



Středoškolská technika 2010

Setkání a prezentace prací středoškolských studentů na ČVUT

Tvorba www stránek

Jakub Dittrich

**Vyšší odborná škola a Střední škola slaboproudé elektrotechniky
Novovysočanská 48/280
190 00 Praha 9**



OBSAH:

OBSAH	2
1. ÚVOD	3
2. TEORETICKÝ ROZBOR	4
2.1. Rozbor požadavků na školní webové stránky	4
2.1.1. Hlavní cíl školních stránek	5
2.1.2. Cílová skupina návštěvníků	5
2.1.3. Obsahová stránka	5
2.1.4. Redakční rada a správa webu	5
2.2. Realizace webových stránek	7
2.2.1. Webhosting	7
2.2.2. HTML	8
2.2.2.1. Vývoj jazyka	8
2.2.2.2. Popis a koncepce jazyka	9
2.2.2.3. Značky a příklad zdrojového kódu	11
2.2.2.4. Parsování v prohlížečích	12
2.2.2.5. Editory HTML	13
2.2.2.6. Budoucnost HTML	13
2.2.3. CSS	13
2.2.3.1. Syntaxe	14
2.2.3.2. Připojení kaskádových stylů do HTML stránky	15
2.2.3.3. Výhody CSS	15
2.2.3.4. Nevýhody CSS	16
2.2.3.5. Limity CSS	17
2.3. Možnosti volby programových prostředí	18
2.3.1. ASP.NET	18
PHP	19
2.4. Statické stránky	20
2.5. Dynamické stránky	20
2.6. Redakční systémy	22
2.6.1. Redakční systém JOOMLA!	23
2.6.2. Redakční systém Drupal	24
2.6.3. Redakční systém phpRS	25
3. STRUKTURA REALIZOVANÝCH WEBOVÝCH STRÁNEK	26
4. IMPLEMENTACE JINÝCH APLIKACÍ	28
4.1. Textový editor TinyMCE	28
4.2. Fancybox	29
4.3. jQuery	29
5. ZHODNOCENÍ VLASTNOSTI	30
6. ZÁVĚR	31

1. ÚVOD

Zadáním mé maturitní práce bylo vytvoření školních webových stránek s co možná největší kontrolou obsahu a struktury pomocí vytvořeného redakčního systému. Samotnou tvorbou webových stránek jsem se začal zabývat v první ročníku střední školy pomocí samostudia za účely potřeby vytvoření prezentace hudební kapely. To po třech letech zdokonalování a učení se novým věcem vedlo k práci na nových školních webových stránkách.

Vypracováním mé maturitní práce bych chtěl přiblížit teorii o tom, jak vůbec takové webové stránky vznikají a fungují, osvětlit používané principy a metody, a pomocí vytvořené a přiložené přílohy ukázat funkcionalitu stránek samotných.

V mé práci vás seznámím se základními požadavky na tvorbu školních webových stránek, vysvětlím potřebné věci pro jejich chod, základy používaných programovacích jazyků, strukturu realizovaných webových stránek, použité externí aplikace a používané a obecně známe webové redakční systémy.

2. TEORETICKÝ ROZBOR

2.1 ROZBOR POŽADAVKŮ NA WEBOVÉ STRÁNKY

Internetová prezentace je v dnešní době velmi silným nástrojem, ať už za účelem získání informací, aktuálního dění, kontaktu nebo prezentace daného subjektu návštěvníkům. Internet je dnes masově využíván opravdu všude a proto ani prezentace základních, středních nebo vysokých škol není výjimkou. S mého vlastního pohledu bych si troufl říci, že u vysokých škol jsou webové stránky v dnešní době naprostou samozřejmostí, u středních škol by tomu mohlo být podobně, ale rozdíly v použitelnosti, v kvalitě dohledaných informací nebo v kvalitě samotných webových stránek jsou velké a stránky nejsou vždy tak informačně použitelné. U základních škol, dle statistických údajů z roku 2005, bylo využíváno webových stránek jen u 20% z nich, což je při jejich velikém počtu poměrně malý podíl. V dnešní době je toto procento určitě vyšší a stále narůstá. Právě u základních škol je potenciál webových stránek a jejich možnosti informování a komunikace nejlepším řešením jak pro školu, tak i pro rodiče, zvláště u těch, jež jsou rodiči žáků nejnižších ročníků.

2.1.1 HLAVNÍ CÍL ŠKOLNÍCH WEBOVÝCH STRÁNEK

Pro školu, ale i pro webové stránky samotné je špatnou vizitkou, pokud postupem času chátrají a jsou zanedbávány. Platí nepsané pravidlo, že zastaralé stránky či neaktuální informace jsou horší než stránky žádné. Stránky by měly v jistém slova smyslu žít, měly by být tedy aktualizovány, návštěvníci by měli nacházet nový obsah a odcházet s tím pocitem, že až se příště vrátí, opět se něco nového dozví.

Školní webové stránky by měly obsahovat následující informace:

- obecně by měly představovat profil školy, její zaměření a nabídku jednotlivých vzdělávacích oborů případně kroužků, profil absolventů a pedagogické zázemí
- informovat návštěvníky o organizaci školního roku, nejlépe pomocí aktualit, které jsou nejčastějším řešením, mít k dohledání důležité kontakty nebo termíny, jako jsou prázdniny, apod.
- mohou zprostředkovávat přijímací řízení, informovat uchazeče o výsledku pomocí systému, ke kterému se přihlásí prostřednictvím přiděleného hesla
- informace o konaných kurzech a jejich nabídce
- pro rodiče by měla být možnost přístupu ke klasifikaci a výsledkům prací svých dětí, jejich rozvrhu a suplování, tyto funkce dnes obstarávají známé systémy jako je Bakalář, iŠkola nebo Webnotes
- mohou prezentovat a dát vědět o svých projektech nebo spolupráci, tím mít možnost získat granty
- dostupnost studijních materiálů pro žáky školy

Příkladů by šlo uvést ještě více, od přihlášek a vyhlášení různých soutěží, reklamy doplňkové činnosti školy až po výměnu pedagogických zkušeností učitelů.

Stránky by měly ale obecně a hlavně představovat školu, nabídku studia, případné porovnání s ostatními školami a zodpovězení otázky, proč si právě vybrat tuto školu?

2.1.2 CÍLOVÁ SKUPINA NÁVŠTĚVNÍKŮ

Návštěvníky stránek můžeme rozdělit do několika skupin:

- rodiče žáků současných, kteří na škole studují, ale i rodiče potenciálních uchazečů o studium,
- veřejnost,
- sponzoři,
- noví učitelé,
- jiné školy,
- žáci, studenti,
- zřizovatel, česká školní inspekce, kontrolní orgány.

Obsah webu by měl zaujmout každou z těchto zmíněných skupin. Jednotlivé sekce by měly být obsahem zaměřeny právě na ně, přístup a dohledatelnost by tedy měla být více než jednoduchá. Orientaci po stránkách je nejlepší řešit pomocí srozumitelného menu, které nemá měnit svou polohu a musí být stále viditelné na jakékoliv stránce, případné podstránky musí být řešeny vypsáním dalších doklikatelných položek, tzv. submenu.

2.1.3 OBSAHOVÁ STRÁNKA

Důvodem, proč se návštěvníci vrací na stránky, je kvalitní obsah, který tam najdou. Kvalitní obsah je pro stránky to nejdůležitější, měl by být atraktivní a zajímavý, zacílený na určité, již zmiňované skupiny.

