

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/B/2014/2268277197

VÝVOJ WEBOVEJ APLIKÁCIE
V JAZYKU PHP
Bakalárska práca

2014

Martin Miškovič

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

VÝVOJ WEBOVEJ APLIKÁCIE
V JAZYKU PHP
Bakalárska práca

Študijný program: Hospodárska informatika

Študijný odbor: 6292 Hospodárska informatika

Školiace pracovisko: Katedra aplikovanej informatiky

Vedúci záverečnej práce: Ing. Peter Škurla, PhD.

Bratislava 2014

Martin Miškovič

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

Dátum: 26. 04. 2014, Bratislava

.....

Martin Miškovič

ABSTRAKT

MIŠKOVIČ, Martin: *Vývoj webovej aplikácie v jazyku PHP*. – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra aplikovanej informatiky. – Vedúci záverečnej práce: Ing. Peter Škurla, PhD. – Bratislava: FHI, 2014, 39 s.

Cieľom mojej bakalárskej práce je vytvorenie webovej aplikácie pomocou open source riešení, akými sú programovací jazyk PHP, databázový server MySQL a knižnice jQuery, a opísať proces jej vývoja. Hlavný účel aplikácie je vytvorenie verejne prístupného miesta na internete na účely inzercie ponúk pre dobrovoľníkov. Pri vývoji aplikácie sa práca sústreďuje na trojvrstvový aplikačný model. Práca sleduje jednotlivé kroky vývoja webovej aplikácie a popisuje fázy návrhu, programovania a testovania aplikácie. Práca je rozdelená do 4 kapitol. Prvá kapitola je venovaná úvodu do problematiky webových aplikácií a popisuje teoretické základy technológií, ktoré boli na vývoj aplikácie použité. Na konci prvej kapitoly sa práca venuje bezpečnosti webových aplikácií, ktorá býva často zanedbávaná. Druhá a tretia kapitola obsahuje stanovenie cieľov práce a metód, použitých na dosiahnutie cieľov. Záverečná kapitola je rozdelená podľa jednotlivých fáz vývoja aplikácie. Práca opisuje praktické využitie jednotlivých technológií, samotný kód aplikácie vo fáze programovania a základné bezpečnostné prvky aplikácie.

Kľúčové slová:

Webová aplikácia, PHP, databáza.

ABSTRACT

MIŠKOVIČ, Martin: *Web application development in PHP*. – University of Economics in Bratislava. Faculty of Economic Informatics; Department of Applied Informatics. – Supervisor: Ing. Peter Škurla, PhD. – Bratislava: FHI, 2014, 39 s.

The aim of my bachelor's thesis is to create a web application using open source solutions, such as programming language PHP, database server MySQL and jQuery libraries, and to describe application's development process. The main focus of the application is to create a publicly accessible place on the internet for the purposes of the advertisement for voluntary work offers. The three-layer application model was implemented to develop this application. The thesis follows several steps of web application development and describes the stages of design, programming and testing. Thesis is divided into 4 chapters. The first chapter is devoted to the introduction of web applications and describes the theoretical foundations of technologies that have been used during the development. The end of the first chapter is devoted to the web application security, which is often neglected. The second and the third chapter includes the objectives of the thesis and work methods used to achieve these objectives. The final chapter is divided according to the different phases of the application development. This thesis describes the practical use of various technologies, the web application source code in the programming phase and the basic safety features of the web applications in general.

Keywords:

Web application, PHP, database.

Obsah

| | |
|---|-----------|
| ÚVOD..... | 8 |
| 1 SÚČASNÝ STAV RIEŠENEJ PROBLEMATIKY DOMA A V ZAHRANIČÍ..... | 9 |
| 1.1 INZERCA PONÚK PRE DOBROVOĽNÍKOV CEZ WEB | 9 |
| 1.2 ROZMACH WEBOVÝCH APLIKÁCIÍ | 10 |
| 1.2.1 <i>Výhody a nevýhody webových aplikácií.....</i> | <i>11</i> |
| 1.3 SOFTVÉROVÝ BALÍK LAMP | 12 |
| 1.4 APACHE | 12 |
| 1.5 ÚVOD DO JAZYKA PHP | 13 |
| 1.6 RELAČNÉ DATABÁZY | 14 |
| 1.6.1 <i>MySQL a phpMyAdmin.....</i> | <i>15</i> |
| 1.7 JAVASCRIPT | 15 |
| 1.7.1 <i>jQuery</i> | <i>16</i> |
| 1.8 BEZPEČNOSŤ WEBOVÝCH APLIKÁCIÍ | 17 |
| 2 CIEĽ PRÁCE..... | 18 |
| 3 METODIKA PRÁCE A METÓDY SKÚMANIA..... | 19 |
| 3.1 PRÍSTUP RAD | 19 |
| 3.1.1 <i>Analýza potrieb.....</i> | <i>19</i> |
| 3.1.2 <i>Návrh</i> | <i>20</i> |
| 3.1.3 <i>Implementácia.....</i> | <i>20</i> |
| 4 VÝSLEDKY PRÁCE | 21 |
| 4.1 ANALÝZA POTRIEB POUŽÍVATEĽOV | 21 |
| 4.2 NÁVRH ŠTRUKTÚRY APLIKÁCIE | 23 |
| 4.2.1 <i>Dátová vrstva.....</i> | <i>24</i> |
| 4.2.2 <i>Logická vrstva.....</i> | <i>25</i> |
| 4.2.3 <i>Prezentačná vrstva.....</i> | <i>26</i> |
| 4.3 PROGRAMOVANIE APLIKÁCIE NA STRANE SERVERA | 27 |
| 4.3.1 <i>Riadenie prístupu cez .htaccess</i> | <i>27</i> |
| 4.3.2 <i>Základný kameň aplikácie</i> | <i>28</i> |
| 4.3.3 <i>Vývojárom definované funkcie.....</i> | <i>29</i> |
| 4.3.4 <i>Vyhľadávací filter</i> | <i>29</i> |
| 4.3.5 <i>Šablóny na dynamické generovanie stránok.....</i> | <i>30</i> |

| | |
|---|-----------|
| 4.4 PROGRAMOVANIE APLIKÁCIE NA STRANE KLIENTA | 30 |
| 4.4.1 Kaskádové štýly | 31 |
| 4.5 BEZPEČNOSŤ WEBOVEJ APLIKÁCIE | 32 |
| 4.5.1 Zachovanie integrity dát | 32 |
| 4.5.2 Šifrovanie v jazyku PHP | 33 |
| 4.5.3 Cookies a identifikácia používateľov | 34 |
| 4.6 TESTOVANIE | 35 |
| ZÁVER | 36 |
| ZOZNAM POUŽITEJ LITERATÚRY | 37 |
| KNIŽNÉ PUBLIKÁCIE | 37 |
| INTERNETOVÉ ZDROJE | 38 |
| PRÍLOHY | 39 |

Úvod

Internet je v dnešnej dobe nenahraditeľným pomocníkom v mnohých oblastiach osobného aj pracovného života. Každý deň ľuďom umožňuje odosielať elektronickú poštu, telefonovať, spravovať financie cez elektronické bankovníctvo, vyhľadávať informácie, vzdelávať sa, vytvárať vlastný obsah, ale aj zabávať sa pozeraním filmov, videí, hraním hier alebo tráviť čas na sociálnych sieťach. Globálna internetová sieť však okrem zábavy ponúka predovšetkým obrovský potenciál pre realizáciu vlastných nápadov a projektov.

Internetové pripojenie sa postupne dostáva od počítačov a notebookov aj do našich vreciek v podobe čoraz inteligentnejších mobilných telefónov. Dopyt po pripojení na internet a webových službách je na svojom vrchole a neustále rastie. Celosvetový trend využívania webových služieb poukazuje na rastúcu obľubu mobilného internetu. Na prudko rastúci dopyt samozrejme reagujú aj webový vývojári, ktorí vytvárajú stále interaktívnejšie aplikácie. [10]

Vývoj webových aplikácií v súčasnosti patrí medzi najžiadanejšie zamestnania a medzi najrýchlejšie rastúce oblasti programovania. Človek sa môže aj sám naučiť programovať, napríklad s pomocou odborných kníh a internetových školení. Existuje nemálo voľne dostupných projektov ako sú *w3schools.com* alebo *codecademy.com*, ktoré pokrývajú širokú škálu tém pre začiatočníkov aj pokročilých. Navyše sa jazyk PHP dá rýchlo naučiť, a to aj vďaka bohatej dokumentácii, veľkému počtu príkladov zdrojových kódov a silnej komunite.

Ako praktická ukážka vývoja webovej aplikácie je v práci popísaná aplikácia, slúžiaca na inzerciu ponúk pre dobrovoľníkov. Aplikácia funguje ako sprostredkovateľ ponuky na trhu dobrovoľníckej práce a pri jej vývoji boli použité najnovšie technológie, akými sú programovací jazyk PHP vo verzii 5.4, databáza MySQL verzie 5.1, XHTML 1.0, CSS 3, JavaScript a knižnice jQuery 1.10. Programovací jazyk PHP stále patrí k najobľúbenejším jazykom na tvorbu webových aplikácií. PHP, v kombinácii s MySQL databázami a JavaScriptom, využívajú aj najnavštevovanejšie weby akými sú *Yahoo*, *Wikipedia* alebo *WordPress*. [9]

1 Súčasný stav riešenej problematiky doma a v zahraničí

Myšlienka samotného projektu vznikla v skupine deviatich študentov, ktorí sa stretli počas navštevovania kurzu s názvom *Tretí sektor a dobrovoľníctvo v USA*, ktorý organizovalo Centrum severoamerických štúdií na Ekonomickej univerzite v Bratislave. Cieľom projektu je vytvoriť verejné dostupné miesto na internete, kde by mohli neziskové organizácie publikovať ponuky dobrovoľníckej práce. Neziskové organizácie by pomocou takejto webovej aplikácie získali rýchly komunikačný kanál na získanie dobrovoľníkov. Dobrovoľníci by si na jednom mieste mohli vybrať z rôznych ponúk a kontaktovať priamo požadovanú organizáciu.

1.1 Inzercia ponúk pre dobrovoľníkov cez web

Projekty podobného typu už na Slovenskom internete existujú. Portál *dobrovolnictvo.sk* združuje neziskové organizácie a dobrovoľníkov z celého Slovenska. Kolektív ľudí, ktorí stoja za týmto projektom, dokonca organizuje tréningy, prednášky, dielne a rôzne iné podujatia na podporu dobrovoľníctva a zaučenie dobrovoľníkov. Vďaka dostatočne rozmanitému obsahu sa ľahko hľadajú cez internetové vyhľadávače a dobrovoľníci môžu kontaktovať priamo jednotlivé neziskové organizácie. Ako ukazuje prieskum medzi študentmi Ekonomickej univerzity (viď kapitola č. 4.1), portál takéhoto typu môže byť pre potenciálneho dobrovoľníka príliš komplexný a neprehľadný. Na portáli je uvedených veľa ponúk, niektoré aj dlhodobé, zo všetkých kútov krajiny.

Niektoré neziskové organizácie si vedú samostatné webové stránky, ktoré používajú na získanie dobrovoľníkov. Tieto stránky sú však príliš jednostranné, obsahujú ponuky iba od jednej organizácie a môžu byť situované iba do určitej geografickej oblasti. Potenciálny dobrovoľník si na takýchto stránkach môže vybrať iba z obmedzenej ponuky, ktorá je často lokalizovaná iba do určitej oblasti.

V čom sa teda nový projekt odlišuje od už existujúcich? V prvom rade je úzko špecializovaný na cieľovú skupinu, ktorou sú študenti Ekonomickej univerzity v Bratislave. Uverejňované budú iba ponuky dobrovoľníckej práce z okolia Bratislavy, aby na ne študenti mohli reagovať aj počas štúdia a pobytu v hlavnom meste Slovenskej republiky. Okrem technického zabezpečenia je pre projekt dôležitá aj priama spolupráca s neziskovými organizáciami, ohľadom vytvárania atraktívnych ponúk, ktoré by študentov Ekonomickej univerzity zaujali.

1.2 Rozmach webových aplikácií

Vývojári aplikácií sa postupne presúvajú z desktopového prostredia do prostredia internetu. Vývoj webových aplikácií sprevádza sprístupnenie nových webových služieb, ako je napríklad *cloud*. Rozdiel medzi výkonom desktopových a webových aplikácií znižuje neustále sa zvyšujúca rýchlosť pripojenia na globálnu sieť internet a výkonné webové prehliadače, ktoré podporujú hardvérovú akceleráciu. Hardvérová akcelerácia je jedna z techník na zvýšenie výkonu procesora, ktorá sa vykonáva pomocou špecifického hardvéru. Princíp hardvérovej akcelerácie sa využíva hlavne pri procesoroch CPU a GPU. [10]

Novou výzvou pre vývojárov sa stal vývoj mobilných aplikácií. V nie ďalekej minulosti boli mobilné telefóny značne limitované na hlasovú a krátku textovú komunikáciu. To sa samozrejme nedá porovnať so súčasnými inteligentnými mobilnými telefónmi, ktoré využívajú moderné technológie ako sú 3G, či dokonca 4G. Ako uvádza Nicholas C. Zakas v elektronickom článku pre Asociáciu pre výpočtovú techniku (Association for computing machinery) [10], pri mobilnom pripojení nastáva nepríjemne dlhá latencia (časový rozdiel medzi odoslaním požiadavky a odpoveďou servera). Toto oneskorenie obmedzuje používateľa v dosiahnutí optimálneho využitia celej šírky prenosového pásma, aj keď jeho pripojenie je teoreticky schopné také veľké množstvo dát spracovať.

Webové aplikácie na komunikáciu v počítačovej sieti využívajú distribučný model klient – server. Tento model rozdeľuje dva medzi sebou komunikujúce počítače na žiadateľa (klient) a poskytovateľa služby (server). Klientovi stačí na komunikáciu bežný webový prehliadač. Prehliadač pri každom načítaní internetovej stránky pošle novú požiadavku na server, ktorý mu vygeneruje a pošle nový dokument, napríklad vo formáte HTML. [1 s. 17-19] Následne prehliadač zozbiera poskytnuté údaje a graficky ich reprezentuje. Na interakciu s vyobrazenou stránkou prehliadače používajú špecifické programovacie jazyky, z ktorých sa najčastejšie využíva JavaScript. JavaScript sa väčšinou používa iba na zmenu zobrazenia stránky, pričom neodosiela serveru novú požiadavku na stiahnutie údajov.

