



**Univerzita Tomáše Bati ve Zlíně**

## **Webový portál pro výuku hudební teorie**

Web portal for teaching music theory

Petr Látal

---

Diplomová práce

2011



**Univerzita Tomáše Bati ve Zlíně**  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2010/2011

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr LÁTAL**  
Osobní číslo: **A09440**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Učitelství informatiky pro základní a střední školy**

Téma práce: **Webový portál pro výuku hudební teorie**

Zásady pro vypracování:

1. Provedte návrh vhodné databázové struktury celého systému.
2. Naprogramujte základní administrační rozhraní pro správu obsahu celého systému na bázi PHP. Mezi důležité moduly budou patřit moduly pro správu obsahu na včetně uživatelských rolí, vyhledávací moduly, výukový modul a podkladový modul.
3. Vytvořte vhodný grafický návrh portálu.
4. Pokuste se vytvořit modul pro generování testů a jejich vyhodnocovací systém schopný generování notové osnovy s MIDI výstupy.
5. Provedte test vytvořeného systému a vytvořte kompletní dokumentaci ke správě systému.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ZENKL, Luděk. ABC hudební nauky . Ilustrovala Květa Kovářová. 2. vyd. Praha: Polygrafia, 1978. 199 s. ISBN 80-86385-21-3
2. ULLMAN, Larry. HTML : PHP a MySQL. Brno : Computer Press, 2004. ISBN 80-251-0063-4
3. KOLEKTIV AUTORŮ. Mistrovství v PHP 5. Computer Press, 2007. ISBN 978-80-251-1519-0
4. ZENKL, Luděk. ABC hudební nauky . Ilustrovala Květa Kovářová. 2. vyd. Praha: Polygrafia, 1978. 199 s. ISBN 80-86385-21-3
5. HTML [online]. Icit. 2011-31-01]. Dostupný z WWW: [http://cs.wikipedia.org/wiki/HyperText\\_Markup\\_Language](http://cs.wikipedia.org/wiki/HyperText_Markup_Language)
6. PHP [online]. Icit. 2011-31-01]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Php>
7. MySQL [online]. Icit. 2011-31-01]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Mysql>
8. ULLMAN, Larry. HTML : PHP a MySQL. Brno : Computer Press, 2004. ISBN 80-251-0063-4

Vedoucí diplomové práce:

**Ing. Dalibor Slovák**

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

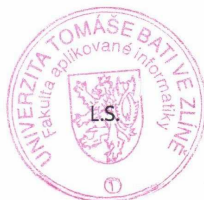
**24. února 2011**

Termín odevzdání diplomové práce:

**13. června 2011**

Ve Zlíně dne 24. února 2011

prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*



## **ABSTRAKT**

Diplomová práce byla vypracována jako webové řešení pro podporu výuky hudební teorie. V teoretické části jsem se zaměřil na popis webového redakčního systému a programovací jazyky spojené s tímto systémem. V praktické části pak bylo třeba vytvořit vhodnou databázovou strukturu kvízů a výukových kurzů. Dále bylo třeba vytvořit prostředí, ve kterém je učitel schopný tvořit notovou osnovu a webově ji prezentovat. Vše je propojeno s výukovým portálem, který umožňuje žákovi se registrovat, požádat si o vstup do příslušného ročníku, čerpat nezbytné výukové informace a testovat své znalosti pomocí testovacího modulu.

Klíčová slova:

HTML 5, PHP, MySQL, WYSIWYG, JQUERY, VEXFLOW, CUFON, AJAX, SMARTY, Open Source Software

## **ABSTRACT**

The thesis was developed as a Web-based support for teaching music theory. The theoretical part focused on the description of web content management systems and programming languages associated with this system. In the practical part, it was necessary to establish an appropriate data-base structure of authors, genres, music samples, or samples. It was necessary to create an environment in which the teacher is able to form a stave and presentate it on the web portal. Everything is connected with the learning portal, which allows the student to register to apply for entry into the class, receive the necessary information, training and test their knowledge through a test module.

Keywords:

HTML 5, PHP, MySQL, WYSIWYG, JQUERY, VEXFLOW, CUFON, AJAX, SMARTY, Open Source Software

Rád bych zde poděkoval panu Ing. Daliboru Slovákovi za zadání této diplomové práce, za vstřícnost a podporu při řešení problémů a koordinaci vstupních a výstupních dat systému.

Prohlašuji, že

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

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.

V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....  
podpis diplomanta

# OBSAH

ÚVOD.....	9
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 POUŽITÉ TECHNOLOGIE A SYSTÉMY .....</b>	<b>11</b>
1.1 CMS .....	11
1.1.1 Výhody CMS.....	12
1.1.2 Komunikace redakčního systému.....	13
1.1.3 Jazykový modul.....	13
1.2 HTML.....	14
1.2.1 Vývoj jazyka.....	14
1.2.2 Verze jazyka.....	14
1.2.3 Popis jazyka.....	16
1.2.3.1 Koncepce.....	16
1.2.3.2 Struktura dokumentu.....	17
1.2.3.3 Příklad zdrojového kódu.....	18
1.2.3.4 Druhy značek .....	18
1.2.4 Parsování v prohlížečích .....	19
1.2.5 Alternativní zařízení.....	20
1.2.6 Editory HTML.....	20
Textové editory .....	20
WYSIWYG editory.....	20
1.3 PHP.....	21
1.3.1 Ukázka kódu.....	21
1.3.2 Některé vlastnosti jazyka PHP .....	22
1.3.3 Významné projekty implementované v PHP .....	23
1.3.4 Výhody a nevýhody PHP.....	24
1.3.4.1 Výhody PHP .....	24
Nevýhody PHP.....	25
1.3.5 Historický vývoj PHP.....	25
1.4 MYSQL.....	27
1.4.1 Uložiště dat.....	28
1.4.2 Správa MySQL přes PhpMyAdmin .....	28
1.4.3 Příklady MySQL a PHP .....	28
1.4.3.1 Výběr databáze.....	28
1.4.3.2 Poslání SQL příkazu .....	29
1.4.3.3 Čtení výsledků SQL dotazu .....	29
1.5 SMARTY .....	30
1.6 JQUERY.....	31
1.6.1 Filozofie .....	31
1.6.2 Funkce .....	32
1.6.3 Funkce \$.....	32
1.6.4 Použití .....	33

1.7	VEXFLOW API .....	34
1.8	CUFON .....	34
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>36</b>
<b>2</b>	<b>SYSTÉM PRO SPRÁVU .....</b>	<b>37</b>
2.1	FUNKCE CMS.....	37
2.1.1	Vytváření obsahu.....	38
2.1.1.1	Vkládání obrázků .....	40
2.1.1.2	Nahrávání obrázků na server .....	42
2.1.1.3	Editace obrázků na serveru .....	43
2.1.2	Správa obsahu v databázi .....	44
2.1.3	Správa novinek .....	45
2.1.4	Uživatelé a práva .....	46
2.1.5	Fotogalerie.....	48
2.1.6	Správa složek .....	49
2.1.7	Správa modulů .....	50
2.1.7.1	Postup při vytváření nového modulu do systému .....	50
2.1.8	Nastavení webu .....	52
2.1.8.1	Nastavení jazyků.....	53
2.1.9	Správa kurzů.....	54
2.1.10	Generátor notových osnov.....	56
2.1.10.1	Tvorba notové osnovy.....	56
2.1.11	Kvízový modul.....	59
2.1.11.1	Tvorba nového kvízu .....	60
2.1.12	Publikování a prezentace.....	63
2.2	ZABEZPEČENÍ SYSTÉMU .....	64
2.3	PRÁCE S VEXFLOW .....	65
2.3.1	Krok 1: Základy pro tvorbu notové osnovy .....	65
2.3.2	Krok 2: Vkládání not.....	65
2.3.3	Krok 3: Vše o modifikátorech .....	67
2.3.4	Tabulky znaků .....	68
2.3.5	Krok 4: Zvuky a notace .....	68
2.3.6	Krok 5: Kytarová osnova .....	70
2.4	NÁVOD JAK PRACOVAT S PORTÁLEM PRO BĚŽNÉHO UŽIVATELE .....	72
	<b>ZÁVĚR .....</b>	<b>77</b>
	<b>CONCLUSION .....</b>	<b>78</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>79</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>81</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>83</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>85</b>



## ÚVOD

Tato diplomová práce byla vytvořena jako podpora výuky hudební teorie. Výukový portál slouží jako prostředí pro získání informací z hlediska základů hudební teorie. Portál je zaměřen na fulltextové vyhledávání, umí vytvářet testy, které následně vyhodnotí. Cílem této práce bylo vytvořit takové webové rozhraní, ve kterém by bylo možné registrovat žáky na portál, možnost zařadit žáky do výukových kurzů, tvořit testy a výukové podklady s kurzem spojené.

Webové rozhraní bude realizováno na serveru pomocí CMS systému využívajícího webové jazyky jako například PHP, MySQL a HTML 5. Veškeré funkce a moduly redakčního systému budou podrobně zdokumentovány. Velký důraz bude kladen na samotný layout a příjemné prostředí pro práci v administraci. Důležitou úlohu zde sehrají hlavně kvízové moduly, kurzový modul a modul pro generování notové osnovy.

Diplomová práce bude koncipována jako manuál s obrázky pro nasazení a bezproblémové využití systému na základních uměleckých školách či konzervatořích. Administrátor by měl být schopný jednoduše vkládat texty do příslušných sekcí, vytvářet potřebné kategorie a plnit systém daty, testy a notovými osnovami. Registrovaný uživatel by pak měl mít jednoduché rozhraní pro přihlašování do kurzů a intuitivní orientaci s vyhledáváním na webu.

## **I. TEORETICKÁ ČÁST**

## 1 POUŽITÉ TECHNOLOGIE A SYSTÉMY

Pro tvorbu diplomové práce bylo využito převážně programovacího jazyka PHP a HTML 5 spolu s databázovým jazykem MySQL. Administrační i uživatelské rozhraní využívá moderního javascriptového API JQUERY a WEXFLOW pro generování notových osnov.

### 1.1 CMS

Zkratka CMS pochází z anglického termínu *Content Management System* a označuje se tak různé systémy pro správu obsahu. CMS je vlastně složitou webovou aplikací, používanou pro vytváření a úpravy obsahu webu bez znalostí tvorby WWW stránek. Výhodou CMS je možnost rychlé reakce, web si můžete aktualizovat sami kdykoliv, ať je třeba neděle nebo svátek.

Pro internetové použití existuje celá řada nejrůznějších CMS, lišících se podle svých schopností. K předchůdcům CMS patří různá administrační rozhraní pro vkládání novinek a aktualit. Složitější CMS umožňují vkládání různých článků a spotů (včetně tabulek a obrázků) do předem určených kategorií, případně úpravu všech textů na webu. Používají se k tomu tzv. WYSIWYG editory. Podobné CMS bývají také součástí intranetů a extranetů.

Častou funkcí CMS je i správa obrázků a celých fotogalerií. Mnohé umí spravovat i audio a video soubory a další multimediální obsah. Výjimkou není ani možnost spravovat diskuze a komentáře. Ty nejlepší CMS dokáží přímo manipulovat se strukturou webu.

Okolo obecné terminologie systémů pro správu obsahu dnes existuje spousta nepřesností. Kromě CMS se rozlišují i další systémy – nejčastěji WCM (zaměřené pouze na správu webu) a DMS (správa dokumentů). Jako synonyma k CMS se používají i termíny redakční systém a publikační systém.<sup>[1]</sup>

Mezi základní funkce CMS systému patří například:

- Tvorba, modifikace a publikace dokumentů (článků) zpravidla prostřednictvím webového rozhraní, často s využitím jednoduchého online WYSIWYG editoru nebo jednoduchého systému formátování textu
- správa souborů jakýchkoliv typů (mp3,doc,pdf)
- správa obrázků či galerií
- kalendářní funkce
- statistika přístupů
- správa registrovaných uživatelů
- správa testů
- správa fulltextového vyhledávání

### 1.1.1 Výhody CMS

Velkou výhodou CMS systému je nepotřebná znalost jazyka HTML díky WYSIWYG editorům.

Většina CMS systémů je Open Source Software (OSS), což znamená, že zdrojové kódy jsou otevřené. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální dostupnost - licenci software, která umožňuje, při dodržení jistých podmínek, uživatelům zdrojový kód využívat, například prohlížet a upravovat. V nepřesném, ale poměrně běžném vyjadřování, se označení open source používá i pro mnoho vlastností, které s otevřeností zdrojového kódu nesouvisí, ale vyskytují se u mnoha open source programů. Například může jít o bezplatnou dostupnost software, vývoj zajišťovaný úplně nebo z podstatné části dobrovolnickou komunitou nebo „nekomerčnost“.<sup>[2]</sup>

### 1.1.2 Komunikace redakčního systému

Redakční systém byl vyvinut tak, aby co nejvíce zjednodušil práci při tvorbě nových modulů, aby komunikace mezi moduly byla zabezpečena proti neoprávněným přístupům a aby každá stránka administrace byla odlehčena o kód, který nemusí být načten. Výsledkem je systém, který komunikuje rychle a maximálně využívá optimalizovaných SQL dotazů.

Každý modul má své místo na ftp serveru ve složce /inc/admin/. Je velmi důležité, aby složka byla unikátní a obsahovala soubor s úpravami (musí se jmenovat upravy.php) a soubor s hláškami (musí se jmenovat hlasky.php). Pokud Vám budou tyto dva soubory chybět, budete na tento nedostatek upozorněni a vyzváni k nápravě. Dále je dobré mít vymyšlenou smysluplnou strukturu souborů pro přidávání, editaci a obecný přehled modulu. V neposlední fázi je pak třeba SuperAdmin účtu, který Vám umožní pomocí „Nastavení Modulů“ vložit nebo editovat daný modul.

### 1.1.3 Jazykový modul

V hlavičce načítaného souboru system.php je načítán jazykový modul, který si nejprve pomocí php kódu vyfiltruje typ země, ze které je stránka načítána a do superglobální proměnné „lang“ vloží hodnotu v podobě dvoupísmenového řetězce :

Česká republika - cs

Slovenská republika – sk

Ostatní země – en

Dále pak se dotáže v databázi tabulky „layout“ a zjistí zda jsou pro daný jazyk vytvořeny jazykové mutace web layoutu. Pokud nenalezne shodu se superglobální proměnnou „lang“, využije výchozího jazyka (v případě portálu hudební nauka online se jedná o češtinu).

Z databáze načte veškerá data a pomocí pole \$langcore[„navez\_bunky“] v kódu zobrazí hodnotu buňky.

Na webu je možné pomocí jednoduchého přepínače či odkazu měnit jazykovou strukturu.

## 1.2 HTML

**HyperText Markup Language**, označovaný zkratkou **HTML**, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.

Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (*Standard Generalized Markup Language*). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

### 1.2.1 Vývoj jazyka

V roce 1989 spolupracovali Tim Berners-Lee a Robert Caillau na propojeném informačním systému pro CERN, výzkumné centrum fyziky poblíž Ženevy ve Švýcarsku. V té době se pro tvorbu dokumentů obvykle používaly jazyky TeX, PostScript a také SGML. Berners-Lee si uvědomoval, že potřebují něco jednoduššího a v roce 1990 byl tedy navržen jazyk HTML a protokol pro jeho přenos v počítačové síti – HTTP (*HyperText Transfer Protocol – přenosový protokol hypertextu*). Zároveň také Tim Berners-Lee napsal první webový prohlížeč, který nazval WorldWideWeb.

V roce 1991 CERN zprovoznil svůj web. Současně organizace NCSA (*National Center for Supercomputer Applications*) vybídla Marca Andreessena a Erica Binu k vytvoření prohlížeče Mosaic; ten vznikl v roce 1993 ve verzích pro počítače IBM PC a Macintosh a měl obrovský úspěch. Byl to první prohlížeč s grafickým uživatelským rozhraním.

Následoval rychlý rozvoj webu, takže bylo nutné pro HTML definovat standardy.

### 1.2.2 Verze jazyka

Verze 0.9

Byla vydána zhruba v roce 1991. Nepodporuje grafický režim (verze, kterou vytvořil Tim Berners-Lee).

Verze 2.0

Zachycuje stav jazyka v polovině roku 1994. Standard vydala komunita IETF (*Internet Engineering Task Force*). Je to první verze, která odpovídá syntaxi SGML. Přidává k původní specifikaci interaktivní formuláře a podporu grafiky.

#### Verze 3.2

Byla vydána 14. ledna 1997 a zachycuje stav jazyka v roce 1996. Připravovaná verze HTML 3.0 nebyla nikdy přijata jako standard, protože byla příliš složitá a žádná firma nebyla schopna naprogramovat její podporu. Standard už vydalo W3C, stejně jako následující verze. Přidává k jazyku tabulky, zarovnávání textu a stylové elementy pro ovlivňování vzhledu.

#### Verze 4.0

Byla vydána 18. prosince 1997. Do specifikace jazyka přibyly nové prvky pro tvorbu tabulek, formulářů a nově byly standardizovány rámy. Tato verze se snaží dosáhnout původního účelu – prvky by měly vyznačovat význam jednotlivých částí dokumentu, vzhled má být ovlivňován připojovanými styly. Některé prezentační elementy byly zavrženy.

#### Verze 4.01

Byla vydána 24. prosince 1999. Tato verze opravuje některé chyby verze předchozí. Podle původního předpokladu se mělo jednat o poslední verzi, po které by se přešlo na XHTML.

#### Verze 5

7. března 2007 byla založena nová pracovní skupina HTML, jejíž cílem je vývoj nové verze HTML. V květnu 2007 bylo odhlasováno, že základem nové specifikace se stanou Web Applications 1.0 a Web Forms 2.0 ze specifikace WHATWG. Jako název nové specifikace bylo odhlasováno HTML 5. Specifikace by měla být hotova v letech 2010-2012, ukončení vývoje specifikace po vyřešení problémů a opravení všech chyb se odhaduje až na rok 2022).

## 1.2.3 Popis jazyka

### 1.2.3.1 Koncepce

Jazyk HTML je od verze 2.0 aplikací SGML (připravovaná verze HTML5 ale již závislost na SGML obsahovat nebude). Je charakterizován množinou značek a jejich atributů definovaných pro danou verzi. Mezi značky se uzavírají části textu dokumentu a tím se určuje význam (*sémantika*) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky (< a >). Část dokumentu tvořená otevírací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. *element* (prvek) dokumentu. Například <strong> je otevírací značka pro zvýraznění textu a <strong>Červená Karkulka</strong> je element obsahující zvýrazněný text. Součástí obsahu elementu mohou být další vnořené elementy. Atributy jsou doplňující informace, které upřesňují vlastnosti elementu.

Značky (zvané tagy) jsou obvykle párové (v XHTML jsou párové všechny), přičemž koncové značka je shodná se značkou počáteční, jen má před názvem znak lomítka. Příklad pro označení odstavce:

```
<p>Text odstavce</p>
```

Některé značky jsou nepárové – nemají žádný obsah a nepoužívají koncovou značku. Příklad pro vykreslení vodorovné čáry:

```
<hr>
```

Tagy mohou obsahovat atributy, které popisují jejich vlastnosti nebo nesou jinou informaci. Příkladem může být odkaz (tag a), jehož atribut href říká, kam se uživatel po kliknutí na něj dostane (v tomto příkladu na stránku <http://example.com>):

```
<a href="http://example.com">text odkazu</a>
```

Nebo jiné možnosti zápisu odkazu - odkaz, který se otevře v novém okně/panelu:

```
<a href="http://example.com" target="_blank">text odkazu</a>
```

Nebo jiné možnosti zápisu odkazu - odkaz, který se otevře v novém okně/panelu, obsahuje titulek (zobrazí se po najetí myší) a bude přiřazen ke třídě CSS:

```
<a href="http://example.com" target="_blank" title="titulek"
class="nazev_tridy">text odkazu</a>
```

Od SGML zdědil jazyk HTML i jiné, méně známé konstrukce pro tvoření elementů. Jedná se o tzv. *zkrácené HTML zápisy*:



- `<element>Obsah/` je totéž jako `<element>Obsah</element>`
- `<element>Obsah</>` je totéž jako `<element>Obsah</element>`
- `<el<el2>Obsah` je totéž jako `<el><el2>Obsah`
- `<ul><li>Adam<>Božena</ul>` je totéž jako `<ul><li>Adam</li><li>Božena</li></ul>`

Všechny tyto zápisy jsou sice podle normy validní a zcela ekvivalentní, ale žádný ze známých prohlížečů zkrácené verze nepodporuje, takže se nedoporučuje je používat.

