200 pixelů a méně se skrývají.

Menu

Navigace po stránkách je zajištěna pomocí dvouúrovňového menu, jehož jednotlivé prvky jsou zvýrazněny větší velikostí písma oproti ostatním textům. V jeho použití jsou současně zahrnuty *hover*¹⁶ animace a *JS* funkce zajišťující změnu šířky jednotlivých podpoložek při změně přiblížení či překlopení displeje.

Po provedení analýzy současných webových stránek byla navrhována níže uvedená struktura menu:



Obr. 13: Struktura menu (zdroj: vlastní)

¹⁶ Aktivují se při najetí kurzoru myši.

Při řešení responzibility dochází pouze k přepočítávání šířky menu a jeho jednotlivých položek. Jeho celková šířka je udána v procentuální hodnotě. Napříč různými rozlišeními displejů zde může dojít pouze k drobným úpravám v odsazení textu a velikosti písma. Komplikace nastávaly pro menší šířky než 600 pixelů, kdy docházelo k velikému zhuštění položek. Z toho důvodu bylo pro menší displeje menu pozměněno do podoby 30 pixelů úzkého pásu, který je pevně umístěn k hornímu okraji obrazovky. Při této změně dochází také k úpravě vyhledávacího formuláře, tlačítka pro změnu jazyka a loga katedry.

Ve výchozím nastavení je menu uživateli nepřístupné. K zobrazení slouží čtvercová ikonka, na Obr. 14 úplně vpravo. Při jejím aktivaci kliknutím dochází k animovanému sestoupení menu, které je do té doby skryto mimo viditelnou část¹⁷ displeje.



Obr. 14: Návrh upravené hlavičky s menu pro menší rozlišení (zdroj: vlastní)

V této verzi jsou hlavní položky zobrazeny ve sloupečku a jsou od sebe odděleny čarou šedivé barvy. Pro mobilní zobrazení byl využit kód třetí strany umístěný v souboru *doubletaptogo.js*. Jedná se o volně licencovaný *jQuery*¹⁸ doplněk ulehčující užívání menu pro dotykové displeje. Každá položka menu je současně také odkazem na jinou stránku. To bohužel způsobuje, že při poklepání na odkazy první úrovně okamžitě dochází k přesměrování, místo zobrazení dalších položek v druhém úrovni menu¹⁹. Výše zmiňovaný doplněk zabraňuje tomuto chování a nejdříve zobrazuje skrytou druhou úroveň, jestliže je přítomna. Na stránku přesměruje pouze při opětovném poklepání na odkaz.

¹⁷ Má nastavené záporné souřadnice.

¹⁸ JavaScriptová knihovna.

¹⁹ Lze tomu na mnoha mobilech předejít, ale okamžité přesměrování je pro návštěvníky velice vyčerpávající.

5.2.2 OBSAH

Textový obsah je umístěn v elementu *main*, hned za hlavičkou stránky, a je rozdělen do tří nezávislých částí.

Spoje

Obdobně jako v hlavičce se také zde se nachází obrázek tištěných spojů. Ten je umístěn do levé horní části rodičovského prvku, tím pádem vytváří dojem, že je přímo spojen s drobečkovou navigací. Jedná se ovšem pouze o grafický prvek designu stránky a v rámci responzibility je skryt při maximální šířce 1060px za účelem získání více místa pro text.

Obsah

Právě zde se nachází obsahově nejdůležitější část každé stránky. Je vytvořena pomocí prvku *article*, jenž má nastavenou procentuální šířku rozdílnou pro různá zařízení. Nedochozí zde k žádným razantním změnám, lišit se může pouze mezera mezi řádky a velikost písma. Podle šířky se přizpůsobují zbylé dvě části – spoje a widget, viz níže.

Z hlediska vzhledu je nutné, aby každý text začínal nadpisem úrovně *h1*. Krom jeho velikosti je totiž ještě podtrhnut zelenou čarou. Tu je v případě potřeby možno editovat v příslušném CSS pravidlu.

Widget

Obsahovou část dokresluje vpravo umístěný widget světle zelená barvy loga. Tento grafický prvek, vytvořen pomocí seznamu odkazů, zobrazuje možnosti kontaktů v případě nějakých nejasností či jiných potřeb, přičemž jednotlivé možnosti jsou od sebe odděleny svislou čarou.

Jeho součástí je také obrázek šipky směřující nahoru. Při kliknutí na toto tlačítko se vykoná funkce JS *goUp()*, která postupně umístí posuvník prohlížeče do jeho základní pozice. Tato akce se vykonává s náhodně dlouhým intervalem a ulehčuje posun na mobilních zařízeních v případě delšího textu.

5.2.3 PATIČKA

Element *footer* je od hlavního obsahu stránky oddělen horizontální šedou čarou o tloušťce 2 pixely. Jsou zde umístěny méně důležité odkazy, celkem do tří samostatných sloupců zarovnaných na střed. Každá jednotlivá skupina je uvedena pomocí nadpisu první úrovně a lze je skrývat a zobrazovat kliknutím při šířce displeje 550 px až 600 px včetně. Při ostatních velikostech je tato funkcionalita znemožněna. Na levé straně se nachází oficiální logo pedagogické fakulty, jež současně slouží jako odkaz na jejich webové stránky. V případě potřeby je možno logo pedagogické fakulty nahradit logem Západočeské univerzity. Dále se zde nachází 20 pixelů vysoký element `div`, určený pro případné vypisování poslední aktualizace. V pravé straně patičky je umístěn obrázek tištěných plošných spojů.

Pro menší rozlišení dochází ke skrytí či posunutí obrázku plošných spojů a odkazy se umístí pod dříve zmíněné katedrální logo vycentrované na střed.

Pro zařízení o šířce 550 až 600 pixelů budou v patičce zobrazeny plošné spoje. Nezabírají ovšem tolik místa jako při větším rozlišení. Nadpisy odkazů jsou umístěny v jednom sloupci, vedle něhož jsou viditelné odkazy. V jednu chvíli je možno vidět pouze jeden sloupec a lze mezi nimi přepínat pomocí kliknutí na nadpisy.

Oproti všem dříve zmíněným částem stránky zde dochází k nejkomplicovanějším změnám vzhledu, které jsou řízeny pomocí JavaScriptových funkcí. Provádějí se při načtení stránky, změně velikosti a změně orientace displeje, obdobně jako funkce upravující šířku menu.

5.3 TESTY RESPONZIBILITY

Webové stránky byly před tvorbou jednotlivých šablon testovány na zařízeních Samsung UE43KU6072, iGET SMART S72 a Microsoft Lumia 550, přičemž byly objeveny níže zmíněné problémy. Kromě těchto problémů je responzibilní chování stránek plně odpovídající předpokladům. Na televizi Samsung UE43KU6072, která má rozlišení čtyřnásobně vyšší než rozlišení FullHD, se webová stránka zobrazovala bez jediné chyby na celkovém designu.

Ikona menu

Na tabletu iGET SMART S72 s operačním systémem Android se nezobrazovala správně ikonka menu v mobilní verzi webových stránek. Ta byla vytvořena pomocí Unicode znaku `☰`. Ten ale pravděpodobně není zahrnut v základních fontech systému Android. Z onoho důvodu je ikonka menu nakonec vkládána jako obrázek formátu *png*, i když to s sebou přináší menší eleganci a také případné komplikovanější úpravy menu.

Pozicování menu

Na mobilním telefonu Lumia 550, a pravděpodobně i na více druhů Microsoft Phone, dochází při posunu stránek k velice zajímavému jevu. Menu je pomocí CSS pravidel pevně umístěno k hornímu okraji displeje při jakékoliv pozice posuvníku prohlížeče. Nicméně při posunu dochází k „*těkavému pohybu*“ menu, jenž je způsobeno oním posunem a snahou menu zůstat stále na nulových souřadnicích. Vzhledem k nízkému zastoupení telefonů s operačním systémem daného typu na tento problém nebyl brán zřetel.

Drobné posunutí tlačítka vyhledávání

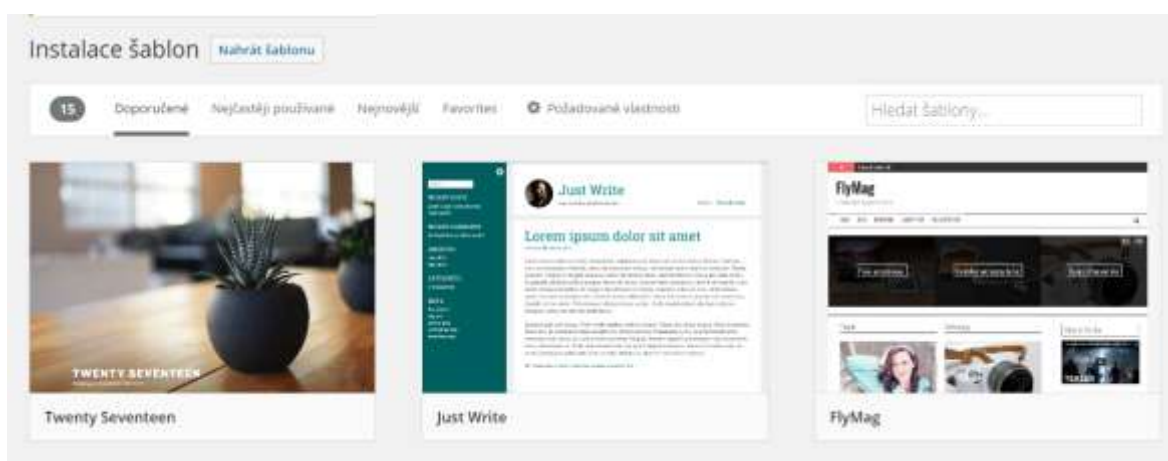
Problémovou částí vývoje webových stránek byla také snaha přizpůsobit je napříč prohlížeči. Pro tento účel byl využit resetovací CSS styl, jenž rozdílne výchozí *margin* a *padding* u všech prohlížeče nadefinuje na nulovou hodnotu a zajistí přesnější podobné vykreslování. Nicméně i přes tento pomocný soubor pravidel bylo nutno využít tzv. *CSS hack* pro nejnovější verzi Mozilla Firefox. Jedná se o specializovaný zápis pravidla, jenž bohužel není validátory brán jako validní zápis, ale v daném prohlížeči zajistí správné zobrazení dle daného pravidla. Leč i přes tato opatření je možné, že může dojít k nepatrným nepřesnostem, neboť webové stránky byly optimalizovány hlavně pro programy Google Chrome verze 56.0 a Mozilla Firefox verze 51.0.1.

6 WORDPRESS

Redakční systém WordPress se řadí mezi nejpoužívanější CMS s rozsáhlou uživatelskou komunitou, která se podílí na jeho vývoji. Tím pádem je k dispozici velice široké spektrum doplňků, od čehož se ovšem odvíjí možnost potenciálních chyb v kódu. (interval.cz, 2011)

6.1 ZÍSKÁVÁNÍ ŠABLON

Instalace obsahuje několik základních šablon sloužících k představení možností redakčního systému. Mimo seznamu již nainstalovaných šablon lze pomocí administrátorského rozhraní prohlížet, vyzkoušet a instalovat nové vzhledy. Tento výčet lze upravovat pomocí filtrů. Možnost správy šablon je oproti ostatním redakčním systémům, o kterých se pojednává v této práci, unikátní.



Obr. 15: Ukázka dostupných šablon přes administrátorské rozhraní (zdroj: vlastní)

Tyto šablony jsou k dispozici také na stránkách www.wordpress.org, v sekci Themes. Nachází se jich zde několik tisíc a je umožněno nahrávat vlastní šablony. Ke každé šabloně je zde uveden popis s náhledem od autora. Lze zde u každého vzhledu nalézt graf ukazující počet stažení za poslední dny, hodnocení a další doplňující informace.

Pro redakční systém WordPress lze najít bezpočet komerčních a nekomerčních šablon na specializovaných stránkách. Např. www.themegril.com či www.colorlib.com.

6.2 MOŽNOSTI EDITACE A PŘIZPŮSOBENÍ

Šablony je možné pomocí redakčního systému WordPress upravovat mnoha způsoby, přičemž každý z nich má svoje klady a zápory.

6.2.1 PŘÍMÁ EDITACE SOUBORŮ ŠABLONY

Pomocí editoru CMS WordPress můžeme upravovat všechny textové soubory nacházející se v adresáři šablony. Jedná se především o PHP a CSS soubory. Jedná se o razantní zásah do struktury šablony, která není nijak kontrolovaná. Hrozí tedy možnost nevyžádaného chování šablony v případě neodborného úpravy. Komplikace také může nastat v případě aktualizace jádra redakčního systému, přičemž by hrozilo přepsání námi dříve vytvořených úprav a navrácení šablony do výchozí podoby.



