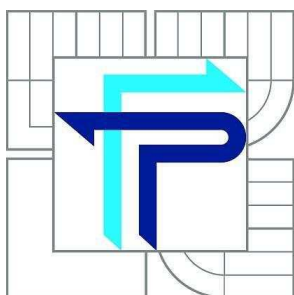


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH INTERNETOVÝCH STRÁNEK

WEBSITE DESIGN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JURAJ KOVÁČIK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. MILOŠ KOCH, CSc.

BRNO 2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Kováčik Juraj

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:

Website Design

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Abstrakt

Bakalárska práca sa zaoberá návrhom a tvorbou internetovej prezentácie pre spoločnosť Your Ideas, s.r.o. Úvodná časť je zameraná na teoretické poznatky z oblasti tvorby internetových stránok, ktoré následne aplikuje pri návrhu webu pre spomenutú firmu.

Abstract

This bachelor's thesis describes the designing and creation of the website produced for Your Ideas, s.r.o. The introduction is aimed at the theories behind website creation which have been subsequently applied when designing the site for the company.

Klíčové slová

Webová stránka, HTML, CSS, Javascript, jQuery, PHP, MySQL, Nette Framework, MVC

Keywords

Web page, HTML, CSS, Javascript, jQuery, PHP, MySQL, Nette Framework, MVC

KOVÁČIK, J. *Návrh internetových stránek*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2011. 54 s. Vedoucí bakalářské práce doc. Ing. Miloš Koch, CSc.

Čestné prehlásenie

Prehlasujem, že predložená bakalárska práca je pôvodná a spracoval som ju samostatne.
Prehlasujem, že citácia použitých prameňov je úplná, že som vo svojej práci neporušil autorské práva (v zmysle Zákona č. 121/2000 Zb., o práve autorskom a o právach súvisiacich s právom autorským)

V Brne, 27.05.2011

.....

Pod'akovanie

Týmito slovami si dovoľujem poďakovať pánu doc. Ing. Milošovi Kochovi, CSc., vedúcemu tejto práce za odbornú pomoc, vecné rady a pripomienky pri spracovaní práce.

Obsah

Úvod.....	9
1 Vymedzenie problému a cieľ práce	10
1.1 Vymedzenie problému.....	10
1.2 Cieľ práce	10
2 Teoretické východiska	11
2.1 Technológie tvorby webových stránok.....	11
2.1.1 HTML.....	11
2.1.2 CSS.....	12
2.1.3 Javascript.....	12
2.1.4 Ajax	13
2.1.5 PHP.....	14
2.1.6 MySQL.....	15
2.2 Nette framework	15
2.2.1 Adresárová štruktúra	16
2.2.2 MVC, MVP architektúra	17
2.2.3 Dibi.....	19
2.3 Bezpečnosť webových aplikácií	21
2.3.1 XSS.....	21
2.3.2 SQL injection	22
3 Analýza súčasného stavu	23
3.1 Základné informácie o firme	23
3.2 IT infraštruktúra spoločnosti.....	24
3.3 SWOT analýza.....	25
3.4 Požiadavky na webovú aplikáciu.....	26
3.5 Voľba technológii	27
4 Vlastný návrh riešenia	29
4.1 Databázový návrh	29
4.1.1 Obmedzenie práv.....	29
4.1.2 Jednotlivé databázy	31
4.2 Modely aplikácie	32

4.2.1	Model DataBase	33
4.2.2	Model Lang	35
4.2.3	Model DetectLang.....	36
4.2.4	Model Menu	37
4.2.5	Model Validate.....	37
4.2.6	Model Customer.....	37
4.2.7	Model SpracujSpravu.....	38
4.2.8	Model SpracujObjednavku.....	38
4.3	Prezentéry aplikácie.....	39
4.3.1	BasePresenter	39
4.3.2	ErrorPresenter.....	40
4.4	Šablóny	41
4.4.1	Šablóna layoutu	41
4.4.2	Šablóna objednávky	41
4.4.3	Šablóna pre emaily	42
4.4.4	Ostatné šablóny	43
4.5	Javascript	43
4.5.1	Konfigurátor objednávky	44
4.5.2	Úvodná stránka.....	45
4.6	Kaskádové štýly.....	46
4.7	Umiestnenie na server.....	46
4.7.1	Optimalizácia pre vyhľadávače.....	46
4.8	Ekonomické zhodnotenie.....	47
4.8.1	Náklady	47
4.8.2	Prínosy.....	49
5	Záver	50
Zoznam použitej literatúry		51
Knižné publikácie		51
Elektronické zdroje		51
Zoznam obrázkov		53
Zoznam tabuliek		54

Úvod

Internet od dôb svojho vzniku v 60. rokoch minulého storočia prešiel rozličnými transformáciami, ktoré z neho urobili celosvetovo a masovo využívané médium. Poskytuje živnú pôdu pre širokú paletu protokolov a služieb, ktoré zasahujú snáď do každého odvetvia ľudského života.

Jednou zo spomenutých služieb sú internetové stránky. V prvopočiatkoch bol obsah webových stránok jednoduchý, statický a previazaný prostredníctvom odkazov. Postupne vznikli jazyky interpretované na strane serveru, prehliadače návštevníkov začali podporovať prehrávanie animácii či videa.

Ľuďom sa tak dostal do rúk prostriedok, ktorý využívajú k svojej práci i zábave. Tento fakt si uvedomili aj firmy, ktoré vycítili príležitosť a cez webové prezentácie začali oslovovať zákazníkov. Súčasný stav zašiel tak ďaleko, že o firme, ktorá nemá na internete kvalitnú webovú prezentáciu možno povedať že neexistuje.

Nutnosť prezentovať sa prostredníctvom internetu si uvedomila aj spoločnosť Your Ideas, s.r.o. Samotnou tvorbou internetovej stránky pre spomenutú firmu ako aj technológiami k tvorbe stránok využívanými sa zaoberám vo svojej bakalárskej práci.

1 Vymedzenie problému a cieľa práce

1.1 Vymedzenie problému

Bakalárska práca sa zaoberá návrhom a realizáciou internetovej stránky pre spoločnosť Your Ideas, s.r.o, ktorá doposiaľ oficiálnou prezentáciou na internete nedisponovala. Samotná realizácia bude pozostávať predovšetkým z tvorby databázového modelu a programového kódu aplikácie.

1.2 Cieľ práce

Cieľom práce je navrhnúť a vytvoriť internetovú stránku, ktorá bude plnohodnotnou prezentáciou služieb a produktov spoločnosti Your Ideas, s.r.o, a súčasne bude slúžiť ako prostredník pre komunikáciu so zákazníkmi. Čiastkovým cieľom práce je tiež poskytnúť informácie a prehľad o súčasných technológiách, ktoré sú využívané k tvorbe moderných internetových stránok.

2 Teoretické východiska

V nasledujúcej časti poukážem na teoretické východiská, ktoré sú potrebné k tvorbe webových stránok. Predstavím technológie používané k tvorbe moderných, dynamických prezentácií. Neopomeniem ani súčasné bezpečnostné nástrahy, ktoré číhajú na dnešné webové stránky.

2.1 Technológie tvorby webových stránok

K tvorbe moderných stránok sa používajú rozličné technológie. Poukážem na tie z nich, ktoré sú považované za základ tvorby webových prezentácií, konkrétne HTML, CSS, javascript. Ďalej predstavím jazyk PHP a MySQL, ktoré sú základom mnohých dynamicky generovaných stránok.

2.1.1 HTML

HTML je akronym od HyperText Markup Language. Je to jazyk, ktorý popisuje formátovanie dokumentu a jeho obsah. Skladá sa zo značiek, medzi ktorými sú uzavreté časti textu a jedná sa o akýsi štandardný typ dokumentu na Internete. (2)

Užívateľ k tomu, aby mohol pristupovať k html dokumentu umiestneného na strane servera musí použiť software – najčastejšie prehliadač. Komunikácia medzi prehliadačom a serverom prebieha prevažne pomocou HTTP protokolu (HyperText Transfer Protocol). Po zadaní adresy vzdialeného servera užívateľom sa naviaže HTTP spojenie. Server vráti užívateľovi dokument a následne ho prehliadač zobrazí. (2)

HTML jazyk však nebol navrhnutý pre tvorbu sofistikovaných užívateľsky prívetivých rozhraní či interaktívnych webových aplikácií. Až postupom času sa vyvinuli technológie akými sú napríklad javascript, AJAX alebo PHP, ktoré umožnili posunúť hranice webových stránok smerom dopredu.

2.1.2 CSS

Kaskádové štýly (v skratke CSS) sa snažia o úplné oddelenie formy (definície štýlov) od samotného obsahu webovej prezentácie. Oddelenie týchto dvoch vrstiev zjednodušuje prácu s obsahom dokumentu, zvyšuje prehľadnosť a čitateľnosť dokumentu. Umožňujú tvorbu správne semantického zdrojového kódu, ktorý zvyšuje prístupnosť stránok. (1)

Na nasledujúcom príklade sa pokúsím ilustrovať rozdiel medzi už zastaraným „inline“ štýlovaním elementu a používaním štýlov CSS.