1.2.1 Výhody a nevýhody webových aplikácií

Webové aplikácie majú oproti desktopovým aplikáciám radu výhod pre používateľov aj vývojárov. Ak chce používateľ využívať desktopovú aplikáciu, musí ju nainštalovať na každé zariadenie, na ktorom ju chce spustiť. Táto aplikácia potom používa výpočtové prostriedky daného zariadenia a tým spomaľuje prípadnú prácu s ďalšími aplikáciami. Pri inštalovaní novej verzie desktopovej aplikácie je opäť potrebné vykonať zmeny na každom samostatnom zariadení. Na druhej strane má používateľ všetky potrebné súbory pre spustenie a beh aplikácie na lokálnom dátovom úložisku a inštalovanie nových verzií aplikácie má pod svojou kontrolou. [13]

Z pohľadu používateľa sa môže zdať webová aplikácia ako pohodlnejšia voľba. Všetko, čo potrebuje na spustenie aplikácie, je webový prehliadač a pripojenie na internet. Webové aplikácie sú platformovo nezávislé a využívajú známe štandardy značkových jazykov ako je HTML alebo XHTML. Používateľ teda vie spustiť webovú aplikáciu z ľubovoľného miesta a zariadenia, za predpokladu, že má prístup k internetovému pripojeniu a štandardnému webovému prehliadaču. K dispozícii má vždy aktuálnu verziu aplikácie a ušetrí aj miesto na lokálnom pevnom disku v jeho zariadení. Používateľ však stráca kontrolu nad inštaláciou nových verzií aplikácie. Vždy bude mať k dispozícii iba najnovšiu verziu, s ktorou sa musí naučiť pracovať. [13]

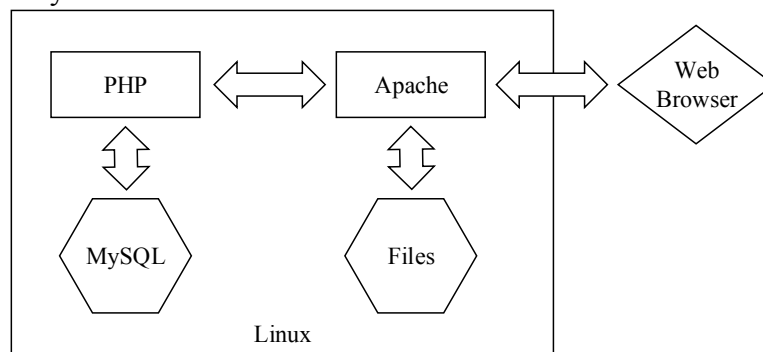
Webové aplikácie sú nezávislé na platforme, keďže používajú štandardy, ktoré v súčasnosti podporujú všetky najpoužívanejšie prehliadače. Z tohto pohľadu je vývojár oslobodený od vyvíjania aplikácie pre rôzne platformy a nemusí sa zaoberať podporou starších verzií. Z finančného hľadiska si vývojár za poskytnutie služby webovej aplikácie môže pýtať mesačný poplatok, z čoho získava pravidelné príjmy, oproti vstupnej investícii do desktopovej aplikácie. Vývojár má tiež k dispozícii najmocnejšiu zbraň pre úspešný rozvoj webovej aplikácie. Pomocou rôznych nástrojov ako *ClickTale* vie vývojár monitorovať správanie používateľov v aplikácii a na základe týchto údajov zmeniť aplikáciu tak, aby sa čo najjednoduchšie používala a vyhovovala čo najväčšiemu počtu používateľov.

Aj napriek používaným štandardom sa často stáva, že jednotlivé prehliadače zobrazujú obsah rôzne. Preto je potrebné aplikáciu otestovať na všetkých najpoužívanejších webových prehliadačoch. Vývojár musí navyše vziať do úvahy rôzne rozlíšenia obrazoviek používateľov a ovládanie pre dotykové zariadenia.

1.3 Softvérový balík LAMP

LAMP je akronym pre serverový softvérový balík v kombinácii Linux, Apache, MySQL a PHP. Softvérový balík LAMP ponúka cenovo lacnú a výpočtovo výkonnú variantu pre webové servery, keďže ponúka neplatený, voľne dostupný softvér. Najdôležitejším elementom je konkrétna linuxová distribúcia nainštalovaná na serveri. Okrem Linuxu a Apache sú súčasťou balíka LAMP ešte databázový server, ktorý môže byť napríklad MySQL alebo MariaDB, a modul pre konkrétny programovací jazyk, či už to bude PHP, Perl, Python alebo všetky dohromady. [14]

Obr. 1.1: Softvérový balík LAMP



Zdroj: [14]

1.4 Apache

Apache je skrátenejší názov pre Apache HTTP Server, open source softvér, ktorý beží na webovom serveri a obsluhuje HTTP požiadavky, odoslané internetovými prehliadačmi na strane klienta. Apache hrá kľúčovú rolu v rozdeľovaní úloh. Analyzuje požiadavku klienta a vykoná všetky potrebné operácie, aby požiadavku uspokojil. Ak sú od klienta požadované aj súbory s PHP skriptami, odošle požiadavky na konkrétne súbory modulu PHP, ktorý ďalej môže komunikovať s databázovým serverom.

Vývoj Apache HTTP servera začal vo februári 1995, kedy si mnohí webmasteri vytvárali vlastné opravy alebo nastavy vtedy najpoužívanejšieho servera HTTP daemon. Malá skupina týchto webmasterov sa stretla s cieľom zjednotiť vykonané zmeny. Apache vznikol ako nastava HTTPd, s pridanými opravami chýb, ktoré boli zozbierané od jednotlivých webmasterov. Na projekte sa stále podieľajú stovky nadšencov, ktorí k vývoju Apache prispievajú nápismi, zdrojovými kódmi alebo dokumentáciou. [15]

1.5 Úvod do jazyka PHP

Programovací jazyk PHP (Hypertext Preprocessor, pôvodne Personal Home Page) je voľne dostupný skriptovací jazyk, ktorý bol špeciálne navrhnutý na tvorbu webových aplikácií a dynamické generovanie internetových stránok. PHP vzniklo v roku 1995 a stále patrí medzi najobľúbenejšie skriptovacie jazyky. Pôvodne čisto procedurálny programovací jazyk rozšíril svoje základy o objektovo orientované programovanie od verzie PHP 3, ktorá vyšla v roku 1998. Objektovo orientovaná paradigma vývoja softvéru už bola dávno na svete. Tvorcovia PHP však argumentovali tým, že skriptovací jazyk je plnohodnotný a silný nástroj aj ako čisto procedurálny jazyk. Rovnakú funkcionality webovej aplikácie v jazyku PHP vieme zabezpečiť dvoma spôsobmi: pomocou funkcií alebo objektov. Výhodou je zrejmá skutočnosť, že vývojár má k dispozícii obidva prístupy a môže si vybrať ten, ktorý mu viac vyhovuje. Napriek všetkým výhodám objektovo orientovaného programovania sa vždy nájde blok kódu, ktorý je efektívnejší, rýchlejší a pamäťovo menej náročnejší, ak sa vykoná procedurálne. [2 s. 22-24]

PHP patrí do skupiny CGI (Common Gateway Interface) skriptovacích jazykov, ktoré poskytujú určité rozhranie medzi serverom a aplikáciou, ktorá generuje obsah webu. Týmto spôsobom sa dá zabezpečiť dynamické generovanie stránky a jej napojenie na databázovú vrstvu, ktorá môže predstavovať jeden súbor alebo celú databázu. CGI skriptovacie jazyky sú základom pre bezproblémové rozširovanie webovej aplikácie. Bez nich by musela byť každá samostatná stránka prítomná na serveri v HTML kóde, čo by vážne znehodnocovalo strávený čas pri aplikovaní zmien na každý súbor. [1 s. 18-19]

PHP je skriptovací jazyk, ktorý sa spúšťa na serveri ešte pred uspokojením požiadavky klienta. PHP kód môže byť uložený v samostatných súboroch s príponou *php* alebo je integrovaný do kódu HTML. Samotný kód je interpretovaný PHP modulom, ktorý z neho vygeneruje požadované dáta v HTML, JSON, alebo inom formáte a tie sú serverom odoslané klientovi. [1 s. 27, 39] Na komunikáciu medzi aplikáciami sa často používa formát JSON alebo XML. JSON je iba blok textu využívajúci dvojice názov a hodnota atribútu. XML navyše umožňuje hierarchické delenie elementov a je ľahko čitateľný pre človeka aj stroj. Aplikácie si vymieňajú dáta cez vonkajšie rozhrania. Na popísanie rozhraní webových služieb sa využíva jazyk WSDL (Web Service Description Language), ktorý je tiež založený na XML. Formát XML sa stal základom pre ďalšie formáty dokumentov, akými sú napríklad RSS, SOAP alebo XHTML. [3 s. 248-250]

1.6 Relačné databázy

Každá webová aplikácia pracuje s určitými údajmi, ktoré môžu mať rôznu štruktúru a môžu byť uložené v rôznych formách. Na uchovanie dát môže aplikácia využívať súbory, ktoré môžu byť štruktúrované napríklad štandardom XML alebo uložiť dáta do poľa v PHP súbore. Tieto možnosti sa však zdajú byť príliš ťažkopádne pri komplexnom obsahu, sťahovaní veľkého množstva údajov alebo častých zmenách obsahu.

Najčastejšie sa na uchovanie dát pre webové aplikácie používajú relačné databázy, ktoré sú založené na teórii relačnej algebry. Na prácu s relačnými databázami však nie je nevyhnutné túto teóriu ovládať. Relačné databázy sa skladajú z tabuliek, ktoré predstavujú relácie a opisujú určitú entitu reálneho sveta. [4 s. 208] Ako príklad si za entitu môžeme zvoliť zamestnancov ľubovoľnej firmy. Tabuľky majú klasickú formu, sú tvorené riadkami a stĺpcami. Každý stĺpec predstavuje atribút entity, ktorý opisujeme a riadok predstavuje jednu konkrétnu entitu. Na príklade by stĺpce predstavovali vlastnosti zamestnanca – jeho meno, priezvisko, pracovné zaradenie, plat a iné. Riadky predstavujú záznam o každom jednom zamestnancovi. Každého jedného zamestnanca je potrebné jednoznačne identifikovať pomocou primárneho kľúča. Primárny kľúč sa musí skladať z jedného alebo viacerých atribútov v danej tabuľke. Treba však zabezpečiť, aby bol primárny kľúč unikátny, teda aby v jednej tabuľke neexistovali dva alebo viac záznamov s rovnakým primárnym kľúčom.

Jedna z najväčších výhod relačných databáz je možnosť popisovať vzťahy medzi jednotlivými entitami alebo tabuľkami. Pomocou vzťahov sú tabuľky priamo prepojené a je možné ich spájať na základe cudzích kľúčov. Ten sa odkazuje na primárny kľúč v inej tabuľke s ktorou je prepojený. [4 s. 209-210]

V relačných databázach poznáme tri základné typy vzťahov. Prvým vzťahom je typ $1:1$, ktorý hovorí, že k jednému riadku z tabuľky A prislúcha práve jeden riadok z tabuľky B (zamestnanec môže pracovať na práve jednom pracovisku). Druhým vzťahom je $1:n$. Tento vzťah opisuje napríklad skutočnosť, že jeden zákazník môže vytvoriť veľa objednávok. Posledným vzťahom v relačných databázach je typ $m:n$. Napríklad ak sa vo firme riešia projekty, na ktoré sú pridelení zamestnanci. Jeden zamestnanec však môže byť priradený ku viacerým projektom. Vzťah $m:n$ nevieme v relačnom modeli priamo opísať. Musí sa vytvoriť nová tabuľka, pomocou ktorej sa vzťah $m:n$ rozbije na dva vzťahy $1:n$, obvykle kombináciou primárnych kľúčov z oboch tabuliek. [4 s. 207-211]

1.6.1 MySQL a phpMyAdmin

MySQL je obľúbený SQL databázový relačný server, s voľne dostupným zdrojovým kódom. MySQL server sa najčastejšie používa s Apache servermi a PHP modulmi, je však implementovaný vo viacerých programovacích jazykoch, akými sú PHP, C++, Perl, či Python, a je podporovaný na viacerých platformách (Windows, Linux, Solaris). [6 s. 7]

Pri vývoji webových aplikácií s použitím technológií PHP a MySQL je veľmi užitočná aplikácia phpMyAdmin, ktorú si obľúbia nielen začínajúci programátori. phpMyAdmin je vizuálny SQL klient s voľne dostupným zdrojovým kódom, ktorý umožňuje interaktívnu tvorbu databáz cez používateľsky príjemné grafické rozhranie. Táto aplikácia okrem bežných prác s tabuľkami a SQL príkazmi, dokáže generovať PHP aj SQL kódy, na základe vytvorenej štruktúry tabuliek. phpMyAdmin je obzvlášť vhodný pre začínajúcich vývojárov, ktorým umožní interaktívne vytvorenie databázy. Pri každej zmene si vývojár vie sledovať SQL príkazy vykonaných zmien na databáze, ktoré aplikácia automaticky generuje. [6 s. 109-110]

1.7 JavaScript

Najzaujímavejším stupňom vývoja webových aplikácií na strane klienta je nepochybne vytvoriť používateľovi možnosť aktívne narábať s aplikáciou. Skutočne interaktívna sa webová aplikácia stane až s použitím JavaScriptu. Pomocou JavaScriptu dokážeme pretvoriť celý vzhľad webovej aplikácie – schovávať HTML elementy, meniť definície kaskádových štýlov elementov, presúvať a animovať objekty a mnoho ďalších užitočných akcií. Pri vývoji grafického používateľského prostredia sa medze nekladú a vďaka JavaScriptu je možné vytvoriť naozaj čokoľvek. [7 s. 89]

Bežné správanie webového prehliadača je najskôr načítanie HTML obsahu webovej stránky a potom spustenie JavaScriptu na príslušný HTML element. Toto riešenie však pri väčšom objeme skriptov blokuje prenášanie HTML dokumentu, hlavne pri pomalšom pripojení na internet. JavaScript môže byť naprogramovaný na sledovanie určitej akcie (funkcia *EventListener*), ktorá sa za určitých podmienkach vykoná. Napríklad po presunutí kurzora nad tlačidlo zmení farbu jeho pozadia. Zbehnutie skriptu môže byť aktivované kliknutím myši, presunutím kurzora alebo vstupom z klávesnice. [7 s. 90-96]