Kvalitu obsahu jde pomyslně rozdělit na dvě skupiny. Jedním příkladem jsou stránky, kdy je obsah tvořen většinou stručnou informací o škole, která návštěvníkovi neřekne nic bližšího o skutečném dění ve škole. Najdete zde adresu školy, pár fotek a nakonec kontakt, jako je email nebo telefon. Pro skutečné informace je tedy potřeba požádat mailem nebo zavolat přímo do školy.

Druhým příkladem jsou weby, které jsou obsahově nabitě, plné aktuálního dění ve škole. Tímto příkladem jsou i mnou vytvářené stránky, kde vám jsou po vybrání z hlavního menu nabídnuty další podsekce, ve kterých najdete mnoho informací. Tyto sekce jsou upravovány, vytvářejí se nové a návštěvník se vždy něco dozví - to je důvod, proč by se měl vracet, stránky žijí. Není tedy dnes v módě jakýmkoliv způsobem udržovat statické, neaktualizované weby.

2.1.4 REDAKČNÍ RADA

Pro údržbu a správný chod webových stránek by si škola měla vytvořit vlastní tzv. redakční radu. Tu by měli tvořit zodpovědní pracovníci na pozicích:

- senior manager – má všechna přístupová práva a zodpovědnost za obsah,
- project manager – zajišťuje aktuálnost a pravidelnou aktualizaci stránek,
- editor – stará se o obsahovou stránku,
- sub-editor – ručí za jazykovou správnost, provádí jazykové korektury.

Pro dobrou funkčnost webu zvláště na velkých školách, může být toto rozčlenění velmi přínosné a efektivní. Avšak není podmínkou, aby rada byla takto hierarchicky členěna a skládala se z tolika členů. Důležité je hlavně to, aby lidé na této práci spolupracovali.

Pokud na stránkách pracuje jen jeden člověk, web vzniká poměrně rychleji, příslušný pracovník se nemusí s nikým domlouvat, není brzděn kritickými připomínkami, má jasnou představu, kterým směrem web vést a za tou se snaží jít.

Ve větší skupině je komunikace a diskuze složitější a taková tvorba namáhavější a delší. Výsledek ale většinou stojí za to – čím pestřejší je skupina s nápady, tím více je ve výsledku pestřejší web. Pokud je web veden jen jedním člověkem, potom vše stojí a padá na jeho hlavu. Navíc, udržovat web aktuální a zajímavý, je pro jednoho člověka velmi, velmi náročné. Pokud onemocní nebo odejde, je potom náhrada těžko dohledatelná. Ve skupině, která si předem určí úkoly a stanoví jasný cíl, se tedy pracuje lépe a snadněji.

2.2 REALIZACE WEBOVÝCH STRÁNEK

Vytvoření a prezentace webových stránek na internetu není v dnešní době žádným problémem. Jejich založení a pozdější správa může být dnes díky intuitivním nástrojům vytvořených různými firmami otázkou pár kliků počítačové myši. Tyto weby jsou potom zobrazovány jako domény třetího řádu (např.: nazev.domena.cz), obsah je generován do šablony a uživatel mění jen obsah. Tato možnost tvorby, respektive spíše správy, je dobrá například pro založení svého vlastního blogu nebo pro jednorázovou prezentaci. Nevýhodou je, že do stránky jsou většinou umísťovány reklamy od poskytovatele.

Pokud má prezentace dobře vypadat, chcete mít prostor s vlastní doménou, projekt se může zadat firmě, která za částku v tisících a deseti tisících korun vytvoří vámi požadované stránky.

Poslední možností je potom samozřejmě tvorba stránek osobou nezávislou, jako tomu bylo v případě těchto školních stránek, kde bylo využito mých schopností k tvorbě webu. Výhodou byla stálá komunikace, usměrňování práce, doplňování požadavků, dohromady velmi dobrá spolupráce.

Jediné co je k tvorbě webových stránek potřeba, je prostor, kde budou fyzicky uloženy, případný přístup k databázím, a samozřejmě znalosti k tomu příslušné. Potom lze naprogramovat cokoliv, co bude mít zákazník na mysli.

2.2.1 WEBHOSTING

Pro chod webových stránek je potřeba mít je někde umístěny. To znamená, mít prostor, kde se budou fyzicky nacházet – webhosting. Existuje několik řešení, při tvorbě školních stránek, mohou být uloženy na školním webovém serveru, pokud je k dispozici, mohou využívat webhostingu u komerčního poskytovatele nebo použít server nějaké spřízněné instituce.

ŠKOLNÍ WEBSERVER

Pokud chceme využívat školního serveru, musíme brát v potaz dvě věci. První z nich je zatížení serveru a konektivita, tedy rychlost připojení k internetu. Pokud návštěvnost stránek bude řádově v desítkách přístupů denně, součástí obsahu na stránkách nebude nějak náročně streamovaný soubor, jako je video či jiný datově náročný materiál, pak můžeme bez potíží použít školní server.

KOMERČNÍ WEBHOSTING

Komerční webhosting je na českém trhu poměrně bohatě dostupný. Firmy nabízejí různé tarify, které se mohou i cenově lišit. V potaz bychom měli brát velikost nabízeného prostoru, tedy kolik megabajtů lze na web umístit. Nabízené tarify mohou mít velikosti od stovek megabajtů až po desítky gigabajtů, které jsou ale pro potřeby školního webu naprosto zbytečné. Další věci jsou nabízené databáze, emailové schránky, ftp přístupy, všechny tyto jmenované mohou mít omezení, avšak jsou i takové firmy, které mají až na kapacity všechny tyto služby neomezené. Webhosting by měl garantovat dostupnost webu, který bývá většinou udáván v procentech, běžně kolem 99%, frekvenci zálohování, přístup ke statistikám návštěvnosti a možnost spouštění vlastních scriptů. Ceny webhostingů se pohybují ve stovkách korun účtovaných za jeden rok provozu.

SPŘÍZNĚNÁ INSTITUCE

Spřízněnou institucí u webhostingu je myšlena například nějaká organizace, u které je web uložen a škola za ní nemusí platit. Příkladem může být obec, která škole poskytne místo na svém webserveru. Potíží však může být, že možnosti a funkčnost mohou být v tomto případě omezeny. Se všemi požadavky se administrátor stránek musí obracet na správce daného serveru, pro kterého to může být přítěž. Doporučované jsou tedy spíše komerční webhostingy, u kterých máte za zaplacený paušál plnou podporu a okamžitě reagují na vámi zadané požadavky.

2.2.2 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), který je dnešnímu HTML velmi podobný. Vývoj HTML byl také ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

2.2.2.1 VÝVOJ JAZYKA

Jazyk HTML a jeho protokol HTTP (Hyper Text Transfer Protocol – přenosový protokol hypertextu) pro přenos v síti byl navržen v roce 1990 za účely zjednodušení doposud používaných protokolů (TeX, PostScript, SGML). Spolu s první verzí HTML vznikl i první prohlížeč nazvaný jako WorldWideWeb. První web zprovoznilo středisko CERN v roce 1991, poté následoval rychlý rozvoj webu a bylo nutné pro HTML definovat standarty.

VERZE JAZYKA

Verze 0.9

- byla vydána v roce 1991,
- nepodporovala grafický režim (verze, kterou vytvořil Tim Berners-Lee, zakladatel).