Obr. 16: Editor souborů (zdroj: vlastní)

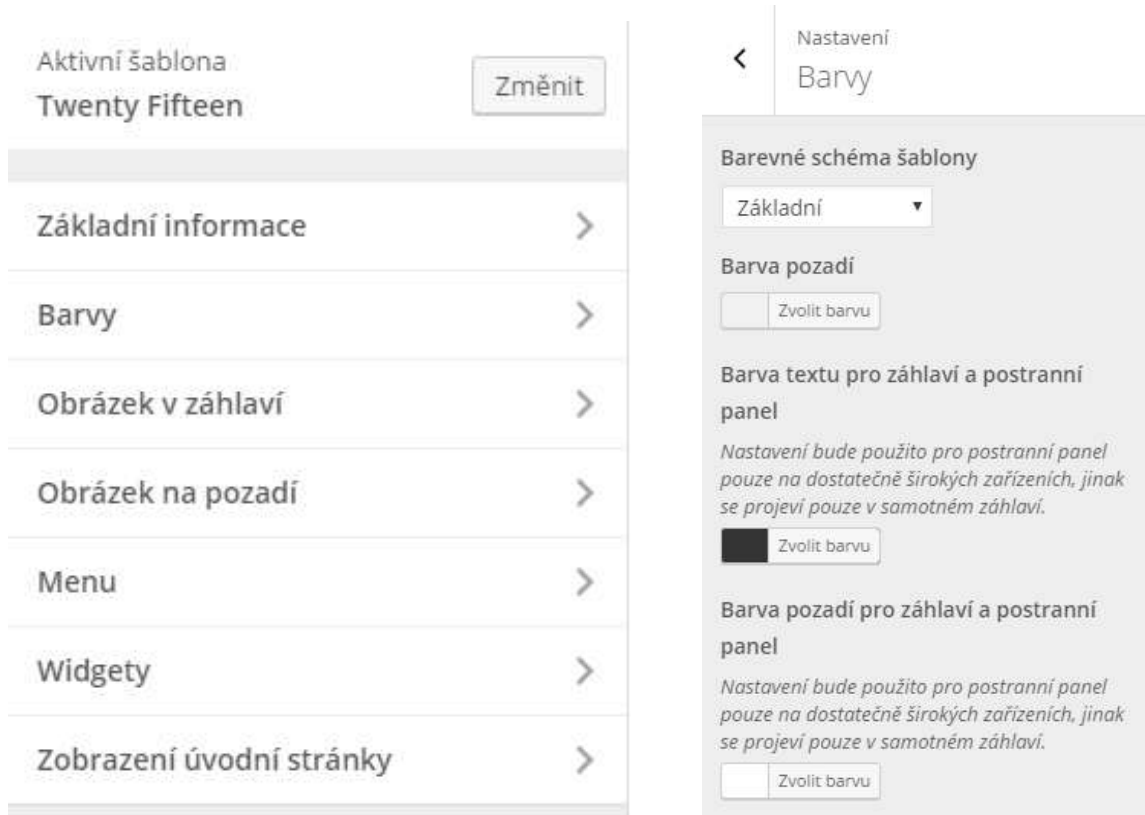
K dispozici je také možnost zobrazení manuálu WP funkcí. Tento manuál se nachází na internetových stránkách www.developer.wordpress.org.



Obr. 17: Zobrazení manuálu a tlačítko pro ukládající úpravu souboru (zdroj: vlastní)

6.2.2 PŘIZPŮBENÍ ŠABLONY

Tento nástroj umožňuje měnit barevná schémata, fonty, obrázky na pozadí, aj., avšak možnosti přizpůsobení jsou závislé na provedení šablon.



Obr. 18a a obr 18b: Možnosti přizpůsobení šablony Twenty Sixteen (zdroj: vlastní)

Šablona KVD není o pokročilé editace rozšířena, a proto lze upravovat pouze standardní prvky jako základní informace (popis, ikona webu), menu a způsob zobrazení úvodní stránky.

6.3 TVORBA VLASTNÍ ŠABLONY

Před tvorbou šablony pro redakční systém WordPress, dále již jen WP, je nutno vzít v potaz strukturu navrhované stránky. Šablona WP totiž obsahuje soubory zastupující jednotlivé části webu, které vzájemným složením vytvoří požadovaný obsah, podrobněji vysvětleno níže. Z našeho designu je patrné, že stránka bude složena ze tří hlavních částí – hlavičky, obsahu a patičky. Tyto tři hlavní části je možno dále rozčlenit na další jednotlivý obsah.

6.3.1 STRUKTURA SOUBORŮ ŠABLONY

V první řadě je nutno založit složku, kde se bude šablona nacházet. Složku, pojmenovanou názvem šablony, vytvoříme na umístění *wp-content/themes/* v adresáři WP. Současně jsou zde potřeba tři soubory - *Footer.php*, *Hlavní soubor index.php* a *Header.php*. Z těchto tří souborů je nejdůležitější *index.php*, protože bez něho nebudou do šablony načítána žádná data. *Header.php* a *footer.php* jsou brány jako soubory doplňujícího rázu, neboť jestli nebudou existovat, doplní si WP svoji vlastní generovanou hlavičku a patičku stránek.

Pro šablonu lze ještě vytvořit soubor *Functions.php*, sloužící k definici vlastních funkcí, využití funkcí WP či jejich přepsání. Jeho obsah by bylo možno umístit do dříve zmíněných souborů, ale jedná se o pracnější provedení na správu a případnou úpravu.

6.3.2 HLAVNÍ SOUBOR INDEX.PHP

Soubor *index.php* můžeme považovat za jeden z nejdůležitějších souborů z celé šablony. Zobrazující se stránka vznikne vložením pomocných souborů²⁰ na požadované místo v tomto souboru, čehož lze docílit pomocí WP funkcí, např. *get_header*, a *get_footer()*.

Vkládání příspěvků se realizuje pomocí cyklu *while*, který vytváří seznam dostupných příspěvků, viz Obr. 19.

Úvodní stránkou redakčního systému WP většinou bývá přehled několika posledních přidávaných příspěvků. Jestliže chceme jako úvodní stránku mít pouze jeden příspěvek, je potřeba upravit nastavení šablony v administračním rozhraní, v přizpůsobení šablony. Současně s tím je nutno vytvořit příspěvek sloužící jako úvodní stránka.

²⁰Např. *header.php*, *footer.php* a *sidebar.php*.

```

<?php if ( have_posts() ) : ?>
    <h1><?php single_post_title(); ?></h1>
<?php
    while ( have_posts() ) :
        the_post();
        the_content();
    endwhile;
else: ?>
    <h1>Chyba</h1>
    <p><?php _e( 'Bohužel nebyla nalezena žádná odpovídající stránka.' ); ?></p>
<?php endif; ?>

```

Obr. 19: Kód vkládající příspěvky (zdroj: vlastní)

Kód vkládající příspěvky jsem rozšířil o jednoduchou podmínku kontrolující existenci požadované stránky. Jestliže ta není splněna, vypíše se zpráva „*Bohužel nebyla nalezena žádná odpovídající stránka.*“. Tato varovná zpráva plní funkci stránky 404, která se zobrazuje při nenalezení daného obsahu na serveru. V opačném případě se vypíše obsah se svým názvem v nadpisu úrovně *h1*.

Sidebar

Obdobně jako ostatní nabídky na stránkách, i obsah bočního panelu je vytvořen pomocí menu. Ovšem vytvoření menu s obsahem obrázků WP neumožňuje bez instalace specializovaných doplňků. V šabloně KVD je použití obrázků jako odkazů umožněno pouze na banner a to pomocí mnou implementované funkce `get_image_menu($location)`, podrobnosti níže v kapitole *Functions.php*.

6.3.3 HEADER.PHP

Tento soubor obsahuje potřebný kód vypisující se do každé stránky. Jedná se především o element `<head>`, který obsahuje *meta* informace stránky a načítání CSS a JS souborů. Při načítání externím souborů bylo využito dvou WP funkcí - `get_template_directory_uri()` a `esc_url()`, jejichž kombinací se vypisuje tzv. *escapovaná*²¹ cesta šablony doplněná o cestu k souborům. Mimo to se zde nahrává také tzv. *favicon*²² stránek a jejich titulek. V něm je použita WP funkce `wp_title()` vracející titulek aktuálně prohlížené stránky. Před samotným koncovým tagem `</head>` se volá WP funkce `wp_head()`, která do šablony vypisuje `<link>` potřebných souborů pro WP.

²¹ Bezpečné zobrazování uvozovek a středníků v řetězci znaků.

²² Ikona webu.

Soubor dále obsahuje pouze počáteční tag `<body>` a hlavičku stránek obsaženou ve značkách `<header>` a `</header>`. Jedná se tedy o seznam odkazů, logo sloužící jako odkaz na index, vyhledávací formulář, drobečkovou navigaci, přepínač jazyků a hlavní navigaci stránek.

Seznam odkazů

Vypisování seznamu odkazů v hlavičce je realizován pomocí menu. To se vypisuje v souboru `header-url.php`, jenž je následně do `Header.php` vkládán pomocí WP funkce `get_template_part($name)`. Protože WP realizuje vypisování menu pomocí HTML elementů `` a ``, bylo potřeba vypisování menu řešit pomocí mé vlastní funkce, která by položky vypisovala pouze jako odkazy `<a>`. K tomuto účelu slouží ve `functions.php` funkce nazvaná jako `get_header_url_menu($location)`.

Logo

Katedrální logo je ve stránkách řešeno jako pozadí elementu `<div>` s CSS třídou `logoKVD`. Umístění indexu stránek je získáno pomocí WP funkcí `esc_url()` a `get_home_url()` a je realizováno pomocí standardního odkazu.

Vyhledávací formulář

Vyhledávací formulář se v šablonách vkládá pomocí WP funkce `get_search_form()`. Ovšem tím docílíme vložení defaultního vyhledávacího formuláře. Tuhle možnost lze překrýt v souboru `searchform.php`, jenž bude obsahovat námi požadovaný HTML formulář. Tento formulář ale musí mít totožný identifikační prvky jako defaultní, neboť při potvrzení vyhledání se data předávají obslužnému kódu WP. Jedná se především o atributy formuláře určující metodu předání dat, jeho id a také id textového inputu s vyhledávaným řetězcem.

Hlavní menu

Hlavní navigace stránek je realizována pomocí menu nazvané Header Menu, do něhož lze přes administrátorské rozhraní WP vkládat jednotlivé položky. Dvouúrovňové menu se vytváří také přes toto rozhraní, takže zde nebudu popisovat postup pro jeho vytvoření. V šabloně je pro navigaci nejdůležitější následující kód uvedený na Obr. 20.

```

<nav>
  <div class="navIN" id="nav">
    <div class="sipka" onClick="animace();"><!--<span>#9776;</span--></div>
    <div class="respo_odel"></div>
    <?php wp_nav_menu( array( 'theme_location' => 'header-menu',
                              'container' => 'ul',
                              'menu_class' => 'navmenu' ) ); ?>
  </div>
  <div class="navGreen"></div>
</nav>

```

Obr. 20: Segment hlavičky, hlavní menu (zdroj: vlastní)

Kromě HTML elementů `<div>` a `<nav>` lze na obrázku spatřit také PHP kód obsahující pouze volání WP funkce `wp_nav_menu($arg)`. Ta obsahuje jako parametr pole hodnot s určitými názvy indexů.

- `theme_location`: Určuje lokaci v šabloně, která se použije k vypisování.
- `container`: Mění obalující element menu. Předdefinovanou hodnotou je `<div>`.
- `menu_class`: Definuje třídu elementu ``, jenž vytváří samotné menu.

Tento kód vypisoval téměř vhodnou strukturu hlavního menu. Sice došlo k přidání atributů třídy a id v elementech ``, ale to nemělo žádný vliv na vzhled. Problémovější částí byla druhá úroveň menu, u které nešlo změnit třídu vnořeného ``. Tento problém se jednoduše vyřešil pouze úpravou názvu dané třídy v CSS souboru použitý v šabloně.

Obtížněji bylo vyřešeno přidání obalujícího elementu `` s třídou `ohran`. Ten vykresluje světle zelený rám o tloušťce 1px, čímž slouží k vizuálnímu oddělení jednotlivých položek menu. Za tímto účelem byla použita WP funkce `add_filter($tag, $funkce)`, jež upravuje danou funkci uvedenou jako řetězcový parametr `$tag`, viz Obr. 21.

```

add_filter( 'wp_nav_menu_objects', function( $items ) {
    foreach ( $items as $item ) {
        if ( $item->menu_item_parent ) {
            $item->title = '<span class="ohran">' . $item->title . '</span>';
        }
    }
    return $items;
});

```

Obr. 21: Funkce `add_filter()` upravující vypisování menu (zdroj: vlastní)

Během jejího vykonání jsou procházeny položky menu a jsou dotazovány na id svého rodiče (za rodiče je považována jiná položka). Jestliže nemá položka rodiče, id se rovná nule a tím pádem se podmínka neprovede. V opačném případě je její název upraven do požadovaného tvaru.