1.7.1 jQuery

Knižnice sú zbierkou preddefinovaných funkcií, ktorých hlavným cieľom je znížiť časovú náročnosť vývoja aplikácie pri súčasnom znížení potrebného počtu riadkov kódu. Tento cieľ dosahujú definovaním často využívaných funkcií, ako je napríklad základný pohyb HTML elementov, rozbaľovanie zoznamov a animácie, simulácia princípu *Drag&Drop*, rôzne tlačidlá, menu alebo kalendáre. [7 s. 98]

Písanie vlastných skriptov je dobrý programátorský prístup a vývojár sa pri ňom veľa naučí. Postupom času sa však tento prístup stane ťažkopádnym a otravným, keď musí vývojár použiť v každom projekte takmer rovnaké funkcie, iba s malými zmenami. Knížnice však vedia poskytnúť priestor na rýchlu produkciu webovej aplikácie, a sú vhodné predovšetkým pre prácu v tímoch. Hlavnou nevýhodou použitia knižníc je zvýšenie času, potrebného na načítanie stránky, čo môže byť veľkou prekážkou pre mobilné zariadenia. [7 s. 98]

jQuery patrí medzi najobľúbenejšie knížnice JavaScriptu. Od svojho vzniku sa veľmi rýchlo presadila a v auguste 2012 už okupovala približne päťdesiat percent zo vzorky milióna najnavštevovanejších internetových stránok. jQuery abstrahuje od bežne používaných funkcií a poskytuje jednotný výsledný efekt na všetkých webových prehliadačoch. jQuery funguje na princípe definovania všeobecných metód so schopnosťou prijímať rôzne typy argumentov pre komplexné úlohy. [7 s. 98-99]

Okrem jQuery existuje aj špeciálna knížnica jQuery Mobile, ktorá však nie je samostatnou knížnicou. jQuery Mobile je v skutočnosti nadstavba jQuery knížnice, ktorá vývojárom poskytuje nové nástroje, optimalizované na vývoj mobilných aplikácií. K používaniu tejto knížnice je však stále potrebný základ, ktorý je obsiahnutý v jQuery. Okrem jQuery existujú knížnice, ako napríklad Zepto.js alebo jQTouch, ktorých veľkosť je približne dvadsaťpäť percent z veľkosti jQuery, pritom sú s jQuery kompatibilné. Spomenuté alternatívy knižníc však nemajú takú dobrú podporu webových prehliadačov ako jQuery. Preto je potrebné si dopredu premyslieť, ktorým smerom sa bude vývojár uberať. [7 s. 100]

1.8 Bezpečnosť webových aplikácií

Webové aplikácie sú uverejnené na internete, kde sa ľahko môžu stať terčom počítačovej kriminality. Úlohou vývojára je zabezpečiť každú časť webovej aplikácie pred úmyselným aj neúmyselným zneužitím. Internet je voľne dostupné a otvorené miesto. Otvorenosť so sebou prináša možnosť návštevy používateľov, ktorých zámery nie sú ušľachtilé. Bezpečnosť nie je jeden z doplnkov, ktoré vývojár do aplikácie pridáva. So zabezpečením webovej aplikácie musí vývojár počítať od úplného začiatku vývoja a implementovať ju do jadra aplikácie. [4 s. 361-362]

V súčasnej dobe nie je problém vyskladať si výkonné počítače, ktoré dokážu šifrovanie hesiel prelomiť stále rýchlejšie. Vhodným riešením by bolo, okrem stále silnejšieho šifrovania, požadovať od používateľa viacstupňové prihlasovanie pomocou rôznych hesiel alebo pravidelné zmeny hesla tak, aby sa nezhodovali s už raz použitými heslami. Takto prehnaná bezpečnosť zmenší pravdepodobnosť prelomenia, ale radikálne zníži aj spokojnosť používateľa s využívaním aplikácie. Vývojár preto musí nájsť ideálnu strednú cestu tak, aby bola webová aplikácia bezpečná a príjemne sa používala. [4 s. 362]

Jednou z možných techník je vytvorenie dvojnásobnej kontroly hodnôt, ktoré sa zadávajú do formulára. Prvá časť je ošetrená HTML kódom alebo JavaScriptom, na určenie prípustných hodnôt alebo vytvorenie regulárneho výrazu. Na samotný HTML kód ani JavaScript sa však nedá úplne spoľahnúť. Kontrola JavaScriptom sa môže jednoducho zrušiť a čo sa týka prípustných hodnôt v HTML formulári, tie sú odosielané cez textový HTTP protokol, ktorý je možné jednoducho zmeniť predtým, ako sa dostane k serveru. Obidve možnosti však poskytujú používateľsky príjemné prostredie na overovanie hodnôt. Druhú časť tvorí PHP kód na strane servera, ktorý ešte raz skontroluje vstupné hodnoty. [4 s. 363-364]

Základom bezpečnosti aplikácie je zabezpečiť databázu pred neoprávneným prístupom. To znamená povoliť prístup iba oprávneným osobám a povoliť im narábať len s tou časťou obsahu, na ktoré majú právo. Pojem bezpečná databáza nadobúda na dôležitosť, ak je vo webovej aplikácii verejne prístupné vyhľadávanie, ktoré sa môže stať pomôckou pri vložení škodlivého SQL kódu (SQL injection). Vloženie škodlivého kódu prebieha cez vyhľadávací HTML formulár, ktorého vstup je použitý na vytvorenie SQL príkazu. Jednoducho stačí ukončiť daný príkaz bodkočiarkou a za ním napísať príkaz na vymazanie všetkých záznamov z konkrétnej tabuľky. PHP potom namiesto jedného SQL príkazu pošle na databázový server dva a údaje sú stratené. [4 s. 365-372]

2 Cieľ práce

Hlavným cieľom práce je vytvoriť webovú aplikáciu v jazyku PHP, ktorá bude slúžiť na inzerciu ponúk pre dobrovoľníkov. Poskytnúť neziskovým organizáciám jednoduchý a rýchly prístup k získaniu nových dobrovoľníkov z radov študentov Ekonomickej univerzity v Bratislave. Webová aplikácia bude slúžiť ako prostredník medzi ponukou a dopytom na trhu dobrovoľníckej práce.

Na študentov, ako záujemcov o dobrovoľníctvo, má aplikácia pôsobiť príjemne a jednoducho, aby si vedeli rýchlo vyhľadať konkrétnu ponuku. Pomocou webovej aplikácie budú mať študenti možnosť kontaktovať priamo neziskovú organizáciu, ktorá ponuku inzerovala či už elektronickou poštou alebo telefonicky. Pre neziskové organizácie je nutné pripraviť používateľsky príjemnú administráciu, aby mohli bez problémov publikovať nové ponuky a časom aj staršie ponuky spravovať – upravovať, aktivovať, deaktivovať, mazať.

Vedľajším cieľom práce je oboznámiť čitateľa so svetom webových aplikácií a najnovších technológiách, ktoré sa využívajú na ich vývoj. Opísať základnú trojvrstvovú architektúru webových aplikácií a vysvetliť niektoré programovacie techniky v prostredí programovacieho jazyka PHP. Ďalším vedľajším cieľom práce je analyzovať danú webovú aplikáciu z pohľadu používateľa aj vývojára a poukázať na niektoré zaujímavé funkcionality aplikácie.

3 Metodika práce a metódy skúmania

Objektom skúmania práce je webová aplikácia v jazyku PHP, ktorá bola navrhnutá ako klasický trojvrstvový model. Aplikácia kombinuje silu programovacieho jazyka PHP, robustnosť databázového servera MySQL a pružnosť najnovších knižníc jQuery. Webová aplikácia je schopná poskytnúť každému návštevníkovi stránky zoznam dostupných ponúk pre dobrovoľníkov. Každý návštevník môže využiť robustný vyhľadávací filter na nájdenie konkrétnej ponuky. Pre neziskové organizácie aplikácia ponúka jednoduchú a rýchlu cestu k získaniu nových dobrovoľníkov. Na komunikáciu s používateľom využíva HTML formuláre na jednoduché pridávanie alebo zmenu už existujúcich ponúk. Aplikácia poskytuje neziskovým organizáciám vlastnú administráciu. V administrácii si môžu zobraziť vedenú evidenciu všetkých ponúk, ktoré môžu pridávať, meniť, mazať a zobrazovať počty prihlásených uchádzačov.

3.1 Prístup RAD

Na vytvorenie webovej aplikácie bol použitý prístup rapidneho vývoja aplikácií (RAD - Rapid application development). Metodológia RAD slúži na urýchlenie vývoja aplikácií, pričom podstatnú úlohu v nej hrajú práve budúci používatelia aplikácie. Výhodou RAD oproti štandardnému vývoju životného cyklu aplikácií (SDLC – Standard development life cycle) je jeho rýchlosť a flexibilita. SDLC je viac systematický a opatrnejší prístup, ktorý sa zameriava hlavne na presnosť, zatiaľ čo RAD je jeho skrátaná verzia. Prístup RAD sa skladá z troch fáz, ktoré na seba navzájom nadväzujú a budú predstavené v nasledovných podkapitolách. Budúci používatelia sú zapojení do každého stupňa vývoja aplikácie. [8 s. 163-166]

3.1.1 Analýza potrieb

V prvej fáze je potrebné jasne stanoviť ciele aplikácie. Vstupným bodom pre získanie údajov boli webové formuláre, pomocou ktorých sme získali potrebné údaje od študentov Ekonomickej univerzity. Cez webové dotazníky sme okrem iných informácií zisťovali, či študenti majú záujem o portál pre dobrovoľníkov, aké dodatočné informácie by podľa nich mal portál obsahovať a aké ponuky by na portáli mali byť uverejnené (viď príloha č. 10). V prvej fáze boli zapojení všetci potenciálni používatelia, ktorí sa prieskumu zúčastnili.

3.1.2 Návrh

V druhej fáze sa iteratívne tvorí návrh aplikácie a jej prototyp, ktorý sa následne testuje. Pri programovaní aplikačnej vrstvy v jazyku PHP bol použitý kombinovaný procedurálny a objektovo orientovaný prístup, bez využitia preddefinovaných knižníc alebo rámcov (framework). Pre interné potreby webovej aplikácie boli vytvorené vlastné funkcie, zatiaľ čo pre prístup na MySQL bol využitý objektovo orientovaný prístup. Preferovaný prístup pri tvorbe kódu bol procedurálny, z dôvodu menšej časovej náročnosti vykonania príkazov.

Kvôli zvýšenej bezpečnosti webovej aplikácie bol použitý takzvaný dvojnásobný prístup, ktorý ponúka kompletne riešenie bezpečnosti za rozumnú námahu. Prvá časť, označovaná ako *prístup zhora dole*, odráža opisný postoj k návrhárovi webovej aplikácie. Jeho úlohou je naplánovať zabezpečenie aplikácie tak, že navrhne konkrétne opatrenia. Druhá časť, ktorá je opačným prístup k prvému, sa nazýva *prístup zdola hore*. V druhej fáze vývojár sleduje jednotlivé komponenty (napríklad databázový server) a ich zraniteľnosť proti útokom.

Na návrhu dizajnu aplikácie a jej následnom testovaní sa podieľala výberová skupina deviatich ľudí z radov študentov Ekonomickej univerzity. V skupine boli štyria muži a päť žien vo veku 21 až 24 rokov. Pri tvorbe dizajnu webovej aplikácie bol použitý reagujúci (responsive) prístup, ktorého hlavnou črtou je tvorba HTML objektov s relatívnymi rozmermi. Vďaka takto definovaným rozmerom sa aplikácia vie prispôbiť monitorom s rôznym rozlíšením. Prístup je vhodný aj na vývoj aplikácií na mobilné platformy, ale v ukážkovej webovej aplikácii sme sa zamerali iba na desktopové platformy.

3.1.3 Implementácia

V poslednej fáze bola webová aplikácia implementovaná na webový server s PHP modulom verzie 5.4. Aplikácia je v súčasnosti implementovaná na verejne dostupnej doméne *dobrovolnicieu.eu* a bola finálne testovaná neziskovými organizáciami. Keďže bola webová aplikácia vyvíjaná ako nový a jediný produkt, nebolo potrebné riešiť problém paralelného behu aplikácií rôznych verzií.

4 Výsledky práce

Ešte pred napísaním prvého riadku kódu musí mať vývojár jasnú predstavu o aplikácii, ktorú vytvára. Prvým krokom pri vývoji webovej aplikácie je analýza potrieb používateľov. Čo od aplikácie očakávajú, aké majú predstavy o funkcionalite a čo všetko pomocou nej môžu získať. Používateľov webovej aplikácie je potrebné rozdeliť na dve skupiny: študentov – potenciálnych dobrovoľníkov a inzerujúce neziskové organizácie.

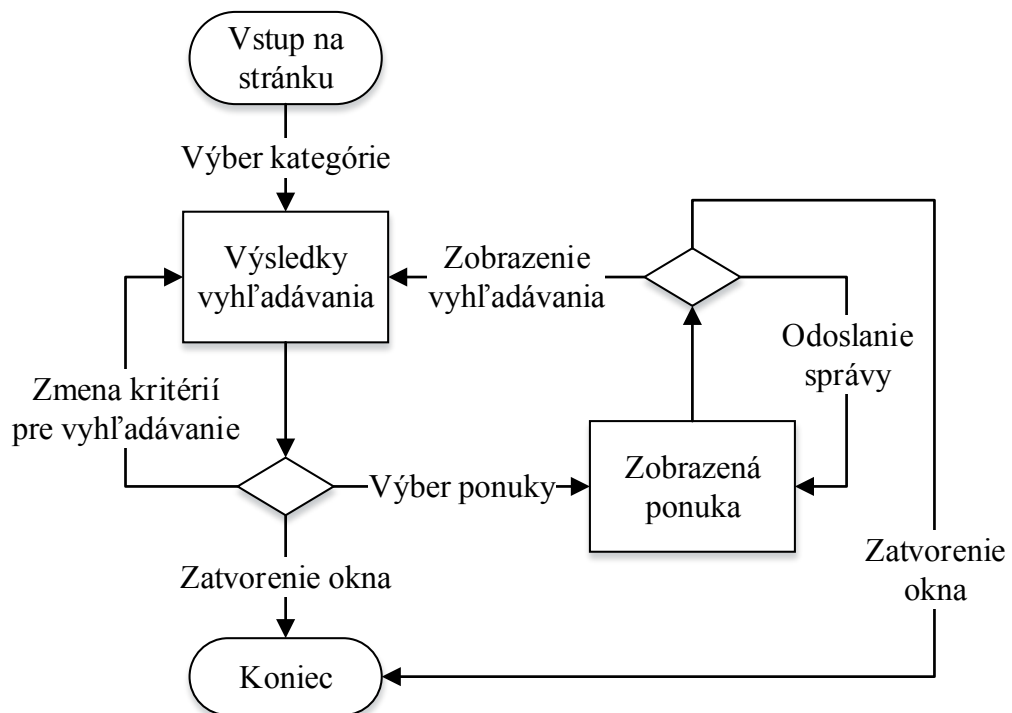
4.1 Analýza potrieb používateľov

S výberovou skupinou deviatich študentov sme vytvorili akúsi prvotnú predstavu o aplikácii, ktorá nám poslúžila ako základ pre vytvorenie internetového dotazníka (viď príloha č. 10). Formou internetového dotazníka sme sa študentov Ekonomickej univerzity pýtali na dobrovoľnícke portály a či by mali záujem využívať špecializovanú aplikáciu na inzerovanie dobrovoľníckych ponúk pre dobrovoľníkov z Ekonomickej univerzity.