Pro každou verzi existuje definice pravidel DTD (*Document Type Definition*). Od verze 4.01 musí být odkaz na deklaraci DTD v dokumentu uveden pomocí klíčového slova `DOCTYPE`. DTD definuje pro určitou verzi elementy a atributy, které lze používat.

Dokument může mimo značkování obsahovat další prvky:

- Direktivy – začínají znaky `<!`, jsou určeny pro zpracovatele dokumentu (prohlížeč).
- Komentáře – pomocné texty pro programátora, nejsou součástí obsahu dokumentu a nezobrazují se (prohlížeč je ignoruje). Příklad komentáře je uveden níže.
- Kód skriptovacích jazyků.
- Definice událostí a kód pro jejich obsluhu.

### 1.2.3.2 *Struktura dokumentu*

Dokument v jazyku HTML má předepsanou strukturu:

- Deklarace DTD – je povinná až ve verzi 4.01, je uvedena direktivou `<!DOCTYPE`.
- Kořenový element – element `html` (značky `<html>` a `</html>`) reprezentuje celý dokument. Kořenový element je povinný, ale otevírací a ukončovací značka samotná povinná není (pokud tyto značky nebudou v těle dokumentu uvedeny, prohlížeč si je sám doplní podle kontextu).
- Hlavička elementu – obsahuje metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl

zobrazení. Hlavička je uzavřena mezi značky `<head>` a `</head>`. Element `head` je opět povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.

- Tělo dokumentu – obsahuje vlastní text dokumentu. Vymezuje se značkami `<body>` a `</body>`. Element `body` je povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.

### 1.2.3.3 Příklad zdrojového kódu

Příklad HTML dokumentu ve verzi 4.01:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <!-- toto je komentář -->
  <head>
    <title>Titulek stránky</title>
  </head>
  <!-- tělo dokumentu -->
  <body>
    <h1>Nadpis stránky</h1>
    <p>Toto je tělo dokumentu</p>
  </body>
</html>
```

### 1.2.3.4 Druhy značek

Značky lze z hlediska významu rozdělit na tři základní skupiny:

- Strukturální značky
  - rozvrhují strukturu dokumentu, příkladem jsou odstavce (`<p>`), nadpisy (`<h1>`, `<h2>`). Dodávají dokumentu formu.
- Popisné (sémantické) značky
  - popisují povahu obsahu elementu, příkladem je nadpis (`<title>`) nebo adresa (`<address>`). Současný trend je orientován právě na sémantické značky, které usnadňují automatizované zpracovávání dokumentů a vyhledávání informací v záplavě dokumentů na webu. Vyvrcholením této snahy je v současné době jazyk XML.

- Stylistické značky
  - určují vzhled elementu při zobrazení, typickým příkladem je značka pro tučné písmo (<b>). Od tohoto druhu značek se postupně upouští, trendem je používání kaskádových stylů, které vzhled popisují odděleně od obsahu dokumentu. Mezi důvody pro neuvádění těchto značek patří především to, že tyto značky jsou orientovány na prohlížení na obrazovce počítače, příliš se však nepočítá s používáním dokumentu jiným způsobem – alternativní prohlížeče pro postižené (čtečky pro slepce), v mobilních zařízeních a podobně. Kaskádové styly umožňují definovat rozdílné zobrazení pro různá zařízení.

#### 1.2.4 Parsování v prohlížečích

Webové prohlížeče jsou programy, jejichž účelem je prezentovat dokument na zobrazovacím zařízení – nejčastěji monitoru počítače. Dokument je prohlížečem načítán a prováděn jeho rozklad (syntaktická analýza) na jednotlivé elementy. Prohlížeč obsahuje tabulku značek, které podporuje. Moderní prohlížeče (Opera, Mozilla Firefox) dokonce umožňují „vytvářet“ vlastní tagy a elementy a umožňují také jejich stylování pomocí kaskádových stylů. Jelikož však nejrozšířenější prohlížeč, Windows Internet Explorer, toto nedovoluje a obsah neznámých elementů zobrazí zcela normálně, bez stylu, vlastní elementy se prakticky nepoužívají.

Každému elementu je poté přiřazen styl (způsob zobrazení). Styly mohou být uvedeny ve stylovém předpisu. Vlastnosti stylů, které nejsou předepsány, doplní prohlížeč podle implicitního stylu, který má zabudován. Některé prohlížeče umožňují uživateli implicitní styly definovat.

Novější prohlížeče pracují obecně ve dvou základních režimech:

- *Standardní režim* – režim snažící se dodržovat definované standardy;
- *Quirk mód* – režim zaměřený na zpětnou kompatibilitu, i pokud takové chování není v souladu se standardy.

Tyto režimy chování zavedl Internet Explorer ve své páté verzi z důvodu zpětné kompatibility. Microsoft při vytváření nové verze prohlížeče chtěl, aby se v něm zobrazovaly správně

stránky, které již existují, ale také nové stránky, které jsou psány podle standardů. To, jaký režim prohlížeč zvolí, závisí především na uvedení direktivy `<!DOCTYPE`, protože většina starších stránek ji vůbec neobsahuje. Později toto chování částečně přebrala Mozilla i Opera, kde však rozdíly mezi režimy nejsou natolik markantní. Existence nestandardního režimu je výsledkem vývoje, kdy si výrobci jednotlivých prohlížečů přizpůsobovali definici HTML podle svých potřeb a prohlížeče podporovaly nestandardní elementy a syntaxi. Řada těchto „vylepšení“ byla následně přejímána do standardů, některé však byly zase v dalších verzích vyřazeny.

### 1.2.5 Alternativní zařízení

Vzhledem k rozvoji jak na poli software, tak i hardware přibýlo mnoho nových možností zpracování HTML dokumentů. Proto se pojem „prohlížeč“ stal zavádějícím (jsou např. i hlasové čtečky, či tiskárny) a W3C začalo používat termín *user agent*. V tuto chvíli ještě neexistuje jednotný český termín, používá se buď doslovný překlad – „uživatelský agent“, nebo např. „interpret“, „interpretr“, „zařízení“ či jiné.

### 1.2.6 Editory HTML

Editory HTML jsou programy pro snadnou tvorbu webových stránek. Například PSPad, Macromedia Dreamweaver a další.

#### *Textové editory*

Editorem HTML může být ve své podstatě jakýkoliv program pracující s textem. V praxi se však používají mnohem sofistikovanější programy. Běžný editor HTML zvládá barevnou syntaxi (barevně rozlišuje jednotlivé části kódu jako například HTML značky či atributy a prostý text), dokáže napovídat značky, zná chytré tabulátory nebo zvládá validovat dokument podle předepsané DTD. Mezi takové nejrozšířenější editory HTML u nás určitě patří PSPad.

#### *WYSIWYG editory*

WYSIWYG je zkratka od anglického „What you see is what you get“, v překladu „Co vidíš, to dostaneš“. Tyto WYSIWYG editory pracují na opačném principu než textové

editory – ve WYSIWYG editoru pracujete s již hotovou stránkou a obecně neplatí, že by uživatel tohoto editoru musel znát jazyk HTML. Ve WYSIWYG editoru si může uživatel poskládat stránku jak se mu zlíbí a program následně vygeneruje požadovaný kód HTML. Mezi nejznámější takovéto editory tohoto typu patří *Adobe Dreamweaver* nebo *Expression Web* (novější verze *Microsoft Frontpage*).<sup>[3]</sup>

### 1.3 PHP

**PHP** (rekurzivní zkratka *PHP: Hypertext Preprocessor*, „PHP: Hypertextový preprocesor“, původně *Personal Home Page*) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což lze využít při tvorbě webových aplikací. PHP lze použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou většinou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti (interpret PHP skriptu je možné volat pomocí příkazové řádky). Syntaxe jazyka je inspirována několika programovacími jazyky. PHP je nezávislý na platformě, skripty fungují bez větších úprav na mnoha různých operačních systémech. Podporuje mnoho knihoven pro různé účely - např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP...)

PHP se stalo velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. V kombinaci s operačním systémem Linux, databázovým systémem (obvykle MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Pro tuto kombinaci se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP nebo Perl.

#### 1.3.1 Ukázka kódu

Takto v PHP vypadá skript Hello world:

```
<?php
    echo "Ahoj, světe!";
```

?>

### 1.3.2 Některé vlastnosti jazyka PHP

- Jazyk PHP je dynamicky typový tzn. že datový typ proměnné se určí v okamžiku přiřazení hodnoty.
- Kvůli tomu má PHP dva operátory porovnání - '==' a '==='; při použití prvního dochází před porovnáním ke konverzi, při použití druhého je výraz pravdivý jen když jsou oba dva operandy stejného datového typu a jejich obsah má stejnou hodnotu.
- Pole se dají indexovat číselnými indexy (jako v jazyce C), nebo mohou fungovat jako hash-mapa. Stejně pole může obsahovat oba typy indexů.
- Pole jsou heterogenní (stejně pole může obsahovat prvky různých typů).
- Řetězce lze uzavírat do uvozovek (při vyhodnocení se provede nahrazení proměnných uvnitř) nebo do apostrofů (nahrazuje se jen escape sekvence \).

```
// Zde je v proměnné string (tečka je operátor spojování řetězců)
$retez = "Ahoj, světe" . ', mám se dobře' . " a nevadí, že střídám
oddělovače";

// Zde je v proměnné číslo (int)
$cislo = 100;

// Do proměnné je možné dát pole, které obsahuje jak čísla, tak
znaky či další pole
$pole = array('a', 'b', 1, 2, array('první' => 'podpole', 'vytiště-
no'));

// Nenahlásí chybu (jenom varování) a vytiskne 'Array'
print($pole);

// Vytiskne obsah proměnné pole
print_r($pole);

// Test porovnání
$cislo = 100;
$retez = '100';

// Toto porovnání ('==') platí díky automatické typové konverzi
if ($retez == $cislo)
```

```
{
    echo 'Jsou stejné';
}

// Ale porovnání pomocí '===' neplatí, neboť nejsou stejné typy
if ($retez === $cislo)
{
    echo 'To by nešlo';
}
```

- PHP do verze 4.2.0 ve výchozím nastavení automaticky přejímalo veškeré proměnné poslané jakoukoliv metodou (HTTP POST, HTTP GET, HTTP cookie, ale i ze zabudovaného mechanismu sessions) a umožňovalo s nimi dále pracovat jako s globálními - tato možnost představovala bezpečnostní riziko.
- Od verze 4.2.0 lze hodnotu získat z tzv. superglobálních proměnných s garancí původu informace - tedy že data byla odeslána požadovanou metodou. Používání globálních proměnných je stále možné pomocí konfigurační direktivy `register_globals` povolit, ale z bezpečnostních důvodů je to silně nedoporučováno.

```
// odešlu formulář metodou POST, kde do pole s názvem jmeno vepíšu 'Tom'
echo $jmeno; // vrátí 'Tom', funguje pouze v případě povolených globálních proměnných
echo $_POST['jmeno'];

// vrátí 'Tom', superglobální proměnné fungují i při vypnutých globálních proměnných
echo $_GET['jmeno'];

// vypíše se chybové hlášení o neexistenci proměnné a vrátí NULL
// NULL je zvláštní hodnota libovolného typu proměnných pro stav 'nedefinováno'
```

### 1.3.3 Významné projekty implementované v PHP

- software MediaWiki — např. Wikipedie, tato webová encyklopedie
- phpBB — balík pro provoz webového fóra
- WordPress — publikační systém pro provoz blogů a podobných aplikací

- phpMyAdmin —webová aplikace pro správu databázového systému MySQL
- Taxy! — překladač intuitivní syntaxe pro formátování textu na HTML

### 1.3.4 Výhody a nevýhody PHP

#### 1.3.4.1 Výhody PHP

- PHP je specializované na webové stránky
- rozsáhlý soubor funkcí v základní knihovně PHP
- nativní podpora mnoha databázových systémů
- multiplatformost (zejména Linux, Windows)
- možnost využití nativních funkcí operačního systému (možná nekompatibilita s jiným OS)
- strmá křivka učení
- obrovská podpora na hostingových službách – PHP je fakticky standardem, který najdeme všude
- obrovské množství projektů a kódů, které lze zdarma využít (WordPress, phpBB a další)
- poměrně slušná dokumentace
- vynikající, velmi svobodná licence – svobodnější, než GPL



### *Nevýhody PHP*

- jazyk PHP není nikde definován, je popsán pouze jeho implementací
- mírně nekonzistentní vývoj v minulosti, který si sebou PHP nese dosud (často se měnící příkazy atd...)
- nekonzistentní pojmenování funkcí a nejednotné pořadí parametrů
- ač jazyk výborně podporuje výjimky, jeho knihovna je používá jen zřídka
- slabší podpora Unicode, pouze přes PHP knihovnu (v PHP 6 má být Unicode řetězec jako základní typ)
- neumožňuje překlad do byte kódu, PHP skript se při každém požadavku překládá znovu
- ve standardní distribuci chybí ladící (debugovací) nástroj
- po zpracování požadavku neudrží kontext aplikace, vytváří jej vždy znovu (oslabuje výkon)
- nepodporuje jmenné prostory (ve verzi PHP 5.3 jsou podporovány)

#### **1.3.5 Historický vývoj PHP**

PHP bylo původně označení pro Personal Home Page, tedy osobní domácí stránky. Vše začalo v roce 1994, kdy byla napsána binární část Common Gateway Interface (CGI) v programovacím jazyku C. Tuto prvotní část napsal dánský/grónský Programátor Rasmus Lerdorf. Lerdorf zpočátku vytvořil tyto nástroje pro osobní domácí stránky (Personal Home Page) za účelem možné záměny s malou skupinou skriptů v Perlu, které chtěl používat pro údržbu osobní domovské stránky. Nástroje měly zajistit běh úloh jako například zobrazení résumé (obsahu) a zaznamenávání návštěvnosti jeho stránek. Tento binární kód ještě tentýž rok skloubil s jiným programem, který sám napsal. Po spojení s Form Interpreter tak vznikla kombinace PHP/FI, která měla mnohem větší funkčnost. PHP/FI obsahovala širokou implementaci pro programovací jazyk C a navíc tato verze mohla komunikovat s databázemi, což umožnilo tvorbu prvních jednoduchých dynamických webových aplikací.

Lerdorf veřejně vydal PHP 8. června 1995, aby mohl najít co nejvíce chyb a tak zdokonalil kód. Tato verze byla pojmenována jako PHP verze 2 a již měla základní funkčnost jako má dnešní PHP. To zahrnuje například proměnné ve stylu Perlu, zpracování formulářů a možnost vložit HTML kód. Syntaxe byla obdobná jako u Perlu, ale byla omezenější, jednodušší a méně konzistentní.

Zeev Suraski a Andi Gutmans, dva Izraelští vývojáři na Technion IIT, přepsali parser v roce 1997, vytvořili tak základ PHP 3 a změnili název jazyka na rekurzivní zkratku PHP = PHP: Hypertext Preprocessor. Tým vývojářů oficiálně vydal PHP/FI 2 v Listopadu 1997 po měsíčním testování beta verze. Poté začalo veřejné testování PHP 3, a její oficiální uvolnění přišlo v červnu 1998. Zeev Suraski a Andi Gutmans poté začali opětovně přepisování jádra PHP a vydali Zend Engine v roce 1999. Založili firmu Zend Technologies v Ramat Gan, Izrael.

Dne 22. května 2000 byla vydána verze PHP 4 postavená na Zend Engine 1.0. Dne 13. června 2004 byla představena verze PHP 5, která již stojí na novém Zend Engine II. PHP 5 obsahuje nové rysy jako je vylepšená podpora pro objektově orientované programování, PHP Data Objects extension (ta definuje lehké a konzistentní rozhraní pro napojení k databázím) a nesčetné množství výkonových vylepšení. PHP 4 se již dále nevyvíjí a pro tuto verzi se nebudou vydávat ani žádné bezpečnostní aktualizace.

V roce 2008 se stává PHP 5 jedinou stabilní verzí, která se vyvíjí. Později se zjistilo, že zde chybí static binding a proto byly přidány v PHP 5.3. PHP 6 se bude zároveň vyvíjet s verzí PHP 5. Mezi hlavní změny patří odebrání register\_globals, magické uvozovky a safe mode.

PHP ještě plně nepodporuje Unicode nebo multibyte strings; podpora unicode bude zahrnuta až do verze PHP 6. Spousta kvalitních open sourcových projektů pozastavilo podporu PHP 4 ve svých nových kódech od 5. února 2008. Aby jim konsorcium vývojářů PHP usnadnilo přechod na PHP 5 poskytlo jim přechodovou verzi z PHP 4 na PHP 5.

PHP 5 běží jak ve 32-bitovém tak i v 64-bitovém prostředí, ale jedinou oficiální verzí pro Windows je 32-bitová verze, vyžadující mód kompatibility Windows 32-bit při použití IIS v 64-bitovém prostředí Windows. K dispozici je verze třetí strany, která je určena pro 64-bitové Windows.<sup>[4]</sup>

## 1.4 MySQL

**MySQL** je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci.

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet. <sup>[5]</sup>

Přehled podporovaných vlastností:

- cizí klíče (od verze 3.23 podporovány v tabulkách typu InnoDB)
- transakce (od verze 3.23 podporovány v tabulkách typu InnoDB)
- podpora různých znakových sad a časových pásem v datech (od verze 4.1)
- poddotazy (od verze 4.1)
- uložené procedury (od verze 5.0)
- trigger (od verze 5.0)
- pohledy (od verze 5.0)

- práce s metadaty (od verze 5.0)

### 1.4.1 Uložiště dat

MySQL nabízí několik typů databázových tabulek (storage engine), které se liší svými možnostmi, použitím a způsobem ukládání dat do souborů:

- MyISAM - nejpoužívanější, bez podpory transakcí
- InnoDB - podpora transakcí
- BerkeleyDB (BDB)
- MEMORY - práci s daty v paměti
- NDB Cluster - úložiště pro clusterované databáze (od verze 5.0)
- ARCHIVE - komprimované tabulky, bez podpory indexů
- CSV - ukládání dat v prostých textových souborech

### 1.4.2 Správa MySQL přes PhpMyAdmin

Pro jednoduchou správu MySQL databází se používá nástroj PhpMyAdmin. PhpMyAdmin je Open Source program napsaný v PHP, který umožňuje zálohování, vytváření tabulek, vkládání, editaci a mazání záznamů v tabulkách, vytváření databází apod. PhpMyAdmin je pokročilý nástroj pro kompletní správu MySQL systému přes webové rozhraní.

### 1.4.3 Příklady MySQL a PHP

#### 1.4.3.1 Výběr databáze

Abychom mohli s určitou databází pracovat, musíme ji vybrat. Každý databázový server může spravovat mnoho databází. Proto je dobré si hned po připojení databázi vybrat. K tomu slouží funkce **mysql\_select\_db**:

```
mysql_select_db ( název_databáze, identifikátor_spojení )
```

### 1.4.3.2 Poslání SQL příkazu

K zaslání SQL příkazu můžeme použít funkci **mysql\_query**:

```
mysql_query ( sql_příkaz, identifikátor_spojení )
```

Funkce **mysql\_query** vrací buď identifikátor výsledku, pokud SQL příkaz vrací nějaká data, nebo pouze hodnotu **true**, pokud SQL dotaz nevrací žádná data. Při chybě vrací funkce **mysql\_query** vždy hodnotu **false**.

### 1.4.3.3 Čtení výsledků SQL dotazu

Pro čtení výsledků SQL dotazu existují dvě šikovní funkce, a to **mysql\_fetch\_row** a **mysql\_fetch\_array**. Obě se nejčastěji používají s jedním parametrem:

```
mysql_fetch_row ( identifikátor_výsledku )
```

```
mysql_fetch_array ( identifikátor_výsledku )
```

Obě funkce načtou další řádek z výsledků SQL dotazu a vrací buď pole s hodnotami, a nebo **false**, pokud bylo už všechno přečteno. Liší se od sebe jen tím, že **mysql\_fetch\_row** načte hodnoty do pole s číselnými indexy, zatímco funkce **mysql\_fetch\_array** přidá i indexy s názvy sloupců.

**Použití funkce `MySQL_Fetch_Row()`**; pro výpis všech hodnot tabulky:

```
$vysledek = MySQL_DB_Query("php", $sql, $spojeni);  
while($zaznam = MySQL_Fetch_Row($vysledek)):  
    echo $zaznam[0]. "<br>";  
    echo $zaznam[1]. "<br>";  
    echo $zaznam[2]. "<br>";  
    echo $zaznam[3]. "<br><br>";  
endwhile;
```

**Použití funkce `MySQL_Fetch_Array()`**; pro výpis všech hodnot tabulky:

```
$vysledek = MySQL_DB_Query("php", $sql, $spojeni);  
while($zaznam = MySQL_Fetch_Array($vysledek)):  
    echo $zaznam[rc]. "<br>";
```

```
echo $zaznam[jmeno]. "<br>";  
echo $zaznam[adresa]. "<br>";  
echo $zaznam[plat]. "<br><br>";  
endwhile;
```

**Použití funkce MySQL\_Fetch\_Object();** pro výpis všech hodnot tabulky:

```
$vysledek = MySQL_DB_Query("php", $sql, $spojeni);  
while($zaznam = MySQL_Fetch_Object($vysledek)):  
    echo $zaznam->rc. "<br>";  
    echo $zaznam->jmeno. "<br>";  
    echo $zaznam->adresa. "<br>";  
    echo $zaznam->plat. "<br><br>";  
endwhile; [6]
```

## 1.5 SMARTY

Smarty je šablonovací systém pro PHP, který umožňuje oddělení aplikační (samotný kód PHP) a zobrazovací (klasické HTML) logiky. To se hodí především pro větší projekty, kdy programátor PHP a kodér HTML není ve většině případů stejná osoba. Díky Smarty tak kodér úpravami HTML nezničí práci programátora. Kodér tedy vždy pracuje jen s HTML a programátor jen s PHP.

Smarty funguje tak, že kodér upravuje klasickou stránku HTML (šablonu) a v ní jsou určité značky, které jsou dynamicky vytvářené pomocí PHP. Taková šablona může např. vypadat takto (index.tpl):

```
<html>  
  
<body>  
  
    Vítejte {$jmeno}  
  
</body>  
  
</html>
```

Právě značka {\$jmeno} bude zajímat programátora. Ten v jiném skriptu zjistí (třeba z databáze) jméno návštěvníka, které pak pomocí metod třídy Smarty zamění za {\$jmeno} v šabloně. Příklad:

```
<?php
/*Skript vybere z databáze jméno
návštěvníka a přiřadí ho proměnné $name */

$smarty->assign('jmeno', '$name');

$smarty->display('index.tpl');
?>
```

Metoda `$smarty->assign` zamění proměnnou `$name` za značku `{ $jmeno }` v šabloně. Druhá metoda pak načte a zobrazí šablonu (tento skript by nefungoval, museli bychom ještě zahrnout třídu Smarty, ale pro zjednodušení je to dostačující).

Velkou předností je kompilace šablon. Pokud zavoláme předchozí skript v prohlížeči, Smarty vytvoří klasický skript PHP i se zobrazovací logikou do speciálního adresáře. Při příštím volání funkce již není nutné znova značky převádět, zavolá se zkompileovaná verze. Proč to Smarty dělá? Je to především rychlejší.

Ještě mnohem rychlejší než kompilace je cachování obsahu. Cachování je v podstatě kompilace, kdy se nevytváří PHP skripty, ale klasické HTML.<sup>[7]</sup>

## 1.6 JQUERY

Jedná se o malou javascriptovou knihovnu, která klade důraz na integraci mezi HTML a javascript. Jedná se o open source software, díky kterému na stránkách můžete vytvářet neuvěřitelné efekty a velmi efektivně pracovat.

### 1.6.1 Filozofie

Stejně jako CSS odděluje „zobrazovací“ charakteristiky od struktury HTML, jQuery odděluje „chování“ od struktury HTML. Například místo přímé specifikace on-click události přímo v HTML kódu tlačítka by stránka řízená jQuery napřed našla vhodný element tlačítka a potom změnila jeho manipulátor události. Takovéto oddělení chování od struktury se také často nazývá jako princip nevtíravého JavaScriptu.

## 1.6.2 Funkce

jQuery nabízí následující funkce:

- Výběr DOM elementů pomocí otevřeného cross-browser
- Výběr DOM elementů pomocí otevřeného cross-browser selektorového enginu Sizzle, odnože projektu jQuery
- Funkce pro procházení a změnu DOM (včetně podpory pro 1–3 a základní XPath
- Události
- Manipulace s CSS
- Efekty a animace
- AJAX
- Rozšiřitelnost
- Utility – např. informace o prohlížeči nebo funkce each
- Javascriptové pluginy

## 1.6.3 Funkce \$

Jeden z kritických konceptů v jakémkoli jQuery kódu je tzv. funkce „\$“. „\$“ je vlastně alias pro jmenný prostor „jQuery“.

Příklad 1: jQuery nabízí funkci pro oříznutí řetězců. Tato funkce může být použita jako:

```
ret = "   foo   ";  
jQuery.trim(ret); // vrátí "foo"
```

Nebo také jako:

```
ret = "   foo   ";  
$.trim(ret);
```



Tyto kódy jsou shodné. Použití „\$“ místo „jQuery“ je konvence, která je za rychlejší cestu pro přístup ke knihovně jQuery.

Příklad 2: Pro vybrání všech odstavců s třídou „foo“ a pro přiřazení druhé třídy „bar“:

```
$( "p.foo" ).addClass( "bar" );
```

Příklad 3: Pro spuštění funkce „mojefunkce“ ihned po načtení:

```
$(document).ready(function() {  
    mojefunkce();  
});
```

#### 1.6.4 Použití

jQuery většinou existuje jako jeden javascriptový soubor, obsahující všechny funkce pro DOM, Ajax, události a efekty. Do webové stránky může být vložen následovně:

```
<script type="application/javascript" src="/cesta/k/jquery.js"></script>
```

Nejnovější stabilní verze může být také načtena pomocí Google AJAX Libraries API. Tento způsob získávání knihovny má mnoho výhod včetně unifikovaného cachování a snížení odezvy. To může být provedeno následovně:

```
<skript type="application/javascript" src="http://www.google.com/jsapi">  
</script>
```

```
<script type="application/javascript">  
google.load("jquery", "1.3.2");  
</script>
```

Další populární způsob načítání jQuery je načtení přímo ze serverů Gogole:

```
<skript type="application/javascript"  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js">  
</script>
```

V neposlední řadě lze jQuery načíst v aktuální verzi přímo z domovské stránky projektu:

```
<script type="application/javascript" src="http://code.jquery.com/jquery-latest.min.js"></script> [8]
```

## 1.7 VEXFLOW API

Jedná se o open source javascriptové rozhraní pro vykreslování notové osnovy pomocí nejnovějšího jazyka HTML verze 5. Díky verzi 5 je zde dostupný tag `<canvas>`, který umí vykreslovat libovolné křivky a znaky a proto zde není potřeba dalších modulů a funkcí.

Rozhraní pracuje jako klientský skript a využívá moderního API JQUERY. V sobě má integrované funkce pro zobrazení téměř všech variant zápisu notové osnovy. Problémem však je vytvořit takové rozhraní, které by jednoduše umožnilo využít všech funkcionalit.

Bez základních znalostí hudební teorie není jednoduché s API pracovat. Postup jak pracovat s API bude popsán v praktické části.

## 1.8 CUFON

Cufon je javascriptový nástroj na nahrazování fontů na webu. Používá k tomu dva soubory, jeden prováděcí a druhý s vytvořenými fonty. Na stránkách Cufon (<http://cufon.shoqolate.com/generate/>) si můžete vytvořit téměř jakýkoliv font do javascriptové formy. Na příkladu uvedeném níže je vidět zdrojový kód potřebný provádějící transformaci fontů tagu `<h1>`.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <script src="cufon-yui.js" type="text/javascript"></script>
    <script src="Frutiger_LT_Std_400.font.js" type="text/javascript"></script>
    <script type="text/javascript">
```

```
Cufon.replace('h1');  
  
</script>  
  
</head>  
  
<body>  
  
<h1>This text will be shown in Frutiger LT Std.</h1>  
  
</body>  
  
</html>
```

Všechno podstatné je v tagu `<head>`: Jako první je načten prováděcí javascript `cufon-yui.js`, dále pak je načten javascript se samotným fontem.

Po načtení hlavičkových souborů je inicializován skript `Cufon.replace('h1')`, který přímo nahrazuje tagy příslušnými fonty.

V Internet Exploreru je vidět malé zpoždění než se text nahradí, proto je dobré dát těsně před `</body>` následující kód:

```
<script type="text/javascript"> Cufon.now(); </script> [9]
```

## **II. PRAKTICKÁ ČÁST**

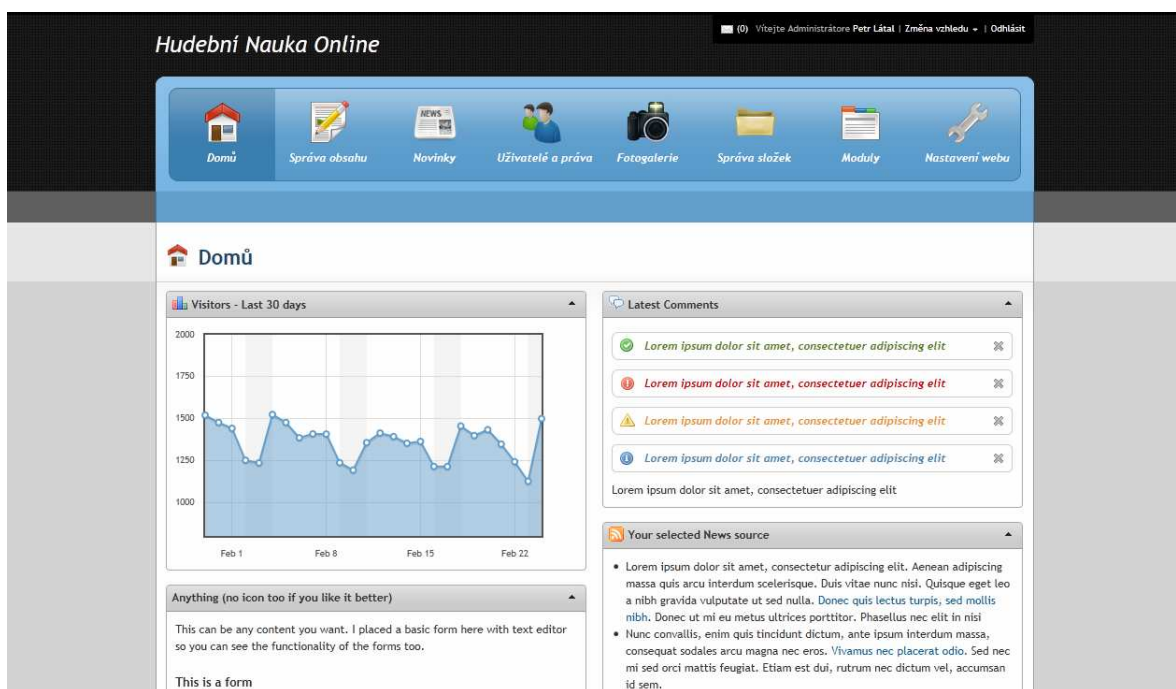
## 2 SYSTÉM PRO SPRÁVU

### 2.1 Funkce CMS

Funkce CMS bychom mohli rozdělit do těchto kategorií:

- Vytvoření obsahu
- Správa obsahu
- Moduly CMS
- Publikování
- Prezentace

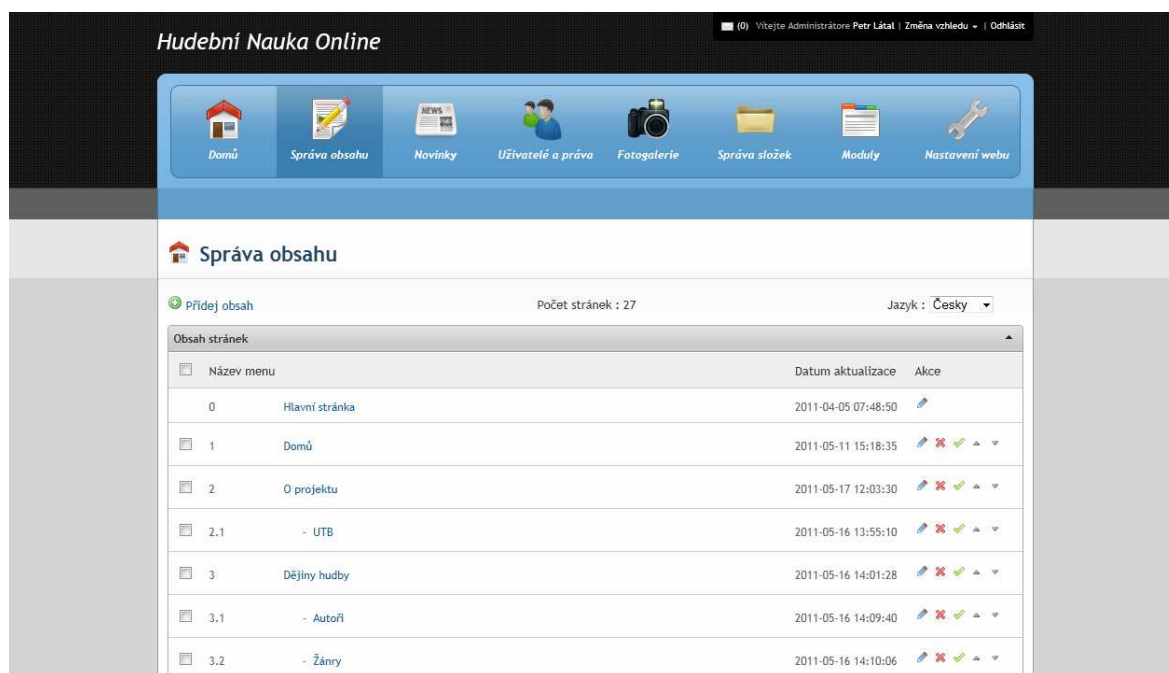
Každá oblast bude rozebrána níže.



Obrázek 1 : Vzhled administrace

### 2.1.1 Vytváření obsahu

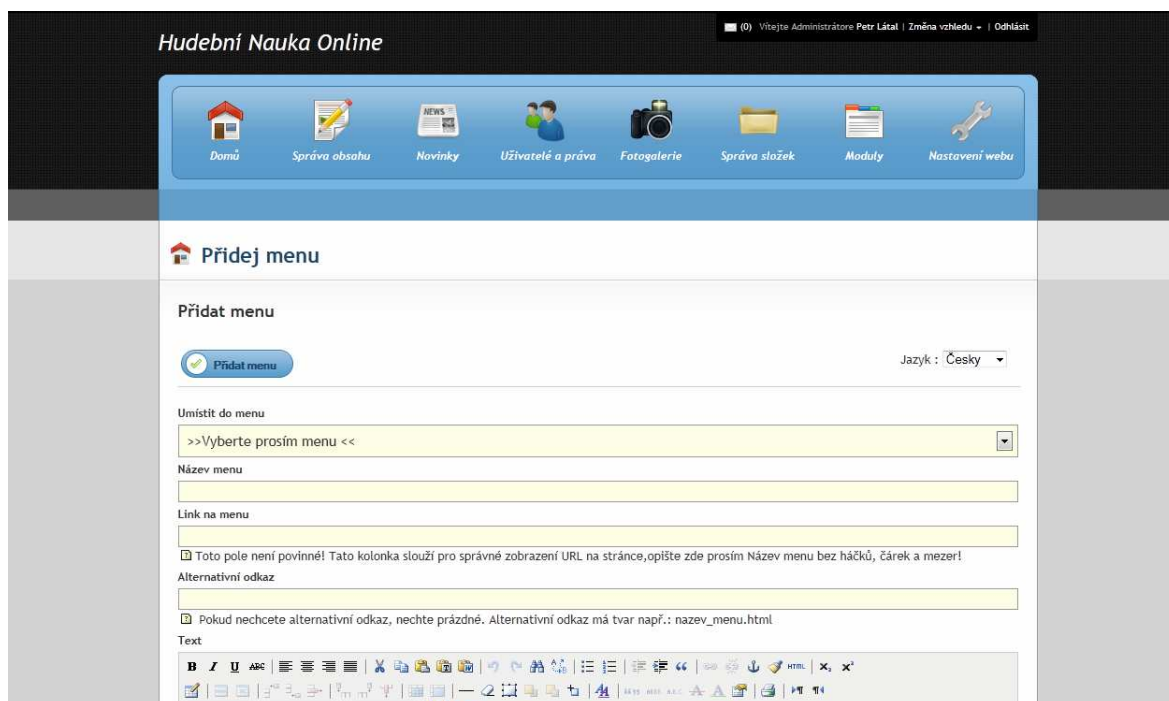
Tvorba obsahu v CMS systému je jednoduchou a efektivní záležitostí. Tento systém má kromě svých základních výhod (jakými jsou například vytváření a editace stránek, jejich úpravy v podobě mazání, povolování či zakazování obsahu a další funkce), také implementovanou vícejazyčnou podporu. Na *Obrázku 2* můžete vidět stromovou strukturu stránek hudebního portálu. Členění je rozvrženo tak, aby se uživatel portálu jednoduše a rychle orientoval v obsahu.



*Obrázek 2 : Ukázka administrace správy obsahu*

Na *Obrázku 3* je zobrazeno rozhraní, ve kterém vkládáme obsah do CMS systému. Základní vlastnosti systému pro vkládání obsahu můžeme rozdělit do 5ti polí uvedených níže.

- název
- link
- alternativní odkaz
- nadřazenost
- jazyk
- obsah



Obrázek 3. : Vkládání nového obsahu

Název je jednoduchým identifikátorem pro orientaci v menu.

Link pak slouží jako ořezaný řetězec názvu bez háčeků, čárek a mezer, jako část adresy pro lepší integraci obsahu do vyhledávačů (jako například Google, Seznam a další).

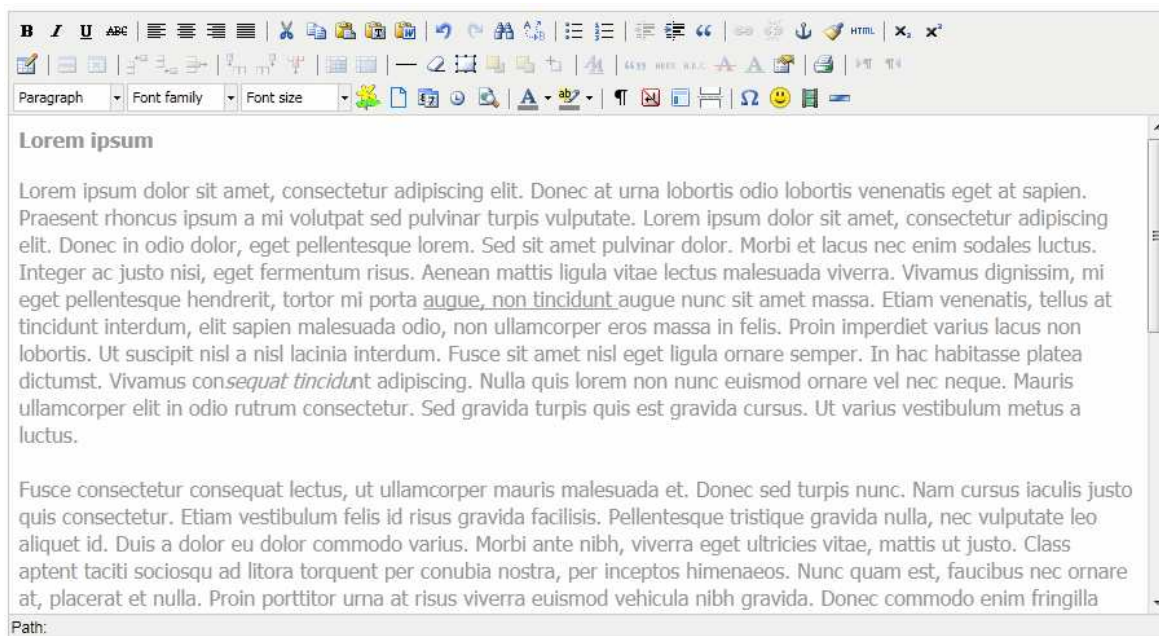
Do alternativního odkazu pak můžeme jednoduše vkládat přesměrování do částí webu, která nejsou pro-linkovány správcem obsahu. Velmi jednoduše tak můžeme přistoupit k modulům, které nejsou integrovány v CMS systému nebo jen jednoduše přesměrovat obsah na externí stránky (jako například na [www.seznam.cz](http://www.seznam.cz)).

Nadřazenost každé stránky je velmi důležitým prvkem obsahu. V případě že chcete, aby byl obsah umístěn v nejvyšší úrovni menu, je potřeba v `select` boxu vybrat položku „Kořenové menu“. V opačném případě pouze vyberete položku, pod kterou má obsah spadat.

Společně s vyplněním obsahu je nezbytné vybrat jazyk, pod který má stránka spadat.

