



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUT OF INFORMATICS

NÁVRH INTERNETOVÝCH STRÁNEK

WEB PAGE DESIGN

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ NOVÁK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. MILOŠ KOCH, CSc.

BRNO 2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Lukáš Novák

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh internetových stránek

v anglickém jazyce:

Web Pages Design

Pokyny pro vypracování:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza problému
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Podle § 60 zákona č. 121/2000 Sb. (autorský zákon) v platném znění, je tato práce "Školním dílem". Využití této práce se řídí právním režimem autorského zákona. Citace povoluje Fakulta podnikatelská Vysokého učení technického v Brně. Podmínkou externího využití této práce je uzavření "Licenční smlouvy" dle autorského zákona.

Seznam odborné literatury:

- CEDERHOLM, Dan. Flexibilní webdesign. 1.vyd. Brno: Computer Press, a.s., 2006. 232 s. ISBN: 80-251-1018-4.
- DOMES, Martin. Tvorba webových stránek. 1.vyd. Brno: Computer Press, a. s., 2006. 192 s. ISBN: 80-251-0920-8.
- GUTMANS, Andi. Mistrovství v PHP 5 . 1.vyd. Brno: Computer Press, a.s., 2005. 520 s. ISBN 80-251-0799-X.
- KRUG, Steve. Webdesign - Nenut'te uživatele přemýšlet. 2.vyd. Brno: Computer Press, a.s., 2006. 168 s. ISBN: 80-7226-892-9.

Vedoucí bakalářské práce: doc. Ing. Miloš Koch, CSc.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/10.





Ing. Jiří Kříž, Ph.D.
Ředitel ústavu



doc. RNDr. Anna Putnová, Ph.D., MBA
Děkanka

V Brně, dne 7. 2. 2010

ANOTACE

Bakalářská práce se zabývá návrhem internetových stránek a možností implementace pokročilých technologií, které jsou spjaté s vývojem webových aplikací. Cílem je vytvořit funkční webovou prezentaci včetně grafického návrhu pro konkrétní firmu.

KLÍČOVÁ SLOVA

WWW, HTTP, HTML, XHTML, CSS, PHP, SEO, MySQL, JavaScript, Apache, databáze, webová prezentace

ANNOTATION

Bachelor thesis deals with design and site deployment options of advanced technologies, which are associated with the development of web applications. The aim is to create a functional web presentation, including graphic design for a specific company.

KEYWORDS

WWW, HTTP, HTML, XHTML, CSS, PHP, SEO, MySQL, JavaScript, Apache, database, web presentation

BIBLIOGRAFIE

NOVÁK, L. Návrh internetových stránek. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2010. 61 s. Vedoucí bakalářské práce doc. Ing. Miloš Koch, CSc.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, a že jsem ve své práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb. O právu autorském a o právech souvisejících s právem autorským).

V Brně dne 20. května 2010

.....
Podpis

PODĚKOVÁNÍ

Děkuji především vedoucímu mé bakalářské práce doc. Ing. Miloši Kochovi, CSc. za metodické vedení, odbornou pomoc a cenné rady a náměty pro vytvoření konečné podoby práce. Dále děkuji svým rodičům za podporu a poskytnuté zázemí. V neposlední řadě patří upřímný dík zaměstnancům společnosti HEIM Trade SE, se kterými jsem měl možnost se setkat, za ochotu, vstřícnost a zvláště jejich čas při podávání tolik důležitých informací z reálného podnikatelského prostředí.

OBSAH

ÚVOD	9
1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ	11
1.1 VYMEZENÍ PROBLÉMU	11
1.2 CÍL PRÁCE	12
2 TEORETICKÁ VÝCHODISKA PRÁCE	13
2.1 INTERNET A SLUŽBA WWW	13
2.2 PREZENTACE OBSAHU A MOŽNOSTI FORMÁTOVÁNÍ	15
2.2.1 ZNAČKOVACÍ JAZYKY HTML, XHTML A XML	15
2.2.2 KASKÁDOVÉ STYLTY CSS	19
2.3 TECHNOLOGIE NA STRANĚ SERVERU	22
2.3.1 PHP	23
2.4 TECHNOLOGIE NA STRANĚ KLIENTA	24
2.4.1 JAVASCRIPT	24
2.4.2 DALŠÍ MOŽNOSTI	25
2.5 DATABÁZE	26
2.6 WEBOVÉ APLIKACE	27
2.6.1 AJAX	27
3 ANALÝZA PROBLÉMU	28
3.1 POSTUP VÝVOJE WEBOVÉ PREZENTACE	28
3.1.1 ANALÝZA A PLÁNOVÁNÍ	28
3.1.2 STRUKTURA A OBSAH	29
3.1.3 GRAFICKÝ NÁVRH	29
3.1.4 BUDOVÁNÍ WEBU	30
3.2 PROPAGACE A OPTIMALIZACE PRO VYHLEDÁVAČE	31
3.3 VÝVOJ WEBOVÝCH STANDARDŮ A PODPORA INTERNETOVÝCH PROHLÍŽEČŮ	33
3.4 VÝVOJOVÉ PROSTŘEDÍ	34
4 VLASTNÍ NÁVRH ŘEŠENÍ	35
4.1 ANALÝZA A PLÁNOVÁNÍ	35
4.2 STRUKTURA A OBSAH	36
4.3 GRAFICKÝ NÁVRH	37
4.3.1 PŘÍPRAVA NÁVRHU	37
4.3.2 TVORBA HTML ŠABLONY A TRANSFORMACE DO CSS	38
4.3.3 PŘÍPRAVA ŠABLONY PRO TISK	43
4.4 BUDOVÁNÍ WEBU	44
4.4.1 NÁVRH DATABÁZE	44
4.4.2 TVORBA ADMINISTRAČNÍHO ROZHRANÍ	45
4.4.3 TVORBA UŽIVATELSKÉHO ROZHRANÍ	49
4.5 OPTIMALIZACE	51
4.6 PREZENTACE WEBOVÉ STRÁNKY NA INTERNETU	54
4.7 EKONOMICKÉ ZHODNOCENÍ	55
ZÁVĚR	56

SEZNAM POUŽITÝCH ZDROJŮ	57
SEZNAM POUŽITÝCH ZKRATEK	60
SEZNAM OBRÁZKŮ	61
SEZNAM TABULEK	61
SEZNAM PŘÍLOH.....	61

Úvod

S možností celosvětového přístupu do elektronického světa se internet stal masovým médiem, které je neuvěřitelně obrovské a zároveň značně levné. Jinak velmi ceněný prostor je zde totiž k dispozici v neomezeném množství. Internet prochází velkým rozvojem, transformuje se jeho uživatelská základna a s ní ruku v ruce i potenciál používání. Vývoj jde stále dopředu a prochází přes prezentaci statických webových stránek, uplatnění skriptovacích jazyků, využití databáze až k internetovým aplikacím, které mají potenciál alespoň z části nahradit ty klasické, desktopové. Některé aspekty, které byly podstatné už při založení celosvětové sítě, však přetrvávají. Jsou to informace.

„Většina přelomových událostí ... je ještě stále před námi.“ Bill Gates

Počítačový obor se velmi rychle mění. Otevírá se nová etapa, pro kterou jsou charakteristické přístroje, které jsou uzpůsobeny přímo a jen pro prohlížení internetových stránek. Další zařízení, jako jsou televize, přehrávače nebo herní konzole, které jsou primárně určeny pro jiný účel, už obsahují možnost připojit se do celosvětové sítě a poskytovat tak uživateli další doplňující služby. Jejich užitek se posouvá na další úroveň a zvyšuje se jejich užitná hodnota. Trend je tedy zpřístupnit internet všude tam, kde to jen jde. Prohlížení webových stránek nebo čtení elektronické pošty tak již není vyhrazeno jen primárně obrazovkám počítačů, ale je možné pro tyto a další činnosti využít display běžného mobilního telefonu nebo televizní obrazovku.

Ovšem nejen pouhá zařízení, která v tomto případě představují jen pomyslnou vstupní bránu, zaznamenávají obrovský boom. I samotný obsah totiž prošel vývojem od statických prezentací, redakčních systémů, přes rozmach blogů až k sociálním sítím. Trendem je umožnit běžným uživatelům spoluvytvářet obsah, jako třeba v diskuzích pod článkem, nebo předat čtenářům do ruky nástroj pro psaní vlastního obsahu. Takovýmto počínům, kdy se v přeneseném významu divák stává hercem, tedy čtenář redaktorem, se říká *Web 2.0*. Tvořit obsah však nutně nemusí znamenat psát článek, je totiž možné také vkládat například vlastní video nebo fotografie.

Ať už je krok k *Web 2.0* obrovskou revolucí nebo jen postupnou evolucí, umožnil, aby se čtenáři velmi jednoduše stali autory bez znalosti jakýchkoli technologií nebo zásad pro prezentaci na internetu. Tento vývoj však může přinést i určitá negativa. Dříve bylo nutné pro publikování na webových stránkách iniciovat obrovskou energii a autor věnoval článku velkou pozornost. Dnes již stačí jen pár kliknutí myši a může se stát, že poskytované informace nebudou tak kvalitní.

Budoucnost lze v oboru IT předpovídat jen velmi těžko. Náznaky, co se může dít, se dají předpovídat na základě informací a možná i potřeb uživatelů či samotných vývojářů a provozovatelů internetových služeb. Po bouřlivém rozvoji ve fázi *Webu 2.0* teď vývoj reflektuje současný stav a je patrná tendence k uzavírání a filtraci obsahu. Velké zahraniční zpravodajské servery a už i některé české začínají například diskusi pod články rušit či nějakým způsobem filtrovat. Dokonce se objevují hlasy o zpoplatnění vybraného obsahu.¹ Podle mého názoru kritický faktor úspěchu takovýchto počínání spočívá hlavně v samotné obsahové kvalitě.

Smyslem následujících kapitol je vymezení primárních problémů a následná sumarizace aktuálních teoretických znalostí, které budou na základě analýzy všech dostupných a dostatečně definovaných informací následně implementovány do reálného prostředí. Naprosto pochopitelná byla pak moje snaha o nalezení prostředí bez zkušeností s předmětnou platformou. Jedině v takovém případě může totiž uplatnění výsledků této práce rozšířit stávající informační procesy a přispět tak ke kvalitativnímu posunu v konkrétním případě reálného podnikatelského subjektu.

¹ Americké deníky *The Wall Street Journal*, *Financial Times* a *The New York Times* již aplikují nebo hodlají zavést systém pro zpoplatnění přístupu na své internetové stránky. [28]

1 Cíle práce, metody a postupy zpracování

1.1 Vymezení problému

Na startovní čáře návrhu jakékoli webové prezentace je nutné určit cíl práce. K dalším nezbytnostem patří také pečlivé plánování. Spojením právě požadovaných cílů s promyšleným plánem vzniká vývojový proces, který lze velmi univerzálně shrnout do několika bodů.

Ze všeho nejdříve je třeba nasbírat potřebné informace. Je nutné zvážit nejen samotný obsah, ale i možnosti použitých technologií. K druhé fázi vývoje se řadí návrh designu a jeho transformace do takové podoby, která je vhodná pro prezentaci na webu. Následné budování pak zahrnuje veškerou programátorskou činnost a zveřejnění práce na internetu. Samotný finiš představuje sledování a vyhodnocování statistik s cílem získat nové návštěvníky a zacílit propagaci.

Počet lidí, kteří využívají internet pro vyhledávání informací, rapidně roste. Investice firmy do webové prezentace je tedy správným krokem, jak oslovit nové zákazníky. Při návrhu je třeba zohlednit obor podnikání firmy, její marketingovou strategii a typy zákazníků. Menším firmám postačuje statický model stránek, který představuje obdobu vizitky v elektronickém prostředí. Pro možnosti prezentace je třeba vybudovat informační architekturu, která rozřazuje podstatné informace do jednotlivých kategorií. Stránky se jistě neobejdou bez názvu společnosti, oboru podnikání, kontaktů a v neposledních řadě referencí zdařilých projektů. Pro větší firmy, nebo ty, které se zabývají jen internetovým podnikáním, jsou příznačné rozsáhlé robustní systémy. K poskytovaným službám se dá přičíst i možnost elektronického podnikání, které představuje alternativní formu tradičních obchodních činností. Papírovou podobu katalogů se zbožím a přímý kontakt prodejce se zákazníkem nahradily webové prezentace a elektronická komunikace. Velký důraz je kladen zejména na bezpečnost operací, neboť většina transakcí se dnes provádí právě elektronickou formou. Poslední dobou se připisuje velká popularita sociálním sítím. Mnoho firem zde proto zakládá své profily, publikuje novinky o produktech a službách. Jejich cíl je jasný. Využít další komunikační kanál, jehož prostřednictvím se lze dostat k potencionálním zákazníkům.

1.2 Cíl práce

Cílem mé bakalářské práce je poskytnout komplex základních informací orientovaných na problematiku zavádění a používání technologií v prostředí vývoje internetových stránek. Primární cíl je ukázat v praxi vývoj webové prezentace. Sekundární cíl je teoreticky popsat technologie a vývojový proces včetně nových trendů v oboru. Práce je rozdělena do čtyř hlavních kapitol.

V první kapitole se zaměřuji na vymezení problému a stanovení cíle práce.

Druhá kapitola se zabývá popisem používaných technologií. Jejich rozřazení je dáno dle typu použití. Je velmi časté, že vede více cest ke stejnému cíli. Proto je v práci rozepsán průřez přes různé používané technologie.

Ve třetí kapitole je detailně rozebrán postup při návrhu internetových stránek včetně faktorů neúspěchu. Dále jsou v této části práce popsány aspekty, které do této problematiky nesporně patří. Jedná se o vývoj webových standardů, podporu internetových prohlížečů, optimalizaci pro prohlížeče a vyhledávače, propagaci a vývojová prostředí.

Získané vědomosti z obou hlavních kapitol pak aplikuji ve čtvrté kapitole na příkladu v praxi. Při vývoji použiji vývojový proces společně se znalostí popsaných technologií. Souhrn teoretických znalostí implementuji do konkrétního prostředí evropské akciové společnosti HEIM Trade SE se sídlem v Brně na Vídeňské ulici. Společnost spravuje v několika lokalitách na území České republiky areály, které po revitalizaci pronajímá dalším podnikatelským subjektům jako výrobní, skladovací, resp. administrativní prostory. Ucelená a přehledná nabídka, tříděná podle lokalit a typů budov, bude předmětem internetové prezentace, která se stane pro zmíněnou společnost soudobou výkladní skříní a novou prezentační aktivitou.