Približne 70 percent respondentov odpovedalo, že dobrovoľnícke portály nevyužíva, napriek tomu až takmer 93 percent študentov by inzerčnú webovú aplikáciu pre dobrovoľníkov z radov študentov Ekonomickej univerzity uvítalo. Zo študentov, ktorí portály využívajú, by až 75 percent respondentov charakterizovalo konkrétny portál ako neprehľadný. Napriek neprehľadnosti väčšina študentov odpovedalo, že portál poskytuje dostatočné množstvo informácií o dobrovoľníctve aj o samotných ponukách. Zaujímavým výstupom dotazníka bolo rozdelenie študentov na dve skupiny v pomere 50:50. Prvá skupina by sa zaujímala o ponuky iba v Bratislave a okolí, zatiaľ čo druhá skupina by sa zaujímala o ponuky na celom území Slovenskej republiky. Študenti tiež rozhodovali o kľúčových slovách, pomocou ktorých môže používateľ efektívne vyhľadávať ponuky. Väčšina respondentov sa zhodla na nasledovných kritériách vyhľadávania: aktuálnosť ponuky, kategória ponuky, názov ponuky, miesto činnosti, popis vykonávanej činnosti a názov neziskovej organizácie.

Aktivitu používateľov – študentov zachytáva vývojový diagram číslo 4.1. Ako uvádza Lukas Mathis v [5 s. 96], vývojové diagramy slúžia na analýzu cieľov používateľa a krokov, akými sa do cieľa vie dostať. Cieľom študenta je nájsť vhodnú ponuku a kontaktovať organizáciu, ktorá ponuku inzerovala. Pre efektívne hľadanie vhodnej ponuky používateľ využije služby vyhľadávacieho filtra. Po zadaní parametrov filter zobrazí zhodujúce sa ponuky. Používateľ si vie ponuky postupne prezrieť a v prípade záujmu neziskovú organizáciu kontaktovať. Pre uľahčenie hľadania ponúk je potrebné rozdeliť ich do určitých skupín alebo kategórií. Tieto kategórie musia byť jasné, výstižné a spolu musia pokryť všetky možné zamerania ponuky. Výberová skupina deviatich študentov sa pri zachovaní RAD prístupu zhodla na nasledovných kategóriách: vzdelávanie, životné prostredie, ľudské práva, sociálna pomoc a voľný čas. Takto rozdelené kategórie pokryjú všetky ponuky, takže neostane ponuka, ktorú by nebolo kam priradiť.

Obr. 4.1: Vývojový diagram vyhľadávania ponúk



Zdroj: [Vlastné spracovanie]

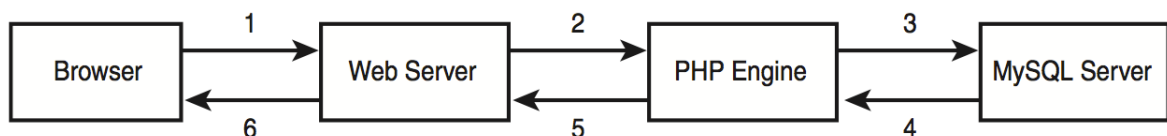
4.2 Návrh štruktúry aplikácie

Základnú architektúru webovej aplikácie môžeme rozdeliť na tri samostatné časti: dátovú, aplikačnú (logickú) a prezentačnú vrstvu. V aplikácii nebudeme používať overovanie prístupových práv v MySQL. Overovanie platného prihlásenia používateľa a získanie jeho práv prebehne na aplikačnej vrstve v PHP skripte.

Obrázok 4.2 zachytáva typickú architektúru webových aplikácií. V nasledujúcich bodoch je popísaná komunikácia medzi jednotlivými prvkami architektúry:

1. Používateľov prehliadač odošle HTTP požiadavku pre konkrétnu stránku. Používateľ môže napríklad cez webový formulár požadovať vyhľadávanie medzi ponukami dobrovoľníckej práce.
2. Webový server požiadavku prijme, analyzuje ju a vyhľadá požadovaný PHP súbor, ktorý odošle PHP modulu na spracovanie.
3. PHP modul začne s vykonávaním príkazov v súbore. Jeden z príkazov je pripojenie sa na databázu. Vtedy PHP vytvorí spojenie s MySQL serverom a odošle dotaz na vrátenie ponúk podľa kritérií, ktoré zadal používateľ.
4. MySQL server prijme dotaz, ktorý vykoná a vráti požadované dáta späť PHP modulu. Dáta budú obsahovať všetky ponuky, ktoré vyhovujú kritériám od používateľa.
5. PHP modul prijme dáta z databázy, dokončí beh skriptu a vytvorí formátovaný HTML dokument, ktorý odošle späť webovému serveru. HTML dokument už bude obsahovať všetky požadované ponuky.
6. Web server odošle prijatý HTML dokument so všetkými ponukami späť klientovi, ktorý ho požadoval. [4 s. 217]

Obr. 4.2: Architektúra webových aplikácií



Zdroj: [4 s. 217]

4.2.1 Dátová vrstva

Dátová vrstva je najspodnejšou vrstvou architektúry a slúži na uchovávanie údajov, ktoré môže aplikácia spracovávať, pridávať, meniť alebo mazať. Na uchovanie dát na serveri aplikácia využíva MySQL databázu verzie 5.1, ktorá poskytuje rýchly a pohodlný prístup k údajom. Pred vytvorením samotnej databázy je potrebné zvážiť, ktoré dáta je potrebné v databáze uchovať na splnenie všetkých požadovaných funkcionalít webovej aplikácie. Štruktúru databázy čiastočne znázorňuje obrázok číslo 4.3. Okrem znázornených základných atribútov obe tabuľky obsahujú ďalšie údaje, ktorých zobrazenie v tomto bode nie je nevyhnutné a ich absencia prispieva k prehľadnosti príkladu.

V prvej tabuľke sú neziskové organizácie, pre ktoré je potrebné uchovať ich prihlasovacie a kontaktné údaje. Stav organizácie slúži na administratívne účely webovej aplikácie. Ako prvý krok je potrebné vykonať kontrolu údajov z registra neziskových organizácií, či sa skutočne jedná o neziskovú organizáciu. Až po úspešnej kontrole administrátor webovej aplikácie povolí neziskovej organizácii prístup do administrácie. Stĺpce ID v oboch tabuľkách predstavujú unikátny celočíselný identifikátor organizácie alebo ponuky.

Druhá tabuľka obsahuje údaje o jednotlivých ponukách dobrovoľníckej práce. Každá ponuka (jeden riadok) však musí obsahovať aj identifikačné číslo organizácie v stĺpci ID_ORG (cudzí kľúč), aby bolo možné tieto tabuľky prepojiť. Každá nezisková organizácia môže mať uverejnených n ponúk, z čoho vyplýva vzťah typu $1:n$. Okrem iných údajov tabuľka obsahuje informácie o stave ponuky (aktuálna, neaktuálna alebo vyradená ponuka), kategórie ponuky a súčet všetkých odoslaných odpovedí. V stĺpci URL je uchovaná kompletná URL adresa každej ponuky.

Tab. 4.1: Tabuľka údajov o neziskových organizáciách

| ID | STAV | MENO | HESLO | NAZOV | MAIL | ... |
|---------------|------|------|-------|-------|------|-----|
| primárny kľúč | | | | | | |

Tab. 4.2: Tabuľka ponúk

| ID | STAV | TYP | ID_ORG | URL | SUCET | ... |
|---------------|------|-----|------------|-----|-------|-----|
| primárny kľúč | | | cudzí kľúč | | | |

Zdroj: [Vlastné spracovanie]

4.2.2 Logická vrstva

Aplikačná alebo logická vrstva predstavuje jadro celej aplikácie. Na vývoj webovej aplikácie je použitý programovací jazyk PHP verzie 5.4, pričom sa využíva kombinovaný procedurálny a objektovo orientovaný prístup. Vo webovej aplikácii je potrebné zabezpečiť prístupové práva na troch úrovniach.

Prvú úroveň predstavuje používateľ – študent, ktorý má rolu host'a a do webovej aplikácie sa neprihlasuje. Jeho jediným právom je prezerat' si inzerované ponuky, ktoré sú rozdelené do piatich základných kategórií. Na načítavanie údajov pre vyhľadávanie slúži vyhľadávací filter, ktorý je najdôležitejšou funkcionalitou celej aplikácie. Filter vie prehľadávať jednotlivé ponuky v databáze podľa zadaných kritérií a kľúčových slov, pričom používateľovi neumožní vykonať na databáze akékoľvek zmeny (vid' príloha č. 1).

Druhú úroveň predstavujú neziskové organizácie. Neziskové organizácie majú po zadaní správnej kombinácie prihlasovacích údajov sprístupnenú administráciu. Prihlásením sa organizácia jednoznačne identifikuje, na základe čoho dostane právo narábať s vlastnými ponukami. Organizácie môžu ponuky vytvárať, upravovať a mazať, pričom každá výmena údajov bude prebiehať cez webové formuláre. Kontrola údajov vo formulároch je zabezpečená JavaScriptom. Zabezpečenie integrity údajov v databáze je ošetrené PHP skriptom, ktorý daný formulár spracováva. Neziskové organizácie si môžu sami upravovať prihlasovacie aj kontaktné údaje a tiež sledovať, koľko používateľov sa prihlásilo na jednotlivé ponuky.

Tretia úroveň slúži pre administrátora webovej aplikácie, ktorý má sprístupnené všetky práva a má vytvorenú vlastnú administráciu. Jeho úlohou je predovšetkým kontrolovať aktivitu neziskových organizácií a overiť ich identitu porovnaním údajov v databáze a registri neziskových organizácií.

4.2.3 Prezentačná vrstva

Najvrchnejšia vrstva slúži na grafickú reprezentáciu dát z databázy, ktoré už boli spracované logickou vrstvou. Prezentačnú vrstvu tvorí webový prehliadač používateľa, ktorý pretvára získané údaje do grafického výstupu. Dizajn webovej aplikácie je kritická časť vývoja, pretože rozhoduje o celkovom dojme aplikácie na používateľa, či služby aplikácie využije aj v budúcnosti a či o aplikácii povie svojim priateľom. Dizajn webovej aplikácie bol navrhnutý v súlade s RAD metodológiou, pričom sme využili vzorku deviatich študentov, ktorí sa podieľali na návrhu štruktúry aj grafického zobrazenia.

Na docielenie príjemného dizajnu aplikácie a zjednotenie sa s farbami Ekonomickej univerzity bola zvolená modro-biela farebná kombinácia a zaoblené rohy (viď príloha č.1). Aplikácia má štandardný tvar s vertikálnym rozdelením. Obsahuje hlavičku, menu s kategóriami, hlavný textový blok a pätičku s dodatočnými informáciami. Kvôli jednoduchému a rýchlemu prístupu k ponukám bolo do vrchnej časti stránky vložené plytké menu so všetkými kategóriami ponúk, ktoré sa odkazuje priamo na vyhľadávací filter. Možná nejednoznačnosť obrázkového menu bola odstránená pridaním názvu kategórie. Implementáciou spoločného obrázkového a textového menu tiež vznikla väčšia interaktívna plocha pre používateľa. [5 s. 121-123]

Z pohľadu študenta alebo potenciálneho záujemcu o dobrovoľníctvo musí byť webová aplikácia jednoduchá na používanie. Študent sa v nej musí vedieť rýchlo zorientovať a vyhľadať potrebné údaje. Filter je umiestnený tak, aby bol používateľovi dostupný na maximálne jedno kliknutie myšou. Zmena pozadia textového menu študentovi okamžite prezradí, v akej kategórii ponúk sa nachádza. Pre neziskové organizácie je potrebné vytvoriť vlastnú administráciu, v ktorej si môžu spravovať svoje ponuky. V administrácii bolo do hlavného textového bloku pridané menu, ktoré predovšetkým slúži na rýchle orientovanie sa pri spravovaní údajov a ponúk.

Na základnú prezentáciu údajov vo webovej aplikácii je využitý značkovací jazyk XHTML verzie 1.0 a kaskádové štýly CSS verzie 3. Kaskádové štýly umožnia efektívne oddeliť obsah webu (tvorený štruktúrou XHTML) od jeho dizajnu. Oddelenie obsahu od dizajnu webovej aplikácie je rozumné hlavne kvôli prípadným zmenám v obsahu alebo dizajne. Pre zvýšenie používateľského pohodlia je vo webovej aplikácii implementovaný JavaScript a knižnica jQuery verzie 1.9.1. JavaScript je využitý napríklad pri prepínaní medzi zobrazením filtra a ponuky, alebo na overovanie údajov pred odoslaním webového formulára.

4.3 Programovanie aplikácie na strane servera

V kapitole číslo 4.2 je popísaný priebeh uspokojenia HTTP požiadavky klienta webovým serverom. V tejto kapitole bude opísaná štruktúra aplikačnej vrstvy, teda aké všetky kroky sa vykonajú v PHP module, kým HTML dokument neodovzdá webovému serveru. Postupne bude popísané využitie jednotlivých stavebných PHP súborov a ich implementácia na ukážkovej webovej aplikácii.