Drobečková navigace

Samotný WP neumožňuje vypsání drobečkové navigace, k tomuto účelu by bylo nutno nainstalovat některý z mnoha doplňků. Na webových stránkách (Musilda, 2013) lze najít PHP funkci vypisující jak samotnou navigaci, tak další podrobné údaje. Tento kód mi s využitím funkcí WP posloužil k vytvoření své vlastní funkce **Chyba! Nenalezen zdroj odkazů.** vypisující drobečkovou navigaci v patřičném tvaru odkazů.

Přepínač jazyků

Přepínání jazykové verze stránek bohužel není možné ze strany šablony zajistit bez většího zásahu do administrátorského rozhraní WP. Pro vícejazyčnou verzi WP by bylo potřeba nainstalovat plugin WPML Multilingual CMS či jeden z alternativních pluginů²³ navržený stránkou www.alternativeto.net. Další možnou variantou je nainstalovat samostatnou instanci WP a pouze mezi nimi přepínat.

Dříve zmíněné pluginy umožňují spravovat jednotlivé komponenty WP dle jazyků, např. menu, a vytvářejí překlady článků.

Instalace podobné instance WP, jen s jiným jazykem, je pravděpodobně méně náročná než instalace pluginů, nicméně je nutno spravovat dvojí obsah.

6.3.4 FOOTER.PHP

Tento soubor, představující patičku stránek, obsahuje pouze tag `<footer>`. V něm se nachází logo pedagogické fakulty, obrázek spojů, tři menu a datum informace o poslední aktualizaci získaný pomocí WP funkce `get_the_modified_date()`.

Tato tři menu lze editovat přes administrátorské rozhraní, obdobně jako hlavní navigaci stránek. Jejich struktura je ovšem trochu odlišná, viz Obr. 22.

```

php  if ( has_nav_menu('pata-prvni') ) {  ?>
<div class="widget">
  <?php
    echo '<h1 onclick="zobraz(\''prvni\'')">'.get_name_menu('pata-prvni').'</h1>';
    wp_nav_menu( array( 'theme_location' => 'pata-prvni',
                       'menu_id' => 'prvni',
                       'container_id' => 'prvni',
                       'container' => 'ul' ) );
  ?>
</div>
<?php }?>

```

Obr. 22: Vypisování prvního sloupečku v patičce stránek (zdroj: vlastní)

²³ Pluginy qTranslate a Ceceppa Multilingua.

Samotné menu se vypisuje pomocí funkce `wp_nav_menu($arg)`, kde se určuje jeho lokace, id menu a id kontejneru, v kterém bude obsažen. Přiřazení id kontejneru je důležité pro vykonávání JS funkce `zobraz(element)`, jenž zobrazuje danou kategorii při zobrazení na malých displejích o velikosti 550 px až 600 px včetně.

Nachází se zde také volání implementované funkce `get_name_menu($location)`, jenž vypisuje název menu, pod nímž si můžeme představit název kategorie obsahující k ní náležící odkazy.

6.3.5 FUNCTIONS.PHP

Tento soubor slouží k implementaci vlastních funkcí, pomocí nichž lze nahradit funkcionalitu některých pluginů. Také je zde možno přepisovat či upravovat defaultní funkce samotného WP. Níže vypisuji dle mého názoru nejdůležitější obsah.

register_my_menu()

Pomocí této funkce je do šablony nadefinováno šest navigačních menu. Jedná se vlastně o šest po sobě následujících volání WP funkce `register_nav_menu()` s parametry `$location` a `$description`, čímž se v šabloně nadefinují objekty menu. Ty je možno spravovat před administrační rozhraní.

Pomocí řádku `add_action('init', 'register_my_menu')` se tato funkce volá při inicializaci.

get_name_menu(\$location)

Tato funkce vrací jméno menu, které se určuje pomocí parametru funkce. Toho je docíleno pomocí funkce `get_nav_menu_location()`, jenž vytvoří pole objektů umístění. Pomocí parametru vybereme námi požadované menu a vrátíme obsah vlastnosti title.

Využívá se v patičce šablony pro vypisování názvu menu do tagu `<h1>`.

get_image_menu(\$location)

Obdobně jako v předchozí funkci `get_name_menu($location)` se zde pomocí parametru funkce získají data požadovaného menu. V cyklu `foreach` se následně vypisují jeho položky ve tvaru: `<img="..." src="..." />`

Je použita pro zobrazování obrázků sloužící jako odkazy. Její nevýhodou je, že název položky menu musí korespondovat s názvem obrázku umístěného v adresáři `obr` naší šablony. Jedná se o obrázky formátu `png`.

get_header_url_menu(\$location)

Jedná se o obdobnou variantu předchozí funkce. Položky menu se vypisují v jednodušším formátu, pouze jako odkazy.

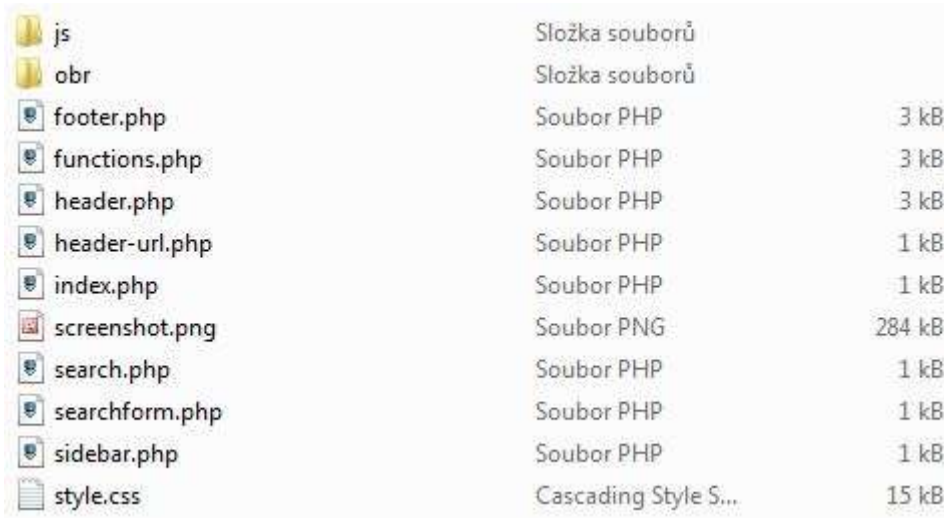
6.3.6 SEARCH.PHP

Tento soubor je svojí strukturou velice podobný souboru *index.php*. Plní také velice podobnou funkci. Pomocí něho jsou vypisovány výsledky hledání a jsou uživateli zobrazeny v upravené podobě.

6.3.7 OSTATNÍ SOUBORY ŠABLONY

Kromě výše zmíněných souborů, z nichž se skládá struktura požadované stránky, se v samotné šabloně nacházejí ještě další složky a soubory. Jedná se o pravidla kaskádových stylů, složku obrázků nutných pro design, pomocné JS skripty a obrázkový soubor *screenshot.png*, jenž v administrátorském rozhraní WP slouží jako ukázkový náhled šablony.

Výsledná struktura je zobrazena na Obr. 23.



js	Složka souborů	
obr	Složka souborů	
footer.php	Soubor PHP	3 kB
functions.php	Soubor PHP	3 kB
header.php	Soubor PHP	3 kB
header-url.php	Soubor PHP	1 kB
index.php	Soubor PHP	1 kB
screenshot.png	Soubor PNG	284 kB
search.php	Soubor PHP	1 kB
searchform.php	Soubor PHP	1 kB
sidebar.php	Soubor PHP	1 kB
style.css	Cascading Style S...	15 kB

Obr. 23: Struktura šablony pro redakční systém WordPress (zdroj: vlastní)

7 JOOMLA

Redakční systém Joomla se společně WordPressem a Drupalem řadí mezi trojici nejpoužívanějších CMS. Má velice obsáhlou komunitu uživatelů s vývojářů. Oproti WordPressu, který se zaměřuje na blogové internetové stránky, lze s jeho pomocí vytvářet různé druhy webových stránek. Např. e-shop, portfolio, blog, aj. (interval.cz, 2011)

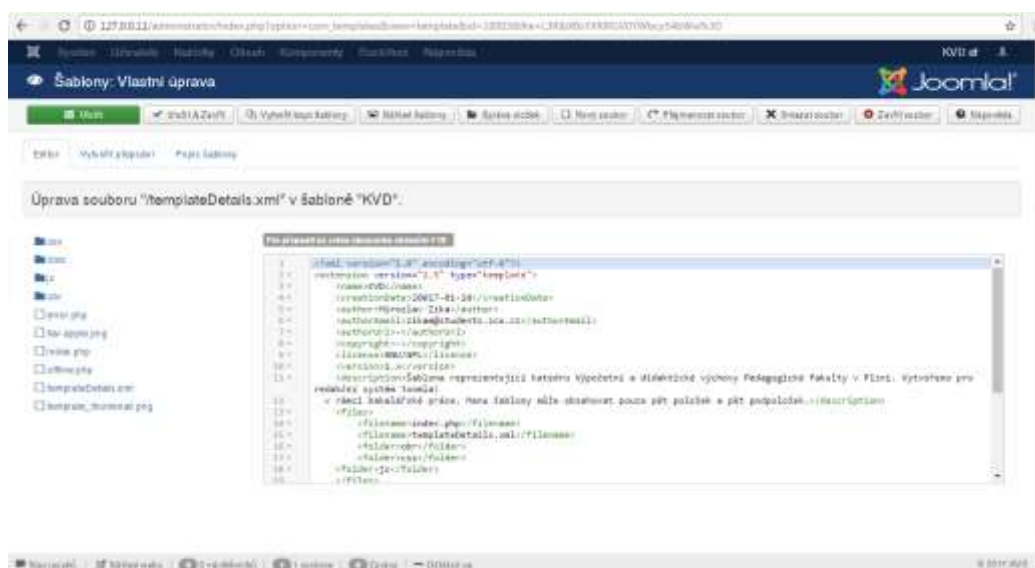
7.1 ZÍSKÁVÁNÍ ŠABLON

Nedostatek šablon je jedna z nevýhod tohoto redakčního systému. Na oficiálních stránkách www.joomla.com je v sekci Templates uveřejněno několik šablon, avšak jejich množství je velice malé. Z toho důvodu je nutno při hledání širšího spektra vzhledů věnovat pozornost společnostem, které se specializují na tvorbu šablon. Dalšími zdroji jsou například stránky www.joomshaper.com, www.siteground.com a www.gavick.com či www.astemplates.com, kde lze stáhnout také šablony pro redakční systém WordPress.

Přes administrátorské rozhraní lze prohlížet pouze šablony fyzicky umístěné v adresáři redakčního systému.

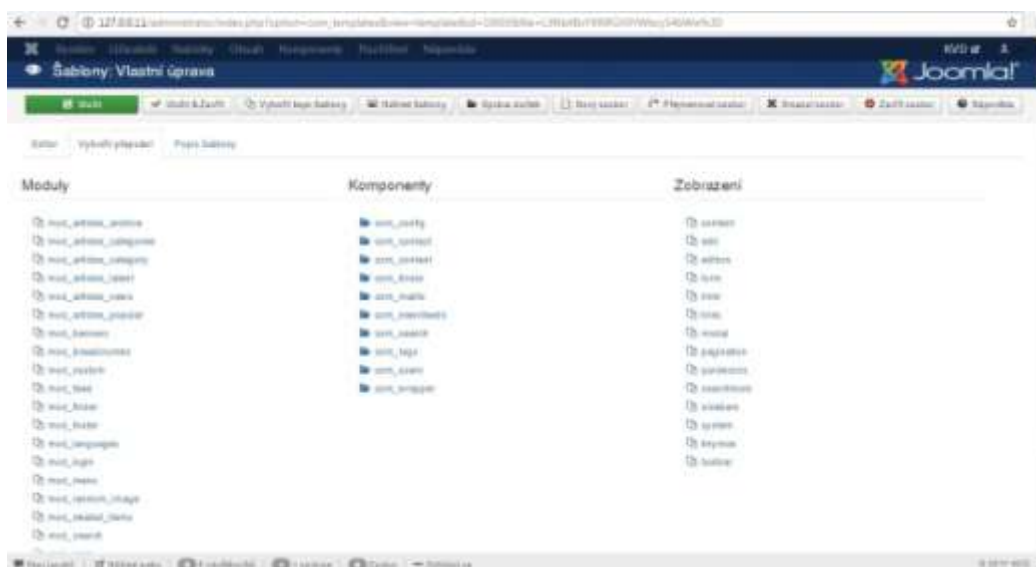
7.2 MOŽNOSTI EDITACE A PŘÍZPŮSOBNÍ

V případě potřeby je možné upravovat dokumenty šablony. Tato možnost se provádí pomocí vlastní úpravy šablony, jež umožňuje přímou editaci z administrátorského rozhraní, viz Obr. 24.