Verze 2.0

- zachycuje stav jazyka v polovině roku 1994,
- standard vydala roku 1995 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 (frames),
- tato verze se snaží dosáhnout původního účelu – prvky by měly vyznačovat význam (sémantiku) jednotlivých částí dokumentu, vzhled má být ovlivňován připojovanými styly,
- některé prezentační elementy byly zavrženy.

Verze 4.01

- byla vydána 24. prosince 1999,
- 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, jejímž 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 (odkdy ji začnou vývojáři webových aplikací používat), ukončení vývoje specifikace po vyřešení problémů a opravení všech chyb se odhaduje až na rok 2022.

2.2.2.2 POPIS A KONCEPCE JAZYKA

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 je otevírací značka pro zvýraznění textu a Červená Karkulka 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 koncová 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>
```

Jiná možnost zápisu odkazu – odkaz, který se otevře v novém okně/panelu:

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

Další možnost zápisu odkazu – odkaz, který se otevře v novém okně/panelu, obsahuje titulek (zobrazí se obvykle po najetí myši) a má přiřazenou třídu, kterou pak můžeme využít například v 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</li><li>Božena</li></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),
- kód skriptovacích jazyků,
- definice událostí a kód pro jejich obsluhu.

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,
- 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>,
- tělo dokumentu – obsahuje vlastní text dokumentu, vymezuje se značkami <body> a </body>

Pozn.: Všechny tři elementy jsou povinné, ale otevírací a ukončovací značka samotná povinná není, pokud tyto značky nebudou v těle dokumentu uvedeny, prohlížeč si je sám doplní podle kontextu.

2.2.2.3 ZNAČKY A 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>
```

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 (), od tohoto druhu značek se postupně upouští, trendem je používání kaskádových stylů, které vzhled popisují odděleně od obsahu dokumentu, mezi důvody pro neužívání 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.

2.2.2.4 PARSOVÁNÍ V PROHLÍŽEČÍCH

Webové prohlížeče jsou programy, jejichž účelem je prezentovat dokument na zobrazovacím zařízení – nejčastěji monitoru počítače. Dokument je prohlížečem načítán a je 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,
- Quick 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 (a chování) 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 rozdíl 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.

2.2.2.5 EDITORY HTML

Editory HTML jsou programy pro snadnou tvorbu webových stránek. Dají se rozlišit dva druhy:

- 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 z 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 patří Adobe Dreamweaver nebo Expression Web (novější verze Microsoft Frontpage).

2.2.2.6 BUDOUCNOST HTML

Vývoj jazyka HTML byl původně ukončen verzí 4.01. Dle W3C měl další vývoj psaní dokumentů na web patřit jazyku XHTML – následníkovi HTML, využívajícímu univerzální jazyk XML. Některým lidem se však vývoj okolo XHTML nezamlouval. Část z nich včetně některých tvůrců webových prohlížečů, jako například Mozilla Foundation, Opera Software či Apple, založila iniciativu WHATWG, jejímž cílem bylo vytvořit novou verzi HTML, která se posléze začala označovat jako „HTML 5“. 7. března 2007 W3C založilo novou pracovní skupinu HTML, jejímž cílem je v roce 2010 uvolnit specifikaci nové verze HTML. Po hlasování bylo určeno, že nová verze ponese označení HTML 5.0 a že zpočátku bude založena na specifikacích Web Applications 1.0 a Web Forms 2.0 z iniciativy WHATWG, které budou upravovány. Krom toho pracuje W3C dále i na specifikacích XHTML 2.0 a XForms, které původně považovala za jediný budoucí směr vývoje, avšak na konci roku 2009 měla být pracovní skupina XHTML 2.0 rozpuštěna.

2.2.3 CSS

Kaskádové styly (v anglickém originále Cascading Style Sheets se zkratkou CSS) jsou jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány zatím dvě úrovně specifikace CSS1, CSS2, dokončuje se revize CSS 2.1 a pracuje se na verzi CSS3. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu

dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

2.2.3.1 SYNTAXE

Tabulka kaskádových stylů sestává z několika pravidel. Každé pravidlo obsahuje selektor a blok deklarácí. Každý blok deklarácí pak obsahuje seznam deklarácí a každá deklaráce sestává z vlastnosti, následuje dvojtečka : a hodnota vlastnosti. Deklarace končí středníkem ;. Před středníkem ještě může nepovinně být označení !important, které zvýší sílu deklaráce.

Příklad pravidla:

```
body
{
    background-color: white;
    color: black;
    padding: 10px !important;
}
```

Celý blok nazveme pravidlo, „body“ je selektor, to v závorkách je blok deklarácí, řádek „background-color: white;“ je deklaráce samotná, „background-color“ je vlastnost a „white“ je hodnota. Celý kód pak nastavuje barvu pozadí stránky na bílou, barvu textu na černou a okraj na 10 pixelů.

Selektory

CSS definuje mnoho různých selektorů, které obvykle můžeme kombinovat.

Mezi základní patří:

- body – tyto deklaráce budou platit pro všechny výskyty elementu body,
- body p – tyto deklaráce budou platit pro všechny elementy p, které se nachází v elementu body, v jakékoliv hloubce,

- body>div – tyto deklaráce budou platit pro všechny elementy div, které jsou dětmi elementu body, to znamená, že pokud bychom měli element div, který se nachází v <body><blockquote><div>..., tyto deklaráce by pro něj neplatily, protože tento div není přímým potomkem elementu body,
- .trida – tyto deklaráce budou platit pro všechny elementy, které mají v HTML nastavenou třídu trida, to se provádí pomocí HTML atributu class,
- #id – tyto deklaráce budou platit pro všechny elementy, které mají v HTML nastavený identifikátor id. To se provádí pomocí HTML atributu id,
- sel1, sel2, sel3 – selektory můžeme seskupovat pomocí čárek, následující deklaráce pak budou platit pro všechny selektory.

2.2.3.2 PŘIPOJENÍ KASKÁDOVÝCH STYLŮ DO HTML STRÁNKY

Máme několik možných způsobů, jak aplikovat kaskádové styly v HTML dokumentu:

- Přímý inline zápis stylu pomocí atributu style. Tato pravidla budou aplikována pouze na dotýčný element.

```
<p style="color: red; text-decoration: underline">Tento odstavec bude červený a  
podtržený.</p>
```

- Zápis stylů do elementu style. Takové styly se aplikují na celou stránku podle předepsaných selektorů.

```
<style type="text/css">  
#hlavicka  
{  
    width: 200px;  
    height: 450px;  
}  
</style>
```

- Připojení externího souboru pomocí elementu link.

```
<head>  
    <link rel='stylesheet' href='style.css' type='text/css'>  
</head>
```

- Připojení externího souboru pomocí http hlavičky link.