Inline štýl:

```
<p style="font-size: 12pt; color: #aaaccc">Nutne menit v dokumente</p>
```

Použitie kaskádového štýlu:

```
<p>Staci zmenit styl v css. Nie je to prehladnejsie? </p>
```

Vďaka oddeleniu vrstiev je kompletná zmena vzhľadu webovej prezentácie otázkou zmeny jediného css súboru, bez nutnosti prechádzať všetky súbory obsahujúce definíciu vzhľadu vnútri HTML značiek. (1)

Medzi ďalšie, nezanedbateľné výhody patrí možnosť definície štýlov pre rôzne zariadenia akými sú napríklad PDA, tlačiareň, monitor či digitálny projektor. (1)

2.1.3 Javascript

JavaScript (skrátene JS) je interpretovaný jazyk, ktorý sa vykonáva na strane klienta. Vkladá sa ako čistý text do HTML stránok. Je podporovaný väčšinou prehliadačov na veľa platformách bez nutnosti inštalácie prídavného modulu do prehliadača. (2)

Pôvodne bol navrhnutý pre jednoduché skripty, ktoré nemali viac ako pár riadkov. Príkladom je validácia vstupov z formuláru. Avšak takéto použitie prináša

so sebou aj vážne bezpečnostné riziká, nakoľko sa jedná o jazyk interpretovaný na strane klienta.

V súčasnosti je využívaný k tvorbe moderných webových aplikácií u ktorých je trend priblížiť sa bežným desktopovým aplikáciám. (2)

Sila JS okrem iného spočíva v práci s DOM, pomocou ktorého môže manipulovať s rodičovským HTML dokumentom. Dokáže pridávať elementy či meniť ich vlastnosti a to bez nutnosti obnovovania stránky. (2)

2.1.4 Ajax

Skratka AJAX je odvodená od „Asynchronous JavaScript and XML“. Inými slovami povedané, jedná sa o súhrn technológií, ktoré umožňujú vývoj interaktívnych webových aplikácií a to bez potreby znovunačítania stránky. (3)

Písmeno A, teda asynchrónny. Značí, že prehliadač užívateľa je schopný odpoveď od servera spracovať až v okamžiku, keď k nemu príde. Nemusí tak čakať na dáta zo strany servera, čím by dochádzalo k zablokovaniu behu aplikácie. Odpoveď od serveru je spracovaná v okamihu, keď server odošle dáta klientovi. (3)

JavaScript, teda písmeno „J“ v skratke AJAX je technológia, ktorá riadi úkony v prehliadači. Vykonáva spojenie so serverom a spracováva dáta, ktoré server vráti. Stará sa aj o následné zobrazenie či iné využitie. (3)

Jazyk XML predstavuje jeden zo spôsobov, akými môže server komunikovať s klientom - vracat' odpoveď. Ďalšia z možností je napríklad čistý text. (3)

2.1.5 PHP

PHP je platforma, ktorú používa viac ako tretina webových serverov na celej planéte. Jedná sa o skriptovací jazyk, ktorý je spracovávaný na strane serveru a disponuje širokou funkcionalitou. Syntax nesie znaky jazykov C, Java a Perl a je nenáročná na učenie. (3), (12)

Kód spracovávaný na strane serveru poskytuje oveľa širšiu funkcionalitu ako iba odosielanie HTML dokumentov klientovi. Umožňuje vykonávanie dotazov nad databázou, počítanie rozličných výpočtov a podobne. Priebeh komunikácie medzi klientom a serverom je nasledovný. Klient vloží do svojho prehliadača adresu stránky. Na serveri sa vykoná skript, ktorý sa napríklad pripojí k databáze a vyberie zoznam zamestnancov spoločnosti spolu s kontaktmi. Server užívateľovi ako odpoveď na jeho požiadavku pošle HTML výstup, ktorý bude rovnaký, ako keby užívateľ pristupoval k statickému html súboru. (2)

V porovnaní s javascriptom, ktorý sa spracováva na strane klienta, pri PHP užívateľ nevidí jadro skriptu, nevie, akým spôsobom skript získal dáta. Vidí iba ich reprezentáciu vo svojom prehliadači. (12)

PHP podporuje prácu s protokolmi pre prístup k emailom (IMAP, POP3, SMTP), umožňuje využívať protokol pre ukladanie a prístup k dátam na adresárovom serveri (LDAP) a mnoho iných. Dokáže pracovať s XML dokumentmi, podporuje najfrekventovanejšie web servery a databázy. (12)

Spomenutá rozsiahla funkcionalita je zaobalená do jazyka s prívetivou syntaxou, ktorá umožňuje funkcionálny aj objektový prístup. Nemožno sa preto diviť, že stále viac spoločností spolieha vo svojich aplikáciách práve na jazyk PHP. (3)

2.1.6 MySQL

MySQL je relačný databázový server, ktorý je schopný prevádzky na viac ako 20 platformách. Je implementovaný v mnohých vývojových jazykoch ako napríklad PHP, Perl či Python. (4)

Jedná sa o jeden z najpopulárnejších relačných open source SQL management systémov (RDBMS). Svoju popularitu si získal vďaka vlastnostiam, menovite: cena, stabilita, jednoduchosť, funkčná výbava a modulárnosť. (4)

Cena je jeden z najľahšie porovnateľných faktorov. Spoločnosť Oracle, ktorá stojí za vývojom tejto databázy, vydáva jednotlivé verzie pod licenciou GPL. Výhodou takéhoto prístupu je fakt, že tým zamedzuje zložitej cenovej politike, ktorá by sa odvíjala napríklad od počtu jadier procesora prípadne počtu pripojených užívateľov, ako je tomu u konkurenčných produktov. (4)

2.2 Nette framework

Pojmom framework označujeme knižnicu resp. sadu knižníc, ktorá vývojarovi uľahčuje písanie aplikácie a zjednocuje spracovávanie najčastejšie riešených úloh. Jednoduchým príkladom funkcionality frameworku môže byť obsluha formulárov alebo obsluha pripojenia k rôznym typom databáz. (7)

Medzi najznámejšie frameworky patrí Zend nasledovaný Symfony či CakePHP frameworkom. Od roku 2008 je na scéne aj počín českého autora s názvom Nette. Tento framework je v našich končinách čoraz viac obľúbený a to hlavne vďaka českej dokumentácii a komunitnej podpore zo strany „domácich“ programátorov. (7)

Nette framework je objektovo orientovaný framework napísaný v PHP 5. Medzi najvýznamnejšie vlastnosti patrí:

- Bezpečnosť (automatické ošetrovanie vstupu a následná eliminácia bezpečnostných hrozieb ako napríklad Cross-site scripting),

- rýchlosť,
- rozšíriteľnosť pomocou pluginov,
- šablónovací systém,
- MVC model,
- na strane hostingu si vystačí s podporou PHP 5.2+ a .htaccess. (10)

2.2.1 Adresárová štruktúra

Medzi ďalšiu, vyššie nespomenutú výhodu patrí voľnosť vo vytváraní adresárovej štruktúry. Odporúčaná adresárová štruktúra v stabilnej verzii frameworku 0.9.7b je nasledovná: (9)



Obr. 2.1: Adresárová štruktúra (Zdroj: vlastný)

Na prvý pohľad je zrejмый obsah jednotlivých adresárov. V zložke „/app“ je užívateľom napísaná aplikácia vrátane modelov, prezentérov a šablón. Priečinok „/www“ (resp document_root) obsahuje kaskádové štýly, obrázky prípadne javascript. Zložka „/libs“ pozostáva zo samotného jadra Nette frameworku a ďalších knižníc, napríklad pre prácu s databázou.

Konfigurácia správaní aplikácie vrátane definovania pripojenia k databáze sa vykonáva v /app/config.ini. Tento súbor je chránený iba pomocou .htaccess. Z uvedeného vyplýva nutnosť podpory .htaccess na strane webhostingu.

2.2.2 MVC, MVP architektúra

Cieľom MVC architektúry je oddeliť aplikačnú a prezenčnú logiku webovej aplikácie. Aplikácia je rozdelená do samostatných logických celkov. Výhodou takéhoto rozdelenia je ľahko udržateľný zdrojový kód a to aj v prípade, že sa na vývoji podieľa niekoľko ľudí. (5)

Aplikácia sa z pohľadu MVC architektúry rozdeľuje na nasledovné logické časti:

- Model,
- view,
- controller.