2 Teoretická východiska práce

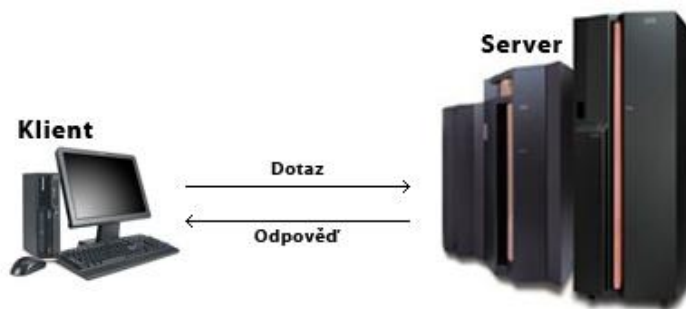
2.1 Internet a služba WWW

Internet označuje globální decentralizovanou počítačovou síť. Prapůvod tak obrovského média tvořilo spojení jen několika páteřních bodů americké armády v 50. a 60. letech minulého století. Vývoj dále pokračoval a zájem několika amerických univerzit předznamenal velký boom v oblasti informačních technologií. Významný rozmach osobních počítačů, uvedení hypertextu pro popis informací a služby WWW a také možnost připojení běžných uživatelů do celosvětové sítě znamenal pro odvětví velmi podstatný zlom. Další etapu ilustruje příchod komerčních subjektů a využití internetu nejen výhradně pro vojenské či akademické účely. [6]

Služba **WWW**, která označuje celosvětové propojení dokumentů, se sestává ze tří základních pilířů, které tvoří protokol HTTP, technologie URL a jazyk HTML. Od roku 1990, kdy tato služba spatřila světlo světa na půdě organizace pro jaderný výzkum CERN, se na tyto základy nabalují ještě další technologie a nové postupy. Další správa a vývoj standardů pak probíhá pod hlavičkou World Wide Web Consortium, zkráceně W3C. [12]

První z hlavních pilířů tvoří protokol **HTTP**. Jeho architektura je založena na komunikaci mezi klientem a serverem, kde jsou dotaz i odpověď vázány pevnými pravidly. Průběh je následující. Uživatel na klientském počítači zadá do adresního řádku webového prohlížeče jedinečnou adresu, případně je adresa převzata z odkazu. Právě jednoznačné adresy pro identifikaci objektu zajišťuje implementace technologie **URL**. Klient následně přeloží zadaný text pomocí DNS serverů na adresu IP a ihned vyšle požadavek. Ten obsahuje záznam o verzi protokolu, URL adresu a metodu, z nichž nejpoužívanější jsou GET a POST. Server požadavek zpracuje a expeduje klientovi odpověď. Kromě samotného obsahu zprávy zahrnuje odpověď také verzi protokolu, stavový kód a hlavičku. Stavový kód je popsán trojmístným číslem a používá se pro zjednodušení komunikace. K určení obsahu přenášeného dokumentu používá HTTP protokol standardu MIME. Proto lze přenášet jak text, který dokážou ze zdrojového souboru zobrazit webové prohlížeče, tak i binární soubory, jako například obrázky. [12]

Limitem protokolu je bezstavová komunikace, která omezuje relaci mezi klientem a serverem jen na jedno spojení. Toto chování je velmi nešikovné, neboť zabraňuje vzniku delší vazby. Lze ho obejít pomocí technologie Cookies. Jedná se o soubory s informacemi, které jsou uloženy na klientovi, a server je dokáže použít pro další zpracování¹. Další omezení vyplývá již z navržené architektury. Server totiž reaguje jen na výzvu klienta, kterému pak odesílá odpověď. Za předpokladu, že je potřeba navrhnout internetovou aplikaci, která vyžaduje automatické aktualizace obsahu bez nutnosti zadat výzvu manuálně, je nutné sáhnout po sofistikovaných nástrojích, které daný problém dokážou vyřešit. Jednou z možností je využití skriptu na straně klienta ve spojení se serverem, který odpověď poskytuje v XML formátu. Obecně tyto techniky přispívají k iluzi desktopové aplikace a umožňují tak přesun k webovým aplikacím.



Obrázek 1 – Protokol HTTP [6]

Zajistit bezpečnost přenášených dat a zamezit možnostem odposlechu sítě je úlohou vrstvy **SSL**. Jedná se o nadstavbu již popsaného HTTP protokolu, který používá zkratku **HTTPS**. Důvodem jeho zavedení je zabezpečení komunikace mezi klientem a serverem. Pomocí šifrování dat zabezpečuje tento uznávaný standard například internetové aplikace elektronického bankovníctví před únikem citlivých údajů a tím minimalizuje hrozbu zneužití celého systému.

Třetím pilířem služby WWW je technologie **HTML**, kterou se podrobně zabývám v následující kapitole.

¹ Ve skutečnosti nenabízí architektura Cookies z dnešního pohledu dostatečnou kapacitu pro ukládání objemných dat. Implementace novějších postupů pro budování webových aplikací je však na blízku.

2.2 Presentace obsahu a možnosti formátování

Základní pravidlem tvorby webové prezentace je oddělení formátování od samotného dokumentu s obsahem. Mezi jednoznačné výhody tohoto uspořádání patří jednodušší správa celé prezentace a také snížení přenášovaných dat. Jazyk, který se primárně stará o logické členění, se nazývá HTML. Všechny konstrukce, které netvoří samotný obsah a popis struktury, ale naopak formátování dokumentu, jsou vykázány mimo do tabulek kaskádových stylů CSS. Spojení obou technologií je dnes pro tvorbu webových prezentací naprosto zásadní.

2.2.1 Značkovací jazyky HTML, XHTML a XML

Zkratka **HTML** označuje značkovací jazyk HyperText Markup Language, který vznikl na základech jazyka SGML. Jazyk používá pro popis struktury dokumentu speciální značky, nazývané tagy. Těmi lze jednoduše určit nadpis, odstavec, odkaz na jiný dokument, tabulku, obrázek a v neposlední řadě strukturovaný seznam. K samotným značkám ve specifikaci ale přibývaly značky další, které podporovaly pouze některé internetové prohlížeče. Tyto změny vznikaly na základě potřeby popsat také grafickou podobu stránky a přinášely na obrazovky počítačů uměleckou hodnotu. Samotná specifikace jazyka však na něco takového nebyla připravena a proto si museli vývojáři pomoci s rozmištěním prvků na stránce hlavně díky vnořeným tabulkám a rámcům. Právě s rozdílem mezi standardy a jejich výkladem a implementací prohlížečů se snaží vypořádat sdružení W3C. Jeho práce nespočívá jen ve vývoji standardů, ale také ve spojení firem zabývajících se problematikou internetových prohlížečů. Toto spojení pak umožňuje snazší prosazení do praxe. [15]

Pro oddělení struktury a grafického popisu bylo nutné odstranit ze specifikace některé značky. Vývoj HTML byl proto zastaven a končí verzí 4.01. Následně začíná éra **XHTML**. Tato nová specifikace upravuje některá pravidla používání, například zápis tagů¹. Dále byly ze specifikace vynechány některé atributy a značky², a to zpravidla ty, které popisovaly jen styl. Tímto krokem se jazyk vrací ke svým kořenům, tedy k popisu

¹ Nepárové tagy končí lomítkem. Všechny atributy musejí mít hodnotu. Všechny tagy malým písmem.

² Mezi zakázané značky patří například: <center>, , <s> a <u>

struktury dokumentů. Zároveň se počítá s doplňkem – kaskádovými styly CSS. Podrobným popisem této technologie se zabývám v následující kapitole 2.2.2.

Další z technologií, která se používá pro popis informací, se nazývá **XML**. Ve srovnání s HTML je univerzálnější a flexibilnější. Nenabízí žádné předem definované značky s předdefinovanými styly, ale dává nástroj popsat data co možná nejkonkrétněji tvorbou vlastních značek. Použití pro tvorbu webové stránky je tak ještě více spjato s kompletním popisem stylu. Proto se XML spíše implementuje tehdy, když je třeba data popsat co nejuniverzálněji, například ve výměně dat. Na základě této technologie fungují služby jako RSS nebo AJAX.

Další vývoj bude velmi zajímavý. Podle dostupných informací se nebude ubírat cestou rozvoje plánované druhé verze XHTML, ale do popředí se dostává již pátá verze HTML. Samotné sdružení W3C spolupracuje s klíčovými hráči¹ na poli internetových prohlížečů, a právě toto spojení je podle mého názoru při rozvoji standardů rozhodující. Standard reaguje na rychle se rozvíjející obsahové požadavky webových stránek. Řeší to, co museli programátoři doposud implementovat velice složitě. **HTML5** přináší řadu novinek, ke kterým patří například uvedení nových elementů², které usnadňují orientaci v dokumentu, nativní podporu offline webových aplikací a v neposlední řadě implementaci elementů audio a video pro reprodukci multimediálního obsahu bez nutnosti instalace dalšího softwaru nebo rozšíření.

Skupina výše popsaných jazyků má pevnou **syntaxi**. Pro tvorbu webových dokumentů se používají značky - tagy, které jsou uzavřeny do speciálních závorek, a na rozdíl od jazyka XML jsou pevně definovány. Tagy mohou být párové a nepárové. Párové tagy uzavírají mezi počáteční a koncovou značkou obsah, který je uzavřeným tagem ovlivněn. Koncové značce předchází kromě speciální závorky ještě lomítko. Samotnou činnost značek doplňují atributy specifické pro každý tag. Obecný zápis je ukázán v tabulce 1.

¹ Microsoft (Internet Explorer), Mozilla Foundation (Firefox), Apple (Safari) a Opera [23]

² <header>, <nav>, <article>, <section>, <aside>, <footer>, <video> a <audio>

Zápis elementů	Popis
<tag>Ovlivněný obsah</tag>	Obecný příklad párového tagu.
<tag atribut="hodnota atributu" />	Obecný příklad nepárového tagu.
Nadpis	Příklad párového tagu – nadpisu.
	Příklad nepárového tagu – vložení obrázku.

Tabulka 1 – Párové a nepárové tagy

Každý webový dokument musí obsahovat základní strukturu po sobě jdoucích značek, tak jak je vyobrazena v tabulce 2. Prvním uvedeným tagem musí být *DOCTYPE* s informacemi o typu a verzi specifikace dokumentu. Následuje značka *<html>*, kterou začíná celý dokument. Poté přichází na řadu definice hlavičky *<head>*. Informace v ní uvedené mohou definovat titulek dokumentu, odkazovat na externí stylotypy a skripty a vymezit meta informace. Pro obsah dokumentu je vymezena značka *<body>*, která slouží jako takový kontejner. Vše, co je zde uvedeno, bude následně vidět v okně webového prohlížeče.

HTML	Popis
<!DOCTYPE html PUBLIC "...">	Definice DTD (hodnota atributu zkrácena).
<html>	Začátek HTML dokumentu.
<head>	Začátek hlavičky.
<title>Internet</title>	Titulek s textem „Internet“
</head>	Konec hlavičky
<body>	Tělo dokumentu.
	Sem se vkládá samotný obsah dokumentu.
</body>	Konec těla dokumentu.
</html>	Konec HTML dokumentu.

Tabulka 2 – Kostra jazyka XHTML

Z uvedené základní struktury je patrné, že kromě tagu *DOCTYPE* jsou ostatní párové a musí být uzavřeny. Je nutné dodat, že dnešní prohlížeče jsou k případným chybám v kódu benevolentní a alespoň některé dokážou opravit. Na druhou stranu může být výsledek zobrazení nepředvídatelný a je nevhodné spoléhat, že chybu za nás opraví jádro webového prohlížeče.

Tagy používané ve webových dokumentech lze třídit na následující skupiny: strukturní, stylové, logické, značky pro seznamy, tabulky, rámce a formuláře. [19]

Zvláštní syntaxi tvoří poznámky, které většinou obsahují doplňující záznamy pro programátora, a ve výsledku se nezobrazují. Mezi další důležité prvky patří značky pro zobrazení formuláře. Ty samy o sobě neumožňují další zpracování a poskytují jen aparát pro serverové nebo na klientovi zpracovávané skripty. K formulářovým prvkům se řadí tlačítka, vstupní textová pole, výběrové a zaškrťovací boxy. Právě implementace formulářů nabízí uživatelům publikovat zpětnou vazbu.

Značka	Popis
<a>	Odkaz
<h1> až <h6>	Nadpisy
<p>	Odstavec
	Obrázek
<table>	Tabulka
<tr>	Řádek tabulky
<td>	Buňka tabulky
	Seznam
	Položka seznamu
<div>	Obecný blokový tag
	Obecný řádkový tag
 	Řádkový zlom
	Obrázek
<form>	Formulář
<input>	Vstupní pole
<textarea>	Textové vstupní pole
<!-- text poznámky -->	Poznámka

Tabulka 3 – Základní tagy jazyka XHTML

Dokument s náležitou strukturou a správnou syntaxí je možné vytvořit v jednoduchém textovém editoru. Webové prohlížeče jej pak identifikují podle přípony *.html* a upraví do čitelné podoby.

2.2.2 Kaskádové styly CSS

Pro objasnění vzniku kaskádových stylů je třeba podívat se do historie, kdy se tato technologie ještě nepoužívala. Primárním účelem jazyka HTML bylo a doposud zůstává popsat strukturu a vazby mezi dokumenty. Pro formátování vzhledu však bylo nutné používat značky, které s popisem struktury neměly nic společného. Rozložení stránky se řešilo pomocí rámců a vnořených tabulek. Právě nedodržování struktury dokumentu vedlo kromě ztížené čitelnosti samotného kódu také k problematické indexaci obsahu. Navíc, a to na rozdíl od již dnes nepoužívaných metod, je při zavedení kaskádových stylů jakákoli změna designu velmi snadná, neboť soubor s definicí stylu se nachází na jednom místě a autor nemusí upravovat každý jednotlivý dokument. Na vzniklou situaci reagovalo konsorcium W3C, které zaštiťuje vývoj webových standardů. Od druhé poloviny devadesátých let začalo vyvíjet jazyk CSS (*Cascading Style Sheet* neboli tabulky kaskádových stylů), který má na starosti popisovat vzhled dokumentu a slouží pouze jako doplněk již dříve uveřejněného jazyka HTML. Nové verze vyvíjených standardů přestávají obsahovat nepodporované značky a postupně tak dochází k užšímu propojení obou technologií. Aktuálně se používá druhá verze specifikace, třetí je stále ve vývoji. [15]

„CSS slouží k popisu prezentace dokumentů, aniž by jakkoli ovlivňovali jeho obsah či strukturu.“ Převzato z [19, s. 1]

Kaskádové styly umožňují vývojářům webu popisovat strukturu dokumentu pomocí jazyka (X)HTML a nabízejí možnost formátovat webové stránky efektivně, univerzálně a kreativně. Představují metodu, jak převést jednoduché i pokročilé grafické návrhy na podobu, kterou dokážou zpracovat internetové prohlížeče.

Samotný vývoj stránky lze rozdělit na dvě části. Nejdříve se popíše struktura dokumentu vhodnými tagy, které představují například nadpisy a odstavce. Následně se vytváří zápis stylu, který například popisuje velikost a barvu textu těchto nadpisů a odstavců. Do HTML dokumentu se pak ještě přidávají identifikátory, selektory třídy a obecné tagy `<div>` a `<span>`, které jsou v některých případech nutné pro přesnou specifikaci obsahu.

Existují základní tři možnosti pro **vkládání stylu** do HTML dokumentu. Tu první představuje vložení zápisu o stylu do atributu *style* podporovaného tagu. Druhou možností je definice pomocí tagu *<style>* v hlavičce HTML dokumentu. Třetím, a v zásadě nejuniverzálnějším řešením, je vyjmutí definice stylů z dokumentu úplně. V hlavičce se pak uvede pomocí tagu *<link>* odkaz na externí soubor s definicí stylopisu. [15]

Vložené zápisy o stylu obsahují jednotlivá **pravidla**. Každé pravidlo se skládá ze selektoru, složených závorek, seznamu vlastností a jim přiřazeným hodnot. [19] Příklad definice kaskádových stylů je možné vidět v tabulce 4.

CSS
<pre>body { background-color:black; } h1 { color:white; padding-left: 5px; } p.odstavec { margin: 5px; } #id { float: left;}</pre>

Tabulka 4 – Příklad definice kaskádových stylů

V následujících odstavcích se pokusím shrnout alespoň základní pravidla, kterými se technologie kaskádových stylů řídí.