Nevyplněním jakéhokoliv povinného pole budete před odesláním formuláře vyzváni k nápravě java-scriptem. Pokud by jste však měli java-script vypnutý, budete upozorněni validací ze strany serveru obdobnou hláškou.

Mezi největší výhody z pohledu vytváření obsahu, patří jednoduché prostředí, které disponuje velmi podobnými funkcemi jako MS Word. CMS systém má integrovaný WYSIWYG plugin TinyMCE s předdefinovaným vzhledem Office 2007 a spoustou dalších funkcí, které usnadňují práci.



Obrázek 4. : Prostředí pro vytváření obsahu

### 2.1.1.1 Vkládání obrázků

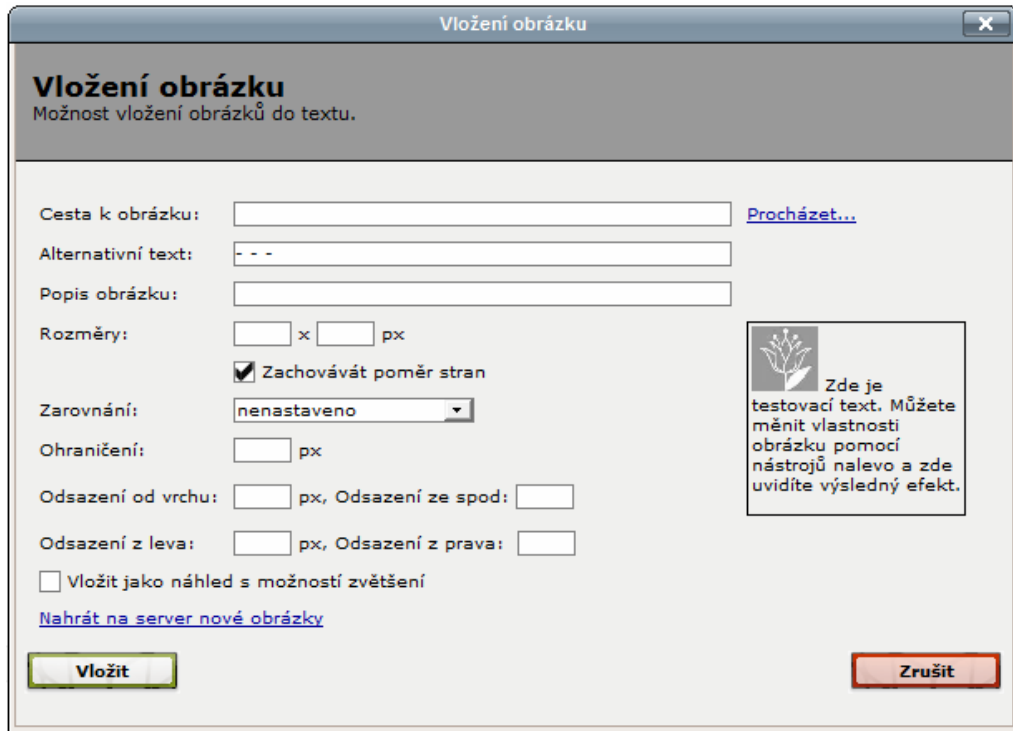
Práce s obrázkem je jednoduchou záležitostí. K vyvolání funkce slouží ikonka na *Obrázku 5* „Vložit/Upravit Obrázek“.



Obrázek 5. : Vložit/Upravit Obrázek

Po kliknutí na ikonku se zobrazí dialogové okno se základními parametry. Pro jednoduchost a rychlost vkládání obrázku do obsahu je třeba vyplnit pouze cestu k obrázku. Pokud nevíte cestu, pomůže odkaz „Procházet“.





Obrázek 6 : Vložit/Upravit Obrázek

Kliknutím na odkaz „Procházet“ se zobrazí další dialogové okno, kde si jednoduše vyberete požadovaný obrázek a kliknutím na něj okno zavřete a vrátíte se do prvního dialogového okna.



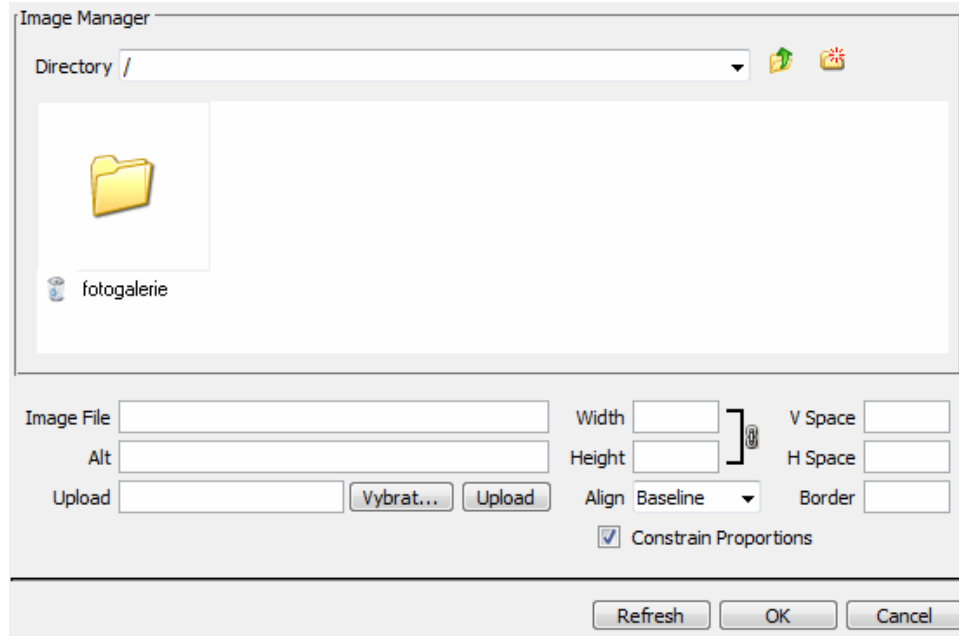
Obrázek 7: Výběr obrázku

Nyní bude cesta k obrázku vyplněna s absolutní cestou k umístění na serveru. Pokud je třeba, vyplňte alternativní text, který slouží jako informace pro prohlížeč, ale i pro Vás. Pokud by se fotka ze serveru smazala, či přesunula na jiné místo, nemusel by si nikdo všimnout, že na daném místě v html kódu fotka byla. Dále je dobré si upravit rozměry fotky a nastavit zarovnání v textu. Výsledný obrázek je možné různě odsadit vůči textu. Nakonec stačí kliknout na tlačítko vložit a obrázek se vloží na místo kurzoru v obsahu.

Obrázek 8: Vložit/Upravit Obrázek

### 2.1.1.2 Nahrávání obrázků na server

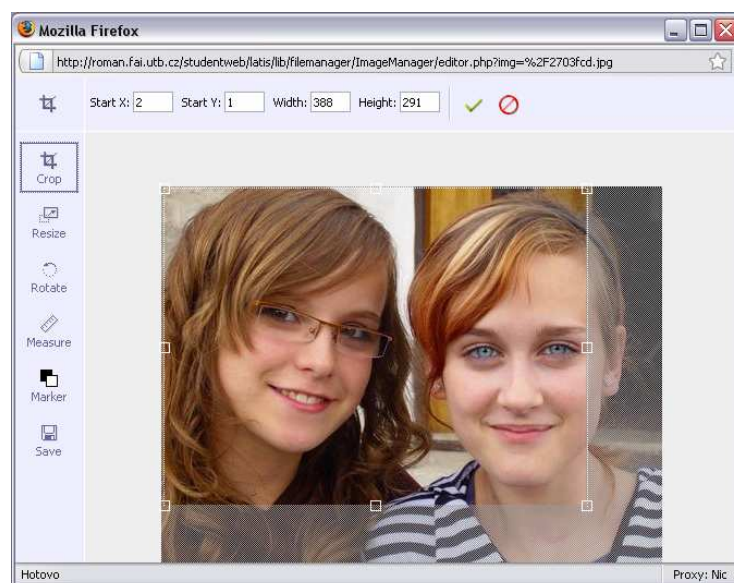
Nahrávání obrázku je zajištěno pomocí ImageManager pluginu upraveného pro tento CMS systém. Jednoduše si zde nalistujeme adresář, do kterého chceme nahrát obrázky, a pomocí položky upload si vybereme obrázek z lokálního disku Vašeho počítače a tlačítkem upload jej nahrajeme do příslušného adresáře. Všechny složky vycházejí z hlavní složky /images/uploads/ na serveru. Správa adresářů je velmi intuitivní a umožňuje jejich jednoduchou úpravu včetně mazání (viz *Obrázek 9*).



Obrázek 9: Správce obrázků

### 2.1.1.3 Editace obrázků na serveru

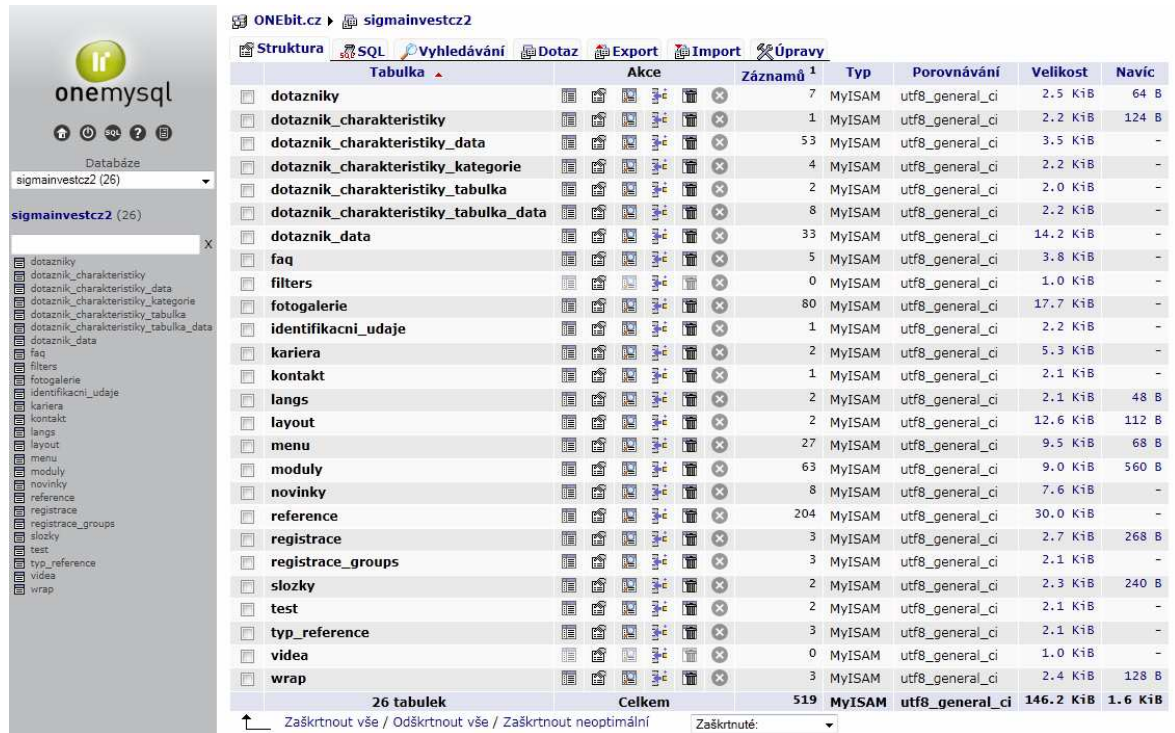
ImageManager má v sobě integrovaný editor obrázků, který umožňuje základní úpravy jako například ořezání, otočení či změnu velikosti obrázku na straně serveru a bez nutnosti žádného instalovaného editoru. Po úpravě fotky lze obrázek na serveru uložit. Obrázek bude uložen do dvou souborů, vám se však zobrazí pouze upravená verze. Na serveru však zůstane pre-editovaný originál, který je v případě přejmenování opět použitelný jako originál.



Obrázek 10: Editor obrázků

## 2.1.2 Správa obsahu v databázi

Data vytvořená pomocí správce obsahu jsou ukládána do MySQL databáze. V tabulce databáze jsou buňky, které obsahují zařazení obsahu do stromové struktury, samotný obsah a vlastnosti týkající se přístupu (například zda je obsah aktivní, kdy byl vytvořen, kým byl vytvořen atd.).



The screenshot shows the 'onemysql' interface with the 'sigmainvestcz2' database selected. The table structure is displayed as follows:

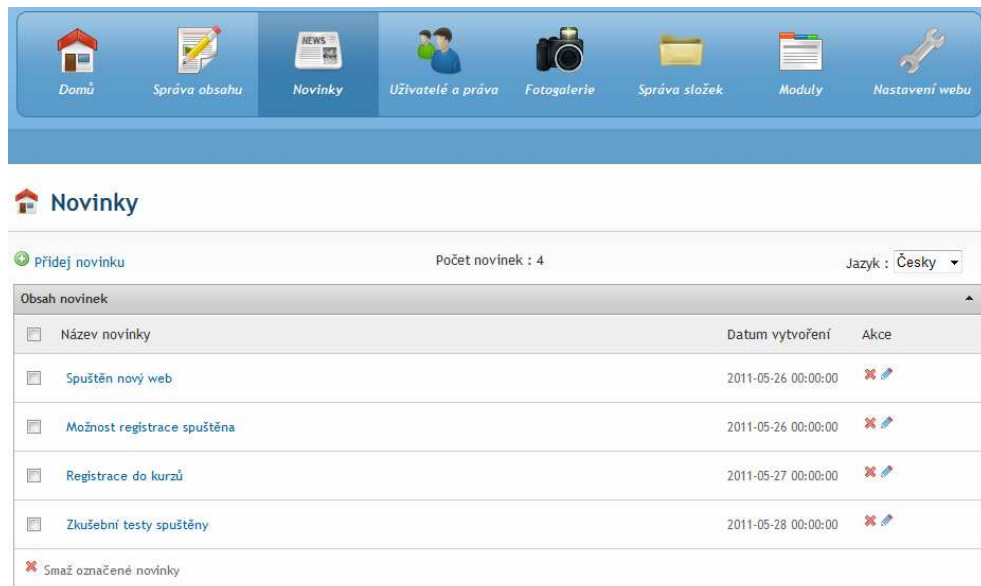
Tabulka	Akce	Záznamů	1	Typ	Porovnávání	Velikost	Navíc
dotazniky		7		MyISAM	utf8_general_ci	2.5 KiB	64 B
dotaznik_charakteristiky		1		MyISAM	utf8_general_ci	2.2 KiB	124 B
dotaznik_charakteristiky_data		53		MyISAM	utf8_general_ci	3.5 KiB	-
dotaznik_charakteristiky_kategorie		4		MyISAM	utf8_general_ci	2.2 KiB	-
dotaznik_charakteristiky_tabulka		2		MyISAM	utf8_general_ci	2.0 KiB	-
dotaznik_charakteristiky_tabulka_data		8		MyISAM	utf8_general_ci	2.2 KiB	-
dotaznik_data		33		MyISAM	utf8_general_ci	14.2 KiB	-
faq		5		MyISAM	utf8_general_ci	3.8 KiB	-
filters		0		MyISAM	utf8_general_ci	1.0 KiB	-
fotogalerie		80		MyISAM	utf8_general_ci	17.7 KiB	-
identifikacni_udaje		1		MyISAM	utf8_general_ci	2.2 KiB	-
kariera		2		MyISAM	utf8_general_ci	5.3 KiB	-
kontakt		1		MyISAM	utf8_general_ci	2.1 KiB	-
langs		2		MyISAM	utf8_general_ci	2.1 KiB	48 B
layout		2		MyISAM	utf8_general_ci	12.6 KiB	112 B
menu		27		MyISAM	utf8_general_ci	9.5 KiB	68 B
moduly		63		MyISAM	utf8_general_ci	9.0 KiB	560 B
novinky		8		MyISAM	utf8_general_ci	7.6 KiB	-
reference		204		MyISAM	utf8_general_ci	30.0 KiB	-
registrace		3		MyISAM	utf8_general_ci	2.7 KiB	268 B
registrace_groups		3		MyISAM	utf8_general_ci	2.1 KiB	-
slozky		2		MyISAM	utf8_general_ci	2.3 KiB	240 B
test		2		MyISAM	utf8_general_ci	2.1 KiB	-
typ_reference		3		MyISAM	utf8_general_ci	2.1 KiB	-
videa		0		MyISAM	utf8_general_ci	1.0 KiB	-
wrap		3		MyISAM	utf8_general_ci	2.4 KiB	128 B
<b>26 tabulek</b>	<b>Celkem</b>	<b>519</b>		<b>MyISAM</b>	<b>utf8_general_ci</b>	<b>146.2 KiB</b>	<b>1.6 KiB</b>

At the bottom of the interface, there are options to 'Zaškrtnout vše / Odškrtnout vše / Zaškrtnout neoptimální' and a 'Zaškrtnuté:' dropdown menu.

Obrázek 11: Data uložená v databázi

### 2.1.3 Správa novinek

Modul funguje jako jednoduché prostředí pro vkládání novinek na hlavní stránku webu. Na *Obrázku 12* můžete vidět jak vypadá struktura vložených novinek v administraci. Fungují zde akce jako například přidání novinky, editace novinky, smazání novinky a hromadné smazání novinky.



Obrázek 12: Správa novinek

Na obrázku *Obrázku 13* můžete vidět, jakým způsobem se vkládá obsah novinek do systému. Podobně jako u obsahu je zde důležité zvolit jazyk, název, datum a text novinky. Při nevyplnění všech políček budete vyzváni validačním skriptem ke znovuvyplnění.

Obrázek 13: Vložení novinky

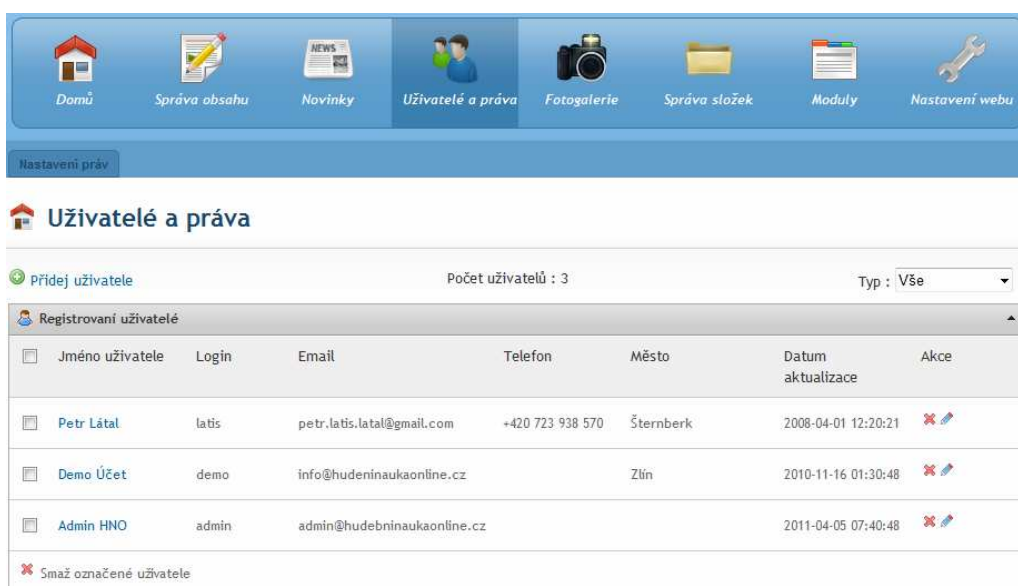
Datum novinky je usnadněno jednoduchým Jquery kalendářem, který Vám pohodlně nabídne při kliknutí do textové oblasti aktuální datum s možností listování měsíci a roky dopředu nebo dozadu. Výstupní formát data je shodný s SQL syntaxí funkce now(), proto se dále s datem pracuje velmi pohodlně. Na *Obrázku 14* je zobrazen náhled Jquery kalendáře.