Link <fancy.css>; rel=stylesheet, <fancy.css>;rel=stylesheet

2.2.3.3 VÝHODY CSS

Používání kaskádových stylů ve srovnání se samotným HTML v praxi přináší výhody:

- rozsáhlejší možnosti formátování, například pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic, CSS má vlastnosti padding a margin, v HTML by bylo potřeba vytvořit složitou konstrukci vnořených tabulek,
- jednodušší údržba webové prezentace - pokud chceme změnit nějaký detail, jako třeba barvu nadpisu, nemusíme složitě procházet HTML kód nebo různé HTML šablony, ale můžeme změnit pouze jednu vlastnost v CSS souboru, který je přilinkován ke všem stránkám,
- oddělení struktury a stylu - v jednom (HTML) dokumentu budeme mít pouze sémanticky označen obsah a v druhém (CSS) dokumentu máme definice vzhledu

stránek, tím dosáhneme snadnějšího strojového zpracování samotného obsahu stránek, do kterého se nám nepletou prvky definující vzhled,

- cachování stylů – webový prohlížeč si může soubor se styly uložit do cache paměti, čímž může být dosaženo zrychlení načtení stránky, na druhou stranu při použití externího CSS souboru dochází k dalšímu HTTP požadavku navíc oproti tomu, když bychom použili buď inline CSS nebo přímo formátování HTML,
- CSS vlastnosti jednotlivých elementů můžeme dynamicky měnit pomocí Javascriptu,
- s pomocí CSS můžeme jednoduše formátovat i jakýkoliv jazyk XML,
- mohou také existovat různé styly pro různá výstupní zařízení, jednoduše tak můžeme nadefinovat různý styl pro tisk, projekci, mobil, PDA, běžný prohlížeč či dokonce pro každý prohlížeč jiný styl, specifikace CSS nezapomínají ani na zrakově postižené – je možno napsat styly pro hlasový syntetizátor nebo hmatovou čtečku Braillova písma
- koncový uživatel si může napsat svůj vlastní CSS styl pro libovolnou stránku, většina prohlížečů nějakým způsobem podporuje uživatelské styly, takže uživatel si může například nastavit, aby měl všechny odkazy na všech webech vždy podtržené nebo aby na tomto konkrétním webu mělo písmo vždy černou barvu,
- v kombinaci s Javascriptem mohou vzniknout účinné bookmarklety, které mohou různě vylepšovat vzhled stránky, například odstranit všechny obrázky na pozadí, změnit pozadí na bílé a písmo na černé apod.

2.2.3.4 NEVÝHODY CSS

Hlavní nevýhodou CSS je ne vždy dostatečná podpora v majoritních prohlížečích. Prohlížeče obsahují v implementaci CSS chyby a je někdy nesnadné napsat kód tak, aby se ve všech (resp. v běžných) prohlížečích výsledek zobrazil stejně. Situace se poslední dobou značně zlepšuje, mj. opravami chyb v prohlížečích a nástupem nových verzí Internet Exploreru, který byl ve své šesté verzi zdrojem častých chyb v CSS.

Občas se používají podmíněné komentáře, které umožňují definovat na úrovni HTML kód, který bude viditelný pouze Internet Exploreru, či naopak.

```
<!--[if IE]>  
<style type="text/css">  
#upozorneni  
{  
  color: blue;  
}  
</style>  
<![endif]-->
```

Tento kód bude interpretován pouze Internet Explorerem, ostatní prohlížeče uvidí obyčejný HTML komentář a interní stylpis tak nebudou interpretovat.

2.2.3.5 LIMITY CSS

- CSS selektory neposkytují přístup k rodičovským prvkům, jako to umožňují sofistikovanější jazyky jako například Xpath, nemůžete například nastylovat jen ty odstavce, které obsahují odkaz,
- horizontální kontrola prvků na stránce je intuitivní a jednoduchá, naopak vertikální stylování už působí problém, například na obyčejné vertikální centrování obsahu se musí používat poměrně složité kombinace a hacky, které ještě nemusí být dopředně kompatibilní,
- CSS neposkytuje možnost pro symbolický zápis proměnné nebo konstanty, všechny hodnoty musí být vepsány přímo v kódu, například pokud se na vícero místech používá stejná barva, nemůže se použít symbolický zápis `barva=red`; a poté už jen psát `proměnnou barva`, všude se musí uvádět přímo hodnota `red`,
- CSS neumí počítat výrazy, prohlížeče například neporozumí výrazu `margin-left: 10% - 3em + 4px;`, webdesigner si toto musí spočítat sám,
- CSS2 nenabízí žádnou možnost pro tvorbu kulatých rámečků nebo jiných kulatých objektů, pracuje pouze s obdélníky,
- CSS2 nenabízí žádnou možnost, jak jednomu elementu přiřadit více obrázků na pozadí.

2.3 MOŽNOSTI VOLBY PROGRAMOVÝCH PROSTŘEDÍ

2.3.1 ASP.NET

ASP.NET je součástí .NET Frameworku pro tvorbu webových aplikací a služeb. Je nástupcem technologie ASP (Active Server Pages) a přímým konkurentem JSP (Java Server Pages).

PRINCIP ASP.NET

Ačkoliv název ASP.NET je odvozen od starší technologie pro vývoj webů ASP, obě technologie jsou velmi odlišné. ASP.NET je založen na CLR (Common Language Runtime), který je sdílen všemi aplikacemi postavenými na .NET Frameworku. Programátoři tak mohou realizovat své projekty v jakémkoliv jazyce podporujícím CLR, např. Visual Basic.NET, JScript.NET, C#, Managed C++, ale i mutace Perlu, Pythonu a další. Aplikace založené na ASP.NET jsou také rychlejší, neboť jsou předkompilovány do jednoho či několika málo DLL souborů, na rozdíl od ryze skriptovacích jazyků, kde jsou stránky při každém přístupu znovu a znovu parsovány.

Koncept ASP.NET WebForms ulehčuje programátorům přechod od programování klasických aplikací pro Windows do prostředí webu: stránky jsou poskládány z objektů, ovládacích prvků (Controls), které jsou protějškem ovládacích prvků ve Windows. Při tvorbě webových stránek je tedy možné používat ovládací prvky jako tlačítko (Button), nápis (Label) a další. Těmto prvkům lze přiřazovat určité vlastnosti, zachytávat na nich události, atd. Tak, jako se ovládací prvky pro Windows samy kreslí do formulářů na obrazovku, webové ovládací prvky produkují HTML kód, který tvoří část výsledné stránky poslané do klientova prohlížeče.

VÝHODY

- díky kompilovanému kódu běží aplikace rychleji a více chyb je zachyceno už při vývoji,
- uživatelsky definované ovládací prvky lze použít jako šablony, čímž se významně redukuje duplicitní kód,
- podobný přístup jako k aplikacím pro Windows zjednodušuje přechod od jednoho prostředí k druhému,
- bohatý výběr ovládacích prvků a knihoven tříd velmi zrychluje vývoj aplikací,
- programátoři mají na výběr velké množství programovacích jazyků,
- schopnost cachovat celou stránku nebo pouze její části podstatně zvyšuje výkon serveru,
- lze jej provozovat na různých operačních systémech i webových serverech, např. IIS (Windows), Apache (Windows, Linux s open source implementací .NETu Monem),
- počínaje verzí 2 generuje ASP.NET validní HTML 4.0 / XHTML 1.0 / XHTML 1.1 kód a JavaScript.

2.3.2 PHP

PHP (Hypertext Preprocessor, 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 skripty jsou většinou prováděny na straně serveru, k uživateli je přenášén až výsledek jejich činnosti. Syntaxe jazyka je inspirována několika programovacími jazyky jako je Perl, C, Pascal nebo Java. 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 jako jsou MySQL, ODBC, Oracle, PostgreSQL, MSSQL a 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.

Takto v PHP vypadá skript Hello world:

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

VÝHODY

- PHP je specializované na webové stránky
- rozsáhlý soubor funkcí v základní knihovně PHP, další funkce v PECL
- nativní podpora mnoha databázových systémů
- multiplatformost (zejména Linux a Microsoft 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 dnes najdeme všude
- obrovské množství projektů a kódů, které lze zdarma využít (WordPress, phpBB a další)
- slušná dokumentace
- velmi svobodná licence – svobodnější než GPL.