4.3.1 Riadenie prístupu cez *.htaccess*

Prvou dôležitou zastávkou každej požiadavky je súbor *.htaccess*, ktorý slúži predovšetkým na riadenie prístupu k súborom a adresárom na serveri. Pomocou neho vieme zablokovať prístup do adresárov na základe IP adresy alebo povinných prihlasovacích údajov. [4 s. 403] Súbor *.htaccess*, uložený v kmeňovom adresári domény, obsahuje dvojice súborových prípon a aplikačných modulov. Takýmto spôsobom je možné definovať konkrétny modul, ktorý sa má použiť na spracovanie súboru s konkrétnou príponou. V ukážke môžeme vidieť, že rôzne typy *php* súborov môžu spracovávať PHP moduly rôznych verzií (viď príloha č. 2).

Ďalšia funkcia súboru *.htaccess* je elegantný *RewriteEngine*, ktorého príkazy *RewriteCond* a *RewriteRule* slúžia na prepisovanie URL adresy. V súbore môže byť nastavené napríklad pridávanie lomítka na koniec URL adresy. *.htaccess* zabezpečí, aby pri obsluhovaní každej požiadavky zo strany klienta, bol vždy ako prvý PHP súbor spracovaný *index.php*, ktorý je základným stavebným kameňom webovej aplikácie. Taktiež dokáže odstrániť prípony súborov ako sú *html*, *php* a vytvoriť na pohľad pekné URL adresy. Cez príkaz *ErrorDocument* vie server zobrazovať vývojárom definované HTML dokumenty, ak nastane príslušná chyba. Napríklad ak webový server nenájde klientom požadovaný dokument, nastane chyba 404 a klienta presmeruje na URL adresu definovanú v súbore *.htaccess*, pri príslušnej chybe.

4.3.2 Základný kameň aplikácie

Základným PHP súborom, ktorý predstavuje vstupný bod webovej aplikácie je *index.php*. V súbore *.htaccess* je definované, že pri každom načítaní stránky z danej domény je ako prvý spracovaný *index*. Ako prvý príkaz je uvedené spustenie *session*, ktorý musí byť vykonaný ako prvý (viď príloha č. 3).

Session umožňuje aplikácii dočasne uchovávať PHP premenné, ktoré ostávajú dostupné aj po zavolaní novej HTTP požiadavky. Takéto premenné môžu slúžiť na uchovanie prihlasovacieho mena alebo kľúčových slov vyhľadávacieho filtra. HTTP protokol nevie uchovávať stav medzi dvoma transakciami a práve tento problém rieši *session*. *Session* je vytvorená pre každého používateľa webovej aplikácie samostatne a používa jedinečné identifikačné číslo, čím sa zabráni sledovaniu ostatných používateľov. Identifikačné číslo aktuálnej *session* môže byť uložené v súbore *cookie* alebo odosielané v rámci URL adresy a je to jediná informácia, ktorá je uložená na strane klienta. [4 s. 509-511]

Ďalej je v súbore *index.php* algoritmus na získanie popisných meta dát z databázy, pretože každá ponuka má definované vlastné meta tagy a tie sú uložené priamo v databáze. Toto riešenie bolo do webovej aplikácie implementované pre zvýšenie viditeľnosti ponúk vo webových vyhľadávačoch. Ak sa URL adresa odkazuje na konkrétnu ponuku v databáze, je potrebné načítať meta dáta ešte pred načítaním šablóny, v ktorej sa už vytvára HTML dokument. URL adresu je možné rozdeliť do jednotlivých úrovní, oddelených lomítkom. PHP kód dokáže presmerovať používateľa na hierarchicky vyššiu úroveň URL adresy v prípade, že požadovaná stránka neexistuje.

PHP modul načíta konfiguračný súbor s definovanými globálnymi premennými, vlastné funkcie a šablónu. Súbor *index.php* je to správne miesto na pridanie všetkých definovaných funkcií, objektov, dát a šablón, ktoré majú byť prístupné naprieč celou webovou aplikáciou.

4.3.3 Vývojárom definované funkcie

Prínosom procedurálneho prístupu k vývoju webovej aplikácie je možnosť definovania potrebných funkcií. Ak sú funkcie v jednom PHP súbore, vieme ich sprístupniť pre celú aplikáciu tak, že pridáme na začiatok súboru *index.php* jeden príkaz (*require*). V takejto konfigurácii je možné volať všetky vývojárom definované funkcie bez nutnosti určenia cesty k súboru. Funkcie môžu vykonávať rôzne úlohy. V ukázkovom príklade sú funkcie využité na prihlasovanie do databázy, konverziu kódovania textových reťazcov alebo odosielanie elektronickej pošty (viď príloha č. 4).

Pre vyhľadávací filter bola vytvorená špecifická funkcia *win2ascii*, ktorá slúži na konverziu textových reťazcov z kódovania Windows 1250 do ASCII. Pri zadaní kľúčových slov používateľ očakáva, že filter vyhľadá zodpovedajúce ponuky bez ohľadu na to, či kľúčové slovo diakritiku obsahuje alebo nie. Pomocou tejto funkcie vieme prekonvertovať napríklad slovo „Šaštín“ na „sastin“. V SQL príkaze *SELECT*, pri prehľadávaní databázy na kľúčové slová, je definovaný príkaz *COLLATE CP1250_GENERAL_CI*, ktorý prinúti ASCII znaky vystupovať v kódovaní Windows 1250, pričom nezáleží na veľkosti písmen. Týmto spôsobom sme získali dva reťazce v zhodnom kódovaní Windows 1250, ktoré MySQL server porovnáva a vyhľadáva zhodu. Pre používateľa webovej aplikácie je príjemnejšie, ak môže vyhľadávať slová aj s diakritikou alebo bez nej.

4.3.4 Vyhľadávací filter

Základom úspechu webovej aplikácie je vyhľadávací filter, ktorý vie vyhľadávať ponuky pomocou viacerých kľúčových slov (viď príloha č. 5). Základom filtra je vyhľadávanie podľa aktuálnosti ponuky a zamerania (kategórie). Jeho skutočná sila však spočíva v možnosti definovania kľúčových slov, rozdelených medzi názov ponuky, miesto, popis ponuky a organizáciu, ktorá ponuku inzerovala. Na základe týchto kľúčových slov vie používateľ pomerne presne vyhľadať konkrétnu ponuku aj pri ich rozsiahlejšom počte. Každý z týchto atribútov ponuky je v databáze uložený samostatne. Pre samotné využitie kľúčových slov na vyhľadávanie bola pre aplikáciu vytvorená funkcia *win2ascii*, ktorej funkcionálnosť je popísaná v predchádzajúcej kapitole.

Filter vie prehľadávať databázu pre každé kľúčové slovo samostatne, ak sú slová oddelené medzerou. Tento spôsob prehľadávania je výhodný v tom, že nezáleží na poradí zadaných kľúčových slov, pretože je pre každé slovo generovaná samostatná podmienka.

4.3.5 Šablóny na dynamické generovanie stránok

Súbor *template.php* v ukázkovom príklade predstavuje hlavnú a jedinú PHP šablónu, ktorá hovorí o štruktúre webovej aplikácie a jednotlivých HTML elementov každej stránky (viď príloha č. 6). Šablónou sa začína vytvárať zdrojový kód každej konkrétnej stránky. Prvým krokom je definovanie typu dokumentu ako je napríklad formát XHTML verzie 1.0, typ Transitional, za ktorým nasleduje samotný XHTML dokument. Hlavička dokumentu obsahuje údaje o obsahu ako je jazyk, textové kódovanie, definovanie kľúčových slov a popisu pre lepšie vyhľadávanie na internete, definovanie skriptov a ciest ku kaskádovému štýlom CSS. Po hlavičke nasleduje telo XHTML dokumentu, ktoré už hovorí o jeho samotnej štruktúre. Telo aplikácie je vertikálne rozdelené na hlavičku, menu, obsah a pätičku. V šablóne je definovaná iba hierarchia jednotlivých elementov, nič však nehovorí o tom, ako sa majú tieto elementy vo webovom prehliadači zobrazovať.

4.4 Programovanie aplikácie na strane klienta

V ukázkovej webovej aplikácii našli najväčšie uplatnenie pomerne jednoduché skripty, prevažne na zobrazenie alebo skrytie určitých elementov a na kontrolu vstupných údajov vo formulároch. Jedným zo skriptov obsluhuje tlačidlo na prepínanie zobrazovania vyhľadávacieho filtra a konkrétnej ponuky, ak je načítaná. Pri zobrazenej ponuke je v hornej časti tlačidlo na zobrazenie HTML formulára, ktorý slúži na odoslanie elektronickej správy neziskovej organizácii, ktorá ponuku inzerovala. Princíp je rovnaký ako pri prepínaní medzi filtrom a ponukou, v tomto prípade je však použitá knižnica jQuery pre používateľsky príjemný efekt. Vďaka funkcii *slideToggle* sa formulár postupne zroluje smerom nadol a celý text ponuky sa posunie o výšku formulára. Opakovaným kliknutím na tlačidlo sa formulár zroluje naspäť smerom hore k tlačidlu.

Vlastný skript bol pre ukázkovú aplikáciu vytvorený na zmenu definície elementov v CSS súbore. Cieľom je vytvoriť prepojenie medzi obrázkom a textom v menu tak, aby sa pri presunutí kurzora nad obrázok alebo text zmenilo pozadie textu. Používateľ tak bude mať aj vizuálnu informáciu o tom, že sa pod obrázkom ukrýva odkaz, na ktorý môže kliknúť. Vytvorenie takéhoto prepojenia s kaskádovými štýlmi je možná jedine vtedy, ak by sa element obrázok v HTML kóde nachádzal pred textom. V opačnom prípade je nutné použiť JavaScript, kde je pomocou funkcie *style* zmenený štýl zobrazovania daného elementu, v ukázkovom príklade textu nad obrázkom (viď príloha č. 6).

4.4.1 Kaskádové štýly

Kaskádové štýly CSS slúžia na definovanie dizajnu jednotlivých HTML elementov, teda akým spôsobom sa budú zobrazovať v prehliadači na strane klienta. Od svojho vzniku umožnili vývojárom zbaviť sa obmedzujúcich stránok, založených na HTML tabuľkách. Pri vývoji webovej aplikácie je použitý reagujúci (responsive) prístup. Hlavným znakom reagujúceho prístupu k tvorbe dizajnu webovej aplikácie je použitie flexibilných dĺžok, zatiaľ čo jeho opak, adaptívny prístup, využíva šablóny s pevnou dĺžkou. Pri reagujúcom prístupe sa využívajú percentuálne definované rozmery, vzhľadom na rodičovský element. Reagujúcim prístupom je možné maximalizovať úžitok používateľa s webovou aplikáciou, bez ohľadu na použité zariadenie a jeho rozlíšenie. [7 s. 53]

Jeden z možných problémov použitia reagujúceho prístupu môže byť kombinácia fixných a flexibilných dĺžok, ktorej použitie vo webovej aplikácii je potrebné vždy dobre premyslieť. Môže totiž nastať situácia, kedy chce vývojár k percentuálnemu rozdeleniu šírky jednotlivých elementov pridať odsadenie s fixnou dĺžkou. Súčet fixných a flexibilných širok všetkých elementov však v takomto prípade môže presiahnuť 100 percent. [7 s. 53-54]

Problémy môžu vyplynúť aj z hierarchie elementov. Ak máme rodičovský element s relatívnou šírkou 75 percent a v jeho vnútri sa nachádza ďalší element, ktorý má mať šírku 40 percent vzhľadom na maximálnu šírku okna aplikácie a nie vzhľadom na šírku rodičovského dokumentu. Pre dosiahnutie cieľa musí mať dcérsky element zadaných 40 percent zo 75 (53,33333 percent), čo často vedie k nie pekným číslam. [7 s. 54]

Na príklade ukážkovej webovej aplikácie je riešený problém s relatívnou šírkou hlavného menu, ktoré sa skladá z obrázkov a príslušného textového popisu (viď príloha č. 7). Celé menu je riešené ako tabuľka s dvoma riadkami, pričom šírka jej buniek je rovnomerne rozdelená na päť častí. Dôležité je však umiestnenie obrázkov a textových popisov do stredu buniek tak, aby boli zachované pôvodné rozmery obrázka. Týmto spôsobom sa aj pri rôznych rozlíšeniach obrazoviek zachovávajú rozmery obrázkov a zmeny v rozlíšení budú vykryté flexibilne širokými bunkami tabuľky. [7 s. 59-62]

4.5 Bezpečnosť webovej aplikácie

Aby mohla byť webová aplikácia umiestnená na verejne prístupnej doméne, bolo do nej nutné implementovať bezpečnostné nástroje, ktoré zaručia zachovanie integrity systému a údajov v databáze. Bezpečnosť webovej aplikácie znamená predovšetkým správne nastavenie prístupových práv používateľov a skutočnosť, aby sa aplikácia vždy správala tak, ako sa od nej očakáva. Na to, aby webová aplikácia splnila svoju funkciu, musí komunikovať s používateľom cez internet. Môžeme povedať, že aplikácia s používateľom komunikuje cez určité rozhranie, ktoré používateľovi poskytuje nástroje na využívanie aplikácie. Pri tvorbe bezpečných aplikácií je v prvom rade potrebné každému používateľovi priradiť prístupové práva a určiť činnosti, ktoré môže vykonávať.

4.5.1 Zachovanie integrity dát

Verejná časť webovej aplikácie v sebe obsahuje vyhľadávanie ponúk pomocou filtra, všeobecné podmienky používania aplikácie, kontakty a iné základné náležitosti. K verejnej časti aplikácie má prístup každý používateľ, respektíve návštevník webových stránok. Návštevník má možnosť vyhľadávať ponuky a v konečnom dôsledku má prístup ku všetkým aktuálne platným ponukám a rovnako aj ku kontaktným údajom inzerujúcej neziskovej organizácie, okrem prihlasovacích údajov. Návštevník má teda priamy nástroj na vyhľadávanie ponúk v databáze, ktorý sa môže jednoducho zneužiť.