Nette framework využíva architektúru MVP, ktorá je postavená na myšlienke MVC. Princíp fungovania je rovnaký, jedná sa iba o akési nahradenie pojmu „controller“ pojmom „presenter“. (10)

Model

Model obsahuje aplikačnú logiku (business logiku) aplikácie. Zaisťuje prístup k dátam a prácu s nimi. Nevie o pohľadoch a presenteroch ktoré s ním pracujú, mal by byť od nich nezávislý. (5)

View

Zobrazuje dáta získané od modelu vo vhodnej podobe. Nestará sa o spôsob, akým model dáta získal a odkiaľ ich získal. V prípade, že dôjde k zmene ukladania dát z textového súboru do relačnej databázy, view (pohľad) sa o tom vôbec nedozvie. (10)

Controller

Controller (presenter) sa stará o spracovanie užívateľských akcií, zaisťuje zmeny v pohľade a modely. Má na starosti celkové previazanie funkčnosti aplikácie. (10)

Celú logiku MVP najlepšie vystihne nasledujúci príklad, ktorý zároveň slúži aj pre porovnanie s funkcionálnym prístupom k PHP bez využitia šablónovacieho systému.

```
<?php
echo "<h1>Zoznam produktov</h1>";

$sql = mysql_query("select produkt, cena from produkty order by
cena");
while ($data = mysql_fetch_array($sql))
{
    echo $data['produkt'] . " za cenu " . $data['cena'];
    echo "<br/>";
}
?>
```

Zdrojový kód 2.1: Ukážka bežného PHP kódu (Zdroj: vlastný)

```
<php
class Produkt extends DibiRow{
    public function zoznamProduktov()
    {
        $sql = dibi::query("select produkt, cena from produkty ");
        return $sql->fetch();
    }
}
```

Zdrojový kód 2.2: Ukážka modelu (Zdroj: vlastný)

```
{if isset($produkty)}
    {foreach $produkty as $produkt}
        {$produkt->produkt} za cenu {$produkt->cena}
    {/foreach}
{/else}
<span class="error">Ziadne produkty sa v DB nenachadzaju</span>
{/if}
```

Zdrojový kód 2.3: Ukážka pohľadu (Zdroj: vlastný)

```
<?php
class Produkty extends BasePresenter{
    public function actionZobraz()
    {
        $model = new Produkt();
        $zoznamProduktov = $model -> zoznamProduktov();
    }
}
```

```
} $this->template->produkty = $zoznamProduktov;  
}
```

Zdrojový kód 2.4: Ukážka prezentéru (Zdroj: vlastný)

Na prvý pohľad sa môže zdať spôsob písania aplikácie prostredníctvom MVP frameworku zdĺhavý. S pribúdajúcimi riadkami kódu celá architektúra dostáva zmysel a aplikácia je prehľadnejšia.

2.2.3 Dibi

Dôležitou súčasťou modernej, dynamickej webovej aplikácie je prístup k dátam, ktoré sú uložené v databáze. Databázových systémov je niekoľko a každý z nich využíva vlastnú syntax a rozdielne funkcie v rámci jazyka PHP. Zatiaľ čo k výberu dát prostredníctvom MySQL sa využíva funkcia *mysql_query()* u Postgresql je to *pg_query()*. (12)

Dibi je databázový layer, ktorý obsahuje ovládače pre 8 rozličných databáz. Medzi najzaujímavejšie z pohľadu vývoja bežných webových aplikácií určite patrí podpora MySQL, MS SQL, PostgreSQL, a SQLite. Zaobaluje funkcie pre prácu s databázami do intuitívneho rozhrania a zjednocuje prístup. Jeho funkcionality tým ani zďaleka nekončí. (8)

Základnými črtami dibi sú:

- Jednoduchosť,
- podpora modifikátorov v dotazoch,
- ľahká integrácia do Nette frameworku.

Jednoduchosť

SQL dotazy sa v dibi zapisujú ako séria parametrov. Cieľom takéhoto riešenia bola predovšetkým prehľadnosť a eliminácia chýb zo strany programátora. (6)

```
$cena = "250";  
  
dibi::query('  
    SELECT [produkt], [cena]  
    FROM [produkty]  
    WHERE [cena] <%i', $cena  
);
```

Zdrojový kód 2.5: Využitie dibi modelu (Zdroj: vlastný)

Príklad demonštruje výber produktov s cenou nižšou ako *\$cena*. Dotaz je ľahko čitateľný, prehľadný. Za zmienku stojí zápis podmienky, v ktorej je použitý modifikátor.

Modifikátory

Namiesto očakávanej hodnoty sa v SQL dotaze používa modifikátor, ktorý určuje dátový typ premennej. Premenné sa uvádzajú až za samotný dotaz, tak ako to znázorňuje predchádzajúci príklad. Ak nie je dátový typ uvedený, dibi sa pokúsi o jeho autodetekciu. (6)

Nasledujúca tabuľka zobrazuje prehľad najčastejšie používaných modifikátorov. Nejedná sa teda o zoznam všetkých dostupných modifikátorov.

Tab. 2.1: Prehľad základných modifikátorov dibi (Zdroj: vlastný)

Modifikátor	Význam
%s	Textový reťazec
%b	Logická hodnota
%i %u	Celé číslo
%f	Desatinné číslo
%sql	SQL dotaz

Integrácia do Nette frameworku

Bezproblémová a rýchla. Za vývojom dibi a Nette frameworku totiž stojí jedna a tá istá osoba, David Grudl.

Celý proces integrácie je možné rozdeliť do troch krokov, ktorými sú:

1. Nahratie dibi do zložky */libs*
2. Definícia prístupu k databáze v */app/config.ini*
3. Inicializácia pripojenia v */app/bootstrap.php*

Následne, po inicializácii dibi layeru je v Nette k dispozícii statická trieda „*dibi::*“, ktorá obsluhuje celú prácu s databázou. (8)

2.3 Bezpečnosť webových aplikácií

Bezpečnosť je jedna z kľúčových vlastností každej webovej aplikácie a treba na ňu brať zreteľ už počas vývoja. Medzi najväčšie bezpečnostné hrozby súvisiace s tvorbou web stránok za využitia PHP a databázy možno označiť XSS a SQL injection.

2.3.1 XSS

XSS prípadne Cross-site scripting je útok, ktorý využíva neošetrené vstupy aplikácie a skripty interpretované na strane užívateľa. Útočník dokáže do zobrazovanej stránky vložiť svoj vlastný spustiteľný kód, ktorý je schopný získať citlivé údaje o návštevníkovi, napríklad z cookies uložených v prehliadači. (10)

Jednoduchým príkladom môže byť nasledujúci kód vložený do formulára a odoslaný na spracovanie.

```
javascript:alert('xss')
```

V prípade, že by sa vyššie uvedený kód po odoslaní spracoval a užívateľovi sa zobrazila hláška s textom „xss“, pravdepodobne by sa jednalo o stránku, ktorá je náchylná k tomuto typu útoku.

Z uvedeného je zrejmé, že voči XSS sa dá brániť jedine dôkladným ošetrovaním všetkých textových reťazcov, ktoré sú zobrazované užívateľovi. Nette framework vo svojom šablónovacom systéme všetky výstupy ošetruje a aplikácia napísaná s jeho využitím by nemala byť zraniteľná XSS útokom. (10)

2.3.2 SQL injection

Bezpečnostná chyba, ktorá útočníkovi umožňuje pracovať s dátami na SQL serveri, ku ktorým by za bežných okolností nemal mať prístup. Vo svete webových aplikácií dochádza k vykonaniu takéhoto typu útoku v prípade neošetrenia vstupných údajov, ktoré sú od užívateľa získané napríklad pomocou formuláru alebo prostredníctvom premenných obsiahnutých v URL adrese stránky. (11)

Absencia validácie vstupných dát môže mať fatálne následky v podobe úniku citlivých informácií. Žiadna spoločnosť nemá záujem na zverejnení informácií o plate svojich zamestnancov prípadne zoznamu zákazníkov. Problém SQL injection čiastočne rieši dibi databázový layer, ktorý ma implementovanú ochranu voči tomuto typu útoku

3 Analýza súčasného stavu

Základom pre správny návrh webovej aplikácie je dôkladná znalosť firmy, pre ktorú je stránka realizovaná. V nasledujúcej kapitole bude predstavená spoločnosť Your Ideas, s.r.o. a jej hlavné produkty. Tieto informácie budú podkladom pre definovanie požiadaviek na aplikáciu a následnú realizáciu.

3.1 Základné informácie o firme

Názov spoločnosti: Your Ideas, s.r.o.
Sídlo: Sokolská 2518/48
960 01, Zvolen
Slovenská Republika
IČO: 44869274
Deň zápisu do OR: 16.07.2009
Základné imanie: 5100 EUR



Obr. 3.1: Logo spoločnosti Your Ideas, s.r.o. (Zdroj: vlastný)

Your Ideas, s.r.o. je mladá spoločnosť, ktorá bola založená v polovici roku 2009 so sídlom vo Zvolene. Má dvoch vlastníkov, pričom každému z nich patrí 50%-ný podiel. Predmetom podnikania je oblasť informačných technológií. Primárne sa zameriava na poskytovanie virtuálnych serverov (VPS) a ich správu. V druhom rade to je tvorba webových prezentácií, informačných systémov podľa požiadaviek klienta a webhosting takýchto riešení

Virtuálne servery

Poskytovanie virtuálnych serverov je sofistikovaná služba, určená prevažne administrátorom systémov. Technológia virtualizácie pre firmu predstavuje možnosť ako mať na jednom fyzickom serveri spustených niekoľko virtuálnych. Z prostredia mimo lokálnu sieť nie je možné rozoznať že sa jedná o virtuálny server.

Príkladom zákazníka takejto služby je administrátor / web programátor, ktorý potrebuje mať k dispozícii server s neustálou prevádzkou. Server, ktorý si nakonfiguroval podľa svojich potrieb – počnúc počtom pridelených IP adries, cez kapacitu diskov, veľkosť pamäte RAM a operačným systémom končiac. Z pohľadu zákazníka odpadá starosť o hardware, jeho fyzické umiestnenie v serverovni a pripojenie na sieť Internet. Naopak, dostáva do rúk produkt, ktorý je „modulárny“ a dokáže v reálnom čase reflektovať na jeho narastajúce potreby.

Spoločnosť zabezpečuje ako doplnkovú službu administráciu takýchto serverov na základe dohody so zákazníkom.

Webhosting a tvorba aplikácií

Možnosť umiestnenia webovej aplikácie na server je ponúkaná v rámci súčasti komplexného balíka služieb prevažne zákazníkom, pre ktorých sa daná aplikácia programuje. Zákazníci, ktorým sa aplikácia vyvíja a následne hostuje na serveroch spoločnosti tak majú zaručené, že prostredie serveru je nakonfigurované pre správny chod ich aplikácie. Cieľom firmy nie je poskytovanie hostingu masám.