Obrázek 14: Vložení novinky

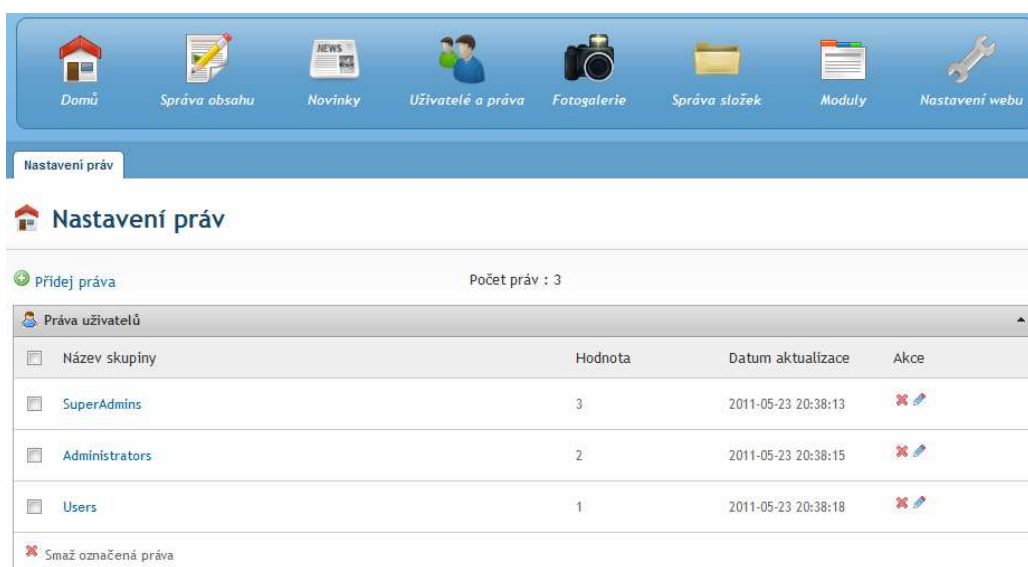
## 2.1.4 Uživatelé a práva

Uživatele systému jsou nedílnou součástí při rozdělování rolí správy systému. V modulu Uživatelé a práva naleznete jak obyčejné uživatele, kteří se registrovali ze stránek webu, tak administrátory, kteří byli vloženi do systému v administraci. Na *Obrázku 15* je zobrazen seznam všech uživatelů webu.



Obrázek 15: Seznam všech uživatelů na webu

Abychom mohli jednoduše každého uživatele kategorizovat, je potřeba si nejprve vytvořit skupiny. Na *Obrázku 16* je záložka s aktuálním rozdělením skupin na webu. Funguje to tak, že každá skupina má přidělenou hodnotu. Čím vyšší číslo má skupina, tím větší má pravomoci. Například skupina SuperAdmins má kromě všech možností skupiny Administrátors, přístup k „Nastavení modulů“. Tato funkce bude popsána později.



*Obrázek 16: Nastavení práv*

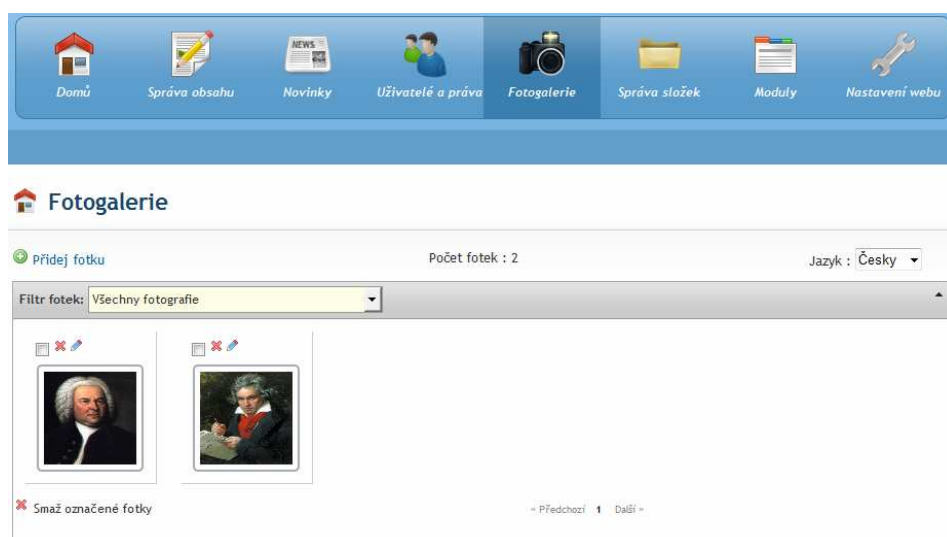
Každého uživatele můžete jednoduše editovat a přidělovat mu různá práva. Na *Obrázku 17* je vidět editace uživatele, která nabízí změnu hesla v horní části a ve spodní části je možnost změny typu uživatele, přístupu a všech osobních informací.

*Obrázek 17: Editace uživatele*



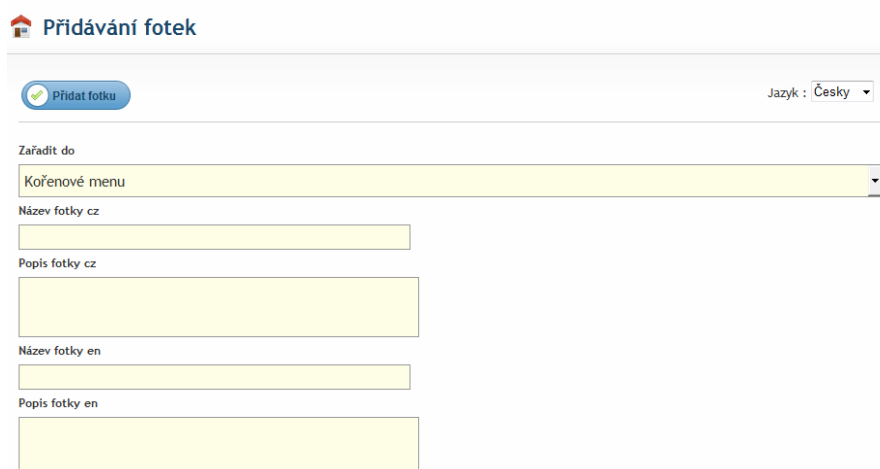
### 2.1.5 Fotogalerie

Modul fotogalerie byl vytvořen za účelem jednoduchého vkládání a členění obrázků do systému. Tento modul je propojený s modulem „správy složek“, protože každou fotografii je potřeba někam zařadit. Pokud si nevytvoříte stromovou strukturu složek, můžete vkládat všechny fotografie do jedné složky, to však může mít za následek zneřehlednění výstupu na webu. Každá fotografie je zpětně editovatelná, mazatelná a je možno ji editovat přímo na serveru. Stránkování modulu je implicitně nastaveno na 30 obrázků na stránku, tato hodnota však lze v systému změnit.



Obrázek 18: Všechny Fotogalerie

Přidávání fotografie je velmi jednoduchou záležitostí, stačí kliknout na odkaz „Přidej fotku“ a vyplnit kategorii, název a popis fotky a vybrat fotku z lokálního počítače.

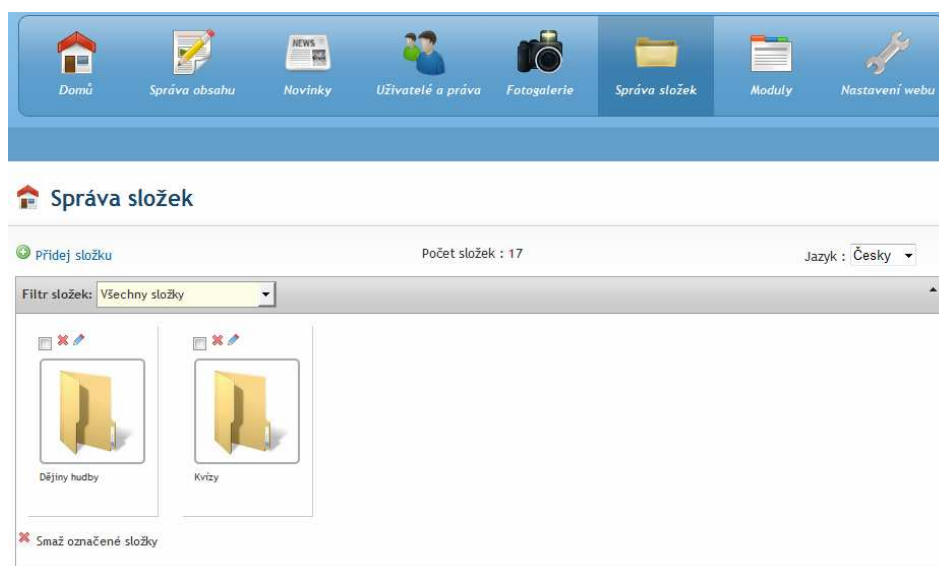


Obrázek 19: Nová fotografie



## 2.1.6 Správa složek

Jedná se o velmi strategický modul celého systému. Každý web je nějakým způsobem kategorizovaný. Podle toho jakou strukturu složek si vytvoříte, takovým způsobem pak budete do systému vkládat fotografie, videa, či propojovat různé moduly (například „kvízový modul“ či „modul kurzů“). Správa složek je tedy databázové rozdělení různých typů obsahu webu. Stránkování modulu je implicitně nastaveno na 30 složek na stránku, tato hodnota však lze v systému změnit.



Obrázek 20: Správa složek

Vkládání složek je realizováno pomocí odkazu „přidej složku“. Zde je opět důležité vybrat kategorii, do které bude nová složka zapadat a vybrat jazyk. Na *Obrázku 21* je zobrazeno vzorové vyplnění nové složky.

Obrázek 21: Tvorba nové složky

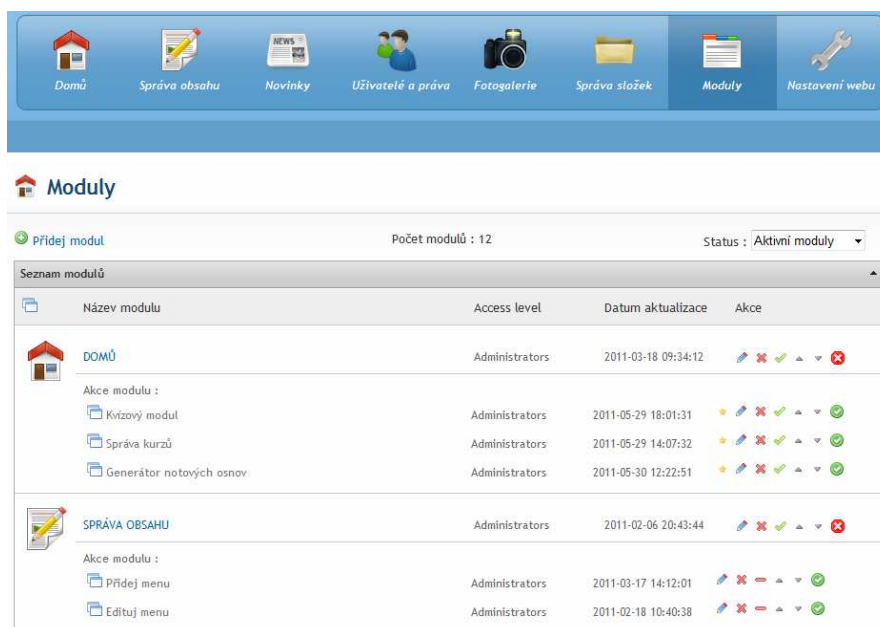
## 2.1.7 Správa modulů

Jedná se o nejdůležitější modul v systému. Bez tohoto modulu by uživatel musel v databázi velmi komplikovaně provazovat nový modul a při nesprávném vyplnění by mohl ohrozit stabilitu systému. Na hlavní stránce modulu jsou zobrazeny pouze aktivní moduly, v pravém horním rohu si však můžete zobrazit všechny moduly, nebo pouze aktivní či neaktivní moduly. U každého modulu jsou vypsány Akce modulu. Těmito akcemi se rozumí jakýkoliv dotaz z administrace na přidání, editaci, či zobrazení jakéhokoliv modulu.

Každá akce modulu je prověřena oprávněním v databázi a nemůže se tedy stát, že by neoprávněná osoba zobrazila či editovala modul, ke kterému nemá přístup.

### 2.1.7.1 Postup při vytváření nového modulu do systému

Na serveru si v adresáři /inc/admin/ vytvoříme složku například: „testmodul“. Do této složky vytvoříme soubory `upravy.php`, `hlasky.php`, dále pak soubor `testmoduladd.php` pro přidávání obsahu, `testmoduledit.php` pro editaci obsahu a `nastaveni_testmodulu.php` pro přehled. Takto vytvořenou strukturu je již potřeba pouze prolinkovat se systémem a vložit funkce, které budou modul řídit.



Obrázek 22: Nastavení modulů v systému

Přihlásíme se jako superadmin do systému a na záložce „Moduly“, klikneme na odkaz „přidat modul“. Vlastnosti každého modulu jsou poměrně složité a uživatel se nemusí ihned orientovat. Pro tyto účely je každá položka popsána a uživatel by tedy neměl mít problém potřebná pole vyplnit. Na *Obrázku 23* můžete vidět část modulu s předvyplněnými informacemi.

**Editace modulu**

Status : Aktivní modul

Umístít do modul stromu  
02 Domů

Access level  
Administrators

Název modulu  
Kvízový modul

Path  
kvíz  
 Fyzická cesta k modulu /inc/modul/. (Např.: modul)

Soubor  
nastaveni\_kvizu.php  
 Název zpracovávaného souboru PHP. (Např.: nastaveni\_modulu.php)

Akce  
nastaveni\_kvizu  
 Hodnota předávaná proměnné PAGE=nastaveni\_modulu. (Např.: nastaveni\_modulu)

Act - v případě odkazování se na editovaný modul

*Obrázek 23: Nastavení modulů v systému*

Ke každému modulu si můžete vytvořit „alias“. Tato možnost byla vytvořena proto, aby jste si v administraci mohli na libovolné záložce vytvářet karty s odkazy na moduly, které nemusí přímo souviset s daným modulem. Tyto moduly jsou ve správci modulů označeny hvězdičkou.

## 2.1.8 Nastavení webu

Jedná se především o jazykový modul, který slouží k jednoduché úpravě jazykového layoutu jak na webu, tak i v administraci. Díky propracovanosti modulu jste schopní přidat do databáze novou buňku s vlastnostmi, které pak můžete kdekoliv na webu aplikovat. Na *Obrázku 24* je zobrazeno prvních 5 položek z jazykového layoutu. Položky jsou kategorizovány jako datový typ, data a název položky.

Položka ID je pouze zobrazena a není možné ji editovat. Obsahuje identifikátor v databázi, při jehož změně by se upravené jazykové hodnoty neuložily do databáze.

Název položky	Datový typ SQL	DATA	Akce
přihlaseení_do_administrace	varchar(50)	Přihlášení do administrace	
sitename	varchar(100)	Hudební Nauka Online	
id	int(11)	1	
lang	varchar(2)	cs	
nav_left_0	varchar(200)	Nabízíme	

Obrázek 24: Nastavení webu

Přidávání nové položky je realizováno pomocí odkazu „přidat položku“ a nabízí stručné, ale výstižné vlastnosti (například datový typ, název bez háčeků a čárek, zařazení v databázi). Po vložení nové položky do databáze je možné položce nastavit hodnotu.

Obrázek 25: Vložení nové položky jazykového layoutu

Kromě jazyka však můžete nastavit i základní informace ke kontaktům či identifikační údaje firmy. Tyto položky jsou však volitelné.

### 2.1.8.1 Nastavení jazyků

Důležitou záložkou v nastavení webu je „Nastavení jazyků“, díky které můžeme na webu vytvořit libovolný počet jazyků. Na *Obrázku 26* jsou vidět aktuálně nastavené jazyky a jejich hodnoty. Přidání nového jazyka je opět realizováno pomocí jednoduchého formuláře.

Jazyk	Název jazyka	Hodnota	Datum aktualizace	Akce
<input type="checkbox"/>	English	en	2011-03-27 15:59:01	
<input type="checkbox"/>	Česky	cs	2011-03-27 15:59:05	

Obrázek 26: Nastavení jazyků

### 2.1.9 Správa kurzů

Tento modul je velmi obdobný modulu Správa složek, protože Vám umožňuje vytvořit si nezávislou stromovou strukturu kurzů. Každý kurz obsahuje název, popis kurzu, heslo pro vstup do kurzu a vedoucího kurzu. Dále pak je možné nastavit zda má být kurz aktivní či ne.

Název kurzu	Seznam registrovaných	Vedoucí kurzu	Datum aktualizace	Akce
1 1. ročník			2011-06-08 10:51:16	[edit] [delete] [status]
1.1 - Cimbál	Registrovaní žáci (1)	Petr Látal	2011-05-29 16:11:46	[edit] [delete] [status]
1.2 - Housle	Registrovaní žáci (2)	Petr Látal	2011-05-25 20:43:38	[edit] [delete] [status]
1.3 - Klarinet	Registrovaní žáci (1)	Petr Látal	2011-05-29 16:11:40	[edit] [delete] [status]
1.4 - Piáno	Registrovaní žáci (3)	Petr Látal	2011-05-29 16:10:48	[edit] [delete] [status]
1.5 - Viola		Petr Látal	2011-05-29 16:06:42	[edit] [delete] [status]
2 2. ročník			2011-05-29 16:09:31	[edit] [delete] [status]

Obrázek 27: Správa kurzů

V seznamu kurzů můžete vidět členění, seznam registrovaných žáků a vedoucího kurzu. Pokud chcete vidět jmenovitě všechny žáky daného kurzu, stačí kliknout na odkaz Registrovaní žáci. Pokud chcete žáka přidat do seznamu, stačí jej přidat pomocí odkazu „přidej uživatele“.

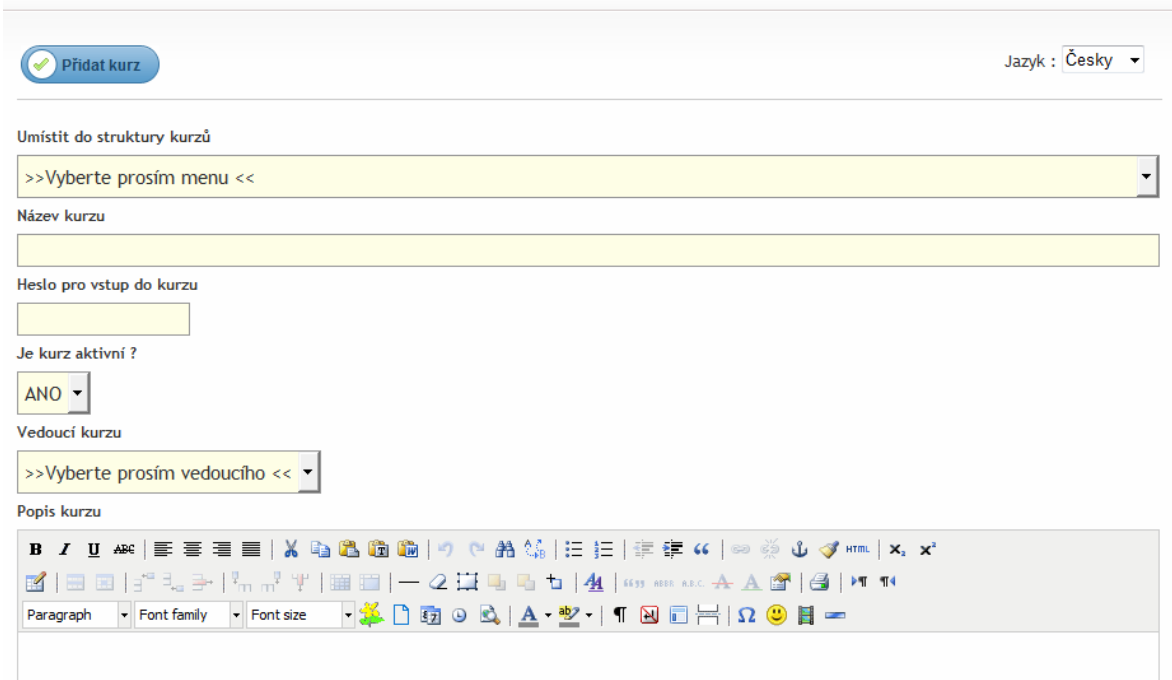
Jméno registrovaného	Název kurzu	Datum zápisu	Akce
Demo Účet	Viola	2011-06-09 23:35:56	[X]
Admin HNO	Viola	2011-06-09 23:35:59	[X]
Petr Látal	Viola	2011-06-09 23:36:02	[X]

Obrázek 28: Registrovaní žáci

Při tvorbě kurzu je potřeba správně vybrat kategorii, napsat název kurzu, vytvořit heslo do kurzu a přidružit vedoucího kurzu. Samozřejmě by bylo dobré vytvořit krátký popis kurzu, který bude zobrazen na webu a bude sloužit jako informace pro uchazeče.

Vedoucího kurzu je možné vybrat ze selectboxu registrovaných uživatelů, proto pokud uživatel neexistuje, je potřeba jej vytvořit v modulu „Uživatelé a práva“ a přidat mu patřičná oprávnění.