Selektory popisují, pro které prvky mají zápisy o stylu platit. Selektorem může být název prvku (převzatý z HTML dokumentu, např. *h1*), třída (*class*) nebo identifikátor (*id*). Uvedené selektory lze kombinovat a vytvářet tak složitější zápisy, které mohou rozšiřovat, ale i naopak zpřesňovat jejich pole působnosti. [19]

Mezi další pravidla se řadí **dědičnost**. Princip popisuje metodu dědění vlastností z rodiče na potomka, která vychází ze struktury (X)HTML dokumentu. Vlastnosti tak mohou být nastaveny jen některým prvkům a další je mohou jednoduše zdědit. Pokud některý prvek nemá definované vlastnosti, jsou mu přiřazeny vlastnosti výchozí¹.

*„Platí pravidlo bližší košile než kabát – pokud se hodnota dědí,
dědí se vždy z nejbližšího nadřazeného prvku.“* Převzato z [19, s. 16]

¹ Výchozí hodnoty stylu pro každý standardem definovaný prvek zabezpečuje internetový prohlížeč.

Stavebním kamenem celé technologie CSS je **kaskáda**, která dala tomuto jazyku svůj název. Pravidlo definuje, jak se musí prohlížeč řídit při zpracování stylů. Různé vlastnosti mohou totiž odpovídat více prvkům, proto platí následující výrok:

*„Přednost mají přesnější, konkrétnější pravidla
před pravidly obecnějšími.“* Převzato z [19, s. 17]

Vlastnosti definované tímto jazykem lze využít od formátování textu až po rozložení celé stránky. Stručný seznam vlastností pro definici stylu je uveden v tabulce 5.

Vlastnost	Popis
Vlastnosti písma	
font-family	Rodina písma
font-size	Velikost písma
Vlastnosti pro nastavení barev	
color	Barva textu
background-color	Barva pozadí
background-image	Obrázek na pozadí
Vlastnosti textu	
text-align	Vertikální zarovnání
text-decoration	Dekorace textu
Vlastnosti boxů	
margin	Velikost vnějších okrajů
padding	Velikost vnitřních okrajů
border	Vlastnosti rámečku.
width	Šířka
height	Výška
float	Nastavení plovoucího objektu
Řízení pozice	
position	Pozicování elementu
left	Posunutí na horizontální ose x
top	Posunutí na vertikální ose y
Poznámka	
/* text poznámky */	Poznámka

Tabulka 5 – Přehled základních vlastností CSS [15]

Další vývoj kaskádových stylů směřuje k uvedení finální podoby specifikace s pořadovým číslem tří. K jeho hlavním výhodám patří modularizace obsahu specifikace do více dokumentů. Toto rozdělení pak umožní snadněji definovat, co aplikace pracující se styly dokáže. Změn je podle rozpracované dokumentace v **CSS3** přichystáno opravdu hodně. Po jejich shrnutí vyplývá, že jejich zavedení usnadní práci a rozšíří obzor webovým vývojářům. [21]

2.3 Technologie na straně serveru

Webový server jsem dosud definoval jako místo, kde jsou uloženy dokumenty, které jsou na požádání odeslány klientovi. Technologie na straně serveru ale dokážou mnohem více, než tuto triviální záležitost. K jejich možnostem se řadí práce s daty a databázemi, interakce na výzvu klienta a mnohé další. Odpověď serveru je ovšem vždy sestavena do takové podoby, aby jí webové prohlížeče rozuměly. Ve výsledku tak webový prohlížeč obdrží klasikou webovou stránku. Veškeré zpracování probíhá na serveru a není závislé na platformě klienta. Právě tato polarizace přináší pro serverové technologie řadu výhod, ke kterým patří například snadná interakce s dalšími serverovými technologiemi a databázemi nebo nezávislost na softwaru a hardwaru klienta. Popsaný model detailně ilustruje obrázek 2. [20]

„Technologie na straně serveru jsou ty, které umožňují mít na serveru logiku pro vytváření webových stránek za běhu“ Převzato z [4, s. 15]

Databáze a serverové technologie umožňují vznik dynamických internetových stránek a jejich spojení představuje jeden z možných modelů webové aplikace. Uvedená grafika dává do souvislosti použití zmíněných komponent a webového klienta společně s výčtem používaných prostředků.



Obrázek 2 – Model webové aplikace [20]

Mezi technologie, které umožňují zpracovávat logiku na straně serveru, patří PHP, které je detailněji popsáno v následující kapitole. PHP má ovšem mnoho konkurentů, jako například ASP.NET, ColdFusion, Perl, Java Server Page nebo Ruby on Rails. I když se technologie mohou velmi lišit, jejich účel a výstup je obvykle totožný.

2.3.1 PHP

PHP je mohutný nástroj umožňující pokročilou tvorbu internetových stránek, pro který se používá rekurzivní zkratka *PHP: Hypertext Preprocesor*. Mezi jeho důležité vlastnosti se řadí mimo jiné vyspělá práce s mnoha databázemi a detailní zpracování dat získaných z formulářů, které jsou pro získání informací od uživatelů velmi důležité. Příklad práce s databází MySQL je uveden tabulce 6.

PHP je zároveň serverovou technologií i skriptovacím jazykem, který je úzce spjat s webovým serverem Apache a operačním systémem Linux¹. Provozovat ji však lze i na jiných operačních systémech i pod jiným webovým serverem, což konkurenti PHP většinou neumožňují. Také díky velké oblibě existuje pro PHP mnoho doplňkových rozšíření. Mezi takové patří například sofistikovaný systém *Smarty*, který nabízí možnost odděleného vývoje PHP a HTML šablony. Nejnovější pátá verze přináší výrazně přepracovanou podporu objektově orientovaného programování a pokročilé zpracování výjimek. Vývoj technologie stále pokračuje a plánovaná šestá verze má přinést lepší podporu *UNICODE* a jmenných prostor (*namespaces*). [17]

Zdrojový kód PHP je zapisován s HTML značkami do jednoho souboru s příponou *.php*. Odlišení v takto kombinovaném dokumentu zajišťuje použití značek `<?php` a `?>`, které reprezentují začátek a konec skriptu.

PHP
<pre><?php mysql_connect(\$server, \$user, \$passwd); mysql_select_db(\$database); \$query = 'SELECT nadpis, clanek FROM articles ORDER BY time;'; \$result = mysql_query(\$query); while (\$row = MySQL_Fetch_Row(\$result)) { echo "<h2>{\$row[0]}</h2>"; echo "<p>{\$row[1]}</p>"; } ?></pre>

Tabulka 6 – Ukázka použití jazyka PHP: práce s databází

¹ Spojení popsanych technologií vystihuje zkratka LAMP (Linux, Apache, MySQL a PHP).

2.4 Technologie na straně klienta

Pro větší funkčnost na straně klienta je třeba implementovat další technologie, které se od serverových velmi liší. Odpověď webového serveru totiž obsahuje celý zdrojový kód, který se spouští až na klientovi. Mezi výhody patří kromě menší zátěže serveru zejména interakce přímo ve webovém prohlížeči.

„Když se HTML zkombinuje se serverovými a klientskými technologiemi, je možné vytvářet velmi výkonná webová řešení“ Převzato z [4, s. 17]

2.4.1 JavaScript

JavaScript je objektově orientovaný programovací jazyk, který podporuje většina dnešních prohlížečů. Přesto je však možné zpracovávání skriptů vypnout, a tak se nelze vždy spolehnout na úplnou podporu. Nejedná se o jazyk kompilovaný a na klienta se přenáší v čisté podobě. Zdrojový kód je tak snadno přístupný a tak není příliš bezpečný. Pro přidání funkcionality do internetových stránek je však dostatečný.

Pomocí JavaScriptu je možné pracovat přímo s webovou stránkou a objekty na ní. Nabídka je velmi široká a zahrnuje například prostředek pro změny kaskádových stylů CSS nebo možnost práce s obsahem stránky. Díky implementaci událostí lze sledovat například kliknutí myši nebo zmáčknutí tlačítka klávesnice a na jejich základě pak provádět přidružené akce. Proto lze například snadno kontrolovat data v textových polích formulářů ještě před odesláním a zbytečně tak nevysílat požadavek na server.

Pro vkládání do dokumentu se používá tag `<script>`, případně je možné uvést odkaz na externí soubor se zdrojovým kódem. Ukázku použití jazyka uvádím v tabulce 7.

JavaScript
<pre><input type="button" onClick="halloWorld();" /> <script type="text/javascript"> function halloWorld() { alert("Ahoj světe!"); } </script></pre>

Tabulka 7 – Ukázka použití jazyka JavaScript: událost onClick

2.4.2 Další možnosti

Dosud jsem se v celé kapitole teoretických východisek zabýval mnoha prostředky, které se užívají při tvorbě internetových stránek. Zmíněné technologie jsou předurčeny k implementaci dynamického webu s použitím databáze, serverových technologií, prezentací na klientovi pomocí HTML a CSS a s případným použitím klientských skriptovacích jazyků. Existují ale i další možnosti prezentace na internetu. Mezi takové patří technologie **Adobe Flash** a **Microsoft Silverlight**.

Obě technologie dokážou prezentovat interaktivní obsah s podporou vektorové grafiky, médií a animací, které předčí implementaci HTML a CSS. A i přes fakt, že Flash i Silverlight nabízejí prostředky pro zpracování kompletní webové prezentace, včetně spolupráce s databázemi, používají se spíše jako doplněk. Ve velké hojnosti lze jejich výstupy nalézt jako reklamní spoty, jednoduché hry nebo přehrávače audio a video formátů. Nová specifikace jazyka HTML s pořadovým číslem pět konsorcia W3C nabízí nativní podporu audio a video přehrávače, proto bude v budoucnu možné vyhnout se použití tohoto druhu technologií alespoň v omezené míře.

„Za pár let ho (Adobe Flash, pozn. a.) nikdo používat nebude, svět se přesune na HTML5.“ Steve Jobs o novém produktu Apple iPad v únoru 2010

Jak už bylo v úvodu této práce zmíněno, prohlížení webových stránek není dnes již doménou pouze klasických počítačů, ale stejně tak lze stránky zpřístupnit na mobilních telefonech a dalších zařízeních. Právě některá zařízení a operační systémy mají s implementací technologií Flash a Silverlight nemalý problém. Navíc musí být na každém klientském počítači nainstalovaný doplněk – plug-in, který dodaný obsah dokáže spustit. Právě proto není vhodné takto prezentovat například obsah nebo navigaci webu. Může se pak totiž stát, že tyto prvky budou nedostupné.

2.5 Databáze

Základ dnes budovaných webových aplikací tvoří objemná data, která je nutné uchovávat a zpracovávat. Nejvýhodnějším řešením je nasadit na tuto práci databázi, které byly k tomuto účelu navrženy. Jejich spojení se serverovými technologiemi hraje klíčovou roli při tvorbě webových aplikací, ke kterým patří například informační systémy nebo elektronické obchody, kde právě data uložená v databázi představují základní kámen celku. Začlenění databáze je možné vidět na obrázku 2 na straně 22, ze kterého je patrné, že serverové technologie umožňují pokládat databázovému serveru dotazy a posléze umožňují přijatá data zpracovat a poslat klientovi.

Databázi je třeba vytvořit na základě získaných požadavků. Při návrhu se postupuje dle principů datového modelování, které zahrnuje tvorbu relací a vztahů mezi nimi. Relace představují tabulky, kde jsou uchovávána data, a vztahy mezi nimi jsou definovány pomocí primárních a cizích klíčů. Dále je nutné zvolit atributy, neboli názvy sloupců relace a jejich datové typy. K vytvoření relačního modelu jsou vývojářům k dispozici normální formy, které vymezují postup, jak vybudovat bezchybný návrh. [12]

Základní přístup k uloženým datům představuje jazyk **SQL**, neboli *Structured Query Language*. Umožňuje vytvářet, modifikovat a mazat relace a záznamy v relacích a také tvořit pokročilé dotazy pro prohlížení záznamů. Příklad použití je vidět v tabulce 8. Kromě základní manipulace s daty lze na databázovém serveru spouštět skripty a vytvářet triggerly a pohledy. To umožňuje převést logiku zpracovávanou technologií serveru přímo ke zdroji dat.

SQL
INSERT INTO users (username) VALUES 'lukas';
SELECT username FROM users;
DELETE FROM users WHERE username='lukas';

Tabulka 8 – Ukázka použití jazyka SQL: práce s databází

Mezi rozšířené databázové servery patří mimo jiné volně šiřitelné MySQL a PostgreSQL a komerční Oracle a Microsoft SQL Server. Výběr databázového stroje ovšem nezáleží jen na požadavcích navrhovaného systému, ale také na použití dalších technologií i operačního systému. Některé totiž nejsou vzájemně kompatibilní.

2.6 Webové aplikace

Zmínky o webových aplikacích padly již v různých částech této práce. Je tedy patrné, že tento termín je dnes používán velmi frekventovaně.

„Webové aplikace vzbuzují dojem, že jednoho dne budou mít stejný vzhled a chování jako jejich vyspělí (výkonni) příbuzní ze světa desktopu.“ Převzato z [4, s. 12]

Jako webovou aplikaci lze klasifikovat spojení serverových technologií a databáze, které má ovšem z pohledu uživatele omezenou funkčnost. Onen nedostatek plyne již z architektury HTTP protokolu, jedním ze základních kamenů internetu. Pro získání nových informací totiž musí uživatel webovou stránku vždy obnovit. A právě AJAX, na jehož základě stojí skriptovací jazyk JavaScript a značkovací jazyk XML, tuto nepříjemnou vlastnost dokáže překonat.

2.6.1 AJAX

Zkratka AJAX vyjadřuje spojení *Asynchronous JavaScript and XML*. Spíše než novou technologii či jazyk ji lze klasifikovat jako postup pro použití vybraných serverových a na klientovi zpracovávaných technologií. Mezi takové patří značkovací jazyky (X)HTML a XML, tabulky kaskádových stylů CSS, skriptovací jazyk JavaScript a použití technologie na straně serveru. K dalším se dále řadí objektově orientovaná prezentace dokumentu DOM, neboli *Document Object Model*, která zajišťuje manipulaci s načtenou HTML stránkou, a objekt *XMLHttpRequest*, který umožňuje provést komunikaci mezi klientem a serverem. [4]

Z výše uvedeného je zřetelně vidět, že technologie používané při vývoji internetových stránek jsou velmi spjaté. I přes použití mnoha jazyků a technologií je výstup velmi snadno popsitelný. AJAX dokáže provádět pro uživatele neviditelná volání na server a následně přijatou odpověď zpracovat a zobrazit. Jeho použití vytváří příjemný komfort a téměř dokonalou iluzi práce s klasickým softwarem. Na webové stránce lze efektivně upravovat dokumenty, prohlížet mapy celého světa nebo pracovat s elektronickou poštou. Vše lze zvládnout bez toho, aby se stránka znovu a znovu načítala.