3.2 IT infraštruktúra spoločnosti

Firemná infraštruktúra pozostáva z vlastných serverov značky SUN, ktoré má spoločnosť umiestnené v dátovom centre v Brne. Do siete Internet sú pripojené prostredníctvom linky o kapacite 1 Gbps . Samozrejmosťou je záložné napájanie prostredníctvom UPS a dieselového agregátu, ktoré zabezpečuje serverhostingová spoločnosť.

Každý server je osadený operačným systémom GNU/Linux, distribúciou CentOS. Prostredie „fyzických“ serverov je upravené, používa sa jadro (kernel), ktoré umožňuje virtualizáciu prostredníctvom Xen. Takto vytvorené prostredie umožňuje využívanie virtuálnych serverov nielen ako produktu pre zákazníkov. Spoločnosť ich používa k oddeleniu ostatných služieb ktoré poskytuje (webhosting), prípadne pre vývoj vlastných projektov. Výsledkom je infraštruktúra, ktorá je škálovateľná a zaručuje vysokú dostupnosť.

Bezpečnosť dát je zabezpečená ich redundantným uložením na viacerých diskoch, ktoré sú zapojené do RAID. Zálohy dát užívateľov sú ukladané na geograficky oddelenom mieste.

3.3 SWOT analýza

Swot analýza hodnotí silné a slabé stránky spoločnosti, príležitosti a hrozby, ktorým čelí na trhu. Umožňuje vyhodnotiť fungovanie firmy, zamerať sa na potlačenie slabých stránok a rozvíjanie príležitostí.

Silné stránky:

- Kvalita poskytovaných služieb,
- prístup k zákazníkom, otvorenosť,
- nízke prevádzkové náklady,
- poloha (firma je všade tam, kde je internet).

Slabé stránky:

- Zadlženosť,
- malý počet stabilných zákazníkov.

Príležitosti:

- Expanzia na český trh,
- rozšírenie portfólia služieb,

- zlepšenie komunikácie so zákazníkmi.

Hrozby:

- Strata hlavných zákazníkov,
- silná konkurencia.

3.4 Požiadavky na webovú aplikáciu

Napriek temer dvojročnej existencii spoločnosti, až do dnešných dní nedisponovala žiadnou webovou prezentáciou. Bolo potrebné správne definovať požiadavky, ktorým v ceste nestáli žiadne technické prekážky ako nutnosť pamätať na spätnú kompatibilitu s prechádzajúcou prezentáciou, prípadne databázou.

Požiadavky vychádzali z plánov spoločnosti, medzi najdôležitejšie patrí expanzia na český trh, otvorenosť voči zákazníkovi a komunikácia s nim. K technickým požiadavkám možno zaradiť SEO optimalizáciu a bezpečnosť. Taktiež bolo nutné pamätať na potrebu budúceho prepojenia databázy s aplikáciou pre zamestnancov spoločnosti.

Jazykové mutácie

Príbuznosť českého a slovenského jazyka dávajú do rúk triumf firmám pôsobiacim na trhoch týchto štátov. Firma podnikajúca na území SR nepotrebuje prezentáciu v českom jazyku aby dokázala osloviť zákazníkov z ČR. Avšak Your Ideas, s.r.o. chce ísť bližšie k návštevníkom svojej stránky a predpokladá, že pre budúceho zákazníka je príjemnejšie ak môže webovú prezentáciu vzhliadnuť vo svojom materinskom jazyku.

Kontaktné formuláre

Zákazník by mal mať k dispozícii formuláre, prostredníctvom ktorých môže kontaktovať jednotlivých pracovníkov spoločnosti. V prípade, že má záujem o objednávku virtuálneho serveru, mal by mu k tomu slúžiť formulár, ktorý by

umožňoval nadefinovať všetky potrebné parametre a automaticky by stanovil cenu zvolenej konfigurácie.

Napojenie na informačný systém spoločnosti

V blízkej dobe spoločnosť predpokladá spojenie webovej prezentácie a dát, ktoré sa v nej používajú s internou aplikáciou pre svojich zamestnancov. Na tento fakt je vhodné pamätať predovšetkým počas návrhu databázovej štruktúry, ktorú bude prezentácia využívať.

Optimalizácia

Prezentácia by mala byť vhodne štruktúrovaná, optimalizovaná pre rozličné druhy prehliadačov a optimalizovaná z hľadiska SEO. Spomenuté typy optimalizácie predpokladajú validný zdrojový kód, používanie kaskádových štýlov, meta značiek a prívetivú, ľahko zapamätateľnú štruktúru URL.

Rozšíriteľnosť

Firemná prezentácia by mala byť v budúcnosti ľahko rozšíriteľná. Pridávanie nových položiek v menu či oddelení spoločností nemôže byť problémom. Bez ohľadu na to, či bude prebiehať prostredníctvom administratívneho rozhrania alebo ručným vložením údajov do databázy.

Bezpečnosť

Aplikácia techník, ktoré zamedzia útokom ako SQL injection, XSS by mala byť samozrejmosťou. V neposlednom rade je potreba aplikáciu ochrániť pred robotmi, ktorí využívajú kontaktné formuláre k odosielaniu nevyžiadanej pošty (tzv. spam).

3.5 Voľba technológií

Voľba technológií vychádza z analýzy požiadaviek na prezentáciu a je ovplyvnená softwarovým vybavením spoločnosti. Limitujúcim faktorom pre výber programovacieho jazyka a databázy je požiadavka, aby aplikácia bola plne funkčná na firemných serveroch.

Kostra aplikácie je tvorená Nette frameworkom využívajúcim jazyk PHP. Zvyšná aplikačná časť je taktiež písaná v tomto jazyku. Dáta sú uložené v MySQL relačnej databáze, ku ktorej sprostredkúva prístup dibi databázový layer.

V časti ktorá sa stará o prezentáciu dát užívateľovi sa používa CSS a javascript. Využíva sa tiež javascript framework jQuery, ktorý zabezpečuje oživenie stránky dynamickými prvkami.

4 Vlastný návrh riešenia

Nasledujúca kapitola poukazuje na technickú realizáciu webovej prezentácie pre spoločnosť Your Ideas, s.r.o. V samotnom návrhu riešenia sa zameriam na podstatné časti aplikácie, ktoré vytvoria ucelený pohľad na to, akým spôsobom funguje. Obsahuje predovšetkým návrh databáz a programovanie jednotlivých častí aplikácie. Grafický návrh pre stránku bol vytvorený externým spolupracovníkom spoločnosti, v práci sa ním preto zaoberať nebudem.

4.1 Databázový návrh

Pri návrhu databázy bolo nutné zohľadniť potrebu budúceho rozširovania a prepojenia dát s vnútorným informačným systémom. Ako najpriateľnejšie riešenie, ktoré by spĺňalo tieto požiadavky sa javila separácia databáz do logických celkov. Vzniklo osem oddelených databáz:

- cities
- customers
- departments
- languages
- orders
- services
- tickets
- web

4.1.1 Obmedzenie práv

Výhodou takéhoto riešenia je predovšetkým bezpečnosť a prehľadnosť. Každá databáza má definovaného vlastného užívateľa, ktorý prístupuje k danej databáze. Navyše, užívateľ má nad tabuľkami databázy presne stanovené operácie, ktoré môže vykonávať. Pre vhodný príklad poslúžia databázy web a orders.

Užívateľ definovaný v MySQL môže pristupovať iba k databáze web a nad všetkými tabuľkami dokáže vykonávať jedine príkaz SELECT. Ak by útočník našiel na stránke napríklad SQL injection zraniteľnosť a získal by prístup k tomuto účtu, jediné údaje ku ktorým sa dostane sú údaje o položkách menu a popisoch odkazov v rámci stránky. Navyše, žiadnu z týchto položiek nedokáže pozmeniť či vymazať.

O niečo horšia situácia nastane v prípade „odcudzenia“ prístupu do databázy s objednávkami. Keďže zákazník môže objednávať prostredníctvom formuláru, počas spracovania objednávky musí dôjsť k zápisu do databázy. MySQL užívateľ pracujúci s touto databázou potrebuje mať nad niektorými tabuľkami definované oprávnenie INSERT. Útočník tak získa možnosť zápisu do týchto tabuliek, ale nemôže z nich údaje vymazať. Nestane sa, že by z databázy odstránil všetky objednávky klientov, maximálne vyvolá poplach v podobe falošných objednávok.