NEVÝHODY

- jazyk PHP není nikde definován, je popsán pouze jeho implementací
- mírně nekonzistentní vývoj v minulosti, který si s sebou PHP nese dosud (někdy často měnící se příkazy či jejich parametry, atd.)

- nekonzistentní pojmenování funkcí, např.: `strpos()`, `strchr()`, ale `str_replace()`, `str_pad()`
- slabší podpora Unicode, pouze přes PHP
- ve standardní distribuci chybí ladící (debugovací) nástroj
- po zpracování požadavku neudrží kontext aplikace, vytváří jej vždy znovu (oslabuje výkon)
- některá bezpečnostní opatření jako `safe_mode`, `open_basedir` jsou implementována ne příliš elegantně

2.4 STATICKÉ STRÁNKY

V dnešní době při obrovském rozmachu internetu a jeho užívání, jsou kladeny daleko větší požadavky na webové stránky nebo aplikace než před několika lety, kdy byl samotný internet ještě v začátcích. Statické stránky jsou dnes tedy téměř nepoužívány, respektive jsou nahrazovány stránkami dynamickými, které umožňují daleko přívětivější a lepší práci, například s obsahem.

Funkčnost statických stránek spočívala v tom, že na straně serveru byly uloženy soubory obsahující HTML kód. Jednotlivé soubory se zobrazovaly zadáním URL adresy v prohlížeči, adresa zde byla zaslána serveru, který následně zaslal celý soubor prohlížeči a ten stránku zobrazil. Mezi jednotlivými stránkami se uživatel pohyboval pomocí odkazů, které právě mění URL adresu a tím zobrazuje jednotlivé soubory.

Tento typ stránek tedy má vždy stejný obsah, jaký je uložen v souborech. Například, pokud máme webovou statickou stránku, kde každý ze souborů obsahuje odkaz, který potřebujeme pozměnit, je třeba provést změny ve všech dokumentech, které daný odkaz obsahují.

Pro velké webové portály nebo stránky, kde je potřeba měnit obsah, je tento typ stránek zcela nepoužitelný, avšak pokud potřebujeme jednoduchou prezentaci, kde se obsah nemění, je tento typ více než vhodný.

2.5 DYNAMICKÉ STRÁNKY

Dynamické stránky jsou dnes nejpoužívanější technologií pro práci s obsahem webových stránek. V porovnání se statickými stránkami jsou při tvorbě značně pracnější, avšak to je vykompenzováno budoucím používáním, které je pro uživatele komfortní a obsluhu stránek zvládne i neškolený uživatel. Není dále potřeba znát žádné programovací jazyky nebo základy HTML značek, jako tomu je u stránek statických.

Základ dynamických stránek tvoří vytvořený program uložený na webovém serveru, dále to jsou databáze a v nich uložená data. Program pomocí parametrů, které jsou nastaveny v URL adrese a odeslány, sestavuje stránku na straně serveru v reálném čase a po dokončení už takto vygenerovanou stránku odesílá zpět uživateli, kde je pomocí prohlížeče zobrazena.

Pro pochopení pojmu program:

Rozumíme tím aplikaci vytvořenou pomocí dané technologie, která nám zajišťuje provoz. Jsou to například technologie ASP, ASP.NET nebo námi používané PHP.

Při tvorbě dynamické stránky se jedná o soubor, který nám například po zobrazení vypíše aktuality. V samotném souboru ale aktuality uloženy nejsou. Je zde vytvořena šablona pomocí HTML značek, do kterých jsou z databáze vypisována data a šablona je defacto jen kopírována. Ve výsledku, kdy server zpracuje všechny požadavky, je stránka odeslaná uživateli obsahově stejná jako stránka statická, obsahuje text, HTML značky pro formátování,

zobrazení médií jako jsou obrázky nebo videa a je tedy de facto identická, avšak v porovnání s původním nezpracovaným souborem je už naplněna žádanými daty, tedy dynamicky zpracována.

Vzorným příkladem dynamických stránek jsou redakční systémy, internetové obchody nebo jako příklad lze uvést i webové mailové klienty, kde jsou data také generována dynamicky.

2.6 REDAKČNÍ SYSTÉMY

CMS (Content Management System – Systém pro Správu Obsahu) neboli redakční systém, je webová aplikace zajišťující správu webového obsahu. Hlavními funkcemi systému je správa souborů, médií, modifikace a publikace dokumentů, správa diskuzních místností, apod.

Obecně se skládá ze dvou částí:

- Webová část
Jsou zde vypisována data, slouží k prohlížení, tedy k tomu, co vidí uživatel navštívených stránek.
- Administrační část
Po přihlášení zde může správce stránek nebo lidé s příslušným oprávněním přístupu spravovat obsah a částečně i chod samotných stránek. Obsahem je myšleno například vkládání článků, přidávání aktualit, modifikace daných sekcí, apod.

K samotné obsluze redakčního systému slouží stejně jako při prohlížení stránek webový prohlížeč a internetové připojení.

Redakční systémy mohou být šířeny jak komerční cestou, kdy je potřeba zaplatit za licenci za používání, tak i cestou takzvaných open-sourceových systémů, které jsou šířeny zdarma a značně tím šetří náklady. Nejčastěji se využívají pro firemní weby, internetové portály, internetové obchody, instituce, apod.

Proč používat redakční systémy a jejich výhody:

Největší výhodou je jejich multifunkčnost, možnost rozšiřovat systém pomocí doplňků, které jsou neustále vytvářeny nebo aktualizovány. Tím je možné získávat nové funkce, stylovat pomocí šablon, kterých je na internetu mnoho a které jdou tvořit i samotnými uživateli.

Jednotlivé systémy mají většinou podobné funkce, kladem všech je například:

- možnost úpravy vzhledu podle šablon,
- rozšíření funkčnosti pomocí doplňků, tzv. plug-inů,
- většinou uživatelská přívětivost, česká lokalizace,
- práce s daty, publikace, mazání a modifikace pomocí WYSIWYG editorů,
- možnost přidělování práv uživatelům.

Nevýhody redakčních systémů:

Každý produkt má i své nevýhody. Já osobně jsem se k používání komerčních nebo open-sourceových redakčních systémů nikdy nedostal, preferuji vlastní tvorbu redakčních systémů na míru pro každé vytvořené webové stránky. Tento postup práce je sice velmi náročný, ale ve výsledku někdy efektivnější. Dostupné redakční systémy jsou dělány pro masy lidí a tím mohou webové stránky ztrácet originalitu.

- kód redakčních systémů bývá nečistý, uživatel nad ním nemá kontrolu, jelikož kódy jsou rozsáhlé a orientace v nich je velice náročná. Tím většinou vzniká i problém se SEO optimalizací daných webových stránek,
- redakční systém je tvořen pro používání tisíci lidmi, je tím sice univerzální, ale pokud uživatel potřebuje nějaké originální funkce, nejsou zde k dispozici,

- při používání doplňků dochází k oslabení zabezpečení systému, jelikož je kód znám a používán mnoha lidmi, hackeři mohou systém studovat a tím se lehčeji dostat do systémů,
- pokud chce mít uživatel originální design nebo stránky samotné, musí si sám vytvořit šablonu a vlastní kostru stránek, jinak hrozí fádnost a obyčejnost.