3 Analýza problému

3.1 Postup vývoje webové prezentace

Postup vývoje webové prezentace lze rozřadit do několika klíčových činností. Na začátku každého projektu je třeba provést analýzu problému, sestavit plán a případně definovat rizika, která mohou nastat. Poté následuje návrh informační struktury, která zahrnuje získání a rozřídění obsahu do jednotlivých kategorií. Po dokončení této části plánu lze přejít ke grafickému návrhu internetové stránky, která ve všech případech zohledňuje obsah i strukturu webové prezentace. Dále je nezbytně nutné provést transformaci grafického návrhu do takové podoby, aby byly internetové prohlížeče schopny webovou stránku zobrazit. Po provedení všech předchozích činností přichází na řadu samotné budování stránek. Do této části patří testování, implementace a použití všech technologií tak, aby výsledek splňoval zadaný cíl práce. V některých případech vývoje internetových stránek je možné již na základě druhého bodu vývojového procesu, tedy sestavení informační struktury, začít projektovat datový model a na něj navazující činnosti a tak není třeba čekat na dokončení grafického návrhu. Obě aktivity mohou probíhat souběžně a posléze je třeba výsledky obou prací spojit a případně přizpůsobit. Už od samého začátku plánování je nezbytné myslet na optimalizaci webu pro prohlížeče a optimalizaci pro vyhledávače. Postupy spjaté s touto problematikou jsou popsány postupně v kapitolách 3.1.3, 3.2 a 3.3, ovšem jejich použití provází celý vývojový proces.

3.1.1 Analýza a plánování

Analýza problému a plánování patří ke klíčovým bodům ve vývoji webových stránek. Je velmi důležité na základě dostupných informací formulovat očekávané funkce prezentace. Veškeré definice cíle práce jsou však v této chvíli návrhu oproštěny od možností jednotlivých technologií, které přicházejí na řadu až vzápětí. Po důkladné analýze je třeba navrhnout takové technologie, které zadaný problém řeší. S výběrem může být spjata i možná nežádoucí nekompatibilita a další rizika. K těm může třeba patřit použití méně podporovaných nebo rozšířených technologií.

Pro prezentaci na internetu je zapotřebí zvolit také vhodnou internetovou doménu. Měla by nejspíše obsahovat název firmy, obchodu nebo služby, kterou společnost nabízí. Pokud firma hodlá vstoupit i na zahraniční trhy, měla by uvažovat i o jiných doménách prvního řádu než je české CZ. Může se jednat například o slovenské SK nebo nejrozšířenější COM. S tímto rozhodnutím samozřejmě souvisí i volba jazykových mutací. Dále je nutné zvolit hosting, neboli místo, kde se prezentace bude fyzicky nacházet. Při zohlednění všech specifíků, ke kterým patří například použité technologie, je možné využít nabídku jedné z externích firem nebo vybudovat vlastní WWW server.

Na základě požadavků od zákazníka a s přihlédnutím ke všem specifíkům právě navrhovaného projektu je nutné sestavit časový plán obsahující důležité milníky a detailní rozpočet.

3.1.2 Struktura a obsah

V tomto bodě vývojového procesu je třeba získat a roztřídit informace do kategorií a vytvořit navigační menu. Dále je potřeba zkusit, jestli bude navržená architektura fungovat dobře v praxi. Uspořádání by mělo být logické a v návaznosti na design stránek. Tvorba obsahu patří ke kritickým faktorům, neboť v praxi se lze setkat se stránkami s kvalitním designem, ale bez podstatných informací. Proto je potřeba dopředu zajistit, kdo a jak bude obsah doplňovat a aktualizovat. Jako řešení se nabízí dva modely. Statický model představuje obdobu papírové vizitky na internetu. Výhodou může být nenáročnost na serverové technologie, naopak negativem fakt, že při každé aktualizaci obsahu musí zasáhnout odborník. Druhou možností je dynamický model, který nabízí možnost aplikace redakčního systému. Ten umožňuje předat samotným uživatelům nástroj pro správu obsahu na webu.

3.1.3 Grafický návrh

Grafický návrh dodává poskytnutým informacím přidanou hodnotu. Stránky proto nejsou jednotvárné a ploché, ale mohou korespondovat s předmětem podnikání, firemními barvami a logem. Při návrhu, ať už s použitím tužky a papíru nebo grafického editoru, je třeba zvážit několik bodů. Primárním místem zobrazení webových stránek jsou monitory o různých rozlišeních, je proto nutné zohlednit při návrhu šířku a délku

stránky. Mezi další zařízení patří mobilní telefony nebo tiskárny, které lze odlišit a přiřadit jim styl ušitý na míru. Po dokončení grafického návrhu probíhá transformace do CSS. Bohužel skoro každý internetový prohlížeč zobrazuje stránku trochu jinak, proto se musí tyto chyby eliminovat. Více o tomto problému je popsáno v kapitole 3.3 Vývoj webových standardů a internetových prohlížečů. Výsledkem práce je šablona, která poslouží pro vkládání obsahu.

„Pokud chceme vytvářet opravdu profesionální aplikace, měli bychom si naše dílo prohlédnout ve všech nejpoužívanějších prohlížečích“ Převzato z [12, s. 25]

3.1.4 Budování webu

Pro samotné budování webu je třeba mít již jasnou představu o tom, jaké technologie a jaké postupy budou využity. Při budování statického webu se v této fázi vývoje vkládá obsah do přichystané šablony a propojují se stránky odkazy.

Tvorba dynamického webu je komplikovanější. Při budování internetových stránek založených na databázi přichází na řadu modelování datového úložiště. Programátor musí mít alespoň základní představu o tom, jaká data budou ukládána, aby dokázal návrh optimalizovat. V některých případech je velmi výhodné tuto část začít ještě před grafickým návrhem nebo současně s ním. Po návrhu databáze následuje aplikace serverových a klientských technologií. Tyto činnosti spojené s programátorskou prací mohou mít za cíl tvorbu například administračního rozhraní, které umožňuje naplnění databáze daty.

Po provedení všech předešlých bodů je nutné prezentaci důkladně otestovat. Zakončením této fáze je spuštění webu na hostingovém serveru.

3.2 Propagace a optimalizace pro vyhledávače

Reklama je sdělení, jehož prostřednictvím se spotřebitelé o produktu dozvídají a přitom se snaží přemluvit zákazníka, aby si zboží koupil. Může také upozorňovat na webové stránky firem nebo různé akce. Důležité je, aby se zákazník o webu dozvěděl a pravidelně ho navštěvoval. Ke splnění těchto cílů je nutná propagace webu.

Mezi výhody internetové reklamy a propagace patří snadná tvorba analýz a vyhodnocování úspěšnosti, flexibilita a zacílení přímo na jednotlivého uživatele. Pomocí dostupných nástrojů lze snadno změřit, kdy a na jakou reklamu uživatelé reagují. Samotný obsah reklamy může být zacílen přímo na jednotlivce. Forma reklamy už není jen statická, jak ji známe z novin, časopisů a billboardů, ale umožňuje realizaci efektních animací. V neposlední řadě jsou výhodou nižší náklady v porovnání s klasickou reklamou.

Reklama ve vyhledávacích a katalogích

Velmi efektivní druh propagace je využití internetových katalogů a vyhledávačů. Oba typy se však od sebe výrazně liší, i když to nemusí být na první pohled rozeznatelné.

Katalog je web, který obsahuje odkazy na jiné webové stránky. Záznam se do příslušné sekce katalogu vkládá manuálně. Fyzicky se ovšem objeví až po kontrole pověřené osoby. Pozice stránky ve výsledku vyhledávání je ovlivněna shodou hledané fráze s již zadanými klíčovými slovy a popiskem stránky.

Vyhledávač je robot, který prochází internet a indexuje nalezený obsah, který ukládá do své databáze. Vyhledávaná fráze je pak porovnávána s obsahem této databáze. Docílení vyšších pozic ve výsledku vyhledávání lze dosáhnout pomocí metody *Search Engine Optimization*, zkráceně SEO. Jedná se o optimalizaci webových stránek pro roboty. I když nikdy nebyly zveřejněny zdrojové kódy vyhledávacích strojů, existuje alespoň obecná představa o tom, jak fungují a této domněnce se metoda SEO drží. Jedná se o rozsáhlý postup, jak stránku optimalizovat tak, aby ji mohl robot bez problému přečíst. Po úpravě internetových stránek je větší pravděpodobnost, že bude ve

výsledcích hledání na předních pozicích. Kdyby však jen toto nestačilo, je možné si místo ve vyhledávači zaplatit.

Metody SEO lze rozdělit na on-page a off-page faktory. První jmenovaný se zaměřuje přímo na obsah webové stránky a off-page faktory se zabývají tím, co na stránce uvedeno přímo není. V následujícím přehledu uvádím podstatné metody obou zmíněných faktorů. V sestavování těchto bodů mi byla nápomocna literatura číslo [25].

On-page faktory

- Volba klíčových slov – velmi důležitá disciplína spjatá s výběrem a použitím klíčových slov, které na stránce budou obsaženy
- Validní (X)HTML kód a použití důležitých tagů – identifikace a rozřazení obsahu do příslušných značek (titulek, meta informace, nadpisy, meta informace a další)
- Oddělení obsahu od vizuální podoby pomocí CSS
- Flash a Silverlight – menu a důležitý obsah by neměly být prezentovány pomocí těchto technologií, špatně se indexují
- JavaScript v externím souboru – zajišťuje čitelnost a přehlednost kódu, prohlížeč pak stahuje skript jen jednou
- Adresa – volba zapamatovatelné adresy, která popisuje obsah dokumentu, u dynamických webů se využívá přepisování adres

Off-page faktory

- Odkazy směřující na stránku a text odkazu

Další možnosti propagace

Distribuce reklamy elektronickou poštou nebo zobrazování reklamních bannerů na obrazovce patří k dalším možnostem propagace. Vyšší návštěvnost lze také zařídit publikováním v internetových médiích, získáváním zpětných odkazů a výměnou reklamních bannerů na tematicky spřízněných webech.

3.3 Vývoj webových standardů a podpora internetových prohlížečů

Webové standardy patří k nedílné součásti vývoje webových stránek. V jejich definici je popsáno, jak má jádro prohlížeče implementované prvky zobrazovat a současně poskytuje programátorovi potřebnou dokumentaci pro vývoj. Z historie evoluce prohlížečů je však patrné, že ne vždy byla podpora dokonalá. Nastíněný stav však přetrvává i v dnešní době. Proto nejsou standardy z pohledu vývojářů tak důležité jako korektní zobrazení stránky v používaných prohlížečích. V některých případech není podstatné právě to, co říkají standardy, ale samotná implementace technologií ve webovém prohlížeči. Setkat se tak lze s kombinacemi, které třeba nejsou validní, ale webová prezentace se zobrazuje ve všech prohlížečích korektně. A to je podle mého názoru podstatné. S touto problematikou je spojen termín validita, což není nic jiného než zhodnocení stránky, zda zdrojový kód odpovídá či neodpovídá standardům. Z těchto výše popsaných důvodů je nutné brát v úvahu optimalizaci webu pro internetové prohlížeče.

Podle mého názoru se web může rozvíjet jen tak rychle, jak se rozvíjejí internetové prohlížeče a nikdy rychleji. Záleží totiž na ochotě uživatelů, zda budou pravidelně instalovat nové verze prohlížečů. V opačném případě budou totiž pořád používat starší verzi, ale obsah, který využívá nejnovější technologie, nebude dostupný. Vzácným příkladem je Internet Explorer 6 od společnosti Microsoft. Jedná se o starý prohlížeč nechvalně známý problémy s bezpečností a parsingem CSS. Dlouhou dobu i po uvedení novějších verzí však internetu dominoval. V dnešní době je již na ústupu. Důvodem může být změna nebo aktualizace operačního systému. Další variantu představuje přechod na nový typ prohlížeče, například také díky osvětě uživatelů. Po návštěvě vybraných webů s tímto prohlížečem se totiž zobrazují varovné zprávy a v některých případech je zamezen přístup úplně, tak jako u Google Apps. K tématu optimalizace webu je dalším hezkým příkladem a oříškem pro webové programátory zároveň výslovná nelibost firmy Apple k Adobe Flash, který obstarává multimediální obsah na straně klienta. Firma má v úmyslu nahradit jej jazykem HTML5, který jsem stručně popsal v teoretických východiscích této práce. S největší pravděpodobností ani jeden z těchto přístupů k interaktivnímu obsahu nezanikne ze dne na den a budou existovat vedle sebe. Posoudit fakt, zda šlo o revoluční krok nebo šlápnutí vedle, může až čas.

3.4 Vývojové prostředí

Již v kapitole obsahující teoretická východiska této práce jsem se velmi stručně zmínil o faktu, že je možné pro návrh webové prezentace vybírat z mnoha, ne vždy kompatibilních, technologií. Velkou nabídku čítají především serverové technologie a databáze. V praktické části této práce jsem se rozhodl implementovat administrační rozhraní, pro jehož vývoj budu potřebovat databázi a technologii serveru. Základní informace o prostředcích pro vývoj uvedu v této kapitole.

Samotný vývoj HTML a CSS lze provádět velmi jednoduše pomocí webového prohlížeče a prostého textového editoru. Na trhu jsou však dostupné takové programy, které dokážou psaní zdrojového kódu zpříjemnit například zvýrazňováním syntaxe nebo našeptáváním kódu. Zvláštní skupinu tvoří skupina WYSIWYG editorů, kde je možné webovou stránku tvořit pomocí grafického rozhraní. Patří k nim mimo jiné programy Adobe Dreamweaver a Microsoft Expression Web. Základní prezentace tak lze jistě vytvořit, ale pro pokročilou tvorbu je nutné zasáhnout přímo do kódu. Následné testování je na použitém editoru nezávislé a není dobré ho podcenit. Správným přístupem je vyzkoušet interpretaci výsledku na různých prohlížečích, ke kterým se řadí Internet Explorer, Mozilla Firefox, Opera, Safari a Google Chrome.

Práce s databází a serverovými technologiemi si ale žádá mnohem více. Pro realizaci projektu rozpracovaného v následující kapitole jsem si vybral volně šiřitelné prostředky, ke kterým se řadí webový server Apache, skriptovací jazyk PHP a databáze MySQL. Pro zahájení práce je nutné všechny tyto technologie nainstalovat a konfigurovat. Nabízí se i možnost vzdáleného testování na hostingovém serveru, jedná se ale z mého hlediska o neefektivní řešení. Nejen že je nutné mít neustálé připojení k internetu, ale rovněž je nezbytné při každé změně soubor znovu nahrát na vzdálený server. Při závěrečném testování je důležité sladit konfiguraci s hostingovým serverem. Po těchto přípravných činnostech je možné zahájit práci přímo na projektu. K vývoji je možné použít jednoduchý textový editor až po robustní vývojové prostředí. Takové může nabídnout správu projektu, pokročilé ladění pomocí integrovaného debuggeru nebo zajistit správu databáze. K silným prostředkům se řadí pro webový vývoj upravené Eclipse PDT, všestranný PSPad nebo legendární Vim.

4 Vlastní návrh řešení

4.1 Analýza a plánování

Těžiště teoretických znalostí budu následně implementovat do zcela reálného podnikatelského prostředí. Na uvedení tohoto projektu do běžné praxe jsem se dohodl s evropskou akciovou společností HEIM Trade SE se sídlem v Brně – Přízřenicích na Vídeňské ulici č. or. 120. Tato společnost provozuje na téže adrese v Brně, ve dvou lokalitách na území hlavního města Prahy, dále pak v Mníšku pod Brdy a v Čelákovících areály s výrobními, provozními, resp. skladovacími a administrativními prostory. Po jejich revitalizaci a modernizaci je pak poskytuje k pronájmu dalším podnikatelským subjektům. Jedná se tedy o formu obchodování B2B, a to teritoriálně na území České republiky, což povede alespoň prozatím výhradně k české jazykové verzi internetových stránek a k výběru domény „cz“.