V jazyku MySQL sa nová databáza vytvára pomocou príkazu:

```
CREATE DATABASE `yi_cities` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Takto vytvorená databáza nemá prideleného žiadneho užívateľa, ktorý s ňou môže manipulovať. Pre vytvorenie užívateľa a definíciu jeho práv k jednotlivým tabuľkám sa používa:

```
GRANT SELECT,INSERT ON yi_cities.cities TO 'user_cities'@'localhost' IDENTIFIED BY 'heslo';
```

Pre správne definovanie prístupových práv je potreba ujasniť si všetky procesy, ktoré budú v prezentácii prebiehať. Môže nastať situácia, že formulár, ktorý potrebuje zapísať dáta do databázy sa k nej pripája prostredníctvom užívateľa bez práva zápisu. V prípade takto oddelených prístupov do databázy je zrejmé, že pre firemnú prezentáciu budú definované iné MySQL účty ako pre internú aplikáciu pre zamestnancov.

4.1.2 Jednotlivé databázy

Databáza cities

Obsahuje 2 tabuľky. Tabuľka štátov je vyplnená anglickými názvami štátov sveta. Užívateľ prostredníctvom objednávkového formuláru zadáva svoju adresu vrátane mesta a PSČ, ktoré je uložené v druhej tabuľke s mestami. Mestá sú prepojené so štátmi prostredníctvom cudzích kľúčov.

Databáza customers

Ako vypovedá názov, databáza obsahuje informácie o zákazníkoch. Jedná sa predovšetkým o kontaktné informácie akými sú telefónne čísla, emaily, kontaktné osoby, adresa spoločnosti a pod. Databáza je využívaná objednávkovým formulárom. Existujúci zákazník sa v procese objednávky môže identifikovať prostredníctvom emailu a nemusí opätovne vkladať svoje kontaktné údaje. Nový zákazník musí vložiť požadované údaje, ktoré sú rozlíšené podľa typu (právnej subjektivity) zákazníka.

Databáza departments

Databáza oddelení obsahuje iba jedinú tabuľku – s názvom a popisom oddelenia, emailovou adresou a telefónnym kontaktom. Využíva sa pri kontaktnom formulári, v ktorom užívateľ môže zvoliť oddelenie, ktoré chce kontaktovať.

Databáza languages

Každá jazyková mutácia stránky obsahuje vlastnú tabuľku. Štruktúry týchto tabuliek sú totožné, obsahujú názov jazykovej konštanty a jej interpretáciu v príslušnom jazyku.

Databáza orders

Všetky objednávky vytvorené prostredníctvom objednávkového formuláru sú uložené práve v tejto databáze. Obsahuje informácie o požadovanej konfigurácii serverov, čase odoslania požiadavky a podobne.

Databáza services

Vzhľadom na predpokladané rozširovanie portfólia služieb spoločnosti, je vhodné mať jednotlivé kategórie produktov uložené v samostatnej databáze. Táto databáza rieši vzťahy medzi produktom a možnosťami volieb, ktoré jednotlivý produkt má. Pre ilustráciu majme dva produkty: virtuálny server a webhosting. V prípade VPS sú možnosti volieb veľkosť HDD a RAM, veľkosť CPU či počet IP adres. Pre webhosting treba definovať množstvo diskového priestoru prípadne počet emailových schránok.

Databáza tickets

Databáza tiketov, alebo tiež správ, ktoré zákazníci poslali prostredníctvom kontaktného formuláru. Správy budú zasielané aj prostredníctvom emailu. Zákazníkovi príde automatická odpoveď a príslušnému kontaktovanému oddeleniu sa odošle zákazníkova požiadavka. Archivácia tiketov v databáze je výhodná pre ľahšiu identifikáciu a komunikáciu s klientom v prípade problémov a tiež z hľadiska rastúcich potrieb spoločnosti. Databáza môže byť použitá ako podklad pre automatický systém pridelovania tiketov zamestnancom.

Databáza web

Obsahom tejto databázy sú položky menu a definície vzťahov medzi nimi (zaradenie do kategórii, poradie v akom dôjde k vykresleniu a pod.). Ďalej sú tu uložené nastavenia objednávkového formuláru (periódy platby a zľavy pri zvolenej perióde). Databáza predpokladá naviazanie na databázu jazykov čím odpadá potreba akokoľvek upravovať túto databázu v prípade rozširovania stránky o ďalšiu jazykovú mutáciu.

4.2 Modely aplikácie

Modely predstavujú srdce, čiže logiku celej aplikácie. Nasledujúca kapitola bude preto zameraná na stručný popis modelov, ktoré obsahuje prezentácia. Ponúkne pohľad na funkcionality jednotlivých modelov.

4.2.1 Model DataBase

Spôsob, akým Nette framework ukladá údaje o pripojení k databáze nepovažujem za najšťastnejší. Definícia užívateľa, hesla a databázy je umiestnená v samostatnej sekcii súboru config.ini v ktorom sa nachádzajú aj ostatné nastavenia aplikácie. Keďže sa jedná o súbor s príponou *ini*, webový server ho neinterpretuje cez PHP ale jeho obsah môže priamo poskytnúť užívateľovi. Jediná ochrana súboru je možná správnym nastavením *.htaccess*.

Vzhľadom na spomenutý fakt bolo vhodné vytvoriť samostatný model, ktorý uchováva údaje o prístupoch k databázam a súčasne dokáže vytvoriť pripojenie prostredníctvom dibi. Za týmto účelom vznikol model DataBase.php s nasledujúcim kódom.

```
class DataBase extends DibiRow
{
    const DB_CITIES      = "DB_CITIES";
    private static $list_of_db = array
    (
        "DB_CITIES" => array(
            'driver'      => 'mysql',
            'host'        => 'localhost',
            'username'    => 'user_web',
            'password'    => 'heslo',
            'database'    => 'yi_cities',
            'charset'     => 'utf8',
            'lazy'        => 'TRUE' )
    );

    public static function connect($database = NULL)
    {
        if ($database !== NULL)
        {
            if(self::existConfig($database))
            {
                dibi::connect(self::$list_of_db[$database], $database);
            }
        }
    }

    private static function existConfig($database)
    {
        if ($database !== NULL)
        {
            foreach (self::$list_of_db as $db => $val)
            {
                if ($db == $database)
                {
                    return TRUE;
                }
            }
        }
    }
}
```

```

        break;
    }
}
else
{
    return FALSE;
}
}
}

```

Zdrojový kód 4.1: Obsah modelu DataBase.php(Zdroj: vlastný)

Ako najvhodnejšie sa javilo ukladanie údajov o databázach do pola. Indexy pola sú totožné s názvami položiek v config.ini čím je zachovaná kompatibilita. K vytvoreniu pripojenia ku konkrétnej databáze tak možno použiť metódu connect(), ktorú obsahuje knižnica dibi. Prvým parametrom metódy connect sú prihlasovacie údaje do databázy, druhým je názov pripojenia. V prípade spojení s viacerými databázami možno jednotlivé pripojenia ľahko identifikovať práve pomocou tohto názvu. Kódom uvedeným nižšie možno v ktoromkoľvek mieste aplikácie iniciovať pripojenie k potrebnej databáze.

```
DataBase::connect(DataBase::DB_CITIES);
```

Takto vytvorené pripojenie možno využiť priamo v dibi, ktorý obsahuje metódu getConnection(). Parametrom je názov spojenia, ktorý odkazuje na názov konštanty vo vnútri modelu DataBase.

```
$main_db = dibi::getConnection(DataBase::DB_CITIES);
$main_db->query('select * from cities');
```

Pripájať sa k databáze vždy pred volaním dotazu je neprehľadné a v budúcnosti môže spôsobiť problémy. Volanie všetkých pripojení k databázam bolo preto umiestnené do súboru bootstrap.php. Pre prístup k databáze v kóde modelov už preto nie je potrebné vytváranie pripojenia, stačí sa naň odkázať pomocou názvu.

Pre ilustráciu uvediem pôvodné riešenie, ktoré je v nette frameworku štandardné. Prístupy k databáze sú v config.ini definované nasledovne:

```
; pripojenie k db
database.driver = mysql
database.database = todolist
database.charset = utf8
database.lazy = TRUE
database.host = localhost
database.username = produkcnilogin
database.password = produkcniheslo
```

Zdrojový kód 4.2: Obsah časti súboru config.ini (Zdroj: vlastný)

V bootstrap-e je pripojenie k databáze realizované nižšie uvedeným príkazom. Možno si všimnúť časť kódu „getConfig('database')“, ktorá v konfiguračnom súbore hľadá prefix „database“. Pomocou prefixov by bolo možné oddeliť pripojenia k rozdielnym databázam.

```
dibi::connect(Environment::getConfig('database'));
```

Po takto inicializovanom pripojení je prístup k databáze možný z ľubovoľného miesta aplikácie to pomocou volania dibi nasledovným spôsobom.

```
dibi::query('select * from zak');
```

4.2.2 Model Lang

Keďže sa definovala požiadavka na prezentáciu s rozličnými jazykovými mutáciami, bolo potrebné určiť spôsob s akým sa bude k jednotlivým jazykom pristupovať. Pre obsluhu práce s jazykmi na stránke vznikol model Lang.php, ktorý obsahuje metódu getText().

Táto metóda má povinný parameter, ktorým je názov jazykovej konštanty. Druhým, nepovinným parametrom je požadovaná jazyková mutácia v podobe skratky daného jazyka. V praxi to znamená, že každá šablóna, prípadne informačná hláška miesto čitateľného textu obsahuje volanie tejto triedy s jazykovou konštantou. Nasledujúci príklad vyberie z databázy obsah konštanty 'INFO_ORDER_SENT' v slovenskom jazyku, ktorý môže znieť „Vaša objednávka bola úspešne odoslaná“.