Obr. 24: Přímá úprava dokumentu (zdroj: vlastní)

Kromě přímé úpravy dokumentu šablony a zobrazení jejího popisu je také umožněno přepsání určitých PHP souborů samotného redakčního systému, čímž lze ovlivnit způsob vykreslování či chování mnoha modulů. Přehled souborů, u nichž je povoleno přepsání, je uveden na Obr. 25.



Obr. 25: Přepisování dokumentů redakčního systému Joomla (zdroj: vlastní)

Vytvoření přepsání je velice jednoduché. Po kliknutí na název prvku se vygeneruje do adresáře šablony požadovaná struktura adresářů a PHP. Jednotlivá struktura je vždy umístěna v odpovídající složce nesoucí název přepisovaného prvku. Všechny tyto složky se nacházejí v rodiči *html*, který se automaticky vytvoří v adresáři šablony – jedná se poté tedy o obsah šablony. Kód přepisujících PHP souborů, jenž zajišťuje zpracování dat do šablony, odpovídá kódu defaultního souboru redakčního systému. Jeho úpravou docílíme námi požadovaných změn.

7.3 TVORBA VLASTNÍ ŠABLONY

V redakčním systému Joomla se nová šablona umísťuje do ní příslušné složky, kterou je potřeba vytvořit ve složce *templates*. Název této složky by měla odpovídat názvu samotné šablony. Právě zde se nacházejí níže popsané a uvedené soubory.

7.3.1 DEFINUJÍCÍ SOUBOR TEMPLATEDETAILS.XML

Popisuje strukturu šablony a obsahuje její základní meta-data jako například jméno autora, jméno šablony, verzi aj., vše uvedené ve značkovacím jazyce XML. Mezi nejdůležitější informace pro nás patří umístění všech složek a souborů nacházející se v rodičovském adresáři šablony. Další nutnou položkou jsou tzv. *position*. Ty nám definují použitelné umístění interaktivních prvků, do kterých lze umísťovat obsah pomocí administrátorského rozhraní. Tento soubor je nutný pro viditelnost šablony.

7.3.2 HLAVNÍ SOUBOR INDEX.PHP

Jedná se o základní soubor, jenž bude využívat pro zobrazování obsahu každá stránka – kromě zobrazování zprávy o chybě a vypnutí stránek, viz dole.

Není zde nutno již předpřipravený HTML dokument rozčleňovat do jednotlivých souborů reprezentující jeho důležité bloky jako u WordPressu. Nicméně je nutno dopsat PHP kód vkládající na dané pozice jejich obsah, je-li uživatelem vytvořen. Na začátku dokumentu je také potřeba dopsat ochranný kód, viz Obr. 26, jenž chrání kód proti přístupu zvenčí.

```
<?php
defined( '_JEXEC' ) or die( 'Nepovolený přístup.' );
?>
```

Obr. 26: Ochranný kód (zdroj: vlastní)

V šabloně je možno použít globální proměnnou `$this`. Byla použita především pro vypisování některých vlastností.

- `$this->baseurl`: vypisuje URL webových stránek.
- `$this->template`: vypisuje název šablony, tedy KVD.
- `$this->language`: vypisuje používaného jazyka.

Pomocí této globální proměnné lze ověřit, zda se na dané pozici nachází nějaký obsah, tzv. modul. Na Obr. 27 je uvedený jednoduchý kód vkládající obsah pozice *druhy_sloupek*, jestliže mu nějaký byl přes administrátorské rozhraní definován. Vložení je provedeno pomocí `jdoc:include`, což je unikátní příkaz šablon redakčního systému Joomla vkládající obsah určený atributem *name*.

```

<?php if ($this->countModules('druhy_sloupek')) : ?>
<div class="widget">
  <jdoc:include type="modules" name="druhy_sloupek" />
</div>
<?php endif; ?>

```

Obr. 27: Kód vkládající obsah pozice *druhy_sloupek* (zdroj: vlastní)

Kromě modulů lze vkládat například *head* obsahující meta informace či samotný obsah, tzv. *component*. Pro vkládání modulů je nejprve zapotřebí v *Definující soubor* *templateDetails.xml* definovat potřebné pozice, do nichž se budou vkládat.

7.3.3 ERROR.PHP A OFFLINE.PHP

Tyto dva soubory slouží k zobrazování chybové hlášky, např. 404, a informační zprávy v případě dočasného či úplného odstavení domény přes administrátorské rozhraní. K tomu lze využít vzhledově rozdílné stránky, ať graficky či strukturou, leč v tomto případě byl použit HTML kód klasických stránek pouze s drobnými grafickými rozdíly zařízenými pomocí CSS. Jediný rozdíl se nachází v obsahu tagu *<main>*, ve které se nachází příslušná informace o chybě.

error.php

V případě zobrazování chybové zprávy pomocí souboru *error.php* do něj redakční systém Joomla neumožňuje vkládat moduly pomocí *<jdoc:modules>*, jimiž jsou v šabloně realizována veškerá menu. Aby stránka v případě chyby nevypadala nepřírozně prázdně, byl vzhled hlavičky graficky upraven, čímž na první pohled vypadá rozbitě, viz Obr. 28 a Obr. 29.



Obr. 28: Vzhled rozbité hlavičky stránky na mobilních zařízeních (zdroj: vlastní)



Obr. 29: Vzhled rozbité hlavičky pro monitory (zdroj: vlastní)

Tato úprava je realizována pomocí vkládání obrázků místo původních ovládacích elementů. Nejsou tedy nijak akčně činná. Ve vzhledu by bylo možno ještě využít další tematické obrázky.

Chybová zpráva se vypisuje pomocí sady řetězcových proměnných samotného redakčního systému. Jedná se o vypsání klasických příčin chyby, např. překlep v adrese či neexistenci požadovaného souboru. Součástí je vytvoření odkazu na indexovou stránku webu.

Veškerý text je zobrazován na základě aktivní jazykové verze prohlížených stránek, jestliže je na stránkách instalováno více jazyků.

offline.php

Velice obdobným způsobem je provedeno zobrazování informační zprávy, že stránky jsou z určitého důvodu odstaveny mimo provoz. Zde je povoleno vkládat moduly, takže stránka není obsahově prázdná jako v předchozím případě. Veškeré odkazy ovšem přesměrovávají uživatele zpátky na `offline.php`. Je možno vypsát buď výchozí text spjatý s vypnutím stránek či svůj vlastní, jenž je možno nastavit přes administrátorské rozhraní.

7.3.4 POSITION

Vzhledem ke způsobu tvorby šablony pro tento redakční systém nebylo již nutno vytvořit žádné další soubory. Tvorba menu, jejich položek a přiřčování do modulů a pozic se provádí výlučně skrze administrátorské rozhraní. Z toho důvodu bych v následujícím textu chtěl stručně tyto pozice vypsát, abych objasnil jejich funkci v šabloně:

- `search`: využita pro vyhledávací formulář,
- `main_menu`: hlavní navigace stránek,
- `header_menu`: seznam pomocných odkazů zobrazujících se v hlavičce,
- `banner_menu`: odkazy v postranním panelu,
- `breadcrumbs`: drobečková navigace,

- jazyk: přepínání jazykových verzí,
- první_sloupek: tři sloupečky odkazů zobrazujících se v patičce stránek,
- druhy_sloupek,
- třetí_sloupek.

Všechny moduly, např. přihlašování, menu, vyhledávání, aj., jsou realizovány defaultním způsobem redakčního systému. Pro vytváření šablony byly přepsány moduly *mod_breadcrumbs*, *mod_language*, *mod_menu* a *mod_search*. Z komponent byly přepsány *com_content* a *com_search*.

Přepsané komponenty

com_content

Tento modul slouží k vypisování hlavního obsahu do stránek, tedy jednotlivých článků. Jeho editace byla nutná, neboť plně neodpovídala požadovaná struktura článku. Kromě toho docházelo k vypisování nepotřebných informací, s kterými se v designu nepočítalo, a proto působily velice rušivě. Byly odstraněny např. informace o času vydání, autoru, jazyku článku, aj. I přes svůj prvotní rušivý vliv byla ponechána ikonka tiskárny sloužící k tisku textu, která byla umístěna do levého horního rohu.

com_search

Jedná se o čtveřici PHP souborů nutných pro stránku po odeslání požadavku při vyhledávání. Jedná se o vyhledávací formulář rozšířený o prvky upřesňující vyhledávání, vypisování chybové hlášky, vytvoření výsledků hledání a řídicí soubor, jenž skládá tyto tři jednotlivé soubory dohromady. Zde byly provedeny úpravy HTML struktury a drobné designové úpravy pomocí CSS.

Přepsané moduly

mod_search

Tento soubor vykresluje hlavní vyhledávací formulář umístěný v hlavičce. Zde bylo nutno přepsat původní strukturu a rozšířit ji o potřebné třídy. Převážně bylo zapotřebí připsat input typu *submit*, který reprezentuje tlačítko s lupou. Jedná se v zásadě o podobnou úpravu, popsanou výše v *com_search*.

mod_breadcrumbs

Obdobně jako u mod_search, také modul pro drobečkovou navigaci byl přepsán převážně pro úpravu jeho HTML struktury. Defaultní drobečková navigace byla vytvářena pomocí odrážkového seznamu ``. K našemu účelu bylo zapotřebí vytvořit pouze po sobě následující odkazy, jež jsou od sebe odděleny pomocí separátoru `>`, pomáhající formulovat pomyslnou mapu stránek.

mod_menu

Ačkoliv tento modul obsahuje pět PHP souborů, rozdílnou strukturu menu v hlavičce, hlavního menu stránky, obrázkové banner menu a tří menu v patičce lze docílit úpravou jediného souboru z nich – *default.php*. Jak již bylo zmíněno dříve, obsah stránky je reprezentován pomocí modulů, přičemž každému modulu lze přiřadit jednu z pozicí šablony. Při vypisování jednotlivých menu stačí pomocí tří podmínek kontrolovat pozice, ve které je modul umístěn. Zjednodušený kód podmínek je uveden na Obr. 30.

```

if($module->position == "banner_menu"){//vytvoření banner menu

    //vypsání prvků ve tvaru: <li><a><img></a></li>

}elseif($module->position == "header_menu"){//vytvoření seznamu odkazů v hlavičce

    //vypsání prvků ve tvaru: <a></a>

}else{//vytvoření hlavní navigace stránky a seznamů odkazů v patičce

    if($module->position == "prvni_sloupek") {
        //úprava proměnné $id a vypsání nadpisu první úrovně s onClick atributem
    }
    if($module->position == "druhy_sloupek") {
        //úprava proměnné $id a vypsání nadpisu první úrovně s onClick atributem
    }
    if($module->position == "treti_sloupek") {
        //úprava proměnné $id a vypsání nadpisu první úrovně s onClick atributem
    }

    //vypsání <ul>
    //vypsání jednotlivých <li> prvků, respektive vytvoření druhé úrovně menu
    //vypsání </ul>
}

```

Obr. 30: Podmínky upravující vypisování menu (zdroj: vlastní)

V případě hlavní navigace je nutno modulu, v němž je menu umístěno, přiřadit přes administrátorské rozhraní třídu *navmenu*. Nedošlo by totiž k přizpůsobování šířky.

The screenshot shows the Joomla! administrator interface for configuring a menu module. The title is 'české menu'. The 'CSS třída nabídky' (Menu CSS class) field is set to 'navmenu'. The 'Pokročilé' (Advanced) tab is selected.

Obr. 31: Přidání třídy modulu pro menu (zdroj: vlastní)

mod_language

Vytvoření vícejazyčného webu pomocí redakčního systému Joomla zahrnuje nutné úpravy v administrátorském rozhraní, leč oproti WordPressu není nutno instalovat žádné doplňky ve formě pluginů. K poskytnutí vícejazyčného obsahu je nutno vykonat následující postup.

1. Instalace jazyků.
2. Vytvoření menu pro každý jazyk.
3. Vytvoření indexu pro všechny jazyky.
4. Vytvoření indexu pro jednotlivé jazyky.
5. Vytvořit přepsání modulu *mod_language* a upravit strukturu dle potřeby.

Úprava tzv. přepínače jazykových verzí je možná úpravou jeho vlastností pomocí Joomla rozhraní. Jazyk je možné reprezentovat názvem či ikonkou vlajky, v závislosti použití rozbalovacího menu. To totiž umožňuje pouze názvy jazyků. V případě standardního seznamu lze vybírat menu vlajkou a názvem.