Aktuální prezentace nabídky společnosti probíhá prostřednictvím outdoorové komunikace nebo plošné mediální inzerce v místě samotném s odkazem na jediný komunikační kanál, a to telefonní kontakt. Doplnění webové adresy k těmto reklamním aktivitám nasměruje potenciálního zákazníka k přehledné prezentaci a společnost tak získá novou komunikační bránu s podnikatelským prostředím. Hned první informace pro návštěvníka tak bude vyčerpávající, ucelená a profesionální. Následující telefonický kontakt už bude konkrétní a tedy mnohem efektivnější.

Projekt připravím tak, aby uživatel mohl ve své internetové prezentaci sám aktualizovat nabídku informací o volných, resp. obsazených kapacitách. Záměrně uvádím i termín „volné“ nebo „pronajaté“ kapacity nebo prostory, neboť se domnívám, že i tento údaj je pro návštěvníka zajímavý. Na jedné straně sice nevyhoví současné poptávce, na straně druhé však tato skutečnost má i neméně důležitou schopnost vypovídací.

Projekt realizuji podle plánu popsaného v kapitole 3.1. Postup vývoje webové prezentace. Po provedené analýze definuji informační strukturu a návrh obsahu. Následně navrhuji grafickou podobu a přenesu ji do jazyka CSS. Poté připravím datový model a implementuji administrační rozhraní. Cílem práce je prezentace vystavená na

internetu. Podle nastíněných specifik a s přihlédnutím k využití administračního rozhraní jsem se rozhodl pro následující technologie. Pro prezentaci struktury a popisu grafiky využiji jazyky XHTML a CSS. Funkce administračního rozhraní naprogramuji pomocí jazyka PHP, správu dat bude zajišťovat databáze MySQL. Vše poběží na webovém serveru Apache, který mi později dovolí využít jeden ze svých modulů. JavaScript bude sloužit jen jako doplněk pro kontrolu vstupních formulářů. Všechny zmíněné prostředky jsem si vybral hlavně díky svým expertním zkušenostem z praxe.

4.2 Struktura a obsah

Mým hlavním cílem pro formulování struktury webu je takové rozvržení, kde je možné dostat se k informacím po co nejmenším počtu kliknutí. Celý web budu této myšlence přizpůsobovat. Po prvním příchodu uživatele se zobrazí úvodní stránka s krátkým popisem firmy a hlavní menu, které čítá města, ve kterých firma podniká. Jeho obsah vypadá následovně:

- Brno
- Praha
- Mníšek
- Čelákovice

Kontakty a slogan budou společně s logem společnosti dostupné na každé zobrazené straně. Samotné logo bude sloužit pro návrat k úvodním informacím o firmě.

Po výběru jednoho z měst bude uživatel přesměrován na stránku s nabídkou firmy, která je rozdělena do následujících kategorií:

- Haly
- Plochy
- Kanceláře

Pro správu celé nabídky navrhnu administrační rozhraní, s jehož pomocí půjde vkládat, editovat a mazat jednotlivé položky. Kontakty, slogan a text úvodní strany bude v prezentaci napevno zabudován ve zdrojovém kódu a pro jeho editaci bude potřeba zásah odborníka. Tyto údaje totiž nebudou předmětem časté změny.

4.3 Grafický návrh

4.3.1 Příprava návrhu

Grafický koncept je velmi důležitý, neboť právě podle něho bývá obvykle webová prezentace hodnocena. Díky firmě HEIM Trade SE, která mi poskytla veškeré potřebné podklady a informace pro zvládnutí tohoto bodu práce, jsem dokázal vytvořit takový návrh, který koresponduje s předmětem podnikání, firemními barvami a logem. Celý proces návrhu lze spojit do několika bodů. Na začátku jsem si vystačil jen s tužkou a papírem, a poté, až když jsem měl jasno v rozložení bloků a jednotlivých textových atributů, jsem převedl náčrt pomocí bitmapového editoru do elektronické podoby. Pak následovalo rozsekání na jednotlivé grafické komponenty, jejichž použití provází celou transformaci do jazyka CSS.



Obrázek 3 – Grafický návrh internetové prezentace

V horní třetině stránky je umístěno logo firmy a slogan. Menu je zarovnané zleva se začátkem horního textu a končí přesně podle pravé hrany obsahové části. Po najetí kurzoru myši se položka menu zvýrazní. Samotný obsah je rozdělen na dva sloupce. Pravý zahrnuje veškeré kontakty a levý úvodní informace, které jsou zarovnány podle

sloganu a navigace. Na úvodní stránce se zobrazuje ještě obrázek se sídlem společnosti, který po výběru jednoho z měst nahradí podřízené menu s nabídkou kategorií.

4.3.2 Tvorba HTML šablony a transformace do CSS

Při tvorbě kaskádových stylů budu vycházet z níže uvedeného strukturovaného dokumentu. Obsah jsem podle typu označil příslušnými HTML tagy a v některých případech jsem přidal identifikátory a selektory třídy. Celý dokument je definován standardem XHTML 1.0 Strict a kódování jsem stanovil na UTF-8. Později se budu znakovou sadou zabývat ještě podrobněji. Za povšimnutí stojí řešení nadpisu první třídy, které následně využiji k zobrazení loga firmy. Pasáže s dlouhým obsahem jsem v následujícím výpisu nahradil pouze popisky v hranatých závorkách.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
  <title>HEIM Trade SE</title>
  <link rel="stylesheet" type="text/css" href="style/styl.css"
media="screen" />
  <meta name="robots" content="index, follow" />
  <meta http-equiv="Content-language" content="cs" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />
</head>
<body>
  <div id="page">
    <div id="header">
      <div id="logo">
        <h1><a href="index.php"><span>HEIM Trade SE</span></a></h1>
      </div>
      <div id="wrapper">
        <h2 id="heslo">nová adresa vašeho podnikání</h2>
        <h2 id="slogan">haly, plochy a kanceláře k pronájmu</h2>
      </div>
    </div>
    <div id="menu"> [... menu ...] </div>
    <div id="content">
      <div id="obsah">
        <div id="sidebar"> [... sub-menu ...] </div>
        <div id="contentbar"> [... obsah ...] </div>
      </div>
      <div id="kontakty"> [... kontakty ...] </div>
    </div>
  </div>
</body>
</html>
```

Po návrhu strukturované šablony přichází na řadu zápis stylopisu, který vychází z grafického návrhu. Všechny pasáže následně uvedeného kódu jsem zapisoval, jak je z HTML šablony patrné, do souboru *styl.css*. Protože předmětem této práce není jen tvorba grafiky a její vyjádření pomocí CSS, nebudu uvádět veškerý výpis zdrojového kódu, ale jen bloky, které souvisí se základním rozvržením stránky, a pasáže, které jsou pro transformaci klíčové.

Prvním krokem převedení návrhu do CSS je nastavení základní vlastnosti stránky, ke kterým patří velikost a typ písma, pozice stránky v prohlížeči, její šířka a barevné pozadí. V grafickém návrhu je text reprezentován pomocí různých typů fontu *Myriad*. Ten ovšem není na všech počítačích dostupný, a proto jsem pro jeho zavedení použil deklaraci *@font-face*, která umožňuje nahrání vlastního fontu. I přes provedené testování a optimalizaci jsem nebyl schopen tento typ fontu aplikovat. Každý z testovaných prohlížečů písmo interpretoval bohužel jinak, a proto jsem byl donucen pod tíhou těchto skutečností použít základní font Arial.

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 16px;
    color: white;
    background-color: #0052a5;
}

#page {
    width: 1000px;
    margin: 0 auto 0 auto;
    text-align: center;
    background-image: url(header.jpg);
    background-repeat: no-repeat;
}
```

Stránka je dle struktury rozdělena na několik částí. První z nich tvoří hlavička webu, která obsahuje logo, slogan a heslo.

```
#header { height: 200px; }

#header #logo {
    float: left;
    position: relative;
    top: 20px;
}
```


V tuto chvíli přichází na řadu postup zobrazení loga společnosti, které bude sloužit jako odkaz na úvodní stranu. Nejdříve je nutné uvést cestu k obrázku a definovat rozměry prvku, který odpovídá právě velikosti loga.

```
#header #logo h1 {  
    width: 165px;  
    height: 165px;  
    background-image: url (logo.png);  
    background-repeat: no-repeat;  
}
```

Druhý krok představuje umístění plochy odkazu přímo na obrázek. V porovnání je velikost plochy menší než u obrázku, protože nezahrnuje vržený stín, který jako odkaz fungovat nebude.

```
#header #logo h1 a {  
    display: block;  
    width: 150px;  
    height: 150px;  
    text-decoration: none;  
}
```

V posledním fázi je nutné skrýt samotný text nadpisu, který je zabalen do značek `<span>`. Text uvedený v nadpisu první třídy se tedy nezobrazí, ale bude dostupný pro vyhledávací roboty a použijí ho i při tvorbě tiskového stylu. Uvedené řešení tedy vyhovuje metodě SEO.

```
#header #logo h1 a span {  
    visibility: hidden;  
    text-decoration: none;  
}
```

V hlavičce webu jsou také umístěny dva slogany. Jejich pozici jsem vyřešil pomocí vloženého identifikátoru a následně jsem ještě upravil definice, které se zejména liší úpravou atributu písma.

Hlavní menu se nachází přímo nad obsahovou částí a je zarovnáno se začátkem sloganů. Obsah navigace je tvořen pomocí seznamových značek `<ul>`, `<li>` a odkazy `<a>`. Položky, které budou fungovat jako odkaz do příslušné kategorie, jsou zobrazeny

velkým písmem. Po najetí kurzoru nebo při výběru jedné z nabídek změní pozadí vybraného odkazu barvu.

```
#menu {
    font-size: 23px;
    margin: 0 0 0 185px;
}

#menu ul li {
    display: block;
    float: left;
    height: 100%;
    margin: 0;
    padding: 3px 3px 0 3px;
    list-style: none;
    text-transform: uppercase;
}

#menu li+li { margin: 0 8px 0 0; }

#menu ul li a {
    color: white;
    text-decoration: none;
    display: block;
    padding: 3px 5px 5px 5px;
}

#menu ul li a:hover { background-color: #0086cc; }

#menu ul li a#visit { background-color: #0086cc; }
```

Hlavní obsahová část se sestává ze dvou bleděmodrých sloupců, pravý z nich obsahuje dokonce barevný přechod. Podmínkou je, aby oba sloupce skončily zároveň. Obsah prezentace je dynamický a lze jen těžko předpovídat, jak bude stránka dlouhá. Proto vytvořím přes oba zmíněné sloupce řez, který se následně bude opakovat až do skončení obsahu. Délka výsledného obrázku získaného řezem je definována součtem délek obsahu, kontaktů a mezery mezi nimi, tedy 1000px, jeho výška je 1px. Tímto postupem docílím toho, že oba sloupce budou končit zaráz.

```
#content {
    background-image: url(wide.jpg);
    background-repeat: repeat-y;
    text-align: left;
}
```

Obsahové bloky jsou pak rozděleny na kontakty a obsah, který čítá ještě podřízené menu a prostor pro samotnou nabídku společnosti.

```
#content #obsah {
    float: left;
    width: 750px;
    background-color: #0086cc;
    min-height: 500px;
}

#content #sidebar {
    float: left;
    width: 145px;
}

#content #contentbar {
    margin: 5px 0 40px 195px;
}

#content #kontakty {
    width: 190px;
    font-size: 11px;
    padding: 0 0 150px 0;
    margin: 0 0 0 800px;
}
```

Při definování dalších kaskádových stylů CSS jsem v pasážích spjatých s podřízenou navigací, formátováním textů, kontaktů a nabídky postupoval analogicky.

4.3.3 Příprava šablony pro tisk

Pro webové prohlížeče je typické, že pracují nejen se styly, které jsou použity pro obrazovky počítačů, ale dovoluují použít více deklarací dle typu použitého média pro zobrazení. Tato vlastnost se hodí pro přípravu tiskové šablony. Nynější návrh se totiž díky použití mnoha grafických prvků příliš nehodí pro výstup na tiskárnu. Pro implementaci je nezbytně nutné nejprve přidat odkaz na stylový soubor do hlavičky HTML šablony.

```
<link rel="stylesheet" type="text/css" href="style/tisk.css"
media="print" />
```

Výstup na tiskárnu lze tak velmi dobře přizpůsobit. Po aplikaci nové definice se nezobrazí žádná rozsáhlá grafika, která není pro tisk příliš vhodná. Pomocí již použitých identifikátorů jsem docílil takového stavu, ve kterém je odstraněna hlavní i podřízená navigace s podmínkou, že vždy se zobrazí aktuálně navštívené město a kategorie. Název firmy, slogan, kontakty a nabídka společnosti jsem po drobných úpravách zachoval. Celá práce proběhla na základě stávající HTML šablony a nebylo potřeba do ní vůbec zasahovat. Z toho právě pramení jedna z výhod kaskádových stylů CSS.

HEIM Trade SE
nová adresa vašeho podnikání
haly, plochy a kanceláře k pronájmu

Brno
Kanceláře

Areál Videňská 264/120 b

Administrativní budova - 19 m ²	Volné
Administrativní budova - 26 m ²	Volné
Administrativní budova - 27 m ²	Volné
Administrativní budova - 36 m ²	Volné
Administrativní budova - 38 m ²	Volné
Administrativní budova - 51 m ²	Pronajato
Administrativní budova - 91 m ²	Pronajato

Videňská 120b
619 00 **Brno**
tel.: 547 122 331
mobil: 602 550 341
e-mail: stepanek@heimtrade.cz

Ka Kablu 531/5
102 00 **Praha**
tel.: 242 403 622
mobil: 606 690 502
e-mail: kaplan@heimtrade.cz

Obrázek 4 – Tisková šablona webové prezentace

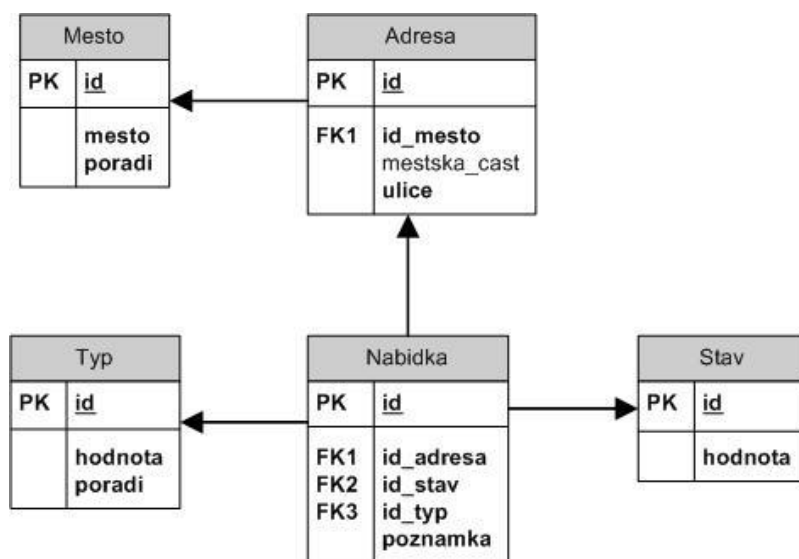
4.4 Budování webu

V této fázi vývoje dynamické webové prezentace přichází na řadu samotné budování administračního rozhraní pomocí zvolených technologií. S ohledem na cíle projektu jsem se rozhodl využít server Apache, technologii PHP a databázi MySQL.