```
Lang::getText('INFO_ORDER_SENT', 'sk');
```

Model je volaný v BasePresenter.php, od ktorého dedia všetky ostatné prezentery. Táto vlastnosť umožňuje ponechať druhý parameter triedy `getText()` nezadaný. V takomto prípade sa nastaví na súčasne využívaný jazyk, ktorý sa zistí z URL stránky.

4.2.3 Model DetectLang

V súvislosti s jazykovými mutáciami bolo potrebné zabezpečiť ešte jednu vlastnosť prezentácie a to aby sa sama prepínala do verzie, ktorú bude návštevník pravdepodobne požadovať. Bežne sa môžeme stretnúť so situáciou, keď česky hovoriaci zákazník príde na stránku prostredníctvom odkazu zo slovenskej stránky. Odkaz pochopiteľne ukazuje na slovenskú jazykovú mutáciu. Takýto stav nie je želaný, model ho rieši automatickým presmerovaním na správnu jazykovú verziu stránky.

Možností, akými sa pokúsiť zistiť skutočný jazyk, ktorým návštevník rozpráva je niekoľko. Najjednoduchší spôsob je získať nastavenie z prehliadača. Táto informácia sa prenáša v http hlavičke a prístup k nej je v PHP nie je žiaden problém. Nejedná sa však o úplne presný spôsob detekcie ale rozhodne je najľahšie implementovateľný. Pre zaujímavosť spomeniem rozšírenie `geoip`, prostredníctvom ktorého možno určiť približnú polohu návštevníka na základe IP adresy a tým aj odhadnúť jazyk. Model `DetectLang` takúto funkcionality neobsahuje, disponuje iba detekciou na základe nastavení prehliadača.

Cookies

Údaje o nastavenej jazykovej verzii sú uchovávané v cookies. Ak užívateľ už raz stránku navštívil, došlo k uloženiu jazykovej verzii do cookies. V prípade, že opätovne navštívi stránku, bude presmerovaný na jazyk, ktorý mal naposledy nastavený.

4.2.4 Model Menu

Ako vypovedá samotný názov, úlohou modelu je starať sa o zobrazenie menu. K uloženiu informácii o menu sa využíva stromová štruktúra, ktorá je vykresľovaná pomocou depth-first search (skrátene DFS) algoritmu. Metódu modelu je možné volať pomocou parametrov, ktoré určujú štart a stop hĺbku pre prehľadávanie menu. Volanie bez parametru umožní zobrazenie všetkých položiek menu.

4.2.5 Model Validate

Každý vstup od užívateľa by mal prejsť kontrolou voči maske, ktorá určuje množinu povolených znakov pre ten-ktorý prvok formuláru. Niektoré prvky je možné validovať prostredníctvom funkcií, ktoré obsahuje Nette framework (napríklad emailovú adresu). Pre zložitejšiu validáciu sú potrebné vlastné funkcie. K tomuto účelu slúži model Validate. Obsahuje statické metódy, ktoré sú volané v prezenteroch pri vytváraní formulárov. Príkladom je nasledujúca metóda umožňujúca validáciu čísla popisného. Pokiaľ zadaný reťazec vyhovuje regulárnemu výrazu, metóda vráti true, v opačnom prípade false.