Tyto úpravy ovšem nijak nemění strukturu jazykového přepínače. Pro náš design stránek by bylo možno využít možnosti zobrazování vlajek jazyků. Pro dodržení původního designu je zapotřebí přepsat modul *mod_language*. Právě zde se nachází kód vypisující tvar přepínače dle nastavení. Abychom docílili zobrazení zkratky jazyka, je místo jeho názvu vypísána zkratka pomocí: `echo(strtoupper($language->sef));`

Nainstalování dvou jazyků a vytvoření menu pro každý z nich umožní kromě zobrazení přepínače také přidružovat položky jednoho menu k odpovídajícím položkám menu v druhém jazyku. Při přidružení odkazuje přepínač přímo na stránku v druhém jazyce. V případě nepřidružení je uživatel přesměrován na pouze na úvodní stránku.

7.3.5 OSTATNÍ SOUBORY ŠABLONY

Kromě výše zmíněných souborů obsahuje šablona nutné soubory CSS, JS a obrázkové soubory. Jako ukázka šablony slouží soubor *template_thumbnail.png*. Obdobně jako v předchozím redakčním systému byla potřeba drobných úprav pravidel CSS. Komplikovaná struktura druhé úrovně menu je tentokrát vyřešena JS kódem uvedeným na Obr. 32

```
$( "ul.podmenu li.menu-item a" ).wrap( '<span class="ohran"></span>' );
$( "ul.podmenu li" ).removeAttr( "class" );
```

Obr. 32: Zabalení položek druhé úrovně do spanu (zdroj: vlastní)

Výsledná struktura šablony je zobrazena na Obr. 33.

css	Složka souborů	
html	Složka souborů	
js	Složka souborů	
obr	Složka souborů	
error.php	Soubor PHP	5 kB
fav-apple.png	Soubor PNG	5 kB
favicon.ico	Ikona	2 kB
index.php	Soubor PHP	5 kB
offline.php	Soubor PHP	5 kB
template_thumbnail.png	Soubor PNG	35 kB
templateDetails.xml	Soubor XML	2 kB

Obr. 33: Struktura šablony pro redakční systém Joomla (zdroj: vlastní)

8 DRUPAL

Tento redakční systém je možno označit jako nejkomplicovanější z výše uvedených, ať architekturou systému či jeho správou. Pro jednoduché weby může být až zbytečně pokročilý. (interval.cz, 2011)

8.1 ZÍSKÁVÁNÍ ŠABLON

Ani u redakčního systému Drupal se v případě dostupnosti šablon nevyskytují žádné razantní změny oproti dříve zmíněným CMS. Pomocí vyhledávače lze nalézt mnoho internetových stránek poskytujících volně dostupné šablony – např. volně licencované vzhledy na www.devsaran.com či placené na stránkách www.themeforest.net. Oficiální webové stránky www.drupal.org obsahují širokou škálu šablon. Jedná se ovšem i o vzhledy pro již zastaralé verze jader, neboť pomocí vyhledávajícího formuláře bylo možno vybrat označení verze od 4.0.x až 9.x²⁴. Dohromady se tedy jednalo o třináct verzí k výběru.

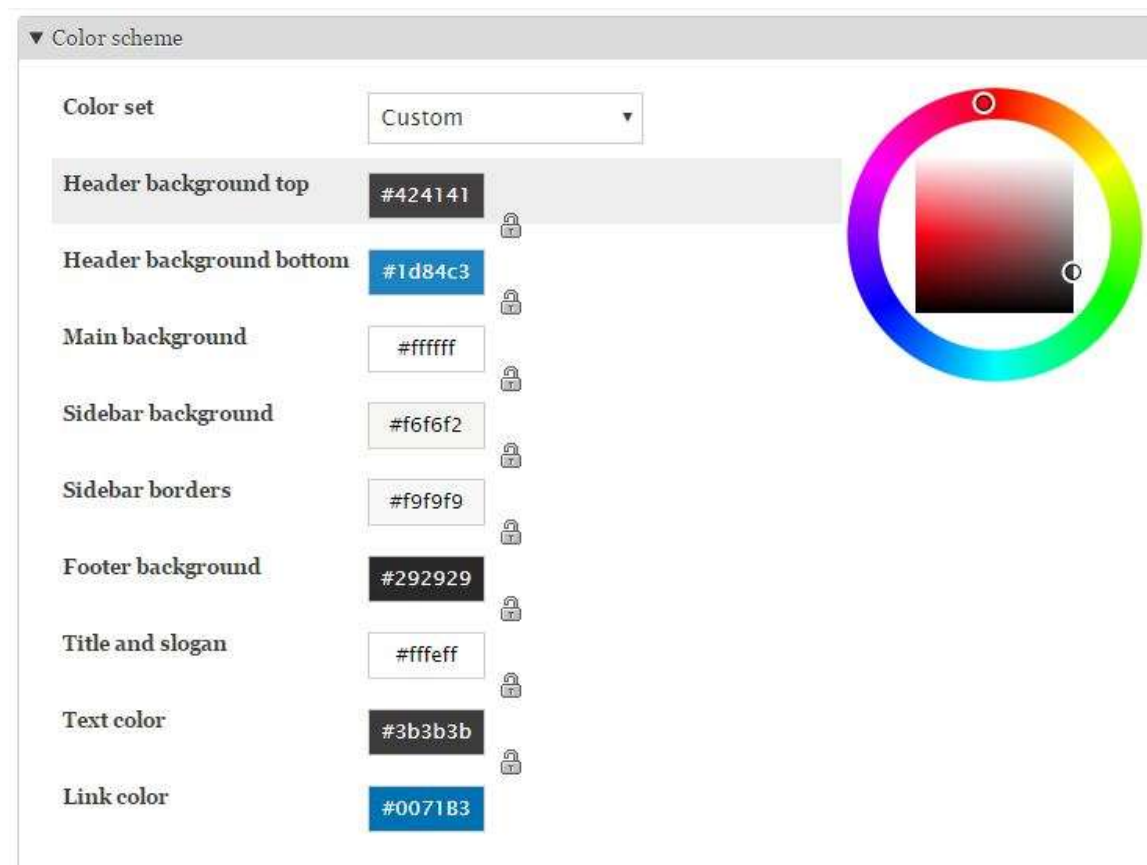
Nutno podotknout, že šablony pro tento redakční systém se dělí na tři skupiny dle jejich uplatnění. Jedná se o šablony pro administrátorské rozhraní a šablony pro klientem přístupnou část stránek, tzv. frontend. Existují také šablony, které je možné aplikovat na obě tyto samostatné části stránek.

8.2 MOŽNOSTI EDITACE A PŘÍZPŮSOBNÍ

Úpravu šablon lze realizovat pomocí přímé editace požadovaného souboru šablony. Jedná se ovšem o velice razantní zásah do samotné šablony. Přes administrátorské rozhraní je možné upravovat globální proměnné vztahující se k zobrazování základních grafických elementů šablony jako je logo a favicon. Lze také umožnit vkládání obrázků do článků a uživatelských komentářů článků.

Na Obr. 34 je zobrazena ukázka upravování barev v šabloně *bartik*. Jedná se o změnu celého barevného setu či jen drobná úpravy barvy použitého setu. Tato funkcionality je podmíněna existencí nutných souborů v šabloně. Jedná se především o textový soubor *color.inc*, obsahující předdefinované barevné sety, a soubor *preview.html* sloužící jako náhled výsledné barevné kompozice stránky.

²⁴ Pro verzi 9.x nebyla k dispozici ke stažení žádná šablona.



Obr. 34: Editor barev šablony (zdroj: vlastní)

Do administrátorského rozhraní je možno nainstalovat modul Theme editor, čímž lze získat ještě více nástrojů umožňující editaci šablony. Jedná se však o vývojáři neověřený modul a jeho používání je pouze na vlastní riziko. Navíc ho lze využít pro Drupal verze 7.x.

8.3 TVORBA VLASTNÍ ŠABLONY Z NÁVRHU

Nainstalované témata vzhledů se nacházejí ve svých přidělených složkách v adresáři `/cores/themes`. Tyto složky, obdobně jako v předchozích případech, jsou pojmenovány názvy šablon.

8.3.1 DEFINUJÍCÍ SOUBOR `.INFO.YML`

Šablony pro redakční systém Drupal 8 mají specifickou strukturu. Jejím základním kamenem jest definiční soubor `názevšablony.info.yml`. Jedná se o textový soubor obsahující meta-data o šabloně, požadované jsou: `name`, `type`, `description` a `core`.

Pro šablonu jsou ale velice důležité údaje v sekci `regions` a v případě použití externích souborů také sekce `libraries`.

```
name: kvd
type: theme
description: 'Šablona reprezentující katedru ...
package: core
version: 1.x

#Regions
regions:
  header_menu: 'Header Menu'
  primary_menu: 'Primary Menu'
  search: 'Search'
  lan_switcher: 'Language Switcher'
  breadcrumb: 'Breadcrumb'
  content: 'Content'
  banner: 'banner'
  footer_logo: 'Footer Logo'
  footer_first: 'Footer First Widget'
  footer_second: 'Footer Second Widget'
  footer_thirt: 'Footer Third Widget'

version: '8.x'
core: '8.x'

screenshot: screenshot.png

libraries:
  - KVD/global-styling
  - KVD/global-scripts
```

Obr. 35: Ukázka .yml souboru (zdroj: vlastní)

8.3.2 REGIONS

Regions plní stejnou funkci jako prvky position v CMS Joomla – slouží k určení místa, ve kterém se budou zobrazovat jim přidružená data přes administrátorské rozhraní. Jak je možno vidět na Obr. 35, v šabloně pro KVD je využito jedenáct těchto elementů.

Nutné je dodržovat předepsaný zápis souboru. Odsazení nové řádky v .yml souboru se nesmí provádět pomocí tabu, ale jedinečně pomocí dvou mezer.

8.3.3 LIBRARIES A SOUBOR .LIBRARIES.YML

Všechny externí soubory lze považovat za knihovny námi vytvořené šablony. Jak lze vidět na Obr. 35, v šabloně KVD se využívají dvě knihovny; *global-styling* a *global-scripts*. Jedná se o pravidla kaskádových stylů a pomocné JS skripty přizpůsobující velikost hlavní navigace a patičky.

KVD.libraries.yml

Tento soubor slouží k definici balíčků souborů, tzv. knihoven, které krom standardních souborů budou v šabloně použity. Na Obr. 36 je znázorněna struktura zápisu knihovny nazvané *global-styling*, jenž obsahuje soubory typu CSS. Poté následuje cesta k souborům a případné doplňují informace v závorkách `{}`. Stejným způsobem je v souboru definována knihovna *global-scripts*, která obsahuje soubor *script.js* a *doubletaptogo.js*.

```
global-styling:
  version: VERSION
  css:
    theme:
      css/style.css: {}
      http://meyerweb.com/eric/tools/css/reset/reset.css: { type: external, minified: true }
```

Obr. 36: Ukázka ze souboru KVD.libraries.yml (zdroj: vlastní)

8.3.4 TEMPLATES

Redakční systém Drupal 8 využívá v systému šablon soubory formátu *.html.twig*. Jedná se o textové soubory šablonového systému Twig. Slouží k oddělení aplikační a prezenční vrstvy, tedy k oddělení PHP kódu od HTML, což s sebou nese nevýhodu transparentnosti kódu.

Struktura navrhnuté šablony vzniká, stejně jako v CMS Joomla, přepisováním příslušných souborů zastupující jednotlivé funkční bloky stránky. Jejich přepsání není bohužel umožněno pomocí administrátorského rozhraní. Tyto soubory je nutno vyhledat buď v samotném jádru Drupalu, nebo pomocí jiných zdrojů.

Šablonový systém umožňuje vytvoření souboru *.html.twig* pro samostatný prvek vytvořený přes administrátorské rozhraní. Např. mějme vytvořený hlavní navigační menu pro anglický jazyk s atributem *Machine name: hlavni-menu-ang*²⁵.



Obr. 37: Ukázka vlastnosti menu (zdroj: vlastní)

Lze k němu přidružit soubor *menu—hlavni-menu-ang.html.twig*, který slouží k zápisu požadované struktury navigace. Tento způsob je ovšem nepraktický, neboť uživatel nemusí vědět, jak je potřeba pojmenovat menu pro zajištění správné funkcionality.

²⁵ Atribut *Machine name* je vytvořen při prvotním založení, nenašel JSem způsob změny.

page.html.twig

Tento soubor můžeme považovat za primární *.html.twig* v celé šabloně, neboť reprezentuje základní stránku. Nejedná se ovšem o celou kostru, ale pouze o obsah tagu `<body>`. Obsahuje tedy bloky `<header>`, `<main>` a `<footer>`. Krom toho se zde nachází také podmínky typu *if* šablonového systému Twig, které se dotazují na existenci v šabloně využívaných *regions*, viz Obr. 38. Tato podmínka je použita u všech regionů, kromě *content*. U něho je předpokládána existence z důvodu jeho reprezentování článku.

```
<div class="odkazy">
  <span>
    {% if page.header_menu %}
      {{ page.header_menu }}
    {% endif %}
  </span>
</div>
```

Obr. 38: Podmínka *if* v šabloně *page.html.twig* (zdroj: vlastní)

html.html.twig

Kostra HTML stránky je definována pomocí tohoto souboru. Jeho výchozí obsah byl pozměněn minimálně, aby nedocházelo ke vkládání administrátorského panelu sloužící správě webu. Tento doplňující obsah totiž příliš zasahoval do původního designu a pro malá zařízení docházelo k překrývání hlavní navigace.