Väčšina komunikácie medzi aplikáciou a používateľom prebieha cez HTML formuláre. Údaje z formulárov sa serveru odosielajú cez textový HTTP protokol, čo nie je bezpečný systém na odosielanie údajov. Odoslanie požiadavky na server cez HTTP protokol sa dá nasimulovať a server navyše nevie aké hodnoty dostal. Preto je nutné kontrolovať hodnoty odosielaných parametrov nielen pred odoslaním, ale aj po prijatí. Pri vyhľadávanom filtri máme zabezpečené iba ohraničenie možných hodnôt pri výbere kategórie ponuky a či chce používateľ zobraziť aktuálne alebo neaktuálne ponuky. Prehľadávanie na kľúčové slová nie je na strane klienta kontrolované. Ďalšie formuláre sa nachádzajú v administrácii pre neziskové organizácie a slúžia na pridávanie a upravovanie ponúk. Formuláre v administrácii navyše obsahujú kontrolu údajov cez JavaScript. Kontrola pomocou JavaScriptu je pre užívateľa príjemná, pretože ho na potrebné zmeny alebo neplatné hodnoty upozorní ešte pred odoslaním údajov na server (viď príloha č. 8). Ani táto kontrola však nie je dostatočná, a tak všetky parametre, ktoré sú z formulárov odoslané, musia byť skontrolované priamo v PHP skripte, ktorý beží na serveri.

4.5.2 Šifrovanie v jazyku PHP

Súkromnú sekciu pre každú neziskovú organizáciu predstavuje administrácia webovej aplikácie. K administrácii má organizácia prístup po zadaní prihlasovacích údajov, ktoré tvorí meno a prihlasovacie heslo. Po overení, že sa jedná o skutočného používateľa, sa mu na základe údajov v databáze pridajú prístupové práva. Prístupové práva organizácii slúžia predovšetkým na prístupovanie k vlastným údajom a všetkým inzerovaným ponukám. Prístupové práva zároveň chránia údaje v databáze pred ich zmenou ostatnými neziskovými organizáciami.

Aby mohli byť prihlasovacie údaje overené, musia sa porovnať s tými, ktoré sú v databáze. Aplikácia povolí neziskovej organizácii registráciu iba v tom prípade, ak zadala prihlasovacie meno, ktoré sa v databáze nenachádza. Týmto krokom sa aplikácia chráni proti prípadným duplicitným záznamom. V databáze však nie je uložené čisté heslo, ktoré by tak bolo veľmi zraniteľné proti prípadnému útoku. Prihlasovacie heslo je ešte pred zapísaním do databázy zašifrované jednosmerným algoritmom. Na generovanie aj overenie hesla aplikácia používa PHP šifrovaciu funkciu *crypt* (viď príloha č. 9). Funkcia *crypt* podporuje viacero šifrovacích algoritmov, medzi ktorými je aj MD5, DES, Blowfish alebo SHA-512. Výber konkrétneho šifrovacieho algoritmu závisí na podpore konkrétneho servera. PHP od verzie 5.3 obsahuje vlastnú implementáciu niektorých algoritmov, ktoré sa použijú ak sú silnejšie ako algoritmy podporované systémom. Pre bezpečne zašifrované heslo je potrebné použiť takzvaný *salt*, ktorého konkrétna forma závisí od použitého šifrovacieho algoritmu. Vo všeobecnosti je *salt* náhodne vygenerovaný reťazec znakov presne určenej dĺžky, ktorý sa použije na šifrovanie hesla. *Salt* sa spolu s čistým heslom zašifruje, čoho výsledkom je zašifrované heslo, ktoré bude pre rovnaké čisté heslo vždy iné. Zašifrované heslo sa spolu so *saltom* uloží do databázy. Šifrovacie algoritmy sú jednosmerné, takže neexistuje možnosť ako zo zašifrovaného hesla získať pôvodné. Pri overovaní hesla sa ono heslo opäť zašifruje pomocou rovnakého šifrovacieho algoritmu a *saltu*, ktorý je uložený v databáze. Ako posledný krok stačí čerstvo zašifrované heslo porovnať s heslom uloženým v databáze a v prípade úspešnej zhody povoliť používateľovi prístup do administrácie. [11]

4.5.3 Cookies a identifikácia používateľov

Nezisková organizácia poskytla webovej aplikácii svoje prihlasovacie údaje, aplikácia ich úspešne overila a poskytla organizácii prístupové práva do administrácie. Aplikácia je však naprogramovaná v PHP a server má krátku pamäť. Server spustí všetky požadované skripty a vráti klientovi čistý HTML dokument. Ako ale docieľiť, aby sa používateľ mohol pohybovať naprieč aplikáciou a odosielať viaceré požiadavky na server bez toho, aby musel opakovane zadávať svoje prihlasovacie údaje?

Na bezproblémové pristupovanie k webovej aplikácii po prihlásení má vývojár dve možnosti. Môže použiť takzvanú *session*, ktorá je vytvorená pre každého používateľa samostatne a dajú sa pomocou nej uchovávať premenné. Tieto premenné sa uchovávajú dovedy, kým má klient vo webovom prehliadači otvorené okno s danou aplikáciou. Na rozdiel od *session*, pre *cookies* vývojár vie definovať presný časový interval, počas ktorého je *cookie* dostupný. [6 s. 399-401]

Existujú rôzne typy *cookies*, najčastejšie sa však vyskytujú ako overovacie *cookies*. Overovacie *cookies* slúžia na testovanie, či je užívateľ prihlásený alebo nie. Tento typ je použitý aj v ukážkovej webovej aplikácii. *Cookies* však môžu slúžiť aj na dlhodobé uchovávanie údajov, kedy slúžia na sledovanie používateľov. Súbor *Cookies* sú iba krátke bloky textu, nie sú to počítačové programy a nie sú spustiteľné ako kód, taktiež nevedia pristupovať k pevnému disku klientovho počítača. Problém predstavujú *cookies* tretích strán, kedy sú používateľské informácie odoslané webovému serveru tretej strany bez súhlasu používateľa. Väčšinou sa jedná o *cookies* firiem, ktoré umiestňujú reklamy na webové stránky svojich zákazníkov. To je aj najčastejší dôvod negatívnej reakcie používateľov na *cookies*. [12]

V ukážkovej webovej aplikácii boli *cookies* využité na uchovanie prihlasovacích údajov. Nastavenie *cookie* je vždy na určitý časový interval. Po jeho uplynutí *cookie* zanikne a používateľ sa musí znova prihlásiť. Keď však klient na server odošle ďalšiu požiadavku, *cookie* mu je predĺžené na svoju pôvodne definovanú hodnotu, aby mohol používateľ pokračovať bez prerušenia (viď príloha č. 9).

4.6 Testovanie

Testovanie webovej aplikácie prebehlo s testovacími ponukami a so štyrmi testovacími subjektami, z ktorých boli všetci študenti Ekonomickej univerzity vo veku 21 až 23 rokov. Po praktickej stránke z pohľadu študentov aj neziskových organizácií testovanie nepreukázalo výrazné nedostatky. Upravené boli JavaScripty, ktoré slúžia na kontrolu formulárov pri vyplňaní údajov neziskovými organizáciami.

Kvôli dodatočnej bezpečnosti boli do všetkých formulárov, ktoré slúžia na zmenu údajov v databáze, implementované prvky *captcha*. Tento prvok chráni aplikáciu pred útokmi softvérových programov – botov, ktorí sú schopní odosielať údaje cez nezabezpečené formuláre. *Captcha* slúži na vykonanie Turingovho testu a teda vie zistiť, či sa za počítačom na strane klienta ukrýva človek alebo „stroj“. Pre človeka je tento test veľmi jednoduchý, stačí mu napísať to, čo vidí na obrazovke. Počítač však takúto analýzu spraviť nedokáže. Veľmi rozšírené sú najmä obrázkové testy. Do ukážkovej aplikácie bol implementovaný jednoduchý matematický test na sčítavanie dvoch pseudonáhodných čísiel z intervalu od nula po deväť (viď príloha č. 8).

Aplikáciu sme mali snahu optimalizovať pre internetové vyhľadávače. Problém nastal pri nedostatočnom obsahu aplikácie. Okrem inzerovaných ponúk v databáze sa v aplikácii nenachádza veľa textu a ponuky predstavujú relatívne dynamický obsah webu, ktorý sa postupne mení. Pri dostatočnom počte inzerovaných ponúk by však tento problém nemal byť zreteľný. Na zlepšenie vyhľadávania sme do databázy pre každú ponuku pridali kľúčové slová a popis obsahu stránky. Tým sme dokázali pre každú samostatnú stránku vytvoriť jedinečné meta popisy obsahu, ktoré pomôžu internetovým vyhľadávateľom lepšie porozumieť obsahu a zefektívniť vyhľadávanie medzi uverejnenými ponukami.

Záver

Vývoj webovej aplikácie bez použitia špecifického frameworku alebo CMS (Content Management System) má viaceré výhody. Ukážková aplikácia zaberá o poznanie menej miesta, spolu s obrázkami veľkosť aplikácie na disku nedosahuje ani 1 megabajt. Rovnako je vylepšený priemerný čas načítania stránok, keďže aplikácia nie je robustná a pozostáva z malého počtu súborov. Webová aplikácia na inzerovanie ponúk pre dobrovoľníkov spĺňa predpokladaný účel využitia a prináša efektívny prístup k vyhľadávaniu ponúk. Aplikácia je dobre dostupná aj z internetových vyhľadávačov, akým je napríklad Google. Ďalšími krokmi pre úspešný rozvoj aplikácie je neustála komunikácia s neziskovými organizáciami a zabezpečenie povedomia študentov Ekonomickej univerzity o existencii samotnej aplikácie, čo však už nie je účelom tejto publikácie.

V práci sú vysvetlené základné techniky na vývoj webových aplikácií s najnovšími dostupnými nástrojmi. Všetky nástroje, ktoré boli na vývoj aplikácie použité, boli open source, teda voľne a zadarmo dostupné. Využitím open source technológií sme znížili cenu vytvorenia webovej aplikácie, ktorá sa tak znížila iba na cenu práce počas návrhu a programovania. Na ďalšie využitie aplikácie stačí využiť prenájom od hostingovej spoločnosti. Cenové kalkulácie pri takto malej aplikácii sú takmer zanedbateľné a závisia priamoúmerne od počtu predpokladaných ponúk uložených v databáze.

Zoznam použitej literatúry

Knižné publikácie

1. KOSEK, J. 1999. *PHP : Tvorba interaktívnych internetových aplikácií*. Praha : Grada Publishing, 1999. 492 s. ISBN 80-7169-373-1.
2. LAVIN, P. 2006. *Object-Oriented PHP : Concepts, Techniques and Code*. San Francisco : No Starch Press, 2006. 216 s. ISBN 978-1-59327-077-3.
3. POKORNÝ, J. 2008. *XML technologie*. Praha : Grada Publishing, 2008. 272 s. ISBN 978-80-247-2725-7.
4. WELLING, L. THOMSON, L. 2008. *PHP and MySQL Web Development*. 4. vyd. Boston : Addison-Wesley, 2008. 1008 s. ISBN 978-0-672-32916-6.
5. MATHIS, L. 2011. *Designed for Use*. Dallas : The Pragmatic Bookshelf, 2011. 344 s. ISBN: 978-1-93435-675-3.
6. NARAMORE, E. et al. 2005. *Beginning PHP5, Apache, and MySQL Web Development*. Indianapolis : Wiley Publishing, 2005. 816 s. ISBN: 978-0-7645-7966-5.
7. GASSTON, P. 2013. *The Modern Web : Multi-Device Web Development with HTML5, CSS3, and JavaScript*. San Francisco : No Starch Press, 2013. 264 s. ISBN: 978-1-59327-487-0.
8. KENDALL, K. KENDALL, J. 2011. *Systems Analysis and Design*. 8. vyd. New Jersey : Prentice Hall, 2011. 600 s. ISBN: 978-0-13-608916-2.






Internetové zdroje

9. WIKIPEDIA: The free encyclopedia. 2014. *Programming languages used in most popular websites* [online], 6. marec 2014 [cit. 12. marec 2014]. Dostupné na internete: http://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites
10. ZAKAS, N. C. 2013. *Building Web sites that perform well on mobile devices remains a challenge* [online], 17. február 2013 [cit. 12. marec 2014]. Dostupné na internete: <http://queue.acm.org/detail.cfm?id=2441756>
11. PESLYAK, A. 2010. *How to manage a PHP application's users and passwords* [online], august 2010 [cit. 10. marec 2014]. Dostupné na internete: <http://www.openwall.com/articles/PHP-Users-Passwords>
12. GRAHAM, CH. *Introduction to Cookies and sessions (and Javascript)* [online], [cit. 17. marec 2014]. Dostupné na internete: http://ocportal.com/docs9/tut_cookies.htm
13. SATALKAR, B. *Advantages of Web Based Applications* [online], [cit. 13. marec 2014]. Dostupné na internete: <http://www.dayasolusi.com/articles/webdevelopment/127-advantages-of-web-based-applications>
14. BROWN, M. 2005. *Understanding LAMP* [online], 1. december 2005 [cit. 25. marec 2014]. Dostupné na internete: <http://www.serverwatch.com/tutorials/article.php/3567741/Understanding-LAMP.htm>
15. THE APACHE SOFTWARE FOUNDATION. *About* [online], [cit. 24. marec 2014]. Dostupné na internete: http://httpd.apache.org/ABOUT_APACHE.html

Prílohy

1. Ukážka dizajnu webovej aplikácie, spolu s vyhľadávacím filtrom.

DOBROVOLNÍCI EKONOMICKEJ UNIVERZITY
informačný portál s aktuálnou ponukou pre dobrovoľníkov

VZDELÁVANIE  **ŽIVOTNÉ PROSTREDIE**  **ĽUDSKÉ PRÁVA**  **SOCIÁLNA POMOC**  **VOLNÝ ČAS** 

Chcem vidieť:
 aktuálne ponuky
 neaktuálne ponuky

Zameranie:
Životné prostredie

Zobraziť výsledok

Vyhľadaj kľúčové slovo:

Názov ponuky:

Miesto:

Popis:

Organizácia:

VÝSLEDKY FILTRA:

Podľa zadanych kritérií nebola nájdená žiadna ponuka.

kto sme | [čas](#) | [kontakt](#) | [všeobecné podmienky](#) | [administrácia](#)