```
/**
 * Validacia cisla popisneho
 * moze obsahovat velke, male (ziadna diakritika!), bodka, pomlcka,
 * medzera (\s), lomka (\/), cisla
 */
public static function myfValCp($item, $arg)
{
    $validCp = "/^[a-zA-Z0-9.-\s\/]{1, ".self::MAX_CP_LENGTH."}$/";
    return preg_match($validCp, $item->getValue()) ? TRUE : FALSE;
}
```

Zdrojový kód 4.3: Príklad validačnej funkcie (Zdroj: vlastný)

4.2.6 Model Customer

Práca so zákazníkom je umiestnená v modeli Customer. Umožňuje predovšetkým vkladanie nového zákazníka do databázy a overovanie už existujúceho na základe zadaného emailu.

4.2.7 Model SpracujSpravu

Model zabezpečuje obsluhu správ, ktoré sú poslané prostredníctvom kontaktného formuláru. Odoslaná správa sa uloží do databázy tickets s údajmi o odosielateľovi (IP adresa, informácie o prehliadači, jazyková verzia stránky a pod). Na základe zvoleného oddelenia sa odošle prostredníctvom emailu osobe, ktorá je k tomuto oddeleniu priradená. Užívateľovi je odoslaný automatický email s informáciou, že na jeho požiadavke sa pracuje.

Obsahuje ochranu voči robotom, ktorí by chceli formulár použiť k zasielaniu nevyžiadanej pošty. Pomocou jednoduchého generátora sú generované príklady na sčítanie a odčítanie jednociferných čísel. Správna odpoveď užívateľa je potrebná k tomu, aby sa odoslala užívateľom napísaná správa.

4.2.8 Model SpracujObjednavku

Užívateľom definovaná objednávka virtuálneho serveru sa odosiela na spracovanie modelu SpracujObjednavku. Prebieha tu uloženie údajov do databázy orders, kontaktovanie zákazníka prostredníctvom automaticky generovaného emailu a odoslanie emailu na obchodné oddelenie spoločnosti s detailnou špecifikáciou objednávky.

Súčasne obsahuje detekciu prípadnej predchádzajúcej objednávky. Celý proces objednania pozostáva z niekoľkých krokov, samostatných formulárov. Môže nastať situácia ako napríklad pád prehliadača alebo strata internetového pripojenia a zákazník nebude schopný dokončiť objednávku. Pri odoslaní „čiastkových“ formulárov sú údaje z nich uložené v sessions a v prípade vyššie spomenutého scenáru sa formulár nastaví podľa údajov prečítaných zo sessions.

4.3 Prezentéry aplikácie

Názvy prezentérov odzrkadľujú štruktúru URL stránky. V prípade, že je pomenovaný napríklad `ServicesPresenter`, je štandardne prístupný z adresy `http://stranka.sk/services`. Webová prezentácia spoločnosti obsahuje nižšie vymenované prezentery, z ktorých možno vyčítať hlavné delenie stránky (položky menu).

- `BasePresenter.php`
- `ContactPresenter.php`
- `ErrorPresenter.php`
- `HomePresenter.php`
- `ReferencesPresenter.php`
- `ServersPresenter.php`
- `SolutionsPresenter.php`

Nasledujúcich pár riadkov venujem opisu dvom z nich, konkrétne `Error` a `Base` prezentéru. Ostatnými sa zapodievať nebudem, ich obsahom je obsluha vykresľovania formulárov, prípadne naplňajú šablóny údajmi.

4.3.1 BasePresenter

Jedná sa o základný prezentér od ktorého dedia všetky ostatné prezentéry. Obsahuje inicializáciu modelov `Lang`, `DetectLang` a `Menu`. Prvý spomenutý slúži na preklad textov vyskytujúcich sa na stránke.

Na základe hodnôt, ktoré vráti model `DetectLang`, prezentér presmeruje URL aplikácie na požadovanú jazykovú mutáciu. V prípade slovenskej je to `http://your-ideas.eu/sk` pre českú verziu `http://your-ideas.eu/cz`.

Volanie menu na tomto mieste umožňuje vloženie potrebných parametrov do metódy modelu. Môže sa jednať o súčasnú jazykovú verziu stránky alebo aktuálny prezentér.

Prístup k menu je potom možný v každom prezentéry bez nutnosti vytvárania opätovnej inštancie modelu.

4.3.2 ErrorPresenter

Stará sa o spracovávanie a zobrazovanie stavových kódov protokolu HTTP. Najčastejšie sa možno stretnúť so stavovým kódom 404, ktorý užívateľa informuje o tom, že požadovaný dokument nebol na serveri nájdený. Tento prezentér obsahuje obsluhu stavových kódov a ku konkrétnemu stavovému kódu priraduje šablónu, ktorá sa zobrazí ako reakcia na vzniknutú udalosť, napríklad na nenájdenný súbor. Výhodou je vlastná, prispôbena chybová hláška, ktorá nemusí byť definovaná v .htaccess ako je to zvykom u webových prezentácií.



Obr. 4.1: Chybová hláška na stránke (Zdroj: vlastný)

Obrázok poukazuje na spôsob, akým funguje zobrazovanie hlášky o neexistujúcom spore na serveri. Design je prebratý od hlavnej stránky. Užívateľ bez straty pohodlia môže kliknúť na ľubovoľnú položku menu, nemusí sa vracieť na úvodnú stránku prezentácie.

4.4 Šablóny

Oddeľujú aplikačnú a prezenčnú vrstvu. Každá akcia prislúchajúca prezentéru má definovanú vlastnú šablónu. Aplikácia zachováva štandardný prístup k šablónam, ktorý je v tvare `/app/templates/NazovPrezenteru/akcia.phtml`. V prípade výslednej adresy `http://your-ideas.eu/sk/servers/order/` sa akciou rozumie „order“ a šablóna je umiestnená v adresári `/app/templates/Servers/order.phtml`

4.4.1 Šablóna layoutu

Špeciálny typ šablóny, ktorý obsahuje definíciu html hlavičky, nalinkované kaskádové štýly pre rôzne zobrazovacie zariadenia a javascript, ktorý sa používa v celej aplikácii. Vytvára tak istú základnú grafickú kostru prezentácie. Je umiestnená v `/app/templates/@layout.phtml` a je potrebné vložiť ju prostredníctvom odkazu do každej šablón, ktorá ju má používať.

Z hľadiska SEO a celkovej prehľadnosti prezentácie by bolo nesprávne používať jeden html tag obsahujúci titul stránky a meta tagy s popisom a kľúčovými slovami naprieč celou aplikáciou bez ohľadu na obsah konkrétnej podstránky. Za týmto účelom sú použité bloky, ktoré slúžia ako modifikátory globálne nastavených vlastností v šablóne layoutu. Každá podstránka má definované vlastné kľúčové slová a popis, ktorý odpovedá jej skutočnému obsahu.

4.4.2 Šablóna objednávky

Zákazník má k dispozícii interaktívny formulár, v ktorom môže presne definovať požiadavky na svoj VPS (veľkosť RAM, CPU, HDD a pod.). Na základe zvolených hodnôt sa vypočítava celková cena serveru. Nemusí kontaktovať nikoho zo spoločnosti aby zistil cenu zvolenej konfigurácie. Práve k tomuto účelu slúži šablóna objednávky. Vykresľuje formulár a dynamicky vypočítava cenu. K výpočtu ceny slúži funkcia písaná v javascript-e. Vždy, keď užívateľ zmení konfiguráciu požadovaného serveru, dôjde k prepočtu ceny.

V ďalšom kroku objednávky sú od užívateľa požadované fakturačné údaje, ktoré sa líšia podľa právnej formy zákazníka. Fyzická osoba nepodnikateľ zadáva iné údaje ako právnická osoba. Na základe voľby právnej formy sú zákazníkovi zobrazené iba tie položky, ktoré sú od neho požadované. Spomenutú funkcionálnosť taktiež zabezpečuje javascript.

Z hľadiska vykresľovania je šablóna zaujímavá tým, že nevykresľuje formulár pomocou príkazu {widget nazovFormularu} ale postupne, po častiach. Takýto prístup umožňuje plnú kontrolu predovšetkým nad grafickou stránkou formulára.

4.4.3 Šablóna pre emaily

Automaticky generované emaily, ktoré sa zákazníkovi posielajú prostredníctvom kontaktného formuláru sú v HTML formáte, ktorého štruktúru obsahujú šablóny pre to určené. Verzia emailu v čistom texte bez HTML sa generuje automaticky, čo je obrovská výhoda a nie je preto potrebné pamätať na zákazníkov, ktorých emailový klient html emaily nepodporuje.

Problém s takto generovanými emailami nastáva v prípade, že obsahujú grafiku. Z nezistených príčin niektoré emailové servery správy s obrázkom väčším ako pár kb označili za spam. Grafické prvky použité v emailoch bolo preto potrebné obmedziť na minimum.



Vážený zákazník, Vážená zákazníčka Juraj Kovacik,

oznamujeme Vám, že sme obrdžali Vašu e-mailovú správu. Naši zamestnanci Vás budú čím skôr kontaktovať.

S úctou

Technická podpora, Your Ideas, s.r.o.

(E-mailová správa bola poslaná automaticky)



your ideas, our solutions

Obr. 4.2: HTML email odosielaný zákazníkom (Zdroj: vlastný)

Na predchádzajúcom obrázku možno vidieť html email, ktorý sa odošle zákazníkovi po kontaktovaní prostredníctvom kontaktného formuláru.

4.4.4 Ostatné šablóny

Doposiaľ spomenuté šablóny boli modifikované nie štandardným spôsobom. Zvyšné šablóny používané v prezentácii nie sú z hľadiska rozboru až také zaujímavé. Obsahujú bežné obsluhy formulárov prípadne vykreslenia textov a využívajú spomenuté modifikátory, ktorými sú nastavované kľúčové slová, popisy stránky (description) a „title tag“ zvlášť pre každú akciu.

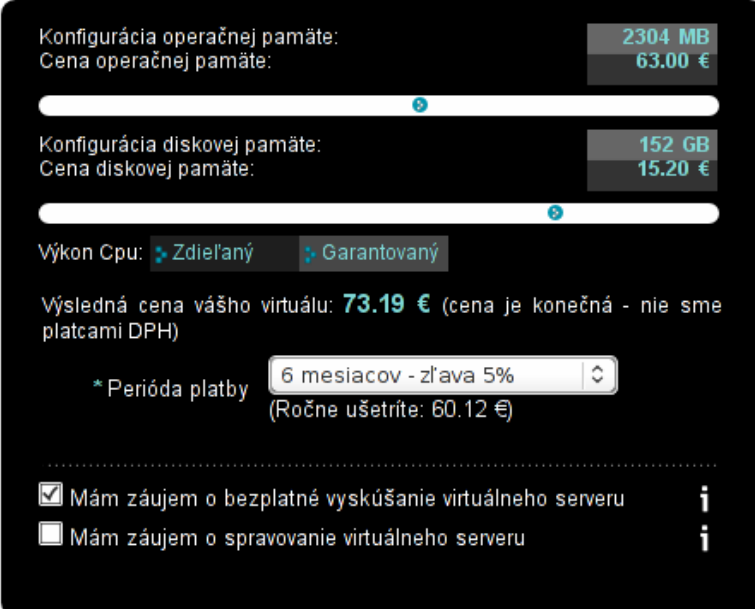
4.5 Javascript

Na stránke je použitý ako interaktívny prvok, ktorý spríjemňuje a najmä uľahčuje užívateľovi prácu. V objednávkovom formulári sa využíva ako praktický prvok predovšetkým k výpočtu ceny. Naproti tomu na úvodnej stránke pôsobí skôr esteticky. Celá funkcionálnosť je umiestnená v jednom súbore, ktorý k svojmu fungovaniu potrebuje prídavnú open source knižnicu jQuery.

4.5.1 Konfigurátor objednávky

Prvou dôležitou časťou, ktorá je realizovaná prostredníctvom javascript a ktorú obsahuje objednávkový formulár je kontrola opustenia stránky. Užívateľ môže omylom zavrieť prehliadač alebo kliknúť na odkaz, ktorým odíde zo stránky a tým stratí údaje doposiaľ vyplnené a ešte neodoslané na server. Pre zamedzenie takejto udalosti slúži funkcia kontrola_opustenia(), ktorá zisťuje, či užívateľ zadal akékoľvek informácie do formuláru. V prípade, že formulár už nie je prázdny a je vyplnené aspoň jedno pole a užívateľ sa rozhodol opustiť stránku, vyvolá sa dialógové okno s informáciou. Opustenie stránky je možné až po potvrdení prostredníctvom tlačidla OK.

Ďalšou, nemenej podstatnou časťou o ktorú sa stará javascript je výpočet ceny na základe zvolenej konfigurácie jednotlivých parametrov. Objednávkový formulár obsahuje niekoľko možností, ktoré si môže užívateľ upraviť podľa svojich potrieb. Za týmto účelom slúžia posuvníky, ktoré pôsobia aj ako obmedzovače. Neumožňujú výber mimo rozsah, ktorý majú nastavený. Vždy po zmene príslušného parametru dôjde k prepočtu aktuálnej ceny prostredníctvom funkcie v javascript-e a užívateľ okamžite uvidí celkovú cenu jeho zvolenej objednávky.