Je zde vkládán soubor *page.html.twig* a to za využití proměnné `{{ page }}`. Krom toho je zde obsažen zápis tagu `<head>`, do kterého jsou vypisovány vložené knihovny *CSS* a *JS*.

block.html.twig

Tato šablona upravuje zobrazovaný článek. Jedná se o triviální zobrazení jeho názvu a obsahu. Byla zde provedena jednoduchá úprava úrovně nadpisu z `<h2>` na `<h1>`.

region.html.twig

V případě všech vypsáných *regions* docházelo k zabalení jejich obsahu do tagu `<div>`, který ve struktuře návrhu designu nebyl použit. Tato podoba HTML kódu je obsažena právě v souboru *region.html.twig*, v němž byl problematický `div` smazán.

menu.html.twig

Menu a jeho položky, které se vytvoří přes administrátorské rozhraní, jsou na webové stránky vkládány pomocí této šablony, či některých z následujících – v závislosti na jejich požadované struktuře. Všechna menu jsou vytvářena pomocí nesetříděného seznamu ``, viz Obr. 39. Kód vytváří rodičovskou hierarchickou podobu seznamů, jestliže dochází ke vnořování dalšího seznamu `` a současně vypisuje položky v podobě tagu ``.

```
{% import _self as menus %}
{% if items %}
  {% if menu_level == 0 %}
    <ul{{ attributes }} {{ dump(menu_name) }}>
  {% else %}
    <ul>
  {% endif %}
  {% for item in items %}
    <li{{ item.attributes }}>
      {{ link(item.title, item.url) }}
      {% if item.below %}
        {{ menus.menu_links(item.below, attributes, menu_level + 1) }}
      {% endif %}
    </li>
  {% endfor %}
</ul>
{% endif %}
```

Obr. 39: Cyklus vypisující menu a jeho položky (zdroj: vlastní)

Hlavní navigace: menu--main.html.twig a menu--hlavni-menu-ang-html.twig

Hlavní navigační menu je nutno vypisovat pomocí speciální šablony, neboť v jeho struktuře jsou oproti zbylým menu použity pomocné třídy *navmenu* a *podmenu*. Z toho důvodu jsou v šabloně vytvořeny tyto dva šablony závislé na atributu *Machine name*. Patrným nedostatkem této implementace je, že šablona je přizpůsobena na již hotový obsah webové stránky. v případě aplikování šablony KVD na jinou webovou stránku se jedná o potenciální chybu šablony a nezobrazování menu.

menu--banner.html.twig

Postranní panel sloužící k zobrazování obrázků ve své nabídce je implementován tímto souborem. Jedná se pouze o vypisování tagů `` obsahující odkaz s obrázkem ve svém těle. Nutnou podmínkou je, aby se titulek položky menu jmenoval jako požadovaný obrázek formátu *.png*, jenž se zde má zobrazovat.

Stejně jako v předchozí kapitole, také tento soubor byl vytvořen na základě *Machine name* předem vytvořeného menu.

block--system-menu-block.html.twig

Zde se nachází HTML zápis pro všechna menu. V předchozích souborech byla vytvářena pouze struktura jejich obsahu, zatímco tento soubor slouží k zobrazení již vytvořeného menu a jeho dodatečných informací. Jedná se především o název menu a prefixy. Vypsání názvu bylo realizováno v patičce webové stránky, kde se nacházejí tři pomocná menu.

block--local-tasks-block.html.twig

Tento soubor byl také upraven minimálně. Jedná se o další obalující kód, jenž vnořuje všechna menu do tagu `<nav>`. Ten bylo potřeba smazat.

block--search-form-block.html.twig

Tento soubor obsahuje zápis podoby vyhledávacího formuláře, jenž přepisuje defaultní vyhledávání implementované Drupalem. Stejně jako v předchozích redakčních systémech je potřeba upravit či přidat atributy jednotlivým ovládacím prvkům. Vzhledem na robustní zápis byly také upraveny CSS pravidla, aby se vztahovaly na tyto upravené elementy.

search-result.html.twig

V případě vyhledávání systém vkládá do regionu `{{ content }}` místo článků výsledky vyhledávání. Jedná se soupis článků, reprezentované útržkem textu a jejich nadpisem, jenž současně slouží jako odkaz. Podobu zápisu lze editovat v této šabloně, přičemž v rámci šablony pro KVD byl upraven pouze úroveň nadpisu z `<h1>` na `<h2>`.

Tyto výsledky lze pomocí rozšířeného vyhledávání dále filtrovat. Vzhled prvků byl ovšem výchozí, a proto ho bylo potřeba pomocí CSS upravit do odpovídající podoby.

Obr. 40: Ukázka vzhledu rozšířených možností vyhledávání (zdroj: vlastní)

breadcrumb.html.twig

V případě drobečkové navigace vkládám Obr. 41 a Obr. 42, aby bylo lépe patrné přepsání její základní podoby. Drobečková navigace byla v návrhu realizována pouze jako řada po sobě následujících odkazů. Systém Drupal jí realizuje pomocí setříděného číslovaného odkazu a současně zapisuje do tagu `<nav>` se zvýrazněním pomocí nadpisu druhé úrovně. Z hlediska sémantiky se jedná o dobře strukturovanou navigační část stránek, ale vzhledem k nízké důležitosti (vzhledem k ostatním navigacím v šabloně) jsem se rozhodl pro původní strukturu a realizovat drobečkovou navigaci pouze pomocí odkazů.

```
{% if breadcrumb %}
  <nav role="navigation" aria-labelledby="system-breadcrumb">
    <h2 id="system-breadcrumb" class="visually-hidden">{{ 'Breadcrumb'|t }}</h2>
    <ol>
      {% for item in breadcrumb %}
        <li>
          {% if item.url %}
            <a href="{{ item.url }}">{{ item.text }}</a>
          {% else %}
            {{ item.text }}
          {% endif %}
        </li>
      {% endfor %}
    </ol>
  </nav>
{% endif %}
```

Obr. 41: Výchozí struktura drobečkové navigace (zdroj: vlastní)

```
{% if breadcrumb %}
  {% for item in breadcrumb %}
    {% if item.url %}
      <a href="{{ item.url }}">{{ item.text }}</a>
    {% else %}
      {{ item.text }}
    {% endif %}
  {% endfor %}
{% endif %}
```

Obr. 42: Upravená struktura drobečkové navigace (zdroj: vlastní)

aintenance-page.html.twig

Tato šablona slouží pro zobrazování tzv. údržbové stránky, kdy jsou stránky odstaveny a je na nich prováděna nějaká činnost. HTML kód je velice podobný šabloně *page.html.twig*. Je rozšířen o třídy upravující vzhled do podoby stránky 404 v šabloně pro redakční systém Joomla. Vkládání informační zprávy o údržbě obstarává redakční systém.

8.3.5 JAZYKOVÉ VERZE

Pro vytvoření vícejazyčné verze šablony je zapotřebí patřičně upravit redakční systém Drupal. Tyto úpravy byly realizovány pomocí návodu sepsaný Stevem Burge na internetových stránkách <https://www.ostraining.com>.

Přepínač jazykových verzí se zobrazuje jako nesetříděný seznam ``, jehož položky `` jsou jednotlivé jazyky sloužící jako odkazy. Korektní zobrazování zkratky jazyka v regionu `{{ plan_switcher }}` probíhá za pomoci JS a CSS pravidla.

Kód JS vytvoří pole obsahující instance s třídou *language-link* – touto třídou jsou označeny tagy `<a>`. Pole se poté prochází cyklem *for*, přistupuje se ke každému prvku a nahrazuje se jeho obsah, název jazyka, prázdným řetězcem.

Zkratky jazyků jsou vkládány pomocí CSS, přičemž aktivní jazyk je skryt.

```
.jazyk ul.links li.cs a:after{color:White; content:"CS";}
.jazyk ul.links li.en a:after{color:White; content:"EN";}
```

Obr. 43: Vkládání zkratk jazyků (zdroj: vlastní)

Správa obsah pomocí nainstalovaných jazyků je velice podobná systému Joomla – k veškerému obsahu lze přidružovat jazyk. Navíc je umožněn ruční překlad jádra CMS, kdy jednotlivé položky lze z angličtiny přeložit do češtiny, viz Obr. 44.



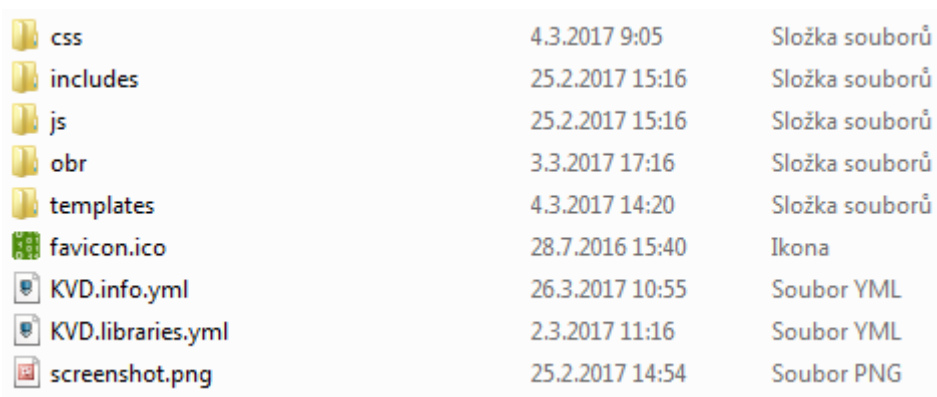
Obr. 44: Po nainstalování více jazyků je možnost překládat názvy a popisky jádra CMS (zdroj: vlastní)

8.3.6 OSTATNÍ SOUBORY ŠABLONY

V případě přizpůsobování zdrojových souborů pro šablonu Drupal bylo potřeba provést mnohem více úprav než v předchozích CMS. Hlavní příčinou je pracné dohledávání šablon upravující požadovaný kód. Proto byla zvolena úprava pomocí *JS* kódu, jenž zajistil potřebné chování. Jedná se především o funkci *predelejPatu()*, v níž se pomocí potomků přistupovalo k nadpisům první úrovně a bylo jim definováno *onClick* funkce skrývající či zobrazující dané menu.

Bohužel se nepovedlo do šablony implementovat kód třetí strany *doubletaptogo.js*, který přispíval k uživatelsky přívětivému menu pro zařízení s malým rozlišením.

Výsledná struktura šablony je zobrazena na Obr. 45.



css	4.3.2017 9:05	Složka souborů
includes	25.2.2017 15:16	Složka souborů
js	25.2.2017 15:16	Složka souborů
obr	3.3.2017 17:16	Složka souborů
templates	4.3.2017 14:20	Složka souborů
favicon.ico	28.7.2016 15:40	Ikona
KVD.info.yml	26.3.2017 10:55	Soubor YML
KVD.libraries.yml	2.3.2017 11:16	Soubor YML
screenshot.png	25.2.2017 14:54	Soubor PNG

Obr. 45: Struktura šablony pro redakční systém Drupal (zdroj: vlastní)

ZÁVĚR

Bakalářská práce byla zaměřena na redakční systémy umožňující správu webových stránek. Jejím primárním cílem bylo seznámit se s tvorbou šablon, s jejich neplacenými i placenými zdroji a možnostmi úprav z hlediska vzhledu a struktury.

V úvodní části byl stručně popsán kódovací jazyk HTML, jeho význam a účel vývoje nové verze označované HTML5. Ve stručnosti byly čtenáři také představeny nové vlastnosti jazyka kaskádových stylů, jehož nejnovější verze je známa pod označením CSS3. Tyto dvě kapitoly byly obsaženy z důvodu poskytnutí potřebných základních pojmů, které jsou nutné při tvorbě moderních webových stránek. Poté se pozornost zaměřila na základy responzibility, důvody její podpory a principy využití knihoven Bootstrap a W3.CSS. Toto téma bylo probráno více do hloubky seznámením se s dotazy na média. V práci byly také popsány jednotlivé prvky výsledného designu pro katedrální stránky KVD.

Následující kapitoly se věnovaly jednotlivým redakčním systémům, přičemž jsou seřazeny podle předpokládané složitosti. Tyto tři kapitoly mají stejnou strukturu podkapitol. Jak bylo již řečeno, tak obsahují zdroje šablon, možnosti editace přes administrátorské rozhraní a nakonec podrobný popis tvorby naší šablony z vlastního navržené a realizované stránky.