2.6.1 REDAKČNÍ SYSTÉM JOOMLA!

Redakčních systémů existuje v dnešní době nepřehledné množství, pro příklad vyberu jen ty nejznámější nebo nejrozšířenější.

JOOMLA!

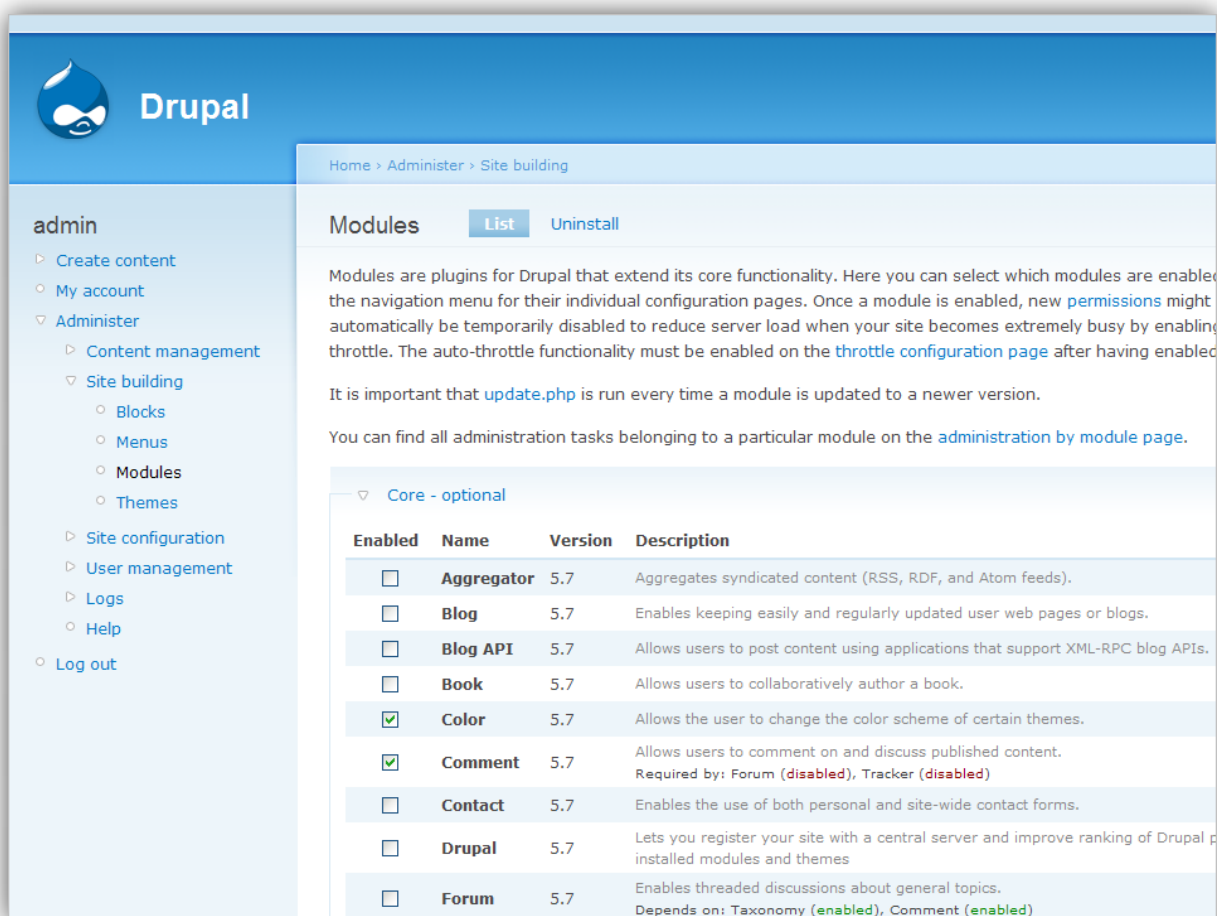
- je open-sourceový, volně šiřitelný redakční systém běžící na technologii PHP s podporou databázi MySQL,
- první jeho verze byla vydána 16. září 2005, dnes je dostupná verze 1.5.

The screenshot displays the Joomla! Administration interface, specifically the Article Manager. The top navigation bar includes 'Site', 'Menus', 'Content', 'Components', 'Extensions', 'Tools', and 'Help'. The main toolbar contains icons for 'Unarchive', 'Archive', 'Publish', 'Unpublish', 'Move', 'Copy', 'Trash', 'Edit', 'New', 'Parameters', and 'Help'. Below the toolbar, there is a filter section with a 'Filter:' input, 'Go', and 'Reset' buttons. The main content area is a table of articles with the following columns: #, Title, Published, Front Page, Order, Access Level, Section, Category, Author, Date, Hits, and ID. The table lists 15 articles, including 'Example Pages and Menu Links', 'What's New In 1.5?', 'Joomla! Overview', 'Extensions', 'Joomla! Features', 'Content Layouts', 'Joomla! Facts', 'The Joomla! Community', 'Support and Documentation', 'Joomla! License Guidelines', 'Platforms and Open Standards', 'Where did the Installers go?', 'What happened to the locale setting?', 'How do I upgrade to Joomla! 1.5?', and 'Only one edit window! How do I create "Read more..."?'. At the bottom, there are pagination controls showing 'Page 1 of 3' and a legend for article states: 'Published, but is Pending', 'Published and is Current', 'Published, but has Expired', 'Not Published', and 'Archived'.

Obr. 2.6.1: Administrační prostředí redakčního systému Joomla!

2.6.2 REDAKČNÍ SYSTÉM DRUPAL

- systém byl původně napsán jedním ze zakladatelů myšlenky CMS Dires Buytaertem, jako většina systém také používá PHP, databáze jsou zde podporovány MySQL a PostgreSQL, je stavěn modulově, na oficiálních stránkách je možné získat moduly pro rozšíření,
- zajímavostí je, že v roce 2009 byl Drupal přijat úřadem prezidenta spojených států Barackem Obamou jako redakční systém pro správu stránek www.whitehouse.gov.



The screenshot displays the Drupal administration interface. The top navigation bar shows the Drupal logo and the word "Drupal". Below it, a breadcrumb trail reads "Home > Administer > Site building". The left sidebar contains an "admin" menu with options like "Create content", "My account", "Administer", "Content management", "Site building", "Site configuration", "User management", "Logs", "Help", and "Log out". The main content area is titled "Modules" and has tabs for "List" and "Uninstall". It contains introductory text about modules and a table of installed modules.

Enabled	Name	Version	Description
<input type="checkbox"/>	Aggregator	5.7	Aggregates syndicated content (RSS, RDF, and Atom feeds).
<input type="checkbox"/>	Blog	5.7	Enables keeping easily and regularly updated user web pages or blogs.
<input type="checkbox"/>	Blog API	5.7	Allows users to post content using applications that support XML-RPC blog APIs.
<input type="checkbox"/>	Book	5.7	Allows users to collaboratively author a book.
<input checked="" type="checkbox"/>	Color	5.7	Allows the user to change the color scheme of certain themes.
<input checked="" type="checkbox"/>	Comment	5.7	Allows users to comment on and discuss published content. Required by: Forum (disabled), Tracker (disabled)
<input type="checkbox"/>	Contact	5.7	Enables the use of both personal and site-wide contact forms.
<input type="checkbox"/>	Drupal	5.7	Lets you register your site with a central server and improve ranking of Drupal p installed modules and themes
<input type="checkbox"/>	Forum	5.7	Enables threaded discussions about general topics. Depends on: Taxonomy (enabled), Comment (enabled)

Obr. 2.6.2: Administrační prostředí redakčního systém Drupal

2.6.3 REDAKČNÍ SYSTÉM phpRS

- jeden ze zástupců českého redakčního systému, je vyvíjen od roku 2001 Jiřím Lukášem, není to tedy výrobek žádné velké firmy, ale dobrovolného programátora,
- můžete si zde všimnout toho rozdílu, že zahraniční systémy jsou po designové stránce určitě uživatelsky přívětivější.