2014 © dobrovolnici.eu

2. Súbor .htaccess so základnými nastaveniami aplikácie.

```
1 # Definovanie PHP modulov
2 AddType application/x-httpd-php54 .php
3 AddType application/x-httpd-php5 .php52
4 AddType application/x-httpd-php .php4
5
6 RewriteEngine On
7 RewriteBase /
8
9 # Podmienky a pravidla na prepisovanie URL adresy
10 RewriteCond %{HTTP_HOST} ^dobrovolnicieu.eu
11 RewriteRule ^(.*)$ http://www.dobrovolnicieu.eu/$1 [R=301,QSA]
12
13 RewriteCond %{THE_REQUEST} ^[A-Z]{3,9}\ /index\.php\ HTTP/
14 RewriteRule ^index\.php$ http://www.dobrovolnicieu.eu/ [R=301,L]
15
16 # Prida na koniec URL adresy lomitko
17 RewriteCond %{REQUEST_FILENAME} !-f
18 RewriteCond %{REQUEST_URI} !\.\.+
19 RewriteCond %{REQUEST_URI} !/$
20 RewriteRule (.*?) %{REQUEST_URI}/ [R=301,L]
21
22 # Presmerovanie na index z /URL1/URL2/URL3/URL4/ na index.php?list=URL1&
act1=URL2&act2=URL3&act3=URL4
23 RewriteRule ^([^\./]+)$ ./index.php?list=$1 [L,QSA]
24 RewriteRule ^([^\./]+)/([^\./]+)$ ./index.php?list=$1&act1=$2 [L,QSA]
25 RewriteRule ^([^\./]+)/([^\./]+)/([^\./]+)$ ./index.php?list=$1&act1=$2
&act2=$3 [L,QSA]
26 RewriteRule ^([^\./]+)/([^\./]+)/([^\./]+)/([^\./]+)$ ./index.php?list=
$1&act1=$2&act2=$3&act3=$4 [L,QSA]
27
28 # Definicia chyboveho dokumentu
29 ErrorDocument 404 http://www.dobrovolnicieu.eu/error-404
```

3. Spojenie všetkých komponentov aplikácie do jedného celku. (1/2)

```
1  <?
2  // spustenie session
3  session_start();
4
5  // vypisovanie chybovych hlasi
6  error_reporting(0);
7
8  // nacitanie konfigu, bez konfigu hodi chybu
9  if (file_exists('./configs/global.php')) require './configs/global.php';
10 else die('ERROR: CONFIG FILE MISSING...');
11
12 // nacitanie funkcii
13 if (file_exists('./configs/functions.php')) require './configs/functions.
14 php';
15 else die('ERROR: FUNCTIONS FILE MISSING...');
16
17 if ( $_GET['list']=='homepage' )
18 {
19     Header( 'HTTP/1.1 301 Moved Permanently' );
20     Header( 'Location: '.$_PAGE['base_url'] );
21     exit;
22 }
23
24 // uvolnenie premennej, ktora zachytava aktualnu kategoriu
25 if (isset($_GET['list']) AND !isset($_GET['act1'])) unset($_SESSION['zam
26 eranie']);
27
28 // nacitanie metatagov z DB
29 if(($_GET['act1']!='') AND (($_GET['list']!='administracia') OR ($_GET['
30 list']!='admin'))
31 {
32     $spojenie = mysql_connection();
33     $query = 'SELECT * FROM PONUKY WHERE URL="'.$_PAGE['base_url'].$_GET
34 ['list'].'/'.$_GET['act1'].'/'";
35     if($spojenie->multi_query($query))
36     {
37         do
38         {
39             if($vysledok = $spojenie->store_result())
40             {
41                 while ($row = $vysledok->fetch_array(MYSQLI_BOTH))
42                 {
43                     $_PAGES[$_GET['list']][$_GET['act1']]['title']=$row['NAZOV']
44 ;
45                     $_PAGES[$_GET['list']][$_GET['act1']]['keywords']=$row['KEYW
46 ORDS'];
47                     $_PAGES[$_GET['list']][$_GET['act1']]['description']=$row['D
48 ESCRIPTION'];
49                     $address=$row['URL'];
50                 }
51             }
52             $vysledok->free();
53         }
54     } while ($spojenie->next_result());
55 }
56 else die('Chyba pri dotazovaní.');
```

3. Spojenie všetkých komponentov aplikácie do jedného celku. (2/2)

```
50     $vysledok->close();
51     $spojenie->close();
52     }
53
54     // ak zvolena stranka/podstranka neexistuje, tak prejdeme o uroveň vyššie
55     // z WEB/LIST/ACT1 na WEB/LIST/ a podobne...
56     if ( isset($address) OR $_GET['list']=="vzdelavanie" OR $_GET['list']=="
57     zivotne-prostredie" OR $_GET['list']=="ludske-prava" OR $_GET['list']=="
58     socialna-pomoc" OR $_GET['list']=="volny-cas")
59     {
60     }
61     else
62     {
63     if ( isset($_GET['list']) and isset($_GET['act1']) and !file_exists('./
64     sections/body/'.$_GET['list'].'body_'.$_GET['act1'].'.php') ) Header(
65     'Location: '.$_PAGE['base_url'].$_GET['list'] );
66     elseif ( isset($_GET['list']) and !isset($_GET['act1']) and !file_exists(
67     './sections/body/body_'.$_GET['list'].'.php') ) Header( 'Location: '.$
68     $_PAGE['base_url'] );
69     }
70
71     // ak požadovaná stránka nie je zvolená alebo neexistuje, tak sa zobrazí
72     // homepage
73     if ( !isset($_GET['list']) ) $_GET['list']='homepage';
74
75     // načítanie meta tagov pre jednotlivé sekcie
76     if (isset($_PAGES[$_GET['list']][$_GET['act1']][$_GET['act2']]['title'])
77     ) $_PAGE['title'] = $_PAGES[$_GET['list']][$_GET['act1']][$_GET['act2']]
78     ['title'];
79     elseif (isset($_PAGES[$_GET['list']][$_GET['act1']]['title'])) $_PAGE['t
80     itle'] = $_PAGES[$_GET['list']][$_GET['act1']]['title'];
81     elseif (isset($_PAGES[$_GET['list']]['title'])) $_PAGE['title'] = $_PAGE
82     S[$_GET['list']]['title'];
83
84     if (isset($_PAGES[$_GET['list']][$_GET['act1']][$_GET['act2']]['keywords
85     '])) $_PAGE['keywords'] = $_PAGES[$_GET['list']][$_GET['act1']][$_GET['a
86     ct2']]['keywords'];
87     elseif (isset($_PAGES[$_GET['list']][$_GET['act1']]['keywords'])) $_PAGE
88     ['keywords'] = $_PAGES[$_GET['list']][$_GET['act1']]['keywords'];
89     elseif (isset($_PAGES[$_GET['list']]['title'])) $_PAGE['keywords'] = $_P
90     AGES[$_GET['list']]['keywords'];
91
92     if (isset($_PAGES[$_GET['list']][$_GET['act1']][$_GET['act2']]['descript
93     ion'])) $_PAGE['description'] = $_PAGES[$_GET['list']][$_GET['act1']][$_
94     GET['act2']]['description'];
95     elseif (isset($_PAGES[$_GET['list']][$_GET['act1']]['description'])) $_P
96     AGE['description'] = $_PAGES[$_GET['list']][$_GET['act1']]['description']
97     ];
98     elseif (isset($_PAGES[$_GET['list']]['title'])) $_PAGE['description'] =
99     $_PAGES[$_GET['list']]['description'];
100
101     // načítanie šablony
102     if (file_exists('./template/'.$_PAGE['template'].'template.php') ) requ
103     ire './template/'.$_PAGE['template'].'template.php';
104     else die('ERROR: TEMPLATE FILE MISSING...');
105
106     ?>
```

4. Programátorom definované funkcie. (1/4)

```
1 <?php
2 // Connects to MySQL DB
3 function mysql_connection()
4 {
5     $connection = new mysqli("localhost", "meno", "heslo", "databaza", 0
6     , "/tmp/mysql51.sock");
7     if($connection->connect_errno) die('Chyba pri pripájaní na DB: '.$
8     connection->connect_error);
9     return $connection;
10 }
11
12 // Performs download query, returns fetched row array
13 function download_query($query)
14 {
15     $connection = mysql_connection();
16     $result = $connection->query($query);
17     if(!$result) die('Chyba pri dotazovaní.');
```

18 \$row = \$result->fetch_array(MYSQLI_BOTH);

19 \$result->close();

20 \$connection->close();

21 return \$row;

22 }

23 // Performs upload query, returns fetched row array

```
24 function upload_query($query)
25 {
26     $connection = mysql_connection();
27     $result = $connection->query($query);
28     if(!$result) die('Chyba pri dotazovaní.');
```

29 \$result->close();

30 \$connection->close();

31 return 1;

32 }

33 // Windows-1250 to ASCII

```
34 function win2ascii($str, $zamena)
35 {
36     $str = StrTr($str,
37     "\xE1\xE4\xE8\xEF\xEC\xE9\xED\xF2\xF3\xF4\xF5\xF6\xF7\xF8\xF9\xFA\xFC\xFD\xFE\xFF\xA0\xA1\xA2\xA3\xA4\xA5\xA6\xA7\xA8\xA9\xAA\xAB\xAC\xAD\xAE\xAF\xB0\xB1\xB2\xB3\xB4\xB5\xB6\xB7\xB8\xB9\xBA\xBB\xBC\xBD\xBE\xBF\xC0\xC1\xC2\xC3\xC4\xC5\xC6\xC7\xC8\xC9\xCA\xCB\xCC\xCD\xCE\xCF\xD0\xD1\xD2\xD3\xD4\xD5\xD6\xD7\xD8\xD9\xDA\xDB\xDC\xDD\xDE\xDF\xE0\xE1\xE2\xE3\xE4\xE5\xE6\xE7\xE8\xE9\xEA\xEB\xEC\xED\xEE\xEF\xF0\xF1\xF2\xF3\xF4\xF5\xF6\xF7\xF8\xF9\xFA\xFB\xFC\xFD\xFE\xFF");
38     $str = StrTr($str,
39     "\xF3\xF6\xF5\xF8\x9A\x9D\x9F\xFA\xFC\xFD\x9E\xF4\xBC\xBE",
40     "\x6F\x6F\x72\x72\x73\x74\x75\x75\x75\x79\x7A\x6F\x4C\x6C");
41     $str = StrTr($str,
42     "\xC1\xC4\xC8\xCF\xCC\xC9\xCD\xC2\xD3\xD4\xD8",
43     "\x41\x41\x43\x44\x45\x45\x49\x4E\x4F\x4F\x52");
44     $str = StrTr($str,
45     "\x8A\x8D\xDA\xDD\x8E\xD2\xD9\xEF\xCF",
46     "\x53\x54\x55\x59\x5A\x4E\x55\x64\x44");
47
48     if($zamena) return strtolower($str);
49     else return $str;
50 }
51 }
```

4. Programátorom definované funkcie. (2/4)

```
55
56 // sends an e-mail, reply to info@dobrovolnicieu.eu
57 function e_mail($to, $subject, $message)
58 {
59     $subject = win2ascii($subject, 0);
60     $headers = "MIME-Version: 1.0"."\\r\\n"."Content-type:text/html; chars
et=windows-1250"."\\r\\n"."From: info@dobrovolnicieu.eu"."\\r\\n"."Reply-To:
info@dobrovolnicieu.eu"."\\r\\n";
61     $message_css = '<html><head><style type="text/css">*{margin:0;padding
g:0;}body{line-height:1.5em;font-family:Georgia, sans-serif;background:n
one;color: #e1eaf0;}a{text-decoration: underline;color: #666666}a:hover{
text-decoration: none;}p{padding-bottom: 1.5em;}clear_both{clear: both;
}h1{margin-bottom: 1em;font-variant: small-caps;font-size: 1.5em;}#all{b
ackground:#73AAE0;background-size: 100%;border-radius:1em;width:90%;min-
width:450px;max-width:1260px;margin:0 auto;}#main{min-width:450px;max-wi
dth:1260px;}#header{margin: 30px 0 0 20px;padding-bottom: 30px;}#header
a.title{padding:20px 0 0 0;display: block;font-variant: small-caps;word-
spacing: 10px;letter-spacing: 2px;text-decoration: none;color: #ffffff;f
ont-size: 2.5em;text-shadow: 0px 2px 1px #4786b6;padding-bottom: 10px;}#
container{min-width:410px;max-width:1220px;margin: 0 20px 0 20px;}#insid
e_container{width: 100%;overflow: hidden;float: left;padding-top: 20px;b
ackground-color: #ffffff;border-radius: 1em;color: #666666;}#content{ove
rflow: hidden;padding: 0 20px 0 20px;}#footer{padding:0 0 20px 20px;colo
r: #B4D6E8;margin: 20px auto 60px 20px;letter-spacing: 0.1em;text-transf
orm: lowercase;}#footer a{color: #B4D6E8;}</style></head><body><div id="
all"><div id="main"><div id="header"><a class="title" href="http://www.d
obrovolnicieu.eu/homepage/" title="Dobrovoľníci EU - Domov" >Dobrovoľ
níci eu</a></div><div id="container"><div id="inside_container"><div id
="content">';
62     $message_css .= $message;
63     $message_css .= greeting();
64     $message_css .= '<p><em>Dobrovoľníci Ekonomickej univerzity<br />C
entrum severoamerických štúdií<br />Ústav medzinárodných programo
v<br />Ekonomická univerzita v Bratislave<br />Dolnozemska cesta 1<br
/>852 35 Bratislava</em></p></div></div></div><br class="clear_both" /><
div id="footer" ><a href="http://www.dobrovolnicieu.eu/administracia/" t
itle="Administrácia" >Administrácia</a> | <a href="http://www.dobrovoľ
nicieu.eu/kontakt/" title="Kontakt" >Kontakt</a> | 2013 © <a href="http
://www.dobrovolnicieu.eu/" title="Domovská stránka" >dobrovolnicieu.eu
</a></div></div></div></body></html>';
65     if(mail($to, $subject, $message_css, $headers)) return true;
66 }
67
68 // sends an e-mail, reply to customer
69 function e_mail_ponuka($to, $reply, $subject, $message)
70 {
71     $subject = win2ascii($subject, 0);
72     $headers = "MIME-Version: 1.0"."\\r\\n"."Content-type:text/html; chars
et=windows-1250"."\\r\\n"."From: ".$reply"."\\r\\n"."Reply-To: ".$reply"."\\r\\n
";
73     $message_css = '<html><head><style type="text/css">*{margin:0;padding
g:0;}body{line-height:1.5em;font-family:Georgia, sans-serif;background:n
one;color: #e1eaf0;}a{text-decoration: underline;color: #666666}a:hover{
text-decoration: none;}p{padding-bottom: 1.5em;}clear_both{clear: both;
}h1{margin-bottom: 1em;font-variant: small-caps;font-size: 1.5em;}#all{b
ackground:#73AAE0;background-size: 100%;border-radius:1em;width:90%;min-
width:450px;max-width:1260px;margin:0 auto;}#main{min-width:450px;max-wi
dth:1260px;}#header{margin: 30px 0 0 20px;padding-bottom: 30px;}#header
```