WordPress je svým uživatelsky přívětivým rozhraním vhodný pro začínající správce webových stránek. Má k dispozici mnoho témat a doplňků, pomocí kterých lze zajistit rozšiřující funkcionalitu bez nutnosti úprav vnitřních souborů šablon. Je ovšem vhodný i v případě rozsáhlejších změn šablony, neboť ty jsou kombinací HTML a PHP rychle srozumitelné.

Velice podobně by šlo charakterizovat také redakční systém Joomla. Administrační rozhraní je komplikovanější, poskytuje ale více možností a funkcí ke správě webu. Možnost přepisování jednotlivých modulů a komponent slouží k přehledné a jednodušší tvorbě či editaci šablony.

Redakční systém Drupal je svoji architekturou složitější než předchozí dva programy určené ke správě obsahu. Jeho šablona je tvořena soubory oddělující prezentační a aplikační vrstvu. Tím pádem její tvorba probíhala velice pomalu a pracně, neboť procházení oficiální anglické dokumentace bylo časově náročné. Nelze ho považovat za vhodný pro jednoduché stránky a ani pro uživatele očekávající jednoduché přizpůsobování šablon.

Závěrem lze dodat, že tvorba šablony pro tyto tři redakční systémy je závislá ryze na uživatelských znalostech o jednotlivých CMS. Protože se jedná o tři nejvíce využívané redakční systémy o široké uživatelské komunitě, lze mnoho nesnází řešit za jejich pomoci na oficiálních fórech či specializovaných stránkách. Při výběru redakčního systému bychom měli ovšem vzít v potaz především jejich funkci a vhodnost pro naše stránky.

RESUMÉ

This thesis is focused on editation and creation of templates for redaction systems like: WordPress, Drupal and Joomla. Its purpose is to educate readers with basic technologies, aspects and to provide principles of responsibility by making their own example.

Bachelor thesis is divided into eight chapters. First chapter is introduction - it defines goals and requirements, which were needed to do in the bachelor thesis.

Three chapters that follow are about ways of creation of web pages through usage of coding language HTML5 and cascading styles CSS3. Their advantages, main features and usage are shown in this chapter. Following this, the bachelor thesis is focusing responsibility of web pages. Open source packages like Bootstrap, W3.CSS and media queries are showcased here.

Fifth chapter is about criteria needed to be taken into consideration when working on graphical part of web page. Reader can learn main features of sections from this chapter. Lastly it is about thoroughly described final design of web pages which were used as source files for template creation.

Last three chapters are dedicated to options of redaction systems and last subchapter is about process of template creation. Structure and usage of particular template files are described there.

SEZNAM LITERATURY

- ALLAN, C. et al., 2010. *Build your own wicked Wordpress themes* [Dokument]. Collingwood. ISBN-13: 978-0-9804552-9-8.
- BOOTSTRAP CORE TEAM, 2011. About. *Get Bootstrap* [online] [cit. 2016-11-11]. Dostupné z: <http://getbootstrap.com/about/>
- CASTRO, E. a B. HYSLON, 2012. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press. ISBN 978-80-251-3733-8.
- Drupal.org, 2005. *Theme Editor* [online], verze 4. 4. 2055 [cit. 2017-02-27]. Dostupné z: https://www.drupal.org/project/theme_editor
- Drupal.org, 2016. *Theming Drupal 7* [online], verze 31. 8. 2016 [cit. 2017-02-25]. Dostupné z: <https://www.drupal.org/docs/7/theming>
- Drupal.org, 2016. *Theming Drupal 8* [online], verze 6. 3. 2017 [cit. 2017-02-26]. Dostupné z: <https://www.drupal.org/docs/8/theming>
- GASSTON, P., 2015. *Moderní web*. Brno: Computer Press. ISBN 978-80-251-4345-2.
- interval.cz, 2011. *10 nejlepších redakčních systémů (CMS)* [online], verze 9.11.2011 [cit. 2017-03-21]. Dostupné z: <https://www.interval.cz/clanky/10-nejlepsich-redakcnich-systemu-cms/>
- Joomla!, 2017. *Joomla! Documentation* [online]. 2016-2017, verze 2.12. 2016 [cit. 2017-02-14]. Dostupné z: https://docs.joomla.org/Main_Page
- KRUG, S., 2006. *Nenuťte uživatele přemýšlet!, 2. aktualizované vydání*. Brno: Computer Press. ISBN: 80-251-1291-8.
- Musilda, 2013. *Vytvoření drobečkové navigace ve WordPressu bez použití pluginu* [online], verze 7. 5. 2013 [cit. 2017-02-08]. Dostupné z: <https://musilda.cz/vytvoreni-drobeckove-navigace-ve-wordpressu-bez-pouziti-pluginu/>
- OStraining, 2015. *5 Steps to Build a Drupal 8 Multi-lingual Site* [online], verze 25. 9. 2015 [cit. 2017-03-07]. Dostupné z: <https://www.ostraining.com/blog/drupal/d8-multi-lingual/>
- SiteGround, 2017. *How to create a WordPress theme?* [online]. 2004-2017 [cit. 2017-02-04]. Dostupné z: https://www.siteground.com/tutorials/wordpress/wordpress_create_theme.htm
- STATCOUNTER GLOBAL STATS, 2016. StatCounter Global Stats. [Mobile and tablet internet usage exceeds desktop for first time worldwide] In: *Mobile and tablet internet usage exceeds desktop for first time worldwide* [online]. 1. 11. 2016 [cit. 2016-12-26]. Dostupné z: <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>
- STATCOUNTER GLOBAL STATS, 2016. StatCounter Global Stats. In: *Comparison in Czech Republic from Aug 2012 to Dec 2016 | StatCounter Global Stats* [online]. 1. 11. 2016 [cit. 2016-12-26]. Dostupné z: <http://gs.statcounter.com/#all-comparison-CZ-monthly-201208-201612>
- w3schools, 2017. *W3.CSS Tutorial* [online]. 1999-2017 [cit. 2016-11-10]. Dostupné z: <http://www.w3schools.com/w3css/>
- w3schools. *Bootstrap 3 Tutorial* [online]. [cit. 2016-11-11]. Dostupné z: <http://www.w3schools.com/bootstrap/default.asp>

WHATWG, 2017. HTML. *WHATWG* [online]. 2016-2017, verze 2. 4. 2017 [cit. 2017-04-02]. Dostupné z: <https://html.spec.whatwg.org/multipage/>

WordPress Codex. *Codex* [online]. [cit. 2017-02-05]. Dostupné z: <https://codex.wordpress.org/>

WordPress Developer. *Developer Resources* [online]. [cit. 2017-02-05]. Dostupné z: <https://developer.wordpress.org/>

SEZNAM OBRÁZKŮ

Obr. 1: Příklady rozvržení webové stránky (zdroj: vlastní)	8
Obr. 2: Ukázka proprietární vlastnosti zakulacení rohů (zdroj: vlastní)	10
Obr. 3: CSS pravidla využívaná pro tisk (zdroj: vlastní)	12
Obr. 4: Využití dotazu na média (zdroj: vlastní)	12
Obr. 5: Možnosti rozčlenění (Zdroj:W3C)	14
Obr. 6: Připojení souboru W3.CSS (zdroj: vlastní)	16
Obr. 7: Možnosti struktury gridů (Zdroj:W3C)	16
Obr. 8: Příklad načítání souborů CSS pro různé šířky (zdroj: vlastní).....	18
Obr. 9: Ukázka zapsání meta tagu viewport (zdroj: vlastní)	18
Obr. 10: Graf zobrazující nárůst prohlížení webu pomocí mobilů a tabletů. (zdroj: (STATCOUNTER GLOBAL STATS, 2016))	20
Obr. 11: Srovnání využití zařízení pro přístup k internetu v České republice. (zdroj: (STATCOUNTER GLOBAL STATS, 2016))	20
Obr. 12: Návrh hlavičky (zdroj: vlastní)	21
Obr. 13: Struktura menu (zdroj: vlastní)	23
Obr. 14: Návrh upravené hlavičky s menu pro menší rozlišení (zdroj: vlastní).....	24
Obr. 15: Ukázka dostupných šablon přes administrátorské rozhraní (zdroj: vlastní).....	28
Obr. 16: Editor souborů (zdroj: vlastní)	29
Obr. 17: Zobrazení manuálu a tlačítko pro ukládající úpravu souboru (zdroj: vlastní).....	29
Obr. 18a a obr 18b: Možnosti přizpůsobení šablony Twenty Sixteen (zdroj: vlastní)	30
Obr. 19: Kód vkládající příspěvky (zdroj: vlastní)	32
Obr. 20: Segment hlavičky, hlavní menu (zdroj: vlastní)	34
Obr. 21: Funkce <code>add_filter()</code> upravující vypisování menu (zdroj: vlastní)	34
Obr. 22: Vypisování prvního sloupečku v patičce stránek (zdroj: vlastní)	35
Obr. 23: Struktura šablony pro redakční systém WordPress (zdroj: vlastní)	37
Obr. 24: Přímá úprava dokumentu (zdroj: vlastní)	38
Obr. 25: Přepisování dokumentů redakčního systému Joomla (zdroj: vlastní).....	39
Obr. 26: Ochranný kód (zdroj: vlastní)	40
Obr. 27: Kód vkládající obsah pozice <code>druhy_sloupek</code> (zdroj: vlastní).....	41
Obr. 28: Vzhled rozbité hlavičky stránky na mobilních zařízeních (zdroj: vlastní).....	41
Obr. 29: Vzhled rozbité hlavičky pro monitory (zdroj: vlastní)	42
Obr. 30: Podmínky upravující vypisování menu (zdroj: vlastní)	44
Obr. 31: Přidání třídy modulu pro menu (zdroj: vlastní)	45
Obr. 32: Zabalení položek druhé úrovně do spanu (zdroj: vlastní)	46
Obr. 33: Struktura šablony pro redakční systém Joomla (zdroj: vlastní).....	46
Obr. 34: Editor barev šablony (zdroj: vlastní)	48
Obr. 35: Ukázka <code>.yml</code> souboru (zdroj: vlastní)	49
Obr. 36: Ukázka ze souboru <code>KVD.libraries.yml</code> (zdroj: vlastní).....	50
Obr. 37: Ukázka vlastnosti menu (zdroj: vlastní)	50
Obr. 38: Podmínka <code>if</code> v šabloně <code>page.html.twig</code> (zdroj: vlastní)	51
Obr. 39: Cyklus vypisující menu a jeho položky (zdroj: vlastní).....	52
Obr. 40: Ukázka vzhledu rozšířených možností vyhledávání (zdroj: vlastní)	53
Obr. 41: Výchozí struktura drobečkové navigace (zdroj: vlastní).....	54
Obr. 42: Upravená struktura drobečkové navigace (zdroj: vlastní).....	54
Obr. 43: Vkládání zkratk jazyků (zdroj: vlastní)	55

Obr. 44: Po nainstalování více jazyků je možnost překládat názvy a popisky jádra CMS (zdroj: vlastní).....	55
Obr. 45: Struktura šablony pro redakční systém Drupal (zdroj: vlastní)	56

PŘÍLOHY

Příloha 1: První návrh designu stránek KVD.

Příloha 2: Druhý návrh designu stránek KVD.

Příloha 3: Třetí návrh designu stránek KVD.

Příloha 4: Čtvrtý návrh designu stránek KVD.

Příloha 5: Schválený návrh designu stránek KVD.

Příloha 6: CD (Text bakalářské práce ve formátu DOCX a PDF).

ZČU | FAKULTA | ÚVOD | STUDIUM | ZAHRANIČNÍ VZTAHY | KONTAKTY

KVD
Katedra výpočetní a didaktické techniky
Pedagogická fakulta ZČU v Plzni

Poslední aktualizace: 26. března 2018 (11:54)

O nás
Lidé
Činnost
Student
Uchazeč

Základní informace

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY (KVD)

adresa: Katedra výpočetní a didaktické techniky
Fakulta pedagogická Západočeské univerzity v Plzni
Klatovská II. 51
308 19 Plzeň

tel: +420 377 636 501

fax: +420 377 636 402

Katedra provádí pedagogické, studijní, výzkumné a další činnosti. Hlavním úkolem je připsat a řídit oboru Učitelství informatiky pro základní a střední školy. Výuka probíhá jak v prezenční a kombinované formě studia tak i v rozšiřujícím studiu. Katedra také zajišťuje výuku certifikačního studia "Informatičtí vychovatelé" pro první i druhý stupeň základních škol.

Katedra se podílí na výuce v bakalářském studijním programu strukturovaného studia: Přirodovědné studia, studijní obor Informatika pro vzdělávání (prezenční forma).

Katedra zrealizuje studium učitelství v navazujícím magisterském studiu: Učitelství informatiky pro ZŠ a Učitelství informatiky pro SŠ (prezenční a kombinovaná forma).

Dále je možno konat státní rigorózní zkoušky a odložit tzv. PhD. u studijních programů učitelství pro základní školy (obor učitelství informatiky pro základní školy) a učitelství pro střední školy (obor učitelství informatiky pro střední školy).