1. [Správa souvisejících článků](#) - [Správa anket](#)

2. od do třídít dle - Celkem článků: 28
 vyhledat text v
omezit na téma
 Zobrazit pouze mé články

3. Volací link článku

4. Titulek článku

5. Úvod
Sem vložte příslušný text včetně HTML syntaxe nebo v případě nevyužití této části vše vymažte - tedy včetně tohoto textu!

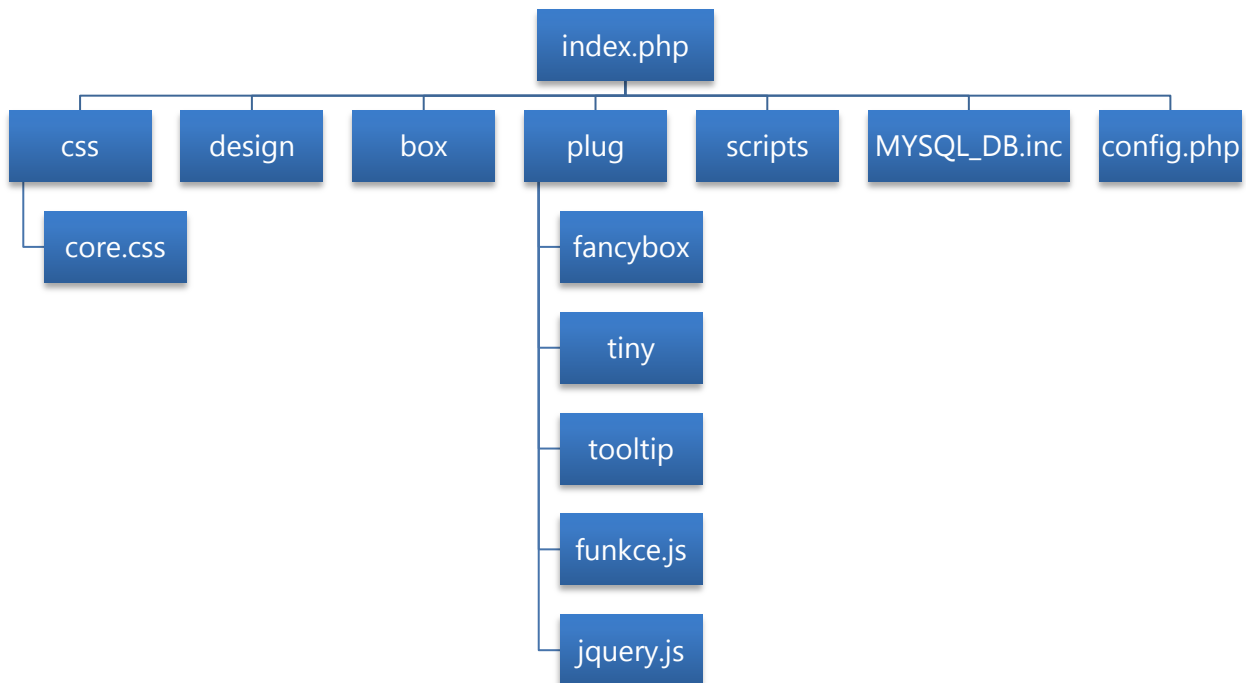
6. Hlavní text
Sem vložte příslušný text včetně HTML syntaxe nebo v případě nevyužití této části vše vymažte - tedy včetně tohoto textu!

7. phpRS značky Ano Ne

8. Téma

Obr. 2.6.3 Administrační prostředí redakčního systému phpRS

3. STRUKTURA REALIZOVANÝCH WEBOVÝCH STRÁNEK



index.php

- je hlavním souborem na webových stránkách,
- je to defaultní soubor, který se otevírá při otevření na stránky, tzn. pokud do prohlížeče zadáte `www.sssep9.cz`, je to to samé, jako kdybyste zadali přímou cestu k tomuto souboru, tj. `www.sssep9.cz/index.php`,
- v tomto souboru se nachází veškerý kód, v hlavičce stránky jsou připojovány potřebné soubory, jako jsou kaskádové styly, javascripty, apod.

css/core.css

- hlavní a jediný načítaný soubor kaskádového stylu pomocí hlavičky, ostatní potřebné jsou připojovány už přímo v souboru

design

- složka obsahující veškerý design, který na stránkách najdete

box

- složka obsahující všechny stránky/soubory, které jsou zobrazovány pomocí fancyboxu, tzn. ty, které jsou načítány ve „vyskakovacích“ oknech

plug

- tato složka obsahuje aplikace, které jsou implementovány do webových stránek,
- složitější, souborově rozmanitější aplikace jsou umístěny v příslušných podsložkách – fancybox, tiny, tooltip, jednotlivé soubory, jako je javascriptová knihovna jquery nebo soubor funkce, jsou umístěny samostatně.

scripts

- nacházející se veškeré vykonávané php scripty, například pro přihlášení, odhlášení uživatele, editaci jednotlivých položek, apod.,
- pokud je nějaká z jmenovaných akcí provedena, prohlížeč otevře tento daný soubor se scriptem, který se provede, na konci scriptu je vždy příkaz k přesměrování na jinou nebo tu samou stránku, odkud byl script spuštěn,
- odkazy na tyto soubory se neuchovávají v historii prohlížečů, neobsahují, deklaraci, hlavičku ani obsah, jde jen o čistý php kód.

MYSQL_DB.inc

- soubor obsahující připojení k databázi,
- při připojování se tento soubor jen zavolá, přečtou se z něho data jako je přihlašovací jméno, heslo a název databáze,
- i když se soubor nachází v kořenovém adresáři webu, tak při zadání cesty k němu do prohlížeče se nezobrazí jeho obsah (přípona *.inc).

config.php

- soubor obsahující nastavení základních proměnných pro celou stránku, je to například proměnná \$tittle, kde se její hodnota vypíše do titulku stránky (u střední školy je to SŠSE, u vyšší odborné školy je to VOŠ, atd.).