## Přidej kurz



Jazyk : Česky


Umístít do struktury kurzů  
>>Vyberte prosím menu <<

Název kurzu

Heslo pro vstup do kurzu

Je kurz aktivní ?  
ANO

Vedoucí kurzu  
>>Vyberte prosím vedoucího <<

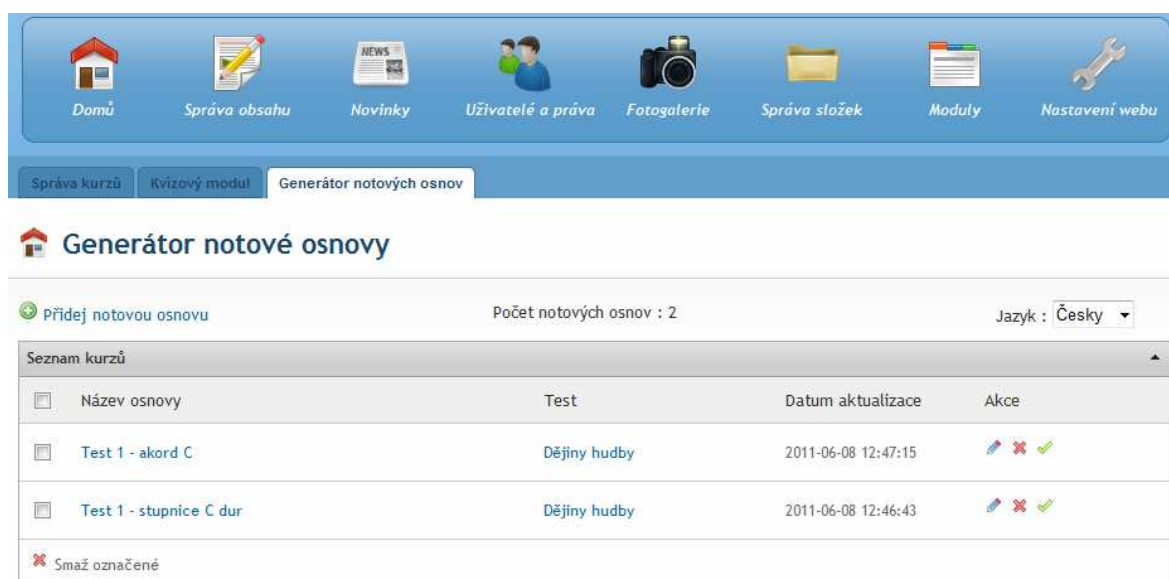
Popis kurzu  


Obrázek 29: Tvorba kurzu

### 2.1.10 Generátor notových osnov

Modul pro generování notových osnov slouží jako co nejjednodušší rozhraní pro tvorbu notové osnovy jak pro kvízový modul, tak pro prezentaci daného výrazu na webu. Mezi hlavní vlastnosti modulu patří název osnovy, test, ke kterému můžete osnovu přidružit (není to však povinné) a status, zda má být osnova aktivní či ne.

Z databázového hlediska však modul obsahuje ještě datovou část, která je modulem vygenerována a uložena do databáze jako hotová osnova připravená k prezentaci.



Obrázek 30: Generátor notových osnov


#### 2.1.10.1 Tvorba notové osnovy

Po kliknutí na odkaz „přidej notovou osnovu“ se objeví první krok dle *Obrázku 31*, který obsahuje obecné informace k osnově.

V levé části jsou obecné informace k osnově. Jako první je potřeba vyplnit jednoznačný identifikátor osnovy. Toto pole je povinné a je potřeba zde vepsat například název osnovy bez háčeků a čárek (přípustný tvar : test-1-akord-c). Na stránce se tvoří tag <canvas>, který zobrazuje javascript jemu přidružený a pokud by na stránce byly dvě a více notových osnov se stejným identifikátorem, osnovy by se nezobrazily! Dále je potřeba vyplnit název osnovy (ten je důležitý jen pro Vás), zařadit osnovu do kategorie (není povinné) a zvolit zda má být osnova aktivní.



V pravé části je pak základní nastavení osnovy. Zde není potřeba nic přepisovat, ale je dobré si osnovu upravovat. Jako první položka je šířka osnovy v pixelech, která udává šířku tagu <canvas>. Dále je zde počet řádků osnovy, zde je dobré si předem říci, kolik řádků osnovy budete chtít. Pozice x a y od osy udává pouze zobrazení prvního znaku vůči notové osnově (Tuto vlastnost není třeba nastavovat). Posledním parametrem je délka notové osnovy v pixelech, která udává samotnou šířku osnovy (osnovu, na které jsou zobrazeny noty). V případě špatného navolení vstupních parametrů se sice osnova vygeneruje a zobrazí, ale je možné, že noty budou buď chybět, protože se zobrazí mnohem dál, než je šířka či délka notové osnovy, nebo se špatně zobrazí.

 **Přidej osnovu**

Jazyk : Česky ▼

**Obecné informace k osnově**

Jednoznačný identifikátor osnovy

Toto pole není povinné! Vepište prosím název osnovy bez háčeků a čárek  
**Například test-1-nota-c**

Název osnovy

Zařazení do kategorie  
>> Vyberte si prosím << ▼

Je osnova aktivní ?

**Nastavení osnovy**

Šířka osnovy v pixelech(např.:850)

Počet řádků osnovy(např.:10)

Pozice x od osnovy v pixelech(např.:10)

Pozice y od osnovy v pixelech(např.:0)

DĚLKA NOTOVÉ OSNOVY v pixelech (např.:500)

Obrázek 31: Tvorba notové osnovy – první krok

Druhým krokem je stisknutí tlačítka „Generuj osnovu“, který nám zobrazí druhou část modulu. V této části je potřeba si zvolit typ klíče, vepsat noty a vygenerovat náhled.

**Obecné informace k osnově**

Typ klíče

NOTY oddělené středníkem (např.: c/4;c/5)

Obrázek 32: Tvorba notové osnovy – druhý krok

Pro jednoduchost je pod polem pro vepisování not vytvořeno pár základních tlačítek, která vloží noty, akordy, pomlky, křížky a další znaky do osnovy. Způsob jakým se noty zapisují do pole bude popsán později.

Po stisknutí tlačítka „vygenerovat náhled“ se Vám zobrazí notová osnova s vepsanými notami. Pokud se Vám zobrazí pouze osnova bez not, znamená to, že jste někde udělali chybu. Pokud se nezobrazí ani notová osnova, pravěpodobně jste nezvolili typ klíče. Notová osnova by pak mohla vypadat podobně jako na *Obrázku 33*.



*Obrázek 33: Vzorová notová osnova – stupnice C dur*

Pro dokončení tvorby osnovy je potřeba udělat poslední krok a sice stisknout tlačítko „uložit notovou osnovu do databáze“ jako na *Obrázku 34*.



*Obrázek 34: Uložení notové osnovy do databáze*

Pokud vygenerování a uložení proběhne v pořádku, budete přesměrováni na hlavní stránku modulu generátoru notových osnov.

### 2.1.11 Kvízový modul

Jedná se o modul, který generuje libovolný typ kvízu nebo testu. Mezi základní vlastnosti tohoto kvízu patří název kvízu, výsledky, kategorie a základní operace jakými je editace, přidávání a mazání kvízů.

Obrázek 35: Kvízový modul – přehled

U každého kvízu máte možnost se podívat na výsledky každého uživatele, který kvíz vyplnil. Je zde vidět úspěšnost, datum pokusu a struktura odpovědí uživatele.

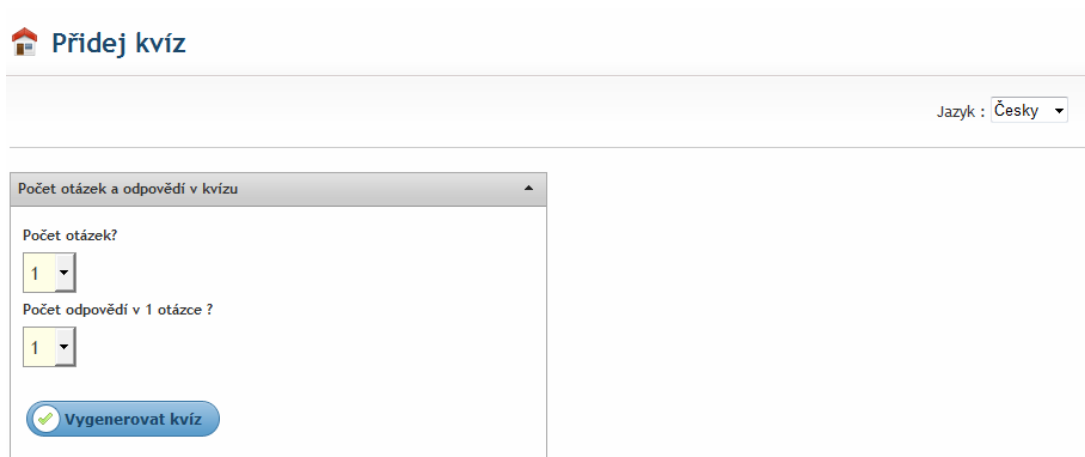
### Výsledky testů

Jméno uživatele	Odpovědi	Úspěšnost	Datum vytvoření	Akce
Petr Látal	Otazka:1 Odpověď uživatele : 1, 2, 3, 4, 5, Otazka:2 Odpověď uživatele : 2, Otazka:3 Odpověď uživatele : 45,	0 %	2011-05-29 17:46:56	✘
Petr Látal	Otazka:1 Odpověď uživatele : 1, 2, 3, 5, Otazka:2 Odpověď uživatele : 1, Otazka:3 Odpověď uživatele : 28,	100 %	2011-05-29 17:47:31	✘
Petr Látal	Otazka:1 Odpověď uživatele : 1, 2, 3, 5, Otazka:2 Odpověď uživatele : 1, Otazka:3 Odpověď uživatele : 28,	100 %	2011-06-04 20:09:15	✘
Petr Látal	Otazka:3 Odpověď uživatele : ,	0 %	2011-06-08 10:53:05	✘

Obrázek 36: Výsledky testů

### 2.1.11.1 Tvorba nového kvízu

Vytvoření kvízu je rozděleno do dvou kroků. V prvním kroku je potřeba navolit počet otázek a počet odpovědí dle *Obrázku 37*.

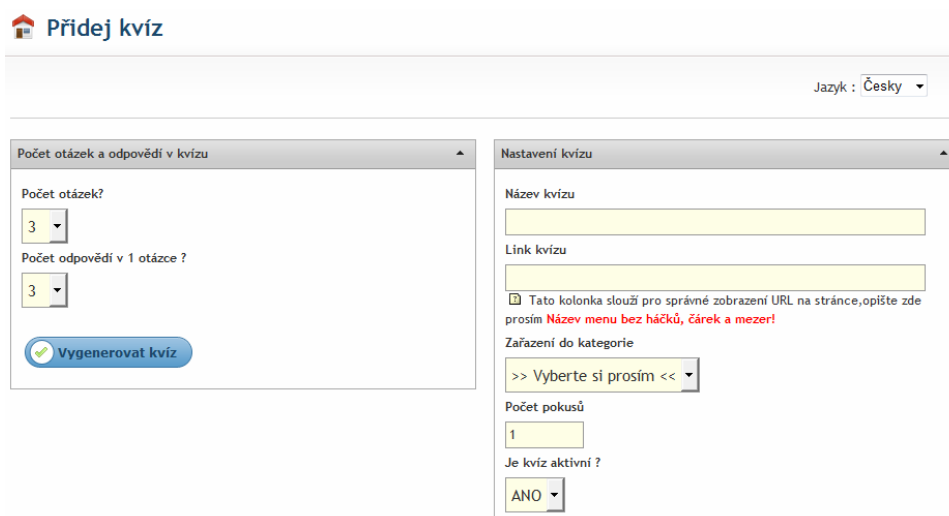


The screenshot shows the 'Přidej kvíz' (Add Quiz) interface. At the top right, there is a language selector set to 'Česky'. The main content area is titled 'Počet otázek a odpovědí v kvízu'. It contains two dropdown menus: 'Počet otázek?' with the value '1' and 'Počet odpovědí v 1 otázce?' with the value '1'. Below these is a blue button with a checkmark icon and the text 'Vygenerovat kvíz'.

Obrázek 37: Tvorba kvízu – první krok

Zde je dobré si počty naddimenzovat, protože v případě, že byste si rozmysleli počet otázek či počet odpovědí, při znovugenerování formuláře by byla data ztracena. V případě takovéto situace raději kvíz uložte a následně editujte, protože při editaci nedojde ke ztrátě aktuálních dat.

Ve druhém kroku stiskněte tlačítko „Vygenerovat kvíz“. Modul se nyní rozšíří o pravou část viz *Obrázek 38*. Zde pouze vyplňte název kvízu, zařaďte jej do kategorie, nastavte počet pokusů a jako poslední zvolte možnost, má-li být kvíz aktivní.



The screenshot shows the 'Přidej kvíz' (Add Quiz) interface, Step 2: 'Nastavení kvízu'. The language selector is still 'Česky'. The left panel is titled 'Počet otázek a odpovědí v kvízu' and shows 'Počet otázek?' set to 3 and 'Počet odpovědí v 1 otázce?' set to 3. The right panel is titled 'Nastavení kvízu' and contains: 'Název kvízu' (empty text field), 'Link kvízu' (empty text field), a checkbox with the text 'Tato kolonka slouží pro správné zobrazení URL na stránce, opište zde prosím **Název menu bez háčků, čárek a mezer!**', 'Zařazení do kategorie' (dropdown menu with '>> Vyberte si prosím <<'), 'Počet pokusů' (text input field with '1'), and 'Je kvíz aktivní?' (dropdown menu with 'ANO'). A blue 'Vygenerovat kvíz' button is at the bottom.

Obrázek 38: Tvorba kvízu – druhý krok

Při vygenerování kvízu se Vám dále zobrazí rozhraní pro vkládání otázek a odpovědí viz *Obrázek 39*. Zde jednoduše vyplníte Vaše otázky a odpovědi. Jelikož si otázky i odpovědi naddimenzujete, je potřeba ještě upravit pár položek. První položkou je typ odpovědi. Zde se vám nabídne typ text, radio a checkbox. Při změně na jakýkoliv typ se zobrazí typ u každé odpovědi.

1. Otázka

Notová osnova

>> Vyberte si prosím <<

Typ odpovědi : Počet odpovědí v otázce : Výčet správných odpovědí :

Text 1

1. V případě odpovědi typu TEXT, nevyplňujte do odpovědi žádné otázky, nebudou zobrazeny, protože u tohoto typu je pouze jeden textbox.

2. Počet odpovědí v otázce - počet vámi vyplněných odpovědí ke každé otázce.

3. Výčet správných odpovědí - například pro správnou odpověď 1,3 a 4 v dané otázce vyplňte : 1,3,4 u odpovědi typu TEXT vepište správnou odpověď.

1. Odpověď

2. Odpověď

3. Odpověď

*Obrázek 39: Tvorba kvízu – otázky a odpovědi*

V případě odpovědi typu TEXT nevyplňujte do odpovědi žádné otázky, ty nebudou zobrazeny, protože u tohoto typu je pouze jedna odpověď typu TEXT.

Dále je zde počet odpovědí. Jedná se o počet odpovědí ke každé otázce.

Výčet správných odpovědí je následující:

- pro správnou odpověď 1,3 a 4 k dané otázce vyplňte: 1,3,4
- u odpovědi typu TEXT vepište správnou odpověď.

Výběrem notové osnovy se pod select-boxem zobrazí Vámi zvolená notová osnova. Pokud u více otázek zvolíte jednu notovou osnovu, nezobrazí se. Jelikož zde budou dva stejné identifikátory tagu <canvas>. Dbejte tedy na to, aby u každé otázky byla jiná notová osnova (respektive osnova s jiným identifikátorem).

Otázky a odpovědi kvízu

1. Otázka

Notová osnova

->Test 1 - akord C

Typ odpovědi : Počet odpovědí v otázce : Výčet správných odpovědí :

Text 1

V případě odpovědi typu TEXT, nevyplňujte do odpovědi žádné otázky, nebudou zobrazeny, protože u tohoto typu je pouze jeden textbox.  
 Počet odpovědí v otázce - počet vámi vyplněných odpovědí ke každé otázce.  
 Výčet správných odpovědí - například pro správnou odpověď 1,3 a 4 v dané otázce vyplníte : 1,3,4 u odpovědi typu TEXT vepište správnou odpověď.

1. Odpověď

2. Odpověď

3. Odpověď

Obrázek 40: Tvorba kvízu – otázka s notovou osnovou

Ukázky typu odpovědi radio a checkbox viz *Obrázek 41* a *Obrázek 42*.

2. Odpověď

3. Odpověď

Obrázek 41: Tvorba kvízu – ukázky typu odpovědi radio

2. Odpověď

3. Odpověď

Obrázek 42: Tvorba kvízu – ukázky typu odpovědi checkbox

### 2.1.12 Publikování a prezentace

Po vytvoření všech obsahů, kurzů a kvízů se vše projeví na stránkách portálu. Postup jak pracovat s portálem bude popsán později.



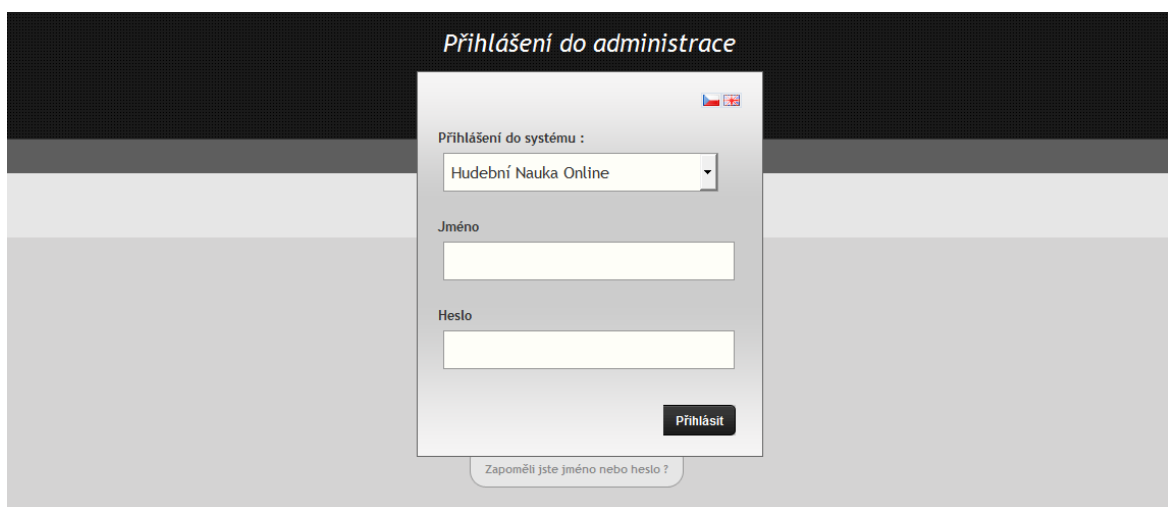
Obrázek 43: Publikace stránek portálu

Velkou výhodou portálu s CMS systémem je zobrazování obsahu stránek s jednotným vzhledem. CMS odděluje obsah od vzhledu a aplikuje jej nezávisle na obsahu. Proto se autoři nemusí starat o vzhled ale pouze o obsah.

Důležitým faktorem portálu je navigace stránek ( neboli „sitemap“). Ta je zajištěna externím generátorem XML Sitemaps (<http://www.xml-sitemaps.com/>). Tento nástroj indexuje veškeré stránky na portále do souboru sitemap.xml a díky tomuto souboru je portál lépe indexovatelný google roboty.