Jednotlivé soubory se zdrojovým kódem jsem logicky roztřídil do adresářové struktury. V kořenovém adresáři se nachází soubor *index.php* a složky *admin*, *core* a *style*. Hlavní soubor *index.php* obsahuje prozatím základní kostru HTML dokumentu a v této fázi do něj bude přibývat PHP kód pro zprovoznění prezentace. Složka *admin* bude obsahovat veškeré soubory spojené s administračním rozhraním, jejichž podrobný popis uvedu také v této kapitole. Jádro prezentace, které obsahuje veškeré funkce pro práci s daty, je umístěno ve složce *core* v souboru *mod_core.php*. Poslední uvedený adresář *style* obsahuje zápisy obsahující CSS a obrázky k prezentaci.

4.4.1 Návrh databáze

Pro realizaci administračního rozhraní je velmi výhodné použít databázi, která je pro práci s daty vytvořena. V této fázi navrhnu relační model databáze a naplním ho daty, která nebude spravovat administrační rozhraní. Výpis SQL příkazů je uveden v přílohách této práce. Návrh datové struktury ilustruje obrázek číslo 5.



Obrázek 5 – Návrh relačního modelu

Veškeré informace o nabídce společnosti sdružuje stejnojmenná tabulka, která kromě cizích klíčů obsahuje ještě dodatečnou poznámku obsahující například výměru kanceláře. Relace *Typ* uchovává hodnoty podřízeného menu, které v tuto chvíli čítají následující data: haly, plochy a kanceláře. Rozlišení, zda je nemovitost například volná či pronajatá, umožňuje relace *stav*. Jelikož firma nabízí své služby ve více městech a na různých adresách, jsou pro tyto účely použity relace *Adresa* a *Mesto*. Ve vybraných tabulkách jsem dodatečně použil atribut *poradi*, ve kterém bude uloženo pořadí položek při zobrazení.

4.4.2 Tvorba administračního rozhraní

Cílem vytvořeného administračního rozhraní webu je vkládat, mazat a editovat nabídku společnosti HEIM Trade SE. Veškeré informace uložené v databázi ovšem nebude spravovat navržená administrace, a proto při změně některých údajů bude nutné provést triviální SQL příkaz. K takovému řešení mě vedl fakt, že některá vložená data se nebudou měnit vůbec nebo jen při zásadní změně podnikání firmy. Navržené schéma databáze je však na případné rozšíření připraveno.

Soubory spojené s prováděním akcí v administračním rozhraní jsou umístěné ve složce *admin*. Vstupním bodem rozhraní umístěným v souboru *index.php* je výpis položek nabídky společnosti, které je možné filtrovat podle adresy. Podle zvolené editační funkce je přivolán soubor s názvem odpovídající šabloně, která obsahuje informace o činnosti každého souboru společně s rozlišením, zda jde o vstupní formulář, označený zkratkou *frm*, nebo o vykonávací soubor se zkratkou *exe*. Po provedení akce se zobrazuje výsledek a přesměruje se zpět na výpis nabídek administračního rozhraní.

Veškeré funkce pro zobrazování na stránce i pro administrační rozhraní jsou zapisovány v souboru *mod_core.php* ve složce *core*, z kterého uvádím následující pasáže kódu. Nejdříve jsem definoval místo, kde jsou uloženy **konstanty** týkající se připojení k databázi. Toto řešení je vhodné v případě, kdy je prezentace umístěna na více místech současně. Obsah souboru s hlavními funkcemi je možné dále upravovat, ale definice konstant zůstávají na hostingovém serveru i testovacím stroji nezměněné.

```
require_once dirname(__FILE__) . '/const.php';
```

Funkce sloužící pro **připojení k databázi** využívá již zmíněné konstanty a musí být volána vždy při práci s databází. To znamená nejen v administračním rozhraní, ale i následně ve výpisech nabídky. Za povšimnutí stojí hned dva SQL dotazy, které umožní korektní zobrazení českých znaků. S touto funkcí souvisí i další funkce s názvem *db_close()*, která práci s databází ukončí.

```
function db_connect() {
    $handler = mysql_connect(DB_HOST, DB_USER, DB_PASS);
    if(!is_resource($handler)) return false;
    $select_db = mysql_select_db(DB_NAME);
    if(!$select_db) return false;
    mysql_query("SET CHARACTER SET utf8");
    mysql_query("SET NAMES utf8");
    return $handler;
}
```

Přihlašování do systému je řešeno pomocí základní HTTP autentizace, která je ve zmíněném protokolu integrována. V návrhu uvažuji takové rozhraní, které bude spravovat právě jeden uživatel, a proto budou přihlašovací údaje uloženy přímo v PHP souboru. Funkce, která autentizaci obsluhuje, musí být volána v každém zabezpečeném souboru. Vstupní bránu do administrace představuje adresa ve tvaru *www.heimtrade.cz/admin*, po jejímž zadání je uživatel vyzván k zadání přihlašovacího jména a hesla. Při úspěšné autentizaci je uživateli umožněn přístup k úpravám informací na webu, v opačném případě je možné přihlašovací proces opakovat.

```
function login_dialog() {
    Header("HTTP/1.0 401 Unauthorised");
    Header("WWW-Authenticate: Basic realm='HeimTrade Admin'");
    Header("Content-Type: text/html; charset=utf8");
    echo "<h2>Neexistující uživatel nebo chybné heslo!</h2>";
    echo "<a href='\".basename($_SERVER['SCRIPT_FILENAME']).\"'>";
        Přihlašte se, prosím.</a>";
    die();
}

function authentication () {
    $userName = isset($_SERVER['PHP_AUTH_USER']) ?
        $_SERVER['PHP_AUTH_USER'] : NULL;
    $userPassword = isset($_SERVER['PHP_AUTH_PW']) ?
        $_SERVER['PHP_AUTH_PW'] : NULL;
    if($userName != "administrator" || $userPassword != "Xq5f.UiK")
        login_dialog();
}
```

Základní **funkce administračního rozhraní** pokrývají požadované cíle tohoto projektu. Předmět jejich činnosti zahrnuje přidávání, editaci a mazání jednotlivých položek obsahujících nabídku společnosti. Nejdříve se budu zabývat tím, jak je systém naprogramován a poté systém popíšu z pohledu uživatele. Pro ukázkou jedné funkce si vyberu činnosti, které jsou spojené s vkládáním nabídky.

```
function vlozit_nabidku($adresa, $typ, $poznamka, $stav) {  
    $query = "INSERT INTO nabidka  
              (poznamka, id_adresa, id_stav, id_typ) VALUES  
              ('{$poznamka}', '{$adresa}', '{$stav}', '{$typ}')";  
    return $result = mysql_query($query);  
}
```

Výběrem jedné z voleb v administračním rozhraní, v tomto případě položky *Vložit nabídku*, se zobrazí editační formulář. Po vyplnění všech údajů je pro jeho odeslání nutné zvolit tlačítko *Uložit*. Následně se provede kontrola zadaných informací pomocí JavaScriptu a v případě jakékoli chyby je uživatel vyzván k jejich opravě. Poté je formulář odeslán určenému PHP skriptu, který zpracovává vstupní proměnné a vkládá informace pomocí funkce *vlozit_nabidku()* do databáze. Když jsou proměnné zadány chybně nebo když se data neuloží do databáze, je zobrazena varovná zpráva. V případě úspěšného uložení dat je uživatel přesměrován na seznam nabídky společnosti, kde je možné vidět nově přidanou položku.

Zbylé funkce spojené s editací a mazáním fungují velmi podobně. Snad jen u editace je nutné nejdříve podle identifikátoru nabídky načíst informace do formuláře a až poté jsou tato data zpracovávána.

Adresa

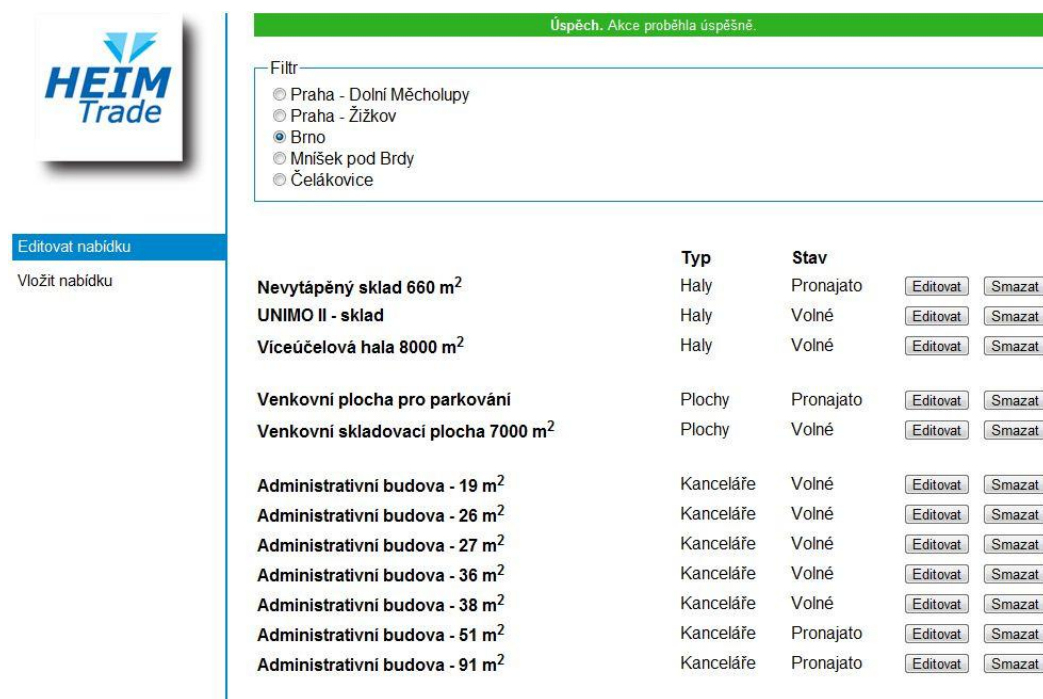
- ☒ Praha - Dolní Měcholupy
- ☐ Praha - Žižkov
- ☐ Brno
- ☐ Mníšek pod Brdy
- ☐ Čelákovice

Typ

- ☒ Hala
- ☐ Plocha
- ☐ Kancelář

Obrázek 6 – Náhled formuláře sloužícího pro vložení nové nabídky

Uživatelské rozhraní administračního systému jsem navrhoval s ohledem na jednoduchost a přehlednost celé aplikace, výsledek mé práce ilustrují obrázky 6 a 7. V levém sloupci je umístěno navigační menu a pravá část zobrazuje grafický interface editačních modulů. V horní části je umístěn informační proužek, který pomocí textu a modré, zelené a červené barvy informuje o právě provedené akci. Vstupy formulářových prvků kontroluje ještě před odesláním na server JavaScript. Toto řešení umožňuje například zobrazit ještě před smazáním položky výzvu, zda má být akce opravdu provedena. Častým problémem při vkládání nové nabídky bude jistě vložení a následné zobrazení výměry. Proto jsem do formuláře přidal tlačítko, které přidá text s HTML zápisem metrů čtverečních do vstupního pole. Tak bude možné výměru zobrazit v každém prohlížeči korektně.



	Typ	Stav		
Nevytápěný sklad 660 m ²	Haly	Pronajato	Editovat	Smazat
UNIMO II - sklad	Haly	Volné	Editovat	Smazat
Víceúčelová hala 8000 m ²	Haly	Volné	Editovat	Smazat
Venkovní plocha pro parkování	Plochy	Pronajato	Editovat	Smazat
Venkovní skladovací plocha 7000 m ²	Plochy	Volné	Editovat	Smazat
Administrativní budova - 19 m ²	Kanceláře	Volné	Editovat	Smazat
Administrativní budova - 26 m ²	Kanceláře	Volné	Editovat	Smazat
Administrativní budova - 27 m ²	Kanceláře	Volné	Editovat	Smazat
Administrativní budova - 36 m ²	Kanceláře	Volné	Editovat	Smazat
Administrativní budova - 38 m ²	Kanceláře	Volné	Editovat	Smazat
Administrativní budova - 51 m ²	Kanceláře	Pronajato	Editovat	Smazat
Administrativní budova - 91 m ²	Kanceláře	Pronajato	Editovat	Smazat

Obrázek 7 – Uživatelské rozhraní administrace webu

4.4.3 Tvorba uživatelského rozhraní

Celý vývojový proces provází volba kódování, která je pro korektní zobrazení diakritiky českého jazyka nezbytná. Díky univerzálnímu použití jsem zvolil kódování **UTF-8**, které jsem definoval již při založení databáze, ve funkci sloužící k připojení k databázi, v hlavičce HTML šablony pomocí tagu `<meta>`, v souborech obsahující zdrojový kód a v hlavičce PHP souborů pomocí funkce *header*.

```
header("Content-Type: text/html; charset=utf8");
```

V projektové fázi nazvané tvorba uživatelského rozhraní se dostávají na řadu činnosti reprezentující spojení již navržené HTML šablony s informacemi umístěnými v databázi. Kompletní řízení obsahu a zobrazení dat bude mít na starosti soubor v kořenovém adresáři pojmenovaný jako *index.php*. Právě zde budou použity funkce zabezpečující mimo jiné připojení a odpojení databáze, výpisy hlavní navigace, podřízeného menu a nabídky. V následujících řádcích uvedu použití těchto základních funkcí společně s popisem logiky fungování prezentace.

Po načtení souboru, který obsahuje řídicí funkce, a po připojení k databázi je nutné definovat a ošetřit **vstupní proměnné**. Právě jejich působením je obsah ovlivněn. V tomto případě jsou vstupy reprezentovány číselnými identifikátory, které definují zvolenou hlavní a podřízenou kategorii.

```
$frm_mesto = isset($_GET['mesto']) ? $_GET['mesto'] : FALSE;
$frm_typ    = isset($_GET['typ']) ? $_GET['typ'] :
               neprazdna_kategorie($frm_mesto);
```

Aplikace je nastavena tak, že pokud není definován identifikátor města, zobrazí se úvodní strana. V opačném případě se vypíše podřízená navigace společně s nabídkou společnosti ve zvoleném městě a kategorii. Tuto činnost zajišťuje právě definice vstupu společně s funkcí *neprazdna_kategorie()*, která najde a vrátí identifikátor z tabulky *typ* s podmínkou, že je kategorii přidružena alespoň jedna nabídková položka. Tímto postupem je dosažen takový stav, že když uživatel zvolí město a zároveň nespecifikuje položku v podřízeném menu, tak se zobrazí nabídka v první neprázdné kategorii.

Pro **výpis položek hlavní navigace** slouží funkce s názvem *vypis_mesta()*, která vrací pole se seřazenými údaji obsahující informace o městech v databázi. Pro tvorbu odkazu v navigaci jsem kromě názvu města a identifikátoru použil již známou funkci *neprazdna_kategorie()*, jejíž popis jsem uvedl v předchozím odstavci.

```
function vypis_mesta() {
    $query = "SELECT id, mesto FROM mesto ORDER BY poradí;";
    $result = mysql_query($query);
    $mesta = array();
    while ($row = mysql_fetch_assoc($result)) {
        $mesta[] = $row;
    }
    return $mesta;
}
```

Výpis podřízeného menu má na starosti funkce *vypis_submenu()*. Při tvorbě odkazů je zde nutné odlišit prázdné kategorie, které neobsahují žádné položky, a ty ostatní. Proto jsem využil funkci *pocet_nabidek()* která na základě *id* města a *id* kategorie vrací počet položek podle zvolených parametrů. Pokud se takto zjistí, že kategorie neobsahuje žádné položky, odkaz v navigaci se zablokuje a text zašedne.

```
function pocet_nabidek($mesto, $typ) {
    $query = "SELECT COUNT(*) as 'pocet' FROM mesto m, adresa a,";
    $query .= "nabidka n WHERE a.id_mesto=m.id AND n.id_adresa=a.id";
    $query .= "AND m.id={$mesto} AND n.id_typ={$typ}";
    $result = mysql_query($query);
    $pole = mysql_fetch_assoc($result);
    return $pole['pocet'];
}
```

Pokud je vybráno jedno z měst v hlavním menu, zobrazí se **výpis nabídky**. Protože jsou položky ještě děleny podle adresy, je nutné využít vnořený cyklus. Výpis pak vypadá tak, že pod každou adresou jsou vypsané nabídkové položky spadající právě pod vybrané město, kategorii a adresu. Za určitých okolností se může stát, že adrese nebudou přiřazeny žádné nabídkové položky. Tuto situaci řeší funkce *pocet_nabidek_na_adrese()*, na jejímž základě je adresa vypsána nebo nikoli.