4. IMPLEMENTACE JINÝCH APLIKACÍ

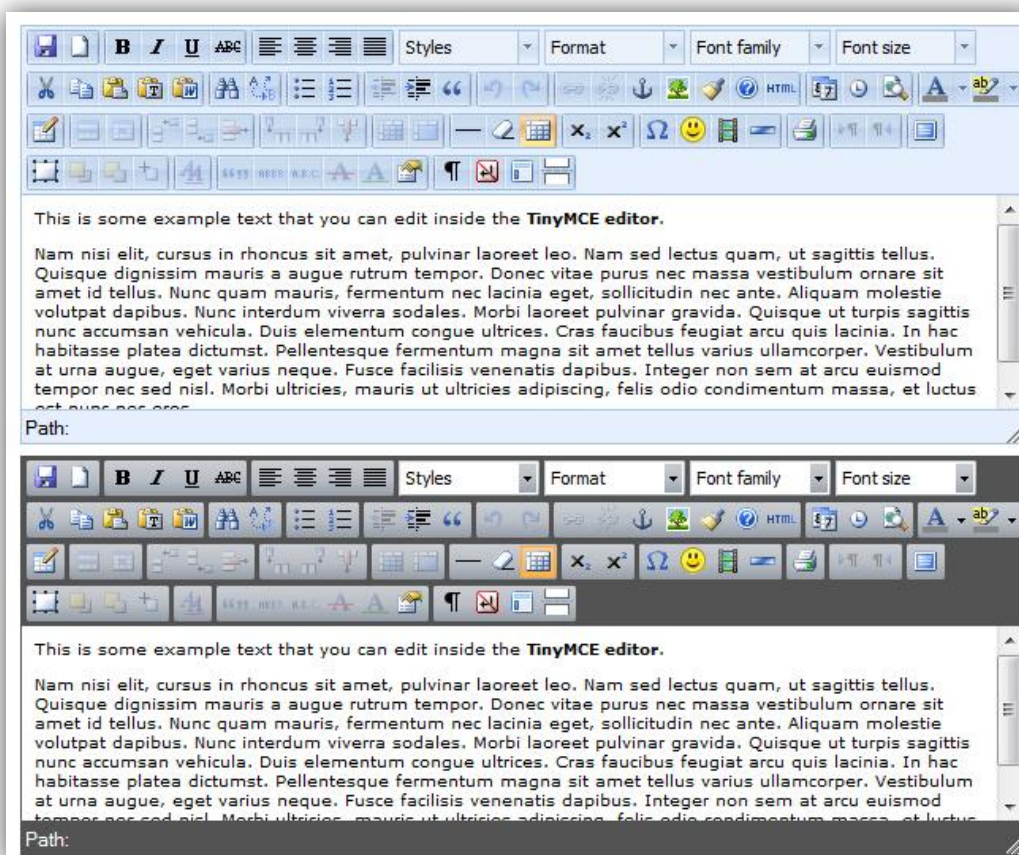
4.1 TEXTOVÝ EDITOR TinyMCE

TinyMCE je jeden z nejpoužívanějších WYSIWYG webových textových editorů. Pro vysvětlení pojmu WYSIWYG. WYSIWYG je akronym anglické věty „What you see is what you get“, v češtině přeloženo jako „co vidíš, to dostaneš“. Je to tedy zkratka pro způsob editace dokumentů, což jednoduše znamená to, že verze zobrazovaná na obrazovce je totožná s výslednou verzí dokumentu nebo verzí tištěnou.

Naprostou klasickými editory v režimu WYSIWYG jsou textové editory Microsoft Word nebo OpenOffice.org Writer.

Námi používaný webový editor je de facto nástavbou pro HTML tag textového pole <textarea>, přidávající potřebná tlačítka, kterými se formátuje daný text. Uživatel tedy nemusí znát HTML tagy (značky) pro formátování textu, jelikož editor dělá tuto práci za něj.

Nevýhodou těchto editorů je poměrně nečistý kód, který aplikace vytváří. Je plný zbytečných elementů, které do HTML vůbec nepatří a výsledný výstupní kód je až 15x větší než kód čistě naformátovaný. Častou editací nebo formátováním zde mohou vznikat chyby, které potom text zdeformují a jediným řešením je kompletní vyčištění formátovacích znaků a znovuzískání čistého textu, který se formátuje znovu. Editor tuto funkci přímo podporuje a čistý text lze zpět získat jediným klikem na tlačítko.



Obr. 4.1 Ukázka editoru Tiny MCE, možností je i přizpůsobit nebo změnit vzhled

4.2 FANCYBOX

Fancybox je javascriptová open source aplikace používající jQuery knihovnu pro zobrazování obrázků, multimédií, textu a vloženého html obsahu. Při zavedení tohoto scriptu do stránek lze zobrazovat obrázky nebo námi určený html obsah, aniž by se zavřel prohlížeč se stávající zobrazovanou webovou stránkou. Lze tím docílit elegantních přechodů při otevírání jednotlivých obrázků, galerií nebo obsahu.

4.3 jQuery

jQuery je javascriptová knihovna kladoucí důraz na interakci mezi html a javascriptem. Byla vydána roku 2006 a je používána a podporována i takovými firmami jako je Google, Microsoft a na svých platformách ji dodává i výrobce Nokia.

Shrňme ve stručnosti trochu laicky, co tato knihovna dělá. Tak jako CSS odděluje zobrazovací charakteristiky od základního html, tak jQuery odděluje ovládací prvky od struktury html. To znamená, že pomocí funkcí napsaných mimo html dokážeme ovládat prvky, které se nám zobrazují. Tímto způsobem jsou na našich webových stránkách řešeny právě takové funkce jako je zobrazování fotek nebo vyskakovacích boxů.

5. ZHODNOCENÍ VLASTNOSTÍ

WEBOVÉ STRÁNKY

Pokud by škola chtěla opravdu profesionální, designově a funkčně dokonalé stránky, musela by se obrátit na firmu, kde by se projekt za desítky tisíc realizoval. Mělo by to samozřejmě své výhody, stránky by mohly být lepší, přívětivější, ale zadání by bylo jednorázové a případně úpravy by byly buď nemožné nebo za určitý příplatek.

V našem případě je výhodou možná komunikace, doladění některých funkcí, tvorba nových funkcí, což dá dohromady snad výsledný dobrý a kvalitní web.

REDAKČNÍ SYSTÉM

Jedinou výhodou, kterou bych chtěl zmínit, je poměrná nechybovost redakčních systémů a tím velká spolehlivost. U vlastních tvořených redakčních systémů se mohou objevovat a objevují se chyby, které se poté musí odstraňovat. V zájmu je samozřejmě, aby byl systém co nejstabilnější a nejspolehlivější, takže je vždy stále co doladovat.

6. ZÁVĚR

Téma zadání mé maturitní práce jsem velmi přivítal. Je to oblast, ve které se velmi rád pohybuji, i když není přímo elektrotechnicky zaměřena. Věřím, že vytvořené webové stránky jsou pro školu prezenčním přínosem a s nasazením nové verze, která má rozšířené funkce a je designově přívětivější, budou plnit tuto úlohu ještě více.

Na samotném produktu, jsem začal pracovat již ve třetím ročníku, kdy škola potřebovala renovovat své webové stránky a já se s doporučením učitelů přihlásil k práci a vytvoření systému dle požadované struktury a požadovaných funkcí.

Z důvodů potřeby renovace byly stránky vypracovány ve velmi krátkém časovém úseku, to mělo dopad na jejich kvalitu a tím občasný výskyt problémů s některými používanými prvky. Na stránkách jsem tedy průběžně pracoval, dělal opravy, úpravy nebo částečně vytváří nové funkce. Systém však nebyl plně navržen tak, aby se dal jednoduše rozšiřovat a doplňovat o nové funkce. Současně s tehdy dostupnou verzí byl vytvářen nový systém, navrhovaný od začátku tak, aby bylo v budoucnu možné jeho rozšíření a lepší správa. Tato nová verze měla být spuštěna již na přelomu roku 2009/2010. Z důvodu časové tísně však nebyl systém zatím dokončen a původní vytvářená verze je dodnes používána.

V mé maturitní práci jsou tedy stránky se současnou funkcionalitou, které byly již vytvořeny, ale od samotného jádra přepracovány, tak i nové funkce, které bylo za úkol do nově vytvářeného systému začlenit.