Na katedře lze studovat i doktorský studijní program Specializace v pedagogice, studijní obor Informatičtí a komunikační technologie ve vzdělávacím prezenční a kombinované formě.

Pracovníci katedry se podílí na řešení různých projektů. Kromě toho organizuje katedra výuku základy výpočetní techniky pro studenty pedagogické fakulty a výuku v dalších předmětech podle požadavků ostatních kateder ZČU v Plzni. Hlavním oblasťou, na které je zaručena pomocnosť, jsou: Programování, teorie počítačů, analýzy a řízeníové prvky a systémy, technika počítačů, řízení a senzorace, zpracování textových a grafických informací, numerické metody, didaktika informatiky a multimediální systémy.

Informace
Historie
Mapa
Virtualizace
Fotogalerie

AS
AS
AS - učitelství
AS - Informatika
AS - Informatika

Copyright © 1992 - 2018 UNB Poutř
Univerzita Ji. P.ka, Česká Republika

Příloha 1: První návrh designu stránek KVD.

ZČU | FAKULTA | ÚVOD | STUDIUM | ZAHRANIČNÍ VZTAHY | KONTAKTY



[O NÁS](#)

[LIDÉ](#)

[ČINNOST](#)

[STUDENT](#)

[UCHAZEČ](#)

Prohledat stránku

Informace
Historie
Mapa
Virtualizace
Fotogalerie

Základní informace

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY (KVD)

Katedra výpočetní a didaktické techniky	
adresa	Fakulta pedagogická Západočeské univerzity v Plzni Klatovská 8 31 300 19 Plzeň
tel	+420 377 636 501
fax	+420 377 636 452

Katedra provádí pedagogické, studijní, výzkonné a další činnosti. Hlavním úkolem je příprava učitelů oboru Učitelství informatiky pro základní a střední školy. Výuka probíhá jak v prezenční a kombinované formě studia tak i v rozšiřujícím studiu. Katedra také zajišťuje výuku certifikačního studia "Informační výchova" pro první i druhé stupni základních škol.

Katedra se podílí na výuce v bakalářském studijním programu strukturovaného studia: Přirodovědná studia, studijní obor: Informatika pro vzdělávání (prezenční forma).

Katedra umožňuje studium učitelství v navazujícím magisterském studiu: Učitelství informatiky pro ZŠ a Učitelství informatiky pro SŠ (prezenční a kombinovaná forma).

Dále je možné konat státní rigorózní zkoušky a uchovávat titul PhDr. u studijních programů: Učitelství pro základní školy (obor Učitelství informatiky pro základní školy) a Učitelství pro střední školy (obor Učitelství informatiky pro střední školy).

Na katedře lze studovat i doktorský studijní program Specializace v pedagogice, studijní obor: Informační a komunikační technologie ve vzdělávání (prezenční a kombinovaná forma).

Pracovníci katedry se podílí na řadě různých projektů. Rovněž toho organizuje katedra výuku základů výpočetní techniky pro studenty pedagogické fakulty a výuku v dalších předmětech podle požadavků ostatních kateder ZČU v Plzni. Hlavními oblastmi, na které je zaměřena jsou: programování, tvorba počítačů, analogové a digitální prvky a systémy, technika počítačů, řízení a simulace, zpracování textových a grafických informací, sumerické metody, didaktika informatiky a multimediální systémy.

ODKAZY

[link odkazu](#)
[hover odkaz](#)
[vlnitý odkaz](#)

NECD

[link odkazu](#)
[hover odkaz](#)
[vlnitý odkaz](#)


NAZEV

[link odkazu](#)
[hover odkaz](#)
[vlnitý odkaz](#)

Podobná stránka: 25. srpna 2013

Příloha 2: Druhý návrh designu stránek KVD.

ZČU | FAKULTA | ÚVOD | STUDIUM | ZAHRANIČNÍ VZTAHY | KONTAKTY




Hledej zde na KVD...

O nás Lidé Činnost Student Uchazeč
 Informace Historie Mapa Virtualizace Fotogalerie

0 nás a Informace

Základní informace

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY (KVD)

Katedra výpočetní a didaktické techniky
adresa: Fakulta pedagogická Západočeské univerzity v Plzni Kláreňská 17, 31 301 19 Plzeň
tel: +420 377 636 501
fax: +420 377 636 452

Katedra provádí pedagogické, studijní, výzkumné a další činnosti. Hlavním cílem je příprava učitelů oboru Učitelství informatiky pro základní a střední školy, výuka probíhá jak v prezenční a kombinované formě studia tak i v rozličných studiích. Katedra také zajišťuje výuku certifikátového studia "Informační výchova" pro první a druhé stupně základních škol.

Katedra se podílí na výuce v bakalářském studijním programu strukturovaného studia: Přirodovědná studia, studijní obor: Informatika pro učitelství (prezenční forma).

Katedra umožňuje studium učitelství v navazujícím magisterském studiu: Učitelství informatiky pro ZŠ a Učitelství informatiky pro SŠ (prezenční a kombinovaná forma).

Dále je možné konat státní rigorózní zkoušky a ucházet titul PhDr. u studijních programů Učitelství pro základní školy (obor Učitelství informatiky pro základní školy) a Učitelství pro střední školy (obor Učitelství informatiky pro střední školy).

Na katedře lze studovat i doktorský studijní program Specializace v pedagogice, studijní obor Informační a komunikační technologie ve učitelství (prezenční a kombinovaná forma).

Pracovníkům katedry se podílí na řešení různých projektů. Kromě toho organizuje katedra výuku základů výpočetní techniky pro studenty pedagogické fakulty a výuku v dalších předmětech podle požadavků ostatních kateder ZČU a FUP. Hlavními oblastmi, na které je zaměřena specializace, jsou: Programování, teorie počítačů, analogové a digitální prvky a systémy, technika počítačů, řízení a simulace, zpracování textových a grafických informací, numerické metody, didaktika informatiky a multimediální systémy.

Kontaktujte nás

?

f

EN

↑

ODKAZY
[link odkazu](#)
[hover odkaz](#)
[vlnitý odkaz](#)

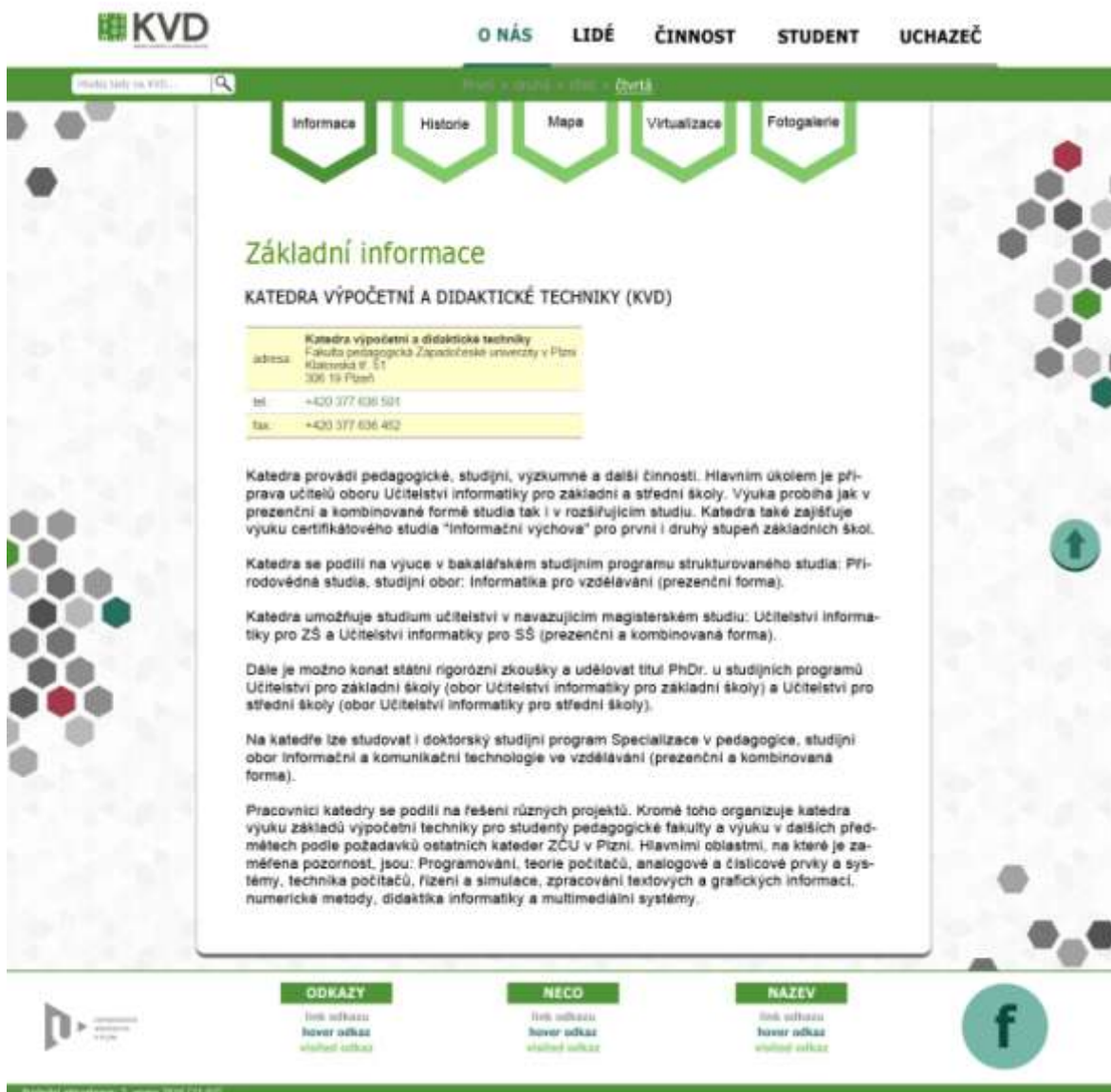
NECD
[link odkazu](#)
[hover odkaz](#)
[vlnitý odkaz](#)

NAZEV
[link odkazu](#)
[hover odkaz](#)
[vlnitý odkaz](#)

Podobly aktualizace: 13. srp 2016



Příloha 3: Třetí návrh designu stránek KVD.



Příloha 4: Čtvrtý návrh designu stránek KVD.

ZČU | FAKULTA | ÚVOD | STUDIUM | ZAHRANIČNÍ VZTAHY | KONTAKTY

KVD
Katedra výpočetní a didaktické techniky

EN

O NÁS LIDÉ ČINNOST STUDENT UCHAZEČ

Informace Historie Mapa Virtualizace Fotogalerie

O nás > Informace

Základní informace

KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY (KVD)

Katedra výpočetní a didaktické techniky
adresa: Fakulta pedagogická Jihočeské univerzity v Plzni Klatovská 17, 311 301 19 Plzeň
tel: +420 377 636 501
fax: +420 377 636 452

Katedra provádí pedagogické, studijní, výzkumné a další činnosti. Hlavním úkolem je příprava učitelů oboru Učitelství informatiky pro základní a střední školy. Výuka probíhá jak v prezenční a kombinované formě studia tak i v rozlišujícím studiu. Katedra také zajišťuje výuku certifikátového studia "Informační výchova" pro první i druhý stupeň základních škol.

Katedra se podílí na výuce v bakalářském studijním programu strukturovaného studia: Přírodovědná studia, studijní obor: Informatika pro vzdělávání (prezenční forma).

Katedra umožňuje studium učitelství v navazujícím magisterském studiu: Učitelství informatiky pro ZŠ a Učitelství informatiky pro SŠ (prezenční a kombinovaná forma).

Dále je možno konat státní rigorózní zkoušky a udělovat titul PhDr. u studijních programů Učitelství pro základní školy (obor Učitelství informatiky pro základní školy) a Učitelství pro střední školy (obor Učitelství informatiky pro střední školy).

Na katedře lze studovat i doktorský studijní program Specializace v pedagogice, studijní obor Informační a komunikační technologie ve vzdělávání (prezenční a kombinovaná forma).

Pracovní katedry se podílí na řešení různých projektů. Kromě toho organizuje katedra výuku základů výpočetní techniky pro studenty pedagogické fakulty a výuku v dalších předmětech podle požadavků ostatních kateder ZČU v Plzni. Hlavními oblastmi, na které je zaměřena pozornost, jsou: Programování, teorie počítačů, analogové a číslicové prvky a systémy, technika počítačů, řízení a simulace, zpracování textových a grafických informací, numerické metody, didaktika informatiky a multimediální systémy.

Kontaktujte nás

?

f

↑

ODKAZY
link odkazu
hover odkaz
všechny odkazy

NECD
link odkazu
hover odkaz
všechny odkazy

NAZEV
link odkazu
hover odkaz
všechny odkazy

Podobný obsah: 13. článek 2016

Příloha 5: Schválený design stránek.