Konfigurácia operačnej pamäte: 2304 MB
Cena operačnej pamäte: 63.00 €

Konfigurácia diskovej pamäte: 152 GB
Cena diskovej pamäte: 15.20 €

Výkon Cpu: Zdieľaný Garantovaný

Výsledná cena vášho virtuálu: **73.19 €** (cena je konečná - nie sme platcami DPH)

* Perióda platby:
(Ročne ušetríte: 60.12 €)

Mám záujem o bezplatné vyskúšanie virtuálneho serveru i
 Mám záujem o spravovanie virtuálneho serveru i

Obr. 4.3: Konfigurátor VPS (Zdroj: vlastný)

O vykresľovanie posuvníka sa stará knižnica jQuery, ktorá zároveň obsluhuje udalosti vyvolané prostredníctvom zmeny posuvníka. Taktiež umožňuje jednoduchý prístup k DOM čo je využité pri skrývaní a zobrazovaní možnosti voľby garantovaného výkonu. Zobrazovanie prvkov formuláru sa využíva aj v ďalšej časti objednávky. V prípade, že užívateľ zvolí možnosť „registrovaný zákazník“, skryje sa formulár s kontaktnými údajmi a miesto neho sa zobrazí formulár na zadanie emailu.

4.5.2 Úvodná stránka

Hneď v úvode je javascript použitý k efektnému prechádzaniu medzi jednotlivými podstránkami bez nutnosti obnovenia celej stránky. Podstránky, možno ich označiť „slajdy“, prezentujú hlavné produkty spoločnosti. K dosiahnutiu výsledného efektu je opäť použitá knižnica jQuery a jej funkcie fadeout() a fadeIn(). Rovnaký efekt je využitý aj v pätičke stránky na prepínanie medzi kontaktmi spoločnosti a informáciami o používaných technológiách.



Obr. 4.4: Úvodná stránka (Zdroj: vlastný)

Stránka na treťom úvodnom „slajde“ je zobrazená na predošlom obrázku. Prezentuje službu outsourcingu IT, prepnutie na ďalšiu službu je možné prostredníctvom šípok na

bokoch prezentácie, prípadne v okne s informáciou o službe. Prechody medzi stránkami má na starosti vyššie spomenutý javascript.

4.6 Kaskádové štýly

Definícia vizuálnej stránky prezentácie je obsiahnutá v CSS súboroch. Žiadna šablóna (okrem šablóny pre emaily) neobsahuje inline štýl, čím je zabezpečená jednoduchá zmena vlastnosti akéhokoľvek prvku na stránke. Prostredníctvom parametru „media“ v html hlavičke sú definované štýly pre jednotlivé zobrazovacie zariadenia. Pre projekciu na monitory, televízory a projektory sa využíva súbor screen.css. Jeho obsahom je definícia všetkých tried a identifikátorov, ktoré sú priradené k html elementom naprieč celou prezentáciou.

Druhým súborom obsahujúcim kaskádové štýly je print.css. Pozostáva z definície typu písma a neobsahuje štýly elementov. Ako napovedá samotný názov, aplikuje sa v prípade, že užívateľ požaduje tlač stránky. Výstupom je textová verzia stránky, ktorá neobsahuje žiadne obrázky.

4.7 Umiestnenie na server

Predchádzajúcich niekoľko kapitol ponúklo pohľad na spôsob, akým aplikácia pracuje. Aby návrh a následné programovanie malo zmysel, bolo potrebné prezentáciu umiestniť na servery spoločnosti a sprístupniť ju tak zákazníkom. Po vytvorení databáz a nastavení prístupov v DataBase.php sa stala stránka plne funkčná a použiteľná. Dostupná je z <http://your-ideas.eu>.

4.7.1 Optimalizácia pre vyhľadávače

Návštevníkmi prezentácie nie sú len ľudia ale aj roboti, ktorí indexujú obsah. Následne ho v získanej podobe ponúkajú užívateľom, ktorí ho vyhľadávajú prostredníctvom

vyhľadávačov. Pre jednoduchší prístup vyhľadávačov sa v „www“ adresári na serveri vytvorili súbory robots.txt a sitemap.xml

Prvý spomenutý umožňuje prístup všetkým robotom k celému obsahu stránky, okrem častí, ktoré sú označené ako disallow. Zakázané sú napríklad Všeobecné obchodné podmienky, ktoré sú umiestnené ako pdf súbor v koreni prezentácie. Ďalšími zakázanými položkami sú jednotlivé kroky objednávkového formuláru.

Prostredníctvom súboru sitemap je vytvorená mapa webu. Informuje vyhľadávacieho robota o všetkých stránkach prezentácie a tým umožňuje indexáciu celého obsahu. Nakoľko sa jedná o prezentáciu, ktorá je statická, k vytvoreniu bolo možné použiť on-line generátor máp.

4.8 Ekonomické zhodnotenie

Nasledujúca kapitola predstavuje ekonomický pohľad na navrhnuté riešenie. Vyčísľuje náklady a prínosy spojené s realizáciou.

4.8.1 Náklady

Náklady spojené s tvorbou webovej prezentácie možno rozdeliť do 3 kategórií, menovite grafický design, programovanie a návrh aplikácie a ostatné.

Grafický design bol vytvorený v rámci grafického manuálu externým spolupracovníkom. Nie je preto možné určiť koľko percent z celkovej čiastky 300 Eur prislúcha samotnej tvorbe vizuálnej časti prezentácie.

Návrh štruktúry aplikácie, celkovej logiky a databázy predstavoval časovú investíciu 5 hodín. Vyčíslené prostredníctvom peňazí, náklady spojené s návrhom tvoria 35 Eur pri hodinovej mzde 7 Eur.

Časovo najrozsiahlejšou činnosťou bolo programovanie aplikácie. Celkový čas strávený tvorbou kódu predstavoval 50 hodín za mzdu 7 Eur/hodina. Fakt, že k tvorbe bol použitý výhradne „slobodný“ software robí sumu 350 Eur konečnú z pohľadu nákladov na programovanie.

Dodatočnými nákladmi boli úpravy zdrojového kódu aplikácie po spustení prezentácie. Oprava chýb, ktoré sa neprejavili počas vývoja prípadne testovania stáli spoločnosť ďalších 35 Eur.

Položkou síce zanedbateľnou z hľadiska financií ale nie z hľadiska dôležitosti je kúpa domény. Ako registrátor bolo zvolené Forpsi, ktoré ponúkalo doménu s koncovkou „.eu“ za necelých 10 Eur.

Nasledujúca tabuľka zobrazuje prehľad nákladov rozdelených podľa vyššie spomenutých kategórii.

Tab. 4.1: Náklady na realizáciu prezentácie (Zdroj: vlastný)

Názov položky	Čiastka v Eur
Grafický design	
Tvorba grafického manuálu	300
Programovanie a návrh	
Návrh aplikácie	35
Programovanie	350
Ostatné	
Oprava chýb	35
Kúpa domény	10
CELKOM	730

Celkové náklady firmy na vytvorenie webovej prezentácie predstavovali 750 Eur vrátane DPH. Po prihliadnutí na fakt, že aplikácia je tvorená na mieru pre spoločnosť, jedná sa o prijateľnú výšku investície.

4.8.2 Prínosy

Ekonomickým prínosom predstaveného riešenia je zvýšenie príjmov spoločnosti. Bezprostredne po spustení prezentácie na internete firma zaznamenala nárast počtu nových zákazníkov. Tí z nich, ktorí kontaktovali spoločnosť prostredníctvom webovej stránky vytvorili v priebehu dvoch mesiacov od spustenia prezentácie zisk, ktorý plne pokryl náklady spoločnosti na tvorbu stránky.

Nezanedbateľným je tiež prínos v podobe ušetreného času. Znížila sa agenda, ktorá bola vynútená zlou informovanosťou budúcich zákazníkov a absenciou webovej prezentácie. Zamestnanci môžu investovať získaný čas do vývoja a skvalitňovania služieb.

5 Záver

V rámci svojej bakalárskej práce som navrhol, vytvoril a umiestnil na internet webovú prezentáciu spoločnosti Your Ideas, s.r.o. Stránka vytvorila platformu pre jednoduchšiu komunikáciu so zákazníkmi, predovšetkým vďaka interaktívnej objednávke služieb. Dôležitá je tiež jej úloha ako prezentácie produktov, ktoré spoločnosť ponúka.

Bezprostredne po zverejnení webovej prezentácie na internete firma zaznamenala citeľný nárast nových zákazníkov a tým aj tržieb. S odstupom času od spustenia spoločnosť hodnotí prínos internetovej stránky veľmi pozitívne.

Zoznam použitej literatúry

Knižné publikácie

- [1] CROFT, J., LLOYD, I., RUBIN, D. *Mistrovství v CSS Pokročilé techniky pro webové designéry a vývojáře*. Brno: Computer Press, 2007. 409 s. ISBN : 978-80-251-1707-7.
- [2] DARIE, C. a kol. *AJAX a PHP Tvoříme interaktivní webové aplikace PROFESIONÁLNĚ*. Brno: Zoner Press, 2006. 320 s. ISBN : 80-86815-47-1.
- [3] GUTMANS, A., BAKKEN, S.S., RETHANS, D. *Mistrovství v PHP 5*. 2. vyd. Brno: Computer Press, 2007. 655 s. ISBN : 978-80-251-1519-0.
- [4] SCHWARTZ, B. a kol. *MySQL profesionálně optimalizace pro vysoký výkon*. Brno: Zoner Press, 2009. 712 s. ISBN : 978-80-7413-035-9.

Elektronické zdroje

- [5] BERNARD, Borek. *Úvod do architektury MVC*. [online]. Zdroják. 7. 5. 2009 [cit. 2011-05-09]. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/uvod-do-architektury-mvc/>>. ISSN : 1803-5620.
- [6] GRUDL, David. *Dibi - pokrokový databázový layer* [online]. PhpFashion. 26. 5. 2006 [cit. 2011-05-09]. Dostupné z WWW: <<http://phpfashion.com/dibi-pokrokovy-databazovy-layer>>.
- [7] GRUDL, David. *Nette Framework: zvyšte svoji produktivitu*. [online]. Zdroják. 10. 3. 2009 [cit. 2011-05-10]. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/nette-framework-zvyste-svoji-produktivitu/>>. ISSN : 1803-5620.
- [8] Nette Foundation. *Dibi is Database Abstraction Library for PHP 5*. [online]. [cit. 2011-05-08]. Dostupné z WWW: <<http://dibiphp.com/cs/>>.
- [9] Nette Foundation. *Dokumentace Nette Framework* [online]. [cit. 2011-05-09]. Dostupné z WWW: <<http://doc.nette.org/cs/>>.

- [10] Nette Foundation. *Hlavní přednosti Nette Framework* [online]. [cit. 2011-05-09]. Dostupné z WWW: <<http://nette.org/cs/hlavni-prednosti>>.
- [11] *SQL Injection (Full Paper)*. [online]. 5. 11. 2009 [cit. 2011-05-06]. Dostupné z WWW: <<http://security-portal.cz/clanky/sql-injection-full-paper>>.
- [12] The PHP Group. *PHP Manual* [online]. [cit. 2011-05-06]. Dostupné z WWW: <<http://www.php.net/manual/en/>>.

Zoznam obrázkov

Obr. 2.1: Adresárová štruktúra (Zdroj: vlastný)	16
Obr. 3.1: Logo spoločnosti Your Ideas, s.r.o. (Zdroj: vlastný).....	23
Obr. 4.1: Chybová hláška na stránke (Zdroj: vlastný)	40
Obr. 4.2: HTML email odosielaný zákazníkom (Zdroj: vlastný)	43
Obr. 4.3: Konfigurátor VPS (Zdroj: vlastný).....	44
Obr. 4.4: Úvodná stránka (Zdroj: vlastný).....	45

Zoznam tabuliek

Tab. 5.1: Prehľad základných modifikátorov dibi (Zdroj: vlastný)	20
Tab. 5.1: Náklady na realizáciu prezentácie (Zdroj: vlastný)	48