4.5 Optimalizace

Optimalizaci pro prohlížeče jsem prováděl přímo ve vývojových fázích zpracovávaného projektu. Z dlouhodobě získávaných znalostí jsem byl schopen už za běhu implementovat postupy vedoucí ke korekci drobných rozdílů mezi jednotlivými prohlížeči. Úspěšné testování proběhlo na internetových prohlížečích Internet Explorer 7 a 8, Mozilla Firefox, Opera, Apple Safari a Google Chrome.

Při **optimalizaci pro vyhledávače** jsem dodržoval všechny zásadní postupy, ke kterým patří volba klíčových slov, validní kód, definice nadpisů, seznamů apod. Dosud však jedna technika zůstala bez povšimnutí. Problémem dynamických webů je jejich „ošklivá“ a jistě špatně zapamatovatelná adresa. V této fázi vývoje totiž vypadá URL adresa pro zobrazení konkrétní nabídky následovně:

```
/index.php?mesto=2&typ=3
```

Funkčnost stránek tímto ale není omezena, a není tedy problém web prezentovat na internetu v takovémto stavu. Ovšem ani uživatel ani vyhledávací robot nezjistí, že právě město s identifikátorem 2 je Brno a typ 3 jsou právě kanceláře. Tento problém je řešitelný pomocí doplňku serveru Apache s názvem *mod_rewrite*, který využiji ve své práci. Jeho aplikace nabízí prostředek pro sofistikované přepisování adres. Mým cílem bude transformovat dynamickou adresu na statickou, ve které budou uvedena klíčová slova. Výsledkem tedy bude adresa, která bude lépe vypovídat o obsahu zobrazené stránky.

```
/brno/kancelare/
```

Podmínkou je, že jak položky hlavního, tak i podřízeného menu musí být unikátní, nesmí se tedy opakovat. To nejlépe zabezpečím zásahem do návrhu databáze.

Prvním krokem pro vybudování nastíněného modelu budou takové úpravy, aby adresa vypadala následovně:

```
/index.php?mesto=brno&typ=kancelare
```

Nejdříve jsem k definicím konstant přidal nové dvě. V první z nich je uložen údaj o kořenovém adresáři webu, který zamezí chybám při odkazování na jiné stránky. Pomocí druhé konstanty bude možné snadno zapínat a vypínat přepisování URL adres.

```
define('DOMENA','http://www.heimtrade.cz');
define('PEKNE_URL',1);
```

Následně je ve zdrojovém kódu nutné upravit vstupní proměnné, ve kterých teď nebudou figurovat číselné identifikátory, ale text. Aby byl zajištěn bezproblémový chod aplikace, musím definovat takové funkce, které tento vstupní text na základě databáze nahradí příslušným identifikátorem.

```
$frm_mesto = isset($_GET['mesto']) ?
    ziskej_id_mesta($_GET['mesto']) : FALSE;
$frm_typ    = isset($_GET['typ']) ?
    ziskej_id_kategorie($_GET['typ']) :
    ziskej_id_kategorie(neprazdna_kategorie($frm_mesto));
```

Mezi nové funkce patří například *ziskej_id_mesta()*, která slouží pro získání identifikátoru města na základě vstupního textu z URL adresy. Na stejném principu pak pracuje i funkce *ziskej_id_kategorie()*. Pro převedení textu získaného z databáze na tvar, který je možné vložit do URL, slouží funkce *priprav_text()*. Například text obsahující město „Mníšek pod Brdy“ se převede na „mnisek-pod-brdy“. Dále jsou definovány funkce pro získání názvu města a kategorie pomocí jejich identifikátoru.

```
function ziskej_id_mesta($mesto) {
    $query = "SELECT id, mesto FROM mesto;";
    $result = mysql_query($query);
    while ($row = mysql_fetch_assoc($result)) {
        if(priprav_text($row['mesto']) == $mesto)
            return $row['id'];
    }
    return false;
}
```

Vše je nyní připraveno k definicím odkazů u hlavní a podřízené navigace. Jejich výpis je prováděn pomocí cyklu, který je uveden v přílohách této práce. Následující ukázka ilustruje pouze část zdrojového kódu spojeného s vytvářením odkazů pro hlavní menu. Výsledná adresa je pak složená z domény, názvu města a kategorie, která obsahuje alespoň jednu nabídkovou položku.

```
$mesto = priprav_text($m['mesto']);
$katgorie = neprazdna_kategorie($m['id'], true);
echo "<li><a href='\".DOMENA.\"/{\$mesto}/{\$katgorie}/{\$style}>
    {\$m['mesto']}</a></li>\n";
```

Na závěr přichází samotné definice, které bezprostředně souvisí s prepisováním URL adresy. Pro jejich zápis je nutné založit konfigurační soubor serveru Apache s názvem *.htaccess*. Jeho obsah je uveden ve výpisu, který následuje tento odstavec.

```
RewriteEngine on

RewriteRule ^admin/$ admin/ [L]

RewriteRule ^([a-z-]+)/([a-z-]*)/$ index\.php?mesto=$1&typ=$2
[L,QSA]
RewriteRule ^([a-z-]+)/$ index\.php?mesto=$1 [L,QSA]

RewriteCond %{REQUEST_URI} !\.[[:alnum:]]+$
RewriteRule ^(.+[^/])$ $1/ [R=301]
```

Působením uvedených zápisů bude zabezpečeno prepisování adres do vymezeného tvaru. Problém s duplicitním obsahem jsem vyřešil vkládáním lomítek za každou adresu, která neobsahuje příponu. K dalším konfiguracím patří definování stránky s chybovým hlášením.

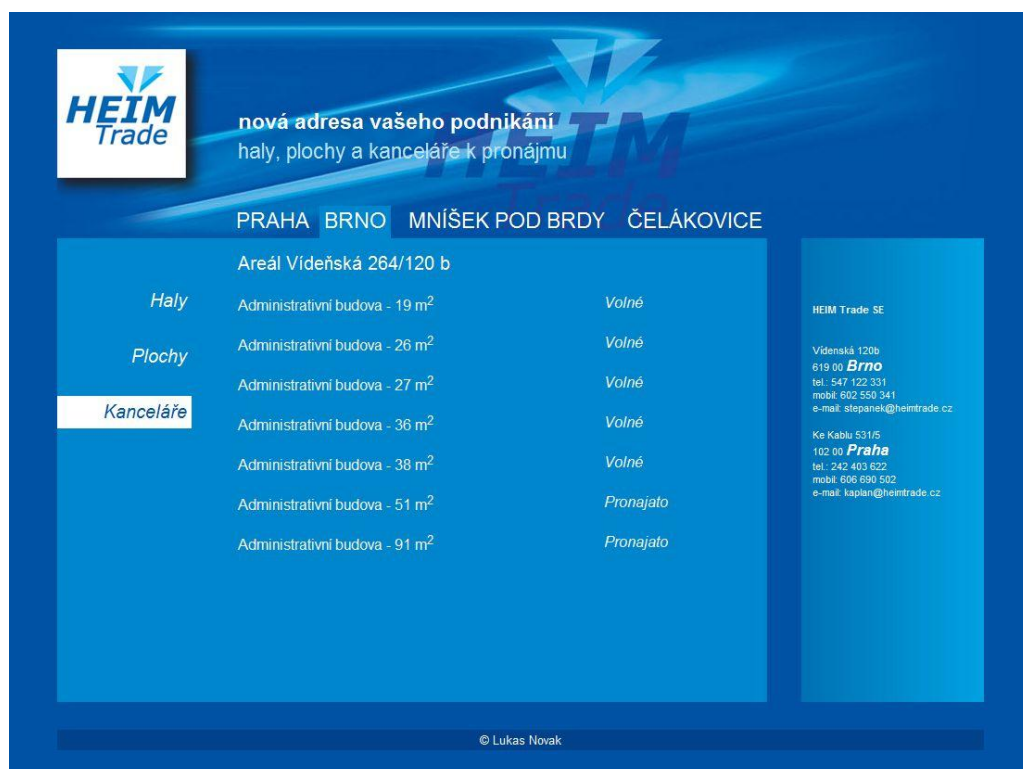
```
ErrorDocument 404 /core/error404.php
```

Veškeré problémy s chybovými zprávami však nejsou vyřešeny. Pokud bude URL adresa odpovídat některým z uvedených předpisů a výsledek se zašle na definovanou stránku *index.php*, tak už je pozdě na řízení chyb ze strany serveru. Proto jsem za definici vstupních proměnných doplnil kód, který slouží pro identifikaci chybějícího obsahu. Pokud tedy skript zjistí, že zadané město nebo kategorie neexistuje, je kromě definice hlavičky se stavovým kódem 404 ještě zobrazen soubor obsahující chybovou zprávu.

```
if(isset($_GET['mesto']) == true &&
    ziskej_id_mesta($_GET['mesto']) == false) {
    header("HTTP/1.0 404 Not Found");
    require "../core/error404.php";
    die();
}
```

4.6 Presentace webové stránky na internetu

Pro finální dokončení práce je zapotřebí webovou prezentaci umístit na internet. Firma HEIM Trade SE dosud nevlastnila svou internetovou prezentaci, ovšem disponuje již doménou *heimtrade.cz*, kterou používá pro e-mailovou komunikaci. Díky této skutečnosti bylo možné na zmíněné doméně pouze aktivovat nové služby spojené s prezentací webových stránek. Pro zdárný chod prezentace jsem vybral takový program hostingové společnosti, který splňuje požadavky, jako jsou podpora serverové technologie PHP, databáze MySQL a možnost přepisování adres pomocí `mod_rewrite`. Pro úspěšnou implementaci jsem na hostingovém serveru vytvořil navrženou databázi a v souboru s konstantami změnil přihlašovací údaje a doménu webu. Výsledek mé práce ilustruje následující obrázek. Vývoj webu se tedy tímto dostal do závěrečné fáze.



Obrázek 8 – Internetová prezentace společnosti HEIM Trade SE

Dalším krokem může být sledování návštěvnosti a vyhodnocování úspěšnosti webových stránek v návaznosti na srovnání výsledků společnosti. Dále je možné relativně bez nákladů rozšířit propagaci webu. O tom podrobně v následující kapitole. Prozatím se totiž používá jen internetová propagace ve vyhledávačích.

4.7 Ekonomické zhodnocení

V současné době je pro jakoukoli firmu prakticky nezbytná prezentace na internetu. Mezi přínosy patří vytvoření nového komunikačního kanálu. Možnost získání nových zákazníků a tudíž bezprostřední ekonomický efekt prostřednictvím této cesty je obtížně měřitelný. Tato platforma totiž není zcela funkční jako nezávislá a vyžaduje ještě další nadstavbovou složku komunikace webové adresy. Předpokládejme, že ta se nebude provádět samoúčelně, ale při každé vhodné příležitosti se k prezentaci názvu, resp. značce společnosti připojí webová adresa. Vizitka, karoserie firemního automobilu, leták nebo outdoorová komunikace firemní značky jsou všechno prostředky dříve používané. V přirozené obměně se jen objeví doplňující údaj. Na tuto formu komunikace lze tudíž pohlížet jako na zcela bezplatnou.

Sumarizace nákladů na zřízení webové prezentace lze pak provést po jednotlivých činnostech, které tvorbu těchto internetových stránek provázely. Patří k nim vytvoření grafického návrhu, jeho transformace do CSS, vytvoření administračního a uživatelského rozhraní, optimalizace pro vyhledávače a náklady na hostingové služby. To jsou náklady pořizovací. Náklady provozní pak každý další rok zahrnují pouze hostingové služby. Platí to však pouze za předpokladu, že tato konkrétní prezentace umožnila vznik virtuálního katalogu, který je dynamický a umožňuje reflektovat aktuální stav společnosti. Nabídka firmy na internetu může být proto stále aktuální, a to bez vnějšího zásahu profesionálního subjektu za úplatu. V případě statické prezentace by bylo nutné kalkulovat v každém následujícím roce vedle hostingových služeb i náklady na změny a doplnění obsahu webové prezentace.

	Statický web	Dynamický web
1. rok	8 000 Kč	15 000 Kč
2. rok	3 000 Kč	1 200 Kč
3. a každý další rok	3 000 Kč	1 200 Kč

Tabulka 9 – Porovnání nákladů statického a dynamického webu

Porovnání nákladů na dynamický a statický web přehledně dokumentuje tabulka č. 9. Jako zdroj finančních veličin byly použity informace od konzultanta této práce.

Závěr

Cílem bakalářské práce bylo vytvořit funkční webovou prezentaci včetně grafického návrhu pro firmu HEIM Trade SE. V aktuálním čase a v tomto konkrétním prostředí jsem uplatnil a aplikoval všechny dosavadní a dostupné znalosti o zadané problematice. Jsem si plně vědom, že odvětví, které je bezprostředně spjato s vývojem a správou internetových stránek, se nesmírně dynamicky vyvíjí. Proto je zapotřebí neustále sledovat dění v oboru a učit se nové metody a efektivní postupy. Stejně tak se může a nepochybně i bude měnit prostředí, ve kterém jsem pracoval a pro které byla tato práce vytvořena. Obchodní společnost je dynamický organismus, který reaguje na podmínky trhu a na nejružnější podnikatelské příležitosti, a to věcně i teritoriálně. Domnívám se, že produkt, který tato práce obsahuje, je na případné vývojové změny připraven a je dostatečně flexibilní. Spolehlivě to však prověří až čas...

Práce na tomto projektu byla pro mne příležitostí k aplikaci celé řady teoretických znalostí do reálného podnikatelského prostředí. Velmi si cením spolupráce s konkrétní firmou, kde jsem měl možnost nasbírat důležité postřehy, připomínky a názory, které byly přínosem pro moji práci a obohatily a rozšířily moje teoretické znalosti o dané problematice.