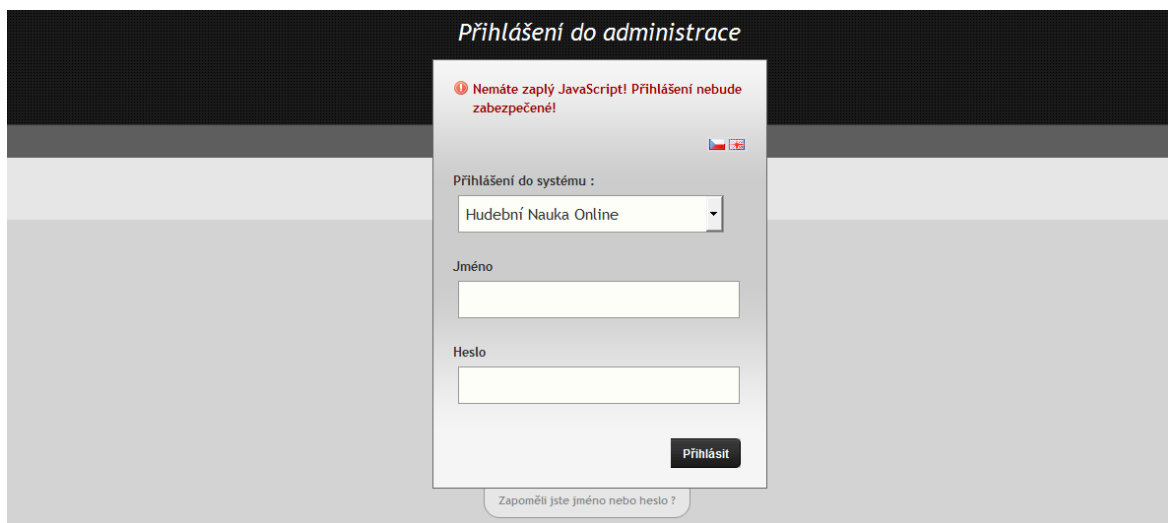
## 2.2 Zabezpečení systému

Přihlášení do administrace je zabezpečeno md5 java-scriptovou funkcí, která nám před odesláním přihlašovacího formuláře zašifruje heslo do md5 podoby a teprve pak heslo odešle na server. Touto cestou je zajištěna mnohem vyšší bezpečnost, jelikož se data přenášejí v kryptované podobě a v případě odposlechu snižují možnost rozluštění zdánlivě jednoduchého formuláře.



Obrázek 44: Přihlášení do administrace

Toto zabezpečení však nefunguje máte-li vypnutý javascript. Systém Vás před přihlášením na tento nedostatek upozorní a pomůže Vám k nápravě viz *Obrázek 45* .



Obrázek 45: Přihlášení do administrace bez zapnutého javascriptu



## 2.3 Práce s VEXFLOW

VexFlow je javascriptová knihovna, která umí generovat libovolný typ notové osnovy. Jedná se o API, které bylo navrženo pro HTML 5 Canvas.

Díky HTML 5 Canvas již není potřeba dalších knihoven či dalších závislostí.

### 2.3.1 Krok 1: Základy pro tvorbu notové osnovy

Absolutním základem je tag canvas o určité šířce a délce.

```
<canvas width=700 height=100"></canvas>
```

K tomu, abychom vytvořili prázdnou notovou osnovu, je potřeba následujícího javascriptového kódu.

```
var canvas = $("div.one div.a canvas")[0];
var renderer = new Vex.Flow.Renderer(canvas,
    Vex.Flow.Renderer.Backends.CANVAS);

var ctx = renderer.getContext();
var stave = new Vex.Flow.Stave(10, 0, 500);
stave.addClef("treble").setContext(ctx).draw();
```

Na prvním řádku je volání objektu canvas pomocí jquery funkce a uložení jejího modelu do proměnné canvas.

Dále je potřeba vygenerovat vlastní pozadí pro notovou osnovu pomocí funkce `Vex.Flow.Renderer()` s parametry objektu proměnné canvas.

Třetí důležitou proměnnou je proměnná `ctx`, která volá funkci `getContext()`, která nám vytvoří prostředí pro notovou osnovu.

Čtvrtou proměnnou je `stave`, která pomocí funkce `Vex.Flow.Stave()` vytvoří délku notové osnovy s pozicemi první noty zleva a zvrchu o Vámi zadaný počet pixelů. Posun je důležitý pro vyšší noty, které se mohou zobrazit těsně za klíčem.

Vše se vykreslí a pomocí funkce `addClef("treble")` bude výsledkem prázdná notová osnova s houslovým klíčem.

### 2.3.2 Krok 2: Vkládání not

Přidávání not do notové osnovy je poněkud složitější proces. Je potřeba pochopit vykreslovací model not.

`StaveNote` je soubor not, které patří do osnovy. Může to být jednoduchá nota, nebo akord. Každá nožička noty může být směrem nahoru nebo dolů v závislosti na tónině. Všechny instance `StaveNote` mají společnou délku trvání.

Tyto StaveNote jsou uskupeny do Voice. Voice mají časovou signaturu a ta musí být společně se skupinkou not na osnově použita pro všechny zvuky na osnově.

Voice jsou spojeny do VoiceGroup. To je zvláště užitečné, když máte více notovou osnovu. Při vykreslování každé noty je vykreslena i správná orientace nožičky vzhledem k notové osnově. A VoiceGroup musí obsahovat alespoň jednu voice.

Nakonec je zde Formatter, tato funkce zajišťuje správné rozložení not na notové osnově.

V kódu uvedeném níže je vykreslena osnova s dvěma notami a akordem

```
var canvas = $("div.two div.a canvas")[0];
var renderer = new Vex.Flow.Renderer(canvas,
    Vex.Flow.Renderer.Backends.CANVAS);

var ctx = renderer.getContext();
var stave = new Vex.Flow.Stave(10, 0, 500);
stave.addClef("treble").setContext(ctx).draw();

// Create the notes
var notes = [
    // nota C.
    new Vex.Flow.StaveNote({ keys: ["c/4"], duration: "q" }),

    // nota D.
    new Vex.Flow.StaveNote({ keys: ["d/4"], duration: "q" }),

    // poznámka (b/4)
    // position of the rest.
    new Vex.Flow.StaveNote({ keys: ["b/4"], duration: "qr" }),

    // akord C
    new Vex.Flow.StaveNote({ keys: ["c/4", "e/4", "g/4"], duration:
"q" })
];

// osnova o 4 notách
var voice = new Vex.Flow.Voice({
    num_beats: 4,
    beat_value: 4,
    resolution: Vex.Flow.RESOLUTION
});

// přidání not do osnovy
voice.addTickables(notes);

// úprava rozložení not
var formatter = new Vex.Flow.Formatter().
    joinVoices([voice]).format([voice], 500);

// vykreslení osnovy
voice.draw(ctx, stave);
```

### 2.3.3 Krok 3: Vše o modifikátorech

Modifikátory v podstatě obohacují noty o různé znaky. Zahrnují `Accidental`, `Vibrato`, `Dot`, `Annotation` a další.

Modifikátory pro jednoduché skupinky not jsou uvnitř `ModifierContext`. To jim umožňuje se inteligentně porovnávat sami se sebou. Například akord obsahuje dvě noty blízko sebe a každá má určitou pozici. Je tedy potřeba zajistit takovou pozici, aby poslední nota měla stejnou pozici jako první nota, ale v jiné oktávě.

Tento kód nám ukáže, jak vypadá takováto situace:

```
var canvas = $("div.three div.a canvas")[0];
var renderer = new Vex.Flow.Renderer(canvas,
    Vex.Flow.Renderer.Backends.CANVAS);

var ctx = renderer.getContext();
var stave = new Vex.Flow.Stave(10, 0, 500);

// Add a treble clef
stave.addClef("treble");
stave.setContext(ctx).draw();

var notes = [
    // Dotted eighth E##
    new Vex.Flow.StaveNote({ keys: ["e##/5"], duration: "8d" }).
        addAccidental(0, new Vex.Flow.Accidental("##").addDotToAll(),
    // Sixteenth Eb
    new Vex.Flow.StaveNote({ keys: ["eb/5"], duration: "16" }).
        addAccidental(0, new Vex.Flow.Accidental("b")),
    // Half D
    new Vex.Flow.StaveNote({ keys: ["d/5"], duration: "h" }),
    // Quarter Cm#5
    new Vex.Flow.StaveNote({ keys: ["c/5", "eb/5", "g#/5"], duration: "q" }).
        addAccidental(1, new Vex.Flow.Accidental("b")).
        addAccidental(2, new Vex.Flow.Accidental("#"))
];

// Helper function to justify and draw a 4/4 voice
Vex.Flow.Formatter.FormatAndDraw(ctx, stave, notes);
```

Další modifikátory sloužící k tomu, abychom viděli jak se mění jejich pozice mezi sebou.

```
var notes = [
    new Vex.Flow.StaveNote(
        { keys: ["g/4", "b/4", "cb/5", "e/5", "g#/5", "b/5"],
          duration: "h" }).
        addAccidental(0, new Vex.Flow.Accidental("bb")).
        addAccidental(1, new Vex.Flow.Accidental("b")).
        addAccidental(2, new Vex.Flow.Accidental("#")).
        addAccidental(3, new Vex.Flow.Accidental("n")).
        addAccidental(4, new Vex.Flow.Accidental("b")).
        addAccidental(5, new Vex.Flow.Accidental("##")),
```

```

    new Vex.Flow.StaveNote({ keys: ["c/4"], duration: "h" })
  ];

  // Pomocná function pro zarovnání a vykreslení 4/4 osnovy
  Vex.Flow.Formatter.FormatAndDraw(ctx, stave, notes);

```

### 2.3.4 Tabulky znaků

Existují speciální tabulky pro zobrazení notových znaků. Tyto tabulky jsou k dispozici ve speciálních souborech.

Tabulky, které jsou k dispozici:

- `Vex.Flow.keyProperties.note_values` - seznam všech not.
- `Vex.Flow.accidentalCodes.accidentals` – seznam náhodných not.
- `Vex.Flow.keySignatures.keySpecs` – seznam validních klíčů.
- `Vex.Flow.durationToTicks` – seznam správných trvání not.

### 2.3.5 Krok 4: Zvuky a notace

VexFlow umí hrát noty, ale jen za předpokladu, že si načte třídu `beam.js`. Tvorba takového kódu bude zobrazena níže.

```

var canvas = $("div.four div.a canvas")[0];
var renderer = new Vex.Flow.Renderer(canvas,
  Vex.Flow.Renderer.Backends.CANVAS);

var ctx = renderer.getContext();
var stave = new Vex.Flow.Stave(10, 0, 500);

// Vytboření houslového klíče a vykreslení
stave.addClef("treble");
stave.setContext(ctx).draw();

var notes = [
  new Vex.Flow.StaveNote({ keys: ["e##/5"], duration: "8d" }).
    addAccidental(0, new Vex.Flow.Accidental("##")).addDotToAll(),
  new Vex.Flow.StaveNote({ keys: ["b/4"], duration: "16" }).
    addAccidental(0, new Vex.Flow.Accidental("b"))
];

var notes2 = [
  new Vex.Flow.StaveNote({ keys: ["c/4"], duration: "8" }),
  new Vex.Flow.StaveNote({ keys: ["d/4"], duration: "16" }),
  new Vex.Flow.StaveNote({ keys: ["e/4"], duration: "16" }).
    addAccidental(0, new Vex.Flow.Accidental("b"))
];

var notes3 = [
  new Vex.Flow.StaveNote({ keys: ["d/4"], duration: "16" }),
  new Vex.Flow.StaveNote({ keys: ["e/4"], duration: "16" }).
    addAccidental(0, new Vex.Flow.Accidental("#")),
  new Vex.Flow.StaveNote({ keys: ["g/4"], duration: "32" }),

```

```

    new Vex.Flow.StaveNote({ keys: ["a/4"], duration: "32" }),
    new Vex.Flow.StaveNote({ keys: ["g/4"], duration: "16" })
  ];

  var notes4 = [ new Vex.Flow.StaveNote({ keys: ["d/4"], duration:
"q" }) ];

  // vytvoření zvuků
  var beam = new Vex.Flow.Beam(notes);
  var beam2 = new Vex.Flow.Beam(notes2);
  var beam3 = new Vex.Flow.Beam(notes3);

  var all_notes = notes.concat(notes2).concat(notes3).concat(notes4
);

  // Helper function to justify and draw a 4/4 voice
  Vex.Flow.Formatter.FormatAndDraw(ctx, stave, all_notes);

  // vykreslení a přehrání zvuku
  beam.setContext(ctx).draw();
  beam2.setContext(ctx).draw();
  beam3.setContext(ctx).draw();

```

Zde jsme vytvořili tři skupiny not se zvukem. Funkce pro přehrávání zvuku not závisí na délce trvání dozvuku spodní noty.

Notování zahrnuje stejný sled operací. K vykreslení not je potřeba vytvořit `StaveTie` instanci a spojit její dvě `StaveNote` dohromady. Od každé `StaveNote` může být akord, ten je však potřeba propojit s indexy not, které chcete propojit.

```

var canvas = $("div.four div.b canvas")[0];
var renderer = new Vex.Flow.Renderer(canvas,
  Vex.Flow.Renderer.Backends.CANVAS);

var ctx = renderer.getContext();
var stave = new Vex.Flow.Stave(10, 0, 500);

// vytvoření osnovy s houslovým klíčem
stave.addClef("treble");
stave.setContext(ctx).draw();

var notes = [
  new Vex.Flow.StaveNote({ keys: ["e##/5"], duration: "8d" }).
    addAcciden-
tal(0, new Vex.Flow.Accidental("##")).addDotToAll(),
  new Vex.Flow.StaveNote({ keys: ["b/4"], duration: "16" }).
    addAccidental(0, new Vex.Flow.Accidental("b"))
];

var notes2 = [
  new Vex.Flow.StaveNote({ keys: ["b/4"], duration: "8" }),
  new Vex.Flow.StaveNote({ keys: ["d/4"], duration: "16" }),
  new Vex.Flow.StaveNote({ keys: ["c/4", "e/4"], duration: "16" }
).
  addAccidental(0, new Vex.Flow.Accidental("b"))
];

var notes3 = [
  new Vex.Flow.StaveNote({ keys: ["c/4", "e/4"], duration: "8" })

```

```

    new Vex.Flow.StaveNote({ keys: ["e/4"], duration: "8" }).
      addAccidental(0, new Vex.Flow.Accidental("#"))
  ];

  var notes4 = [
    new Vex.Flow.StaveNote({ keys: ["d/4"], duration: "8" }),
    new Vex.Flow.StaveNote({ keys: ["e/4"], duration: "8" }),
  ];

  // vytvoření zvuků
  var beam = new Vex.Flow.Beam(notes);
  var beam2 = new Vex.Flow.Beam(notes2);
  var beam3 = new Vex.Flow.Beam(notes3);
  var beam4 = new Vex.Flow.Beam(notes4);

  // Vytvoření spoje mezi poslední notou a první skupinou a první
  notou poslední skupiny

  var tie = new Vex.Flow.StaveTie({
    first_note: notes[1],
    last_note: notes2[0],
    first_indices: [0],
    last_indices: [0]
  });

  // Create another tie between the two chords in the tune
  var tie2 = new Vex.Flow.StaveTie({
    first_note: notes2[2],
    last_note: notes3[0],
    first_indices: [0, 1],
    last_indices: [0, 1]
  });

  var all_notes = notes.concat(notes2).concat(notes3).concat(notes4
);

  // Helper function to justify and draw a 4/4 voice
  Vex.Flow.Formatter.FormatAndDraw(ctx, stave, all_notes);

  // Vytvoření zvuků
  beam.setContext(ctx).draw();
  beam2.setContext(ctx).draw();
  beam3.setContext(ctx).draw();
  beam4.setContext(ctx).draw();

  // Vykreslení spojů
  tie.setContext(ctx).draw();
  tie2.setContext(ctx).draw();

```

### 2.3.6 Krok 5: Kytarová osnova

VexFlow umí také vykreslovat kytarovou osnovu. Mechanizmy pro zobrazení osnovy jsou velmi podobné jako klasická notová osnova. Je však potřeba použít jiných tříd pro generování osnov a not.

```

// Vytvoření kontextu
var canvas = $("div.five div.a canvas")[0];
var renderer = new Vex.Flow.Renderer(canvas,
  Vex.Flow.Renderer.Backends.CANVAS);

```

```
var ctx = renderer.getContext();
ctx.setFont("Arial", 10, "").setBackgroundFillStyle("#eed");

// Vytvoření a vykreslení osnovy
var tabstave = new Vex.Flow.TabStave(10, 0, 500);
tabstave.addTabGlyph();
tabstave.setContext(ctx).draw();

// Tvorba not
var notes = [
  // A single note
  new Vex.Flow.TabNote({
    positions: [{str: 3, fret: 7}],
    duration: "q"}),

  // akord s 3rd string bent
  new Vex.Flow.TabNote({
    positions: [{str: 2, fret: 10},
               {str: 3, fret: 9}],
    duration: "q"}).
  addModifier(new Vex.Flow.Bend("Full"), 1),

  // nota s harsh vibrato
  new Vex.Flow.TabNote({
    positions: [{str: 2, fret: 5}],
    duration: "h"}).
  addModi-
fier(new Vex.Flow.Vibrato().setHarsh(true).setVibratoWidth(70), 0)
];

Vex.Flow.Formatter.FormatAndDraw(ctx, tabstave, notes);
```

Stačilo zaměnit `Stave` za `TabStave` a `StaveNote` za `TabNote`. Přidali jsme také nějaké `bend` a `vibrato` modifikátory.

Jsou zde dvě věci, které je třeba manuálně specifikovat – typ písma, a barvu pozadí. Funkce `formel` se používá k zobrazení `fret` čísel, anotací a jiného textu. Vlastní `clearRect` se používá k vymazání tagu `canvas`.<sup>[18]</sup>

## 2.4 Návod jak pracovat s portálem pro běžného uživatele

V první řadě je dobré vědět, jak se registrovat na portále. V levé části portálu uprostřed najdete přihlašovací formulář a u něj najdete „Nemáte svůj účet? Vytvořte si jej“. Po kliknutí na odkaz „Vytvořte si jej“ se vám zobrazí formulář viz *Obrázek 46*. Zde vyplňte vaše iniciály včetně uživatelského jména, hesla a emailu a formulář odešlete.



The image shows a registration form with the following fields:

- REGISTRACE NOVÉHO UŽIVATELE**
- Přihlašovací údaje**
  - Přihlašovací jméno:
  - Heslo:
  - Potvrzení hesla:
- Osobní údaje**
  - Jméno:
  - Příjmení:
  - Ulice:
  - Město:
  - PSČ:
  - Email:
  - Telefon:
  - ICQ:

*Obrázek 46: Registrace nového uživatele*

Po úspěšném registrování se můžete přihlásit na portál. Po přihlášení by se Vám mělo zobrazit základní menu dle *Obrázku 47*. Zde si pro začátek můžete upravovat vaše nastavení účtu.





**Uživatel :**  
Petr Látal

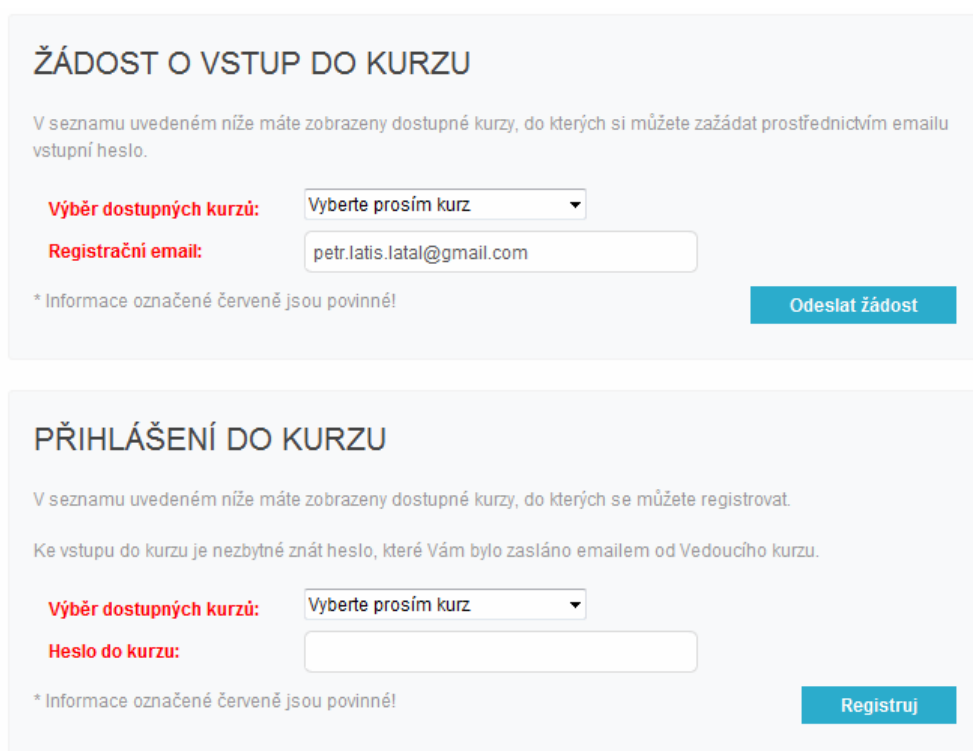
**Poslední přihlášení :**  
2011-06-10 02:43:13

**Výukový kurz :**  
[Přihlášení do kurzu](#)  
[Moje kurzy \( 5 \)](#)  
[Testy \( 2 / 2 \)](#)

**Osobní nastavení :**  
[Nastavení účtu](#)  
[Odhlaš](#)

Obrázek 47: Přihlášení na portál

Jelikož jste se přihlásili poprvé, bude potřeba Vás registrovat do kurzu. Přihlášení provedete kliknutím na odkaz „Přihlášení do kurzu“. Zobrazí se dva formuláře viz *Obrázek 48*.



**ŽÁDOST O VSTUP DO KURZU**

V seznamu uvedeném níže máte zobrazeny dostupné kurzy, do kterých si můžete zažádat prostřednictvím emailu vstupní heslo.

**Výběr dostupných kurzů:**

**Registrační email:**

\* Informace označené červeně jsou povinné!

**PŘIHLÁŠENÍ DO KURZU**

V seznamu uvedeném níže máte zobrazeny dostupné kurzy, do kterých se můžete registrovat.

Ke vstupu do kurzu je nezbytné znát heslo, které Vám bylo zasláno emailem od Vedoucího kurzu.

**Výběr dostupných kurzů:**

**Heslo do kurzu:**

\* Informace označené červeně jsou povinné!

Obrázek 48: Žádost a přihlášení do kurzu

První formulář slouží jako žádost o vstup do kurzu. Zde je potřeba vybrat si kurz, kterého se chcete zúčastnit a dále je potřeba vyplnit registrační email. Email bude vyplněný, pokud jste řádně přihlášení. V případě zadání špatného emailu, nebo emailu, který nebyl zadán při

registraci, budete upozorněni hláškami v horní části prohlížeče. Jestliže vše správně vyplníte, bude Vám na email doručeno heslo do kurzu. Toto heslo Vám samozřejmě může sdělit i vedoucí kurzu. Tento formulář však slouží jako jednoduchý automat pro zaslání hesel do kurzů.

V druhém kroku (kdy už znáte heslo) je potřeba vybrat opět kurz a vepsat heslo do kurzu. Opět Vám bude odeslán informační email o úspěšném registrování se do kurzu a Vy se nyní dostanete k testům a studijním materiálům určeným pro daný kurz.

V přihlašovací menu se Vám zobrazí odkaz „Moje kurzy“, kde po kliknutí na odkaz uvidíte seznam Vašich kurzů s možností odhlášení se z kurzu viz *Obrázek 49*.



MOJE KURZY

Zde je uveden seznam kurzů, do kterých jste přihlášení.

Název kurzu	Datum zápisu do kurzu	Akce
<a href="#">Klarinet</a>	2011-05-25 22:09:04	<input type="button" value="Odhlásit se z kurzu »"/>
<a href="#">Piáno</a>	2011-05-26 01:14:08	<input type="button" value="Odhlásit se z kurzu »"/>
<a href="#">Housle</a>	2011-05-26 01:17:20	<input type="button" value="Odhlásit se z kurzu »"/>
<a href="#">Cimbál</a>	2011-06-04 20:08:48	<input type="button" value="Odhlásit se z kurzu »"/>
<a href="#">Viola</a>	2011-06-09 23:36:02	<input type="button" value="Odhlásit se z kurzu »"/>

*Obrázek 49: Moje kurzy*

V neposlední řadě je zde odkaz „Testy“, ve kterém pro Vás budou připravovány testy vedoucím kurzu viz *Obrázek 50*. Tyto testy by měly plnit funkci klasického testu v hudební nauce každé základní umělecké školy. Každý test je ohodnocen a uložen včetně data vykonání testu. V závislosti na počtu pokusů je i test opravitelný. Počítá se vždy poslední pokus!

**SEZNAM TESTOVÝCH ÚLOH KE SPLNĚNÍ**

Zde je uveden seznam testů, které je potřeba splnit. Po vyplnění a odeslání testu Vám bude test vyhodnocen a uložen do databáze.

Možnost opravy testu bude možná až po domluvě s vedoucím kurzu.

Název testu	Datum vykonání testu	Úspěšnost	Počet pokusů
<a href="#">Cimbál kvíz</a>	2011-06-08 10:53:05	0 %	4 / 5
<a href="#">Dějiny hudby</a>	2011-05-29 18:00:21	100 %	1 / 4

**VŠECHNY VAŠE TESTY**

Seznam testů zadaných pro Vás. Naleznete zde neaktivní testy nebo testy s maximálním dosaženým počtem pokusů.

Procenta u každého testu zobrazují výsledek posledního pokusu.

Název testu	Datum vykonání testu	Úspěšnost	Počet pokusů
Cimbál kvíz	2011-06-08 10:53:05	0 %	4 / 5
Dějiny hudby	2011-05-29 18:00:21	100 %	1 / 4

*Obrázek 50: Testy*


Máte-li v testu více jak jednu správnou odpověď a odpovíte pouze na část odpovědí správně, nezískáte ani tuto část otázky za menší počet bodů. Testy jsou koncipovány tak, že buď odpovíte na odpovědi správně a dostanete 100% bodů z otázky nebo odpovíte špatně a získáte 0% bodů z otázky viz *Obrázek 51*.

## Výsledky testu

Otázka číslo 4 je správně. (odpověď číslo 2)

Otázka číslo 5 je správně. (hodnota : c,d,e,f,g,a,h,c)

Máte **2 z 2** dobře. Vaše úspěšnost je :**100 %**.

 Test byl zaznamenán!

## DĚJINY HUDBY

1. Který významný hudební skladatel napsal 7 symfonií

- Bach
- Betowen
- Dvořák
- Janáček

2. Vypište stupnici C dur (odpovědi oddělte čárkou)

Vyhodnotit

Obrázek 51: Vzorový test s vyhodnocením

## ZÁVĚR

Cílem diplomové práce bylo vytvoření výukového portálu hudební teorie. Bylo třeba vytvořit stránky pro snadnou a rychlou orientaci. Nedílnou součástí celého systému pak bylo vytvoření administrace CMS systému pro pohodlné vkládání obsahu a komplexní správu výukových modulů s generátorem notové osnovy.

V teoretické části jsem popsal jazyky a moderní javascriptové knihovny, které jsem využil při tvorbě celého systému. Základním prvkem byl CMS systém pracující na technologii PHP, MySQL a HTML 5. Dále bylo potřeba popsat javascriptová API JQUERY a VEXFLOW, bez kterých by nebylo možné pracovat s notovými osnovami.

V praktické části jsem popsal všechny funkce a moduly CMS systému. Uživatel by měl být schopný použít tuto práci jako manuál, podle kterého zvládne vytvořit, upravit všechny obsahy portálu a spravovat jak uživatele, tak výukové kurzy, kvízy a notové osnovy. Pomocí modulu pro generování notové osnovy uživateli stačí jen základní znalosti v oblasti hudební teorie a základní syntaxe zápisu not k tomu, aby vytvořil několikařádkovou notovou osnovu, propojil ji s kvízovým modulem a prezentoval na webovém portále. V příloze jsou přiloženy zdrojové kódy 2 notových osnov, které jsem nechal systémem vygenerovat a uložit do databáze. Při správné konfiguraci celého systému, je možné zdrojový kód zkopírovat a po vložení do HTML by se měla zobrazit daná notová osnova.

Závěrem bych rád podotknul, že tento systém je díky API JQUERY obohacen o velmi efektivní práci asynchronního javascriptu s pěknými animacemi. Na celém systému se stále pracuje a veškeré vazby mezi administračním rozhraním CMS systému a samotným API WEXFLOW se stále vyvíjí. Do budoucna by celý systém měl být schopný uspokojit i náročného vyučujícího hudební teorie včetně hudebního podkladu.

## CONCLUSION

The aim of this thesis was to create a learning portal of music theory. It was necessary to create a simple page for easy and quick navigation on the web portal. An integral part of the whole system of administration was the creation of CMS system for easy loading of content and complex management teaching modules with a generator system of notes.

In the theoretical part I described the modern languages and JavaScript libraries that I used for designing the whole system. The basic element of the CMS working on technology PHP, MySQL and HTML 5. It was necessary to describe the JQuery JavaScript API and VEXFLOW, without which it would be possible to work with musical note curriculum.

In the practical part I described all the features and modules of CMS. The user should be able to use this thesis as a manual, that manages to create, edit all the contents of the portal and manage both users and tutorials, quizzes and musical curriculums. Using the module for generating musical notes require just a basic knowledge of music theory and basic syntax to write notes to create multiline stave, connected it with quizze module and presented on the web portal. Please find attached the source code 4 sheet music curriculum, that I let to be generátor by the systém and stored in databáze. With proper configuration of the system, it is possible source-code copy and paste into the HTML and than you should see given stave.

Finally I would like to observe that the system is enhanced by API JQuery is a very effective job of asynchronous JavaScript with nice animations. On the entire system we still works and all links between the CMS administration interface and the API of WEXFLOW are still in progress. In the future, the system should be able to satisfy the ambitious teacher of music theory, including background music.

**SEZNAM POUŽITÉ LITERATURY**

- [1] Co je CMS [online]. [cit. 2011-06-10]. Dostupný z WWW:  
<http://www.voxcafe.cz/clanky/optimalizace-stranek/co-je-cms.html>
- [2] Open Source Software [online]. [cit. 2011-06-10]. Dostupný z WWW:  
[http://cs.wikipedia.org/wiki/Open\\_source\\_software](http://cs.wikipedia.org/wiki/Open_source_software)
- [3] HTML [online]. [cit. 2011-06-10]. Dostupný z WWW:  
[http://cs.wikipedia.org/wiki/HyperText\\_Markup\\_Language](http://cs.wikipedia.org/wiki/HyperText_Markup_Language)
- [4] PHP [online]. [cit. 2011-06-10]. Dostupný z WWW:  
<http://cs.wikipedia.org/wiki/Php>
- [5] MySQL [online]. [cit. 2011-06-10]. Dostupný z WWW:  
<http://cs.wikipedia.org/wiki/Mysql>
- [6] Příklady MySQL [online]. [cit. 2011-06-10]. Dostupný z WWW:  
<http://vyuka.fai.utb.cz/mod/resource/view.php?id=602>
- [7] Co je to SMARTY [online]. [cit. 2011-06-10]. Dostupný z WWW:  
<http://smarty.ronnieweb.net/co-je-smarty.php>
- [8] JQUERY [online]. [cit. 2011-06-10]. Dostupný z WWW:  
<http://cs.wikipedia.org/wiki/JQuery>
- [9] Cufon [online]. [cit. 2011-06-10]. Dostupný z WWW:  
<http://suewebik.net/index.php?/archives/695-Cufon.html>
- [10] ULLMAN, Larry. HTML : PHP a MySQL. Brno : Computer Press, 2004. ISBN 80-251-0063-4.
- [11] HENK C.A : van TILBORG. Fundamentals of cryptology. Norwell : Kluwer Academic, 2000. ISBN 0-7923-8675-2.
- [12] ZELINKA, Ivan. Aplikovaná informatika. Zlín : Ediční středisko, UTB, 1999. ISBN 80-214-1423-5.
- [13] SCHLOSSNAGLE, Georgie. Pokročilé programování v PHP 5. Computer Press, 2004. ISBN 80-86815-14-5.

- [14] LACKO, Luboslav. PHP5 a MYSQL5 Hotová řešení. Computer Press, 2007. ISBN 978-80-251-1695-1.
- [15] KOLEKTIV AUTORŮ. Mistrovství v PHP 5. Computer Press, 2007. ISBN 978-80-251-1519-0.
- [16] CASTRO, Elizabeth. HTML : XHTML a CSS – Názorný průvodce tvorbou WWW stránek. Computer Press, 2007. ISBN 978-80-251-1531-2.
- [17] RESIG, John. Javascript a AJAX : Moderní programování webových aplikací. Computer Press, 2007. ISBN 978-80-251-1824-5.
- [18] VexFlow [online]. [cit. 2011-06-10]. Dostupný z WWW: <http://vexflow.com/docs/tutorial.html>



**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

API	Rozhraní, které obsahuje procedury a funkce knihovny, které může programátor využívat
HTML	HyperText Markup Language
XHTML	HyperText Markup Language s jistým standartem zápisu
WML	Wireless Markup Language
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
FTP	File Transfer Protokol
IMAP	Internet Message Access Protocol
POP3	Post Office Protocol verze 3
LDAP	Lightweight Directory Access Protocol
SQL	Simple Query Language
MySQL	Databázový systém SQL
ODBC	Open Database Connectivity
Oracle	Databázový systém
PostgreSQL	Databázový systém SQL
MSSQL	Databázový systém SQL založený na platformě windows
PHP	Rekurzivní zkratka PHP: Hypertext Preprocessor
POST	http metoda odesílání dat z formuláře
GET	http metoda odesílání dat z formuláře
COOKIE	malé množství dat uložené na Vašem počítači
CMS	Zkratka pro Správu webového obsahu, označují se tak webové redakční systémy
javascript	JavaScript je programovací jazyk, který pracuje na straně klienta

---

JQUERY	Robustní javascriptové API
VEXFLOW	API pro generování notové osnovy
Cache	Dočasné místo pro práci s daty
plugin	Zásuvný modul
tag	Část značek v kódu HTML jazyka
sitemap	Mapa stránek
kernel	Jádro systému
cross-browser	Klientský skript, který vytváří podporu mezi všemi prohlížeči
AJAX	Asynchronní javascript, který umožňuje načítat PHP data z javascriptu.
WHATWG	Web Hypertext Application Technology Working Group
SGML	Standard Generalized Markup Language
CSS	Kaskádové styly
LAMP	spojení Linux, Apache, MySQL a PHP nebo Perl

**SEZNAM OBRÁZKŮ**

<i>Obrázek 1 : Vzhled administrace .....</i>	<i>37</i>
<i>Obrázek 2 : Ukázka administrace správy obsahu.....</i>	<i>38</i>
<i>Obrázek 3. : Vkládání nového obsahu .....</i>	<i>39</i>
<i>Obrázek 4. : Prostředí pro vytváření obsahu.....</i>	<i>40</i>
<i>Obrázek 5. : Vložit/Upravit Obrázek .....</i>	<i>40</i>
<i>Obrázek 6 : Vložit/Upravit Obrázek .....</i>	<i>41</i>
<i>Obrázek 7: Výběr obrázku .....</i>	<i>41</i>
<i>Obrázek 8: Vložit/Upravit Obrázek .....</i>	<i>42</i>
<i>Obrázek 9: Správce obrázků.....</i>	<i>43</i>
<i>Obrázek 10: Editor obrázků .....</i>	<i>43</i>
<i>Obrázek 11: Data uložená v databázi.....</i>	<i>44</i>
<i>Obrázek 12: Správa novinek.....</i>	<i>45</i>
<i>Obrázek 13: Vložení novinky .....</i>	<i>45</i>
<i>Obrázek 14: Vložení novinky .....</i>	<i>46</i>
<i>Obrázek 15: Seznam všech uživatelů na webu.....</i>	<i>46</i>
<i>Obrázek 16: Nastavení práv .....</i>	<i>47</i>
<i>Obrázek 17: Editace uživatele .....</i>	<i>47</i>
<i>Obrázek 18: Všechny Fotogalerie .....</i>	<i>48</i>
<i>Obrázek 19: Nová fotografie .....</i>	<i>48</i>
<i>Obrázek 20: Správa složek.....</i>	<i>49</i>
<i>Obrázek 21: Tvorba nové složky.....</i>	<i>49</i>
<i>Obrázek 22: Nastavení modulů v systému .....</i>	<i>50</i>
<i>Obrázek 23: Nastavení modulů v systému .....</i>	<i>51</i>
<i>Obrázek 24: Nastavení webu .....</i>	<i>52</i>
<i>Obrázek 25: Vložení nové položky jazykového layoutu .....</i>	<i>53</i>
<i>Obrázek 26: Nastavení jazyků .....</i>	<i>53</i>
<i>Obrázek 27: Správa kurzů .....</i>	<i>54</i>
<i>Obrázek 28: Registrovaní žáci.....</i>	<i>54</i>
<i>Obrázek 29: Tvorba kurzu .....</i>	<i>55</i>
<i>Obrázek 30: Generátor notových osnov .....</i>	<i>56</i>
<i>Obrázek 31: Tvorba notové osnovy – první krok.....</i>	<i>57</i>

<i>Obrázek 32: Tvorba notové osnovy – druhý krok</i> .....	57
<i>Obrázek 33: Vzorová notová osnova – stupnice C dur</i> .....	58
<i>Obrázek 34: Uložení notové osnovy do databáze</i> .....	58
<i>Obrázek 35: Kvízový modul – přehled</i> .....	59
<i>Obrázek 36: Výsledky testů</i> .....	59
<i>Obrázek 37: Tvorba kvízu – první krok</i> .....	60
<i>Obrázek 38: Tvorba kvízu – druhý krok</i> .....	60
<i>Obrázek 39: Tvorba kvízu – otázky a odpovědi</i> .....	61
<i>Obrázek 40: Tvorba kvízu – otázka s notovou osnovou</i> .....	62
<i>Obrázek 41: Tvorba kvízu – ukázky typu odpovědi radio</i> .....	62
<i>Obrázek 42: Tvorba kvízu – ukázky typu odpovědi checkbox</i> .....	62
<i>Obrázek 43: Publikace stránek portálu</i> .....	63
<i>Obrázek 44: Přihlášení do administrace</i> .....	64
<i>Obrázek 45: Přihlášení do administrace bez zapnutého javascriptu</i> .....	64
<i>Obrázek 46: Registrace nového uživatele</i> .....	72
<i>Obrázek 47: Přihlášení na portál</i> .....	73
<i>Obrázek 48: Žádost a přihlášení do kurzu</i> .....	73
<i>Obrázek 49: Moje kurzy</i> .....	74
<i>Obrázek 50: Testy</i> .....	75
<i>Obrázek 51: Vzorový test s vyhodnocením</i> .....	76

## SEZNAM PŘÍLOH

Příloha P1 – notová osnova stupnice C dur

Příloha P2 – notová osnova akordu C

## PŘÍLOHA PI: NOTOVÁ OSNOVA STUPNICE C DUR

```
<canvas class="osnovy" id="osnova-test-1-stupnice-c-dur" width="560" height="100"></canvas>
<script>
```

```
    var canvas = $("#canvas#osnova-test-1-stupnice-c-dur")[0];
    var renderer = new Vex.Flow.Renderer(canvas,
        Vex.Flow.Renderer.Backends.CANVAS);

    var ctx = renderer.getContext();

    var stave_0 = new Vex.Flow.Stave(0, 0, 530);
    stave_0.addClef("treble").setContext(ctx).draw();

    // Create the notes
    var notes = [new Vex.Flow.StaveNote({ keys: ["c/4"], duration: "q" }),new
Vex.Flow.StaveNote({ keys: ["d/4"], duration: "q" }),new Vex.Flow.StaveNote({ keys:
["e/4"], duration: "q" }),new Vex.Flow.StaveNote({ keys: ["f/4"], duration: "q" }),new
Vex.Flow.StaveNote({ keys: ["g/4"], duration: "q" }),new Vex.Flow.StaveNote({ keys:
["a/4"], duration: "q" }),new Vex.Flow.StaveNote({ keys: ["b/4"], duration: "q" }),new
Vex.Flow.StaveNote({ keys: ["c/5"], duration: "q" })];

    // Create a voice in 8/4
    var voice0 = new Vex.Flow.Voice({
        num_beats: 8,
        beat_value: 4,
        resolution: Vex.Flow.RESOLUTION
    });

    // Add notes to voice
    voice0.addTickables(notes);

    // Format and justify the notes to 800 pixels
    var formatter = new Vex.Flow.Formatter().
        joinVoices([voice0]).format([voice0], 800);

    // Render voice
    voice0.draw(ctx, stave_0);

</script>
```

## PŘÍLOHA P II: NOTOVÁ OSNOVA AKORDU C

```
<canvas class="osnovy" id="osnova-test-1-akord-c" width="560" height="100"></canvas>
<script>

    var canvas = $("#canvas#osnova-test-1-akord-c")[0];
    var renderer = new Vex.Flow.Renderer(canvas,
        Vex.Flow.Renderer.Backends.CANVAS);

    var ctx = renderer.getContext();

    var stave_0 = new Vex.Flow.Stave(0, 0, 520);
    stave_0.addClef("treble").setContext(ctx).draw();

    // Create the notes
    var notes = [new Vex.Flow.StaveNote({ keys: ["c/4", "e/4", "g/4"], duration:
"q" }));

    // Create a voice in 1/4
    var voice0 = new Vex.Flow.Voice({
        num_beats: 1,
        beat_value: 4,
        resolution: Vex.Flow.RESOLUTION
    });

    // Add notes to voice
    voice0.addTickables(notes);

    // Format and justify the notes to 800 pixels
    var formatter = new Vex.Flow.Formatter().
        joinVoices([voice0]).format([voice0], 800);

    // Render voice
    voice0.draw(ctx, stave_0);

</script>
```