Seznam použitých zdrojů

Monografie

- [1] BRÁZA, Jiří. *PHP 4: Učebnice základů jazyka*. Praha: Grada, 2002. 224 s. ISBN 80-247-0442-0.
- [2] BRÁZA, Jiří. *PHP 4: Praktické příklady*. Praha: Grada., 2003. 224 s. ISBN 80-247-0441-2.
- [3] CEDERHOLM, Dan. *Flexibilní webdesign*. Brno: Computer Press, 2006. 232 s. ISBN: 80-251-1018-4.
- [4] DARIE, Cristian, et al. *AJAX a PHP: tvoříme interaktivní webové aplikace profesionálně*. Brno: Zoner Press, 2006. 320 s. ISBN 80-86815-47-1.
- [5] DOMES, Martin. *Tvorba webových stránek*. Brno: Computer Press, 2006. 192 s. ISBN: 80-251-0920-8.
- [6] DOSTÁLEK, Libor, KABELOVÁ, Alena. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5. aktualizované vydání. Brno: Computer Press, 2008. 488 s. ISBN: 978-80-251-2236-5.
- [7] EISMANNOVÁ, Katrin. *PHOTOSHOP: výběry, masky a montáž*. Brno: Zoner Press, 2006. 543 s. ISBN 80-86815-41-2.
- [8] GUTMANS, Andi. *Mistrovství v PHP 5*. Brno: Computer Press, 2005. 520 s. ISBN 80-251-0799-X.
- [9] KAHNEY, Leander. *Jak myslí Steve Jobs: Tajemství úspěchu společnosti Apple*. 1. rozš. vyd. Brno: Computer Press, 2009. 304 s. ISBN 978-80-251-2794-0.
- [10] KOUT, Pavel. *Praktický JavaScript*. Brno: Zoner Press, 2004. 326 s. ISBN 80-86815-00-5.
- [11] KUČERA, Miroslav, ŽITNÍK, Lukáš. *Inspirativní webdesign*. Praha: Mobil Media, 2003. 200 s. ISBN 80-86593-42-8.
- [12] KOSEK, Jiří. *PHP: Tvorba interaktivních internetových aplikací*. Praha: Grada, 1998. 492 s. ISBN 80-7169-373-1.

- [13] KRUG, Steve. *Webdesign - Nenutíte uživatele přemýšlet*. 2.vyd. Brno: Computer Press, 2006. 168 s. ISBN: 80-7226-892-9.
- [14] LYNCH, Patrick J., HORTONOVÁ, Sarah. *Základní průvodce webdesignem*. Brno: Zoner Press, 2004. 226 s. ISBN 80-86815-05-6.
- [15] MEYER, Eric A. *Eric Meyer o CSS: kompletní průvodce*. Brno: Zoner Press, 2007. 560 s. ISBN 978-80-86815-64-0.
- [16] MEYER, Eric. *Eric Meyer o CSS: pokračujeme s kaskádovými styly profesionálně!*. Brno: Zoner Press, 2005. 285 s. ISBN 80-86815-17-X.
- [17] SCHLOSSNAGLE, George. *Pokročilé programování v PHP 5*. Brno: Zoner Press, 2004. 640 s. ISBN 80-86815-14-5.
- [18] SIROVICH, Jamie, DARIE Christian. *SEO v PHP: Programujeme profesionálně*. Brno: Computer Press, 2008. 380 s. ISBN 978-80-251-2083-5
- [19] STANÍČEK, Petr. *CSS Kaskádové styly: Kompletní průvodce*. 2. vyd. Brno: Computer Press, 2003. 178 s. ISBN 80-7226-872-4.
- [20] ZAKAS, Nicholas, MCPEAK, Jeremy, FAWCETT, Joe. *AJAX: Profesionálně*. Brno: Zoner Press, 2007. 668 s. ISBN 978-80-86815-77-0.

Internetové zdroje

- [21] DUDEK, Jan. *CSS3 - kaskádové styly budoucnosti* [online]. 2005 [cit. 20. 4. 2010]. Dostupný z WWW: <http://interval.cz/clanky/css3-kaskadove-styly-budoucnosti/>.
- [22] HASSMAN, Martin. *Nové značky v HTML5* [online]. 2009 [cit. 6. 2. 2010]. Dostupný z WWW: <http://zdrojak.root.cz/clanky/nove-znacky-html5/>.
- [23] HUNT, Lachlan. *Seznámení s HTML 5* [online]. 2007 [cit. 6. 2. 2010]. Dostupný z WWW: <http://interval.cz/clanky/seznameni-s-html-5/>.
- [24] *Co je Silverlight?* [online]. 2010 [cit. 18. 2. 2010]. Dostupný z WWW: <http://interval.cz/clanky/silverlight-co-je-silverlight/>.

- [25] SMIČKA, Radim. *Optimalizace pro vyhledávače - SEO: jak zvýšit návštěvnost webu* [online]. 2004 [cit. 23. 10. 2009]. 120 s. Dostupný z WWW: <<http://seo.jasminka.cz/seo-kniha.pdf>>.
- [26] *Syntaxe v XHTML* [online]. 2009 [cit. 6. 2. 2010]. Dostupný z WWW: <<http://www.jakpsatweb.cz/html/xhtmll.html>>.
- [27] POŠVIC, Kamil. *XHTML 2 pravděpodobně míří k ledu, prioritou je HTML5* [online]. 2009 [cit. 6. 2. 2010]. Dostupný z WWW: <<http://www.root.cz/zpravicky/xhtmll-2-pravdepodobne-miri-k-ledu-prioritou-je-html-5/>>.
- [28] VŠETEČKA, Roman. *Internetové noviny New York Times zpoplatní pravidelné čtenáře* [online]. 2010 [cit. 20. 2. 2010]. Dostupný z WWW: <http://technet.idnes.cz/internetove-noviny-new-york-times-zpoplatni-pravidelne-ctenare-pxv-/sw_internet.asp?c=A100121_094536_sw_internet_vse>.

Knihovny

- (1) Moravská zemská knihovna v Brně, Kounicova 65a, 601 87 Brno, web: www.mzk.cz, e-mail: mzk@mzk.cz
- (2) Knihovna Jiřího Mahena v Brně, Kobližná 4, 601 50 Brno, web: www.kjm.cz, e-mail: kjm@kjm.cz
- (3) Knihovna Fakulty podnikatelské, Vysoké učení technické v Brně, Fakulta podnikatelská, Útvar vědeckých informací (knihovna) FP VUT, Kolejní 4, 612 00 Brno, web: <http://knihovna.fbm.vutbr.cz>, e-mail: knihovna@fbm.vutbr.cz
- (4) Knihovna Fakulty informačních technologií, Vysoké učení technické v Brně, Knihovna FIT VUT v Brně, Božetěchova 2, 612 66 Brno, web: <http://www.fit.vutbr.cz/lib/>, e-mail: knihovna@fit.vutbr.cz

Seznam použitých zkratek

AJAX	Asynchronous JavaScript and XML
B2B	Obchodní model Business to Business
CERN	Conseil Européen pour la Recherche Nucléaire
CSS	Cascading Style Sheet
DNS	Domain Name Service
DOM	Document Object Model
DTD	Document Type Definition
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IP	Internet Protocol
PHP	PHP: Hypertext Preprocessor, dříve Personal Home Page
SEM	Search Engine Marketing
SEO	Search Engine Optimization
SQL	Structured Query Language
SSL	Secure Sockets Layer
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WYSIWYG	What You See Is What You Get
WWW	World-Wide Web
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

Seznam obrázků

<i>Obrázek 1 – Protokol HTTP</i>	<i>14</i>
<i>Obrázek 2 – Model webové aplikace</i>	<i>22</i>
<i>Obrázek 3 – Grafický návrh internetové prezentace.....</i>	<i>37</i>
<i>Obrázek 4 – Tisková šablona webové prezentace.....</i>	<i>43</i>
<i>Obrázek 5 – Návrh relačního modelu</i>	<i>44</i>
<i>Obrázek 6 – Náhled formuláře sloužícího pro vložení nové nabídky.....</i>	<i>47</i>
<i>Obrázek 7 – Uživatelské rozhraní administrace webu.....</i>	<i>48</i>
<i>Obrázek 8 – Internetová prezentace společnosti HEIM Trade SE.....</i>	<i>54</i>

Seznam tabulek

<i>Tabulka 1 – Párové a nepárové tagy</i>	<i>17</i>
<i>Tabulka 2 – Kostra jazyka XHTML.....</i>	<i>17</i>
<i>Tabulka 3 – Základní tagy jazyka XHTML</i>	<i>18</i>
<i>Tabulka 4 – Příklad definice kaskádových stylů</i>	<i>20</i>
<i>Tabulka 5 – Přehled základních vlastností CSS.....</i>	<i>21</i>
<i>Tabulka 6 – Ukázka použití jazyka PHP: práce s databází.....</i>	<i>23</i>
<i>Tabulka 7 – Ukázka použití jazyka JavaScript: událost onClick.....</i>	<i>24</i>
<i>Tabulka 8 – Ukázka použití jazyka SQL: práce s databází.....</i>	<i>26</i>
<i>Tabulka 9 – Porovnání nákladů statického a dynamického webu</i>	<i>55</i>

Seznam příloh

<i>Příloha 1 – Založení databáze</i>	
<i>Příloha 2 – Naplnění databáze daty</i>	
<i>Příloha 3 – Zobrazení hlavní navigace</i>	
<i>Příloha 4 – Zobrazení podřízené navigace</i>	
<i>Příloha 5 – Zobrazení nabídky</i>	

Přílohy

Příloha 1 – Založení databáze

```
CREATE TABLE mesto (  
    id INT NOT NULL AUTO_INCREMENT,  
    mesto VARCHAR(30) NOT NULL UNIQUE,  
    poradi INT NOT NULL UNIQUE,  
    PRIMARY KEY (id)  
)TYPE=INNODB, COLLATE utf8_czech_ci;  
  
CREATE TABLE adresa (  
    id INT NOT NULL AUTO_INCREMENT,  
    id_mesto INT NOT NULL,  
    mestska_cast VARCHAR(30) NULL,  
    ulice VARCHAR(30) NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (id_mesto) REFERENCES mesto(id)  
)TYPE=INNODB, COLLATE utf8_czech_ci;  
  
CREATE TABLE stav (  
    id INT NOT NULL AUTO_INCREMENT,  
    hodnota VARCHAR(30) NOT NULL,  
    PRIMARY KEY (id)  
)TYPE=INNODB, COLLATE utf8_czech_ci;  
  
CREATE TABLE typ (  
    id INT NOT NULL AUTO_INCREMENT,  
    hodnota VARCHAR(30) NOT NULL UNIQUE,  
    poradi INT NOT NULL UNIQUE,  
    PRIMARY KEY (id)  
)TYPE=INNODB, COLLATE utf8_czech_ci;  
  
CREATE TABLE nabidka (  
    id INT NOT NULL AUTO_INCREMENT,  
    id_adresa INT NOT NULL,  
    id_stav INT NOT NULL,  
    id_typ INT NOT NULL,  
    poznamka VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (id_adresa) REFERENCES adresa(id),  
    FOREIGN KEY (id_stav) REFERENCES stav(id),  
    FOREIGN KEY (id_typ) REFERENCES typ(id)  
)TYPE=INNODB, COLLATE utf8_czech_ci;
```

Příloha 2 – Naplnění databáze daty

```
INSERT INTO mesto(mesto, poradi) VALUES ('Praha', 1);
INSERT INTO mesto(mesto, poradi) VALUES ('Brno', 2);
INSERT INTO mesto(mesto, poradi) VALUES ('Mníšek pod Brdy', 3);
INSERT INTO mesto(mesto, poradi) VALUES ('Čelákovice', 4);

INSERT INTO adresa(id_mesto, mestska_cast, ulice)
VALUES (1, 'Dolní Měchopupy', 'Ke Kablu 531/5');
INSERT INTO adresa(id_mesto, mestska_cast, ulice)
VALUES (1, 'Žižkov', 'Nám. Barikád 1134/3');
INSERT INTO adresa(id_mesto, ulice)
VALUES (2, 'Videňská 264/120 b');
INSERT INTO adresa(id_mesto, ulice)
VALUES (3, 'Mníšek pod Brdy 900');
INSERT INTO adresa(id_mesto, ulice)
VALUES (4, 'Křižíkova 270/17');

INSERT INTO stav(hodnota) VALUES ('Volné');
INSERT INTO stav(hodnota) VALUES ('Pronajato');

INSERT INTO typ(hodnota, poradi) VALUES ('Haly', 1);
INSERT INTO typ(hodnota, poradi) VALUES ('Plochy', 2);
INSERT INTO typ(hodnota, poradi) VALUES ('Kanceláře', 3);
```


Příloha 3 – Zobrazení hlavní navigace

```
<div id="menu">
<ul>

<?php
$mesta = vypis_mesta();
foreach($mesta as $m) {
    $style = "";

    if($frm_mesto == $m['id'])
        $style = " id='visit'";

    if(PEKNE_URL == 0) {
        /* Bez SEO optimalizace URL */
        $kategorie = neprazdna_kategorie($m['id']);
        echo "<li>";
        echo "<a href='index.php?mesto={$m['id']}&typ={$kategorie}'";
            {$style}>{$m['mesto']}</a>";
        echo "</li>\n";
    }
    else {
        /* Implementace SEO optimalizace URL */
        $mesto = priprav_text($m['mesto']);
        $kategorie = neprazdna_kategorie($m['id'], true);
        echo "<li>";
        echo "<a href='".DOMENA."/{$mesto}/{$kategorie}/'{$style}>
            {$m['mesto']}</a>";
        echo "</li>\n";
    }
}
?>

</ul>
</div>
```

Příloha 4 – Zobrazení podřízené navigace

```
<div id="sidebar">

<?
if($frm_mesto) {
    $typ = vypis_submenu();
    echo "<ul>";

    foreach($typ as $t) {
        $style = "";
        if($frm_typ == $t['id'])
            $style = " id='selected'";

        /* Jestli není v kategorii nabídka, položka zašedne */
        $pocet = pocet_nabidek($frm_mesto, $t['id']);
        if($pocet > 0) {
            if(PEKNE_URL == 0) {
                /* Bez SEO optimalizace URL */
                echo "<li>";
                echo "<a href='index.php?mesto={$frm_mesto}&typ={$t['id']}'"
                    "{$style}>{$t['hodnota']}</a>";
                echo "</li>\n";
            }
            else {
                /* Implementace SEO optimalizace URL */
                $mesto = priprav_text(ziskej_nazev_mesta($frm_mesto));
                $kategorie = priprav_text($t['hodnota']);
                echo "<li>";
                echo "<a href='".DOMENA."/{$mesto}/{$kategorie}/'"
                    "{$style}> {$t['hodnota']}</a>\n";
                echo "</li>\n";
            }
        }
        else
            echo "<li><div class='disenable'>{$t['hodnota']}</div></li>";
    }
    echo "</ul>";
}
else {
    echo "<img src='style/sidlo.jpg' alt='HEIM Trade SE' />";
}
```

Příloha 5 – Zobrazení nabídky

```
<div id="contentbar">

<?
if($frm_mesto) {
    $adresa = vypis_adresy($frm_mesto);
    foreach($adresa as $a) {
        $pocet = pocet_nabidek_na_adrese($a['id'],$frm_typ);
        if($pocet > 0) {
            echo "<h2>Areál {$a['mestska_cast']} {$a['ulice']}</h2>";
            $nabidka = vypis_nabidku($frm_mesto,$frm_typ,$a['id']);
            foreach($nabidka as $nab) {
                echo "<div class='nabidka'>";
                echo "<span class='poznamka'>{$nab['poznamka']}</span>";
                echo "<span class='stav'>{$nab['stav']}</span>";
                echo "</div>";
            }
        }
    }
}
else {
    echo "<p>Akciová společnost HEIM Trade byla založena v roce 2004.
Předmětem její činnosti je správa, revitalizace a pronájem
nemovitostí vlastních nebo v majetku třetích osob pro další
komerční využití, a to na území České a Slovenské republiky
(www.heimtrade.sk - stránky se připravují). Právě v návaznosti na
zahájení podnikatelské činnosti na Slovensku byla později změněna
právní forma společnosti na dnes aktuální evropskou
společnost (SE). Filozofií společnosti je poskytnout
podnikatelskému prostředí v lokalitách působení nejen
"střechu nad hlavou", ale i veškerý další nadstavbový servis,
který její zákazníci hledají...</p>";
}
?>

</div>
```