4. Programátorom definované funkcie. (3/4)

```
a.title{padding:20px 0 0 0;display: block;font-variant: small-caps;word-spacing: 10px;letter-spacing: 2px;text-decoration: none;color: #ffffff;font-size: 2.5em;text-shadow: 0px 2px 1px #4786b6;padding-bottom: 10px;}#container{min-width:410px;max-width:1220px;margin: 0 20px 0 20px;}#inside_container{width: 100%;overflow: hidden;float: left;padding-top: 20px;background-color: #ffffff;border-radius: 1em;color: #666666;}#content{overflow: hidden;padding: 0 20px 0 20px;}#footer{padding:0 0 20px 20px;color: #B4D6E8;margin: 20px auto 60px 20px;letter-spacing: 0.1em;text-transform: lowercase;}#footer a{color: #B4D6E8;}</style></head><body><div id="all"><div id="main"><div id="header"><a class="title" href="http://www.dobrovolnicieu.eu/homepage/" title="Dobrovoľníci EU - Domov" >Dobrovoľníci eu</a></div><div id="container"><div id="inside_container"><div id="content">';
74     $message_css .= $message;
75     $message_css .= '</div></div></div><br class="clear_both" /><div id="
"footer" ><a href="http://www.dobrovolnicieu.eu/administracia/" title="A
ministrácia" >Administrácia</a> | <a href="http://www.dobrovolnicieu.
eu/kontakt/" title="Kontakt" >Kontakt</a> | 2013 © <a href="http://www.
dobrovolnicieu.eu/" title="Domovská stránka" >dobrovolnicieu.eu</a></d
iv></div></div></body></html>';
76     if(mail($to, $subject, $message_css, $headers)) return true;
77 }
78
79 // determine greeting depending on time
80 function greeting()
81 {
82     $pozdrav = "<p>Želáme Vám ";
83     $hodina = date("G");
84     if($hodina<5) $pozdrav .= "dobrú noc.";
85     if($hodina<10) $pozdrav .= "krásne ráno.";
86     elseif($hodina<18) $pozdrav .= "produktívny deň.";
87     elseif($hodina<22) $pozdrav .= "príjemný večer.";
88     else $pozdrav .= "dobrú noc.";
89     $pozdrav .= "</p>";
90     return $pozdrav;
91 }
92
93 // Generates salt
94 function generate_salt()
95 {
96     $salt = strstr(base64_encode(mcrypt_create_iv(16, MCRYPT_DEV_URANDOM)
), '+', '.');
97     $cost = 10;
98     $salt = sprintf("$2a%02d$", $cost) . $salt;
99     return $salt;
100 }
101
102 // returns ID for organization or offer
103 function get_id($column, $table)
104 {
105     // retrieve number of rows
106     $connection = mysql_connection();
107     $result = $connection->query('SELECT '.$column.' FROM '.$table.'');
108
109     $row_n = $result->num_rows;
110
111     // retrieve max ID
112     $row = download_query('SELECT '.$column.' FROM '.$table.' ORDER BY '
```

4. Programátorom definované funkcie. (4/4)

```
        .$column.' DESC LIMIT 1;');
112     $row_max = $row[$column];
113
114     // if no ID is missing get another one, else determine first missing
    and use it as new ID
115     unset($login_id);
116     if($row_n!=$row_max)
117     {
118         $result = $connection->query('SELECT '.$column.' FROM '.$table.'
    ORDER BY '.$column.' ASC;');
119         $i=1;
120         while ($row = $result->fetch_array(MYSQLI_BOTH) and empty($login
    _id))
121         {
122             if($row[$column]!=$i) $login_id=$i;
123             $i++;
124         }
125     }
126     else
127     {
128         $login_id = $row_max+1;
129     }
130     $result->close();
131     $connection->close();
132     return $login_id;
133 }
134 ?>
```

5. Prehľadávanie databázy PHP algoritmom. (1/5)

```
1 <?php
2 // vytvorenie query pre DB
3 $spojenie = mysql_connection();
4
5 // zapisanie aktualnej kategorije do pamate
6 if(!isset($_POST['zameranie']) AND isset($_SESSION['zameranie']))
7 {
8     $_POST['zameranie']=$_SESSION['zameranie'];
9 }
10
11 elseif(!isset($_POST['zameranie']) AND !isset($_SESSION['zameranie']))
12 {
13     if($_GET['list']=="vzdelavanie") $_POST['zameranie']=1;
14     elseif($_GET['list']=="zivotne-prostredie") $_POST['zameranie']=2;
15     elseif($_GET['list']=="ludske-prava") $_POST['zameranie']=3;
16     elseif($_GET['list']=="socialna-pomoc") $_POST['zameranie']=4;
17     elseif($_GET['list']=="volny-cas") $_POST['zameranie']=5;
18 }
19 else
20     $_SESSION['zameranie']=$_POST['zameranie'];
21
22 if(isset($_SESSION['zameranie']))
23     $zamer=$_SESSION['zameranie'];
24 else
25     $zamer=$_POST['zameranie'];
26
27 // zapisanie ostatnych premennych
28 if (isset($_POST['stav_1']) OR isset($_POST['stav_2']))
29     $_SESSION['stav']=$_POST['stav_1']+$_POST['stav_2'];
30 else
31     $_SESSION['stav']=1;
32
33 if (isset($_POST['nazov']))
34 {
35     $_SESSION['nazov']=$_POST['nazov'];
36     $_SESSION['nazov_db']=explode(" ", str_replace("*", "%", win2ascii($
37     spojenie->real_escape_string($_POST['nazov']), 1)));
38 }
39 if (isset($_POST['miesto']))
40 {
41     $_SESSION['miesto']=$_POST['miesto'];
42     $_SESSION['miesto_db']=explode(" ", str_replace("*", "%", win2ascii(
43     $spojenie->real_escape_string($_POST['miesto']), 1)));
44 }
45 if (isset($_POST['popis']))
46 {
47     $_SESSION['popis']=$_POST['popis'];
48     $_SESSION['popis_db']=explode(" ", str_replace("*", "%", win2ascii($
49     spojenie->real_escape_string($_POST['popis']), 1)));
50 }
51 if (isset($_POST['organizacia']))
52 {
53     $_SESSION['organizacia']=$_POST['organizacia'];
```


5. Prehľadávanie databázy PHP algoritmom. (2/5)

```
54     $_SESSION['organizacia_db']=explode(" ", str_replace("*", "%", win2a
scii($spojenie->real_escape_string($_POST['organizacia']), 1));
55 }
56
57 // vyhľadavaci formular
58 if (isset($_GET['act1'])) echo '<form method="post" onsubmit="" action="
'.$_PAGE['base_url'].'.$_GET['list'].'/'.$_GET['act1'].'/?s=1">';
59 else echo '<form method="post" onsubmit="" action="'.$_PAGE['base_url'].
$_GET['list'].'/?s=1">';
60 echo '
61     <div id="filter_left">
62     <p>
63     <label for="stav" id="filter_stav"><strong>Chcem vidieť:</strong><
/label>
64     <br />
65     <input type="checkbox" '; if (($_SESSION['stav']==1) OR ($_SESSION[
'stav']==3)) echo 'checked="checked"'; echo ' name="stav_1" value="1" />
aktuálne ponuky
66     <br />
67     <input type="checkbox" '; if (($_SESSION['stav']==2) OR ($_SESSION[
'stav']==3)) echo 'checked="checked"'; echo ' name="stav_2" value="2" />
neaktuálne ponuky
68     </p>
69     <p>
70     <label for="zameranie" id="filter_zameranie"><strong>Zameranie:</st
rong></label>
71     <br />
72     <select tabindex="" name="zameranie" id="zameranie" onfocus="input_
focus(\'zameranie\');" onblur="input_blur(\'zameranie\');" >
73     <option value="0" '; if($zamer==0) echo 'selected="selected"'; ech
o '>všetky</option>
74     <option value="1" '; if($zamer==1) echo 'selected="selected"'; ech
o '>vzdelávanie</option>
75     <option value="2" '; if($zamer==2) echo 'selected="selected"'; ech
o '>životné prostredie</option>
76     <option value="3" '; if($zamer==3) echo 'selected="selected"'; ech
o '>ľudské práva</option>
77     <option value="4" '; if($zamer==4) echo 'selected="selected"'; ech
o '>sociálna pomoc</option>
78     <option value="5" '; if($zamer==5) echo 'selected="selected"'; ech
o '>voľný čas</option>
79     </select>
80     </p>
81     <input tabindex="5" type="submit" class="submit" name="send_filter"
value="Zobrazíť výsledok" />
82     </div>
83     <div id="filter_right">
84     <p><strong>Vyhľadaj kľúčové slovo:</strong></p>
85     <p>
86     <label for="nazov" id="filter_nazov">Názov ponuky:</label>
87     <input tabindex="1" type="text" name="nazov" id="nazov" onchange="a
jax_filter(nazov, this.value)" onfocus="input_focus(\'nazov\');" onblur="
input_blur(\'nazov\');" value="'.$_SESSION['nazov'].' " />
88     </p>
89     <p>
90     <label for="miesto" id="filter_miesto">Miesto:</label>
91     <input tabindex="2" type="text" name="miesto" id="miesto" onfocus="
input_focus(\'miesto\');" onblur="input_blur(\'miesto\');" value="'.$_SE
```

5. Prehľadávanie databázy PHP algoritmom. (3/5)

```
SSION['miesto'].'" />
92     </p>
93     <p>
94         <label for="popis" id="filter_popis">Popis:</label>
95         <input tabindex="3" type="text" name="popis" id="popis" onfocus="in
put_focus(\`'popis\`');" onblur="input_blur(\`'popis\`');" value="'.$_SESSIO
N['popis'].'" />
96     </p>
97     <p>
98         <label for="organizacia" id="filter_organizacia">Organizácia:</lab
el>
99         <input tabindex="4" type="text" name="organizacia" id="organizacia"
onfocus="input_focus(\`'organizacia\`');" onblur="input_blur(\`'organizaci
a\`');" value="'.$_SESSION['organizacia'].'" />
100    </p>
101    </div>
102    </form>
103    <br class="clear_both" />';
104
105    // vytvorenie dotazu
106    $query = "SELECT * FROM ORGANIZACIE INNER JOIN PONUKY ON PONUKY.ID_ORG=0
ORGANIZACIE.ORG_ID WHERE";
107    if($_SESSION['stav']==3) $query.= " (STAV = 1 OR STAV = 2)";
108    elseif($_SESSION['stav']==2) $query.= " STAV = 2";
109    elseif($_SESSION['stav']==1) $query.= " STAV = 1";
110    if($zamer!=0) $query.= " AND TYP = ".$zamer;
111    if($_SESSION['nazov']!=" AND count($_SESSION['nazov_db'])) {
112        $query.= " AND (";
113        foreach($_SESSION['nazov_db'] as $value) {
114            $query.= "NAZOV COLLATE CP1250_GENERAL_CI LIKE \"%".$value."%\"
AND ";
115        }
116        $query = substr($query, 0, -5).)";
117        unset($value);
118    }
119    if($_SESSION['miesto']!=" AND count($_SESSION['miesto_db'])) {
120        $query.= " AND (";
121        foreach($_SESSION['miesto_db'] as $value) {
122            $query.= "MIESTO COLLATE CP1250_GENERAL_CI LIKE \"%".$value."%\"
AND ";
123        }
124        $query = substr($query, 0, -5).)";
125        unset($value);
126    }
127    if($_SESSION['popis']!=" AND count($_SESSION['popis_db'])) {
128        $query.= " AND (";
129        foreach($_SESSION['popis_db'] as $value) {
130            $query.= "POPIS COLLATE CP1250_GENERAL_CI LIKE \"%".$value."%\"
AND ";
131        }
132        $query = substr($query, 0, -5).)";
133        unset($value);
134    }
135    if($_SESSION['organizacia']!=" AND count($_SESSION['organizacia_db']))
{
136        $query.= " AND (";
137        foreach($_SESSION['organizacia_db'] as $value) {
138            $query.= "ORG_NAZOV COLLATE CP1250_GENERAL_CI LIKE \"%".$value."
```

5. Prehľadávanie databázy PHP algoritmom. (4/5)

```
139     %\" AND ";
140     }
141     $query = substr($query, 0, -5).");
142     unset($value);
143 }
144 $query .= " ORDER BY NAZOV ASC;";
145 // vypis zoznamu ponuk
146 echo '<h2>Výsledky filtra:</h2>';
147 if($vysledok = $spojenie->query($query))
148 {
149     $pocet_riadkov=$vysledok->num_rows;
150     if ($pocet_riadkov)
151     {
152         if($spojenie->multi_query($query))
153         {
154             $i=0;
155             echo '<div id="filter_output">';
156             do {
157                 if($vysledok = $spojenie->store_result()) {
158                     while ($row = $vysledok->fetch_array(MYSQLI_BOTH)) {
159                         $i++;
160                         echo '<p><strong>'. $i .'</strong> <a href="'. $row['URL']
161 .'' >'. $row['NAZOV'] .'</a> ('. $row['MIESTO'] .' ) - '. $row['ORG_NAZOV'] .'<
162 /p>';
163
164                         if($_SESSION['stav']==3)
165                         {
166                             echo '<div class="odstup" ><strong>Aktuálna ponuka:
167 </strong> ';
168
169                             if ($row['STAV']==1) echo 'ÁNO';
170                             else echo 'NIE';
171                             echo '</div>';
172                         }
173                         if($zamer==0)
174                         {
175                             echo '<div class="odstup" ><strong>Zameranie:</stron
176 g> ';
177
178                             if ($row['TYP']==1) echo 'vzdelávanie';
179                             elseif ($row['TYP']==2) echo 'životné prostredie';
180
181                             elseif ($row['TYP']==3) echo 'ľudské práva';
182                             elseif ($row['TYP']==4) echo 'sociálna pomoc';
183                             elseif ($row['TYP']==5) echo 'voľný čas';
184                             echo '</div>';
185                         }
186                         echo '<br class="clear_both" />';
187                         if(($_SESSION['stav']==3) OR ($zamer==0)) echo '<br />';
188
189                     }
190                 }
191                 $vysledok->free();
192             }
193         } while ($spojenie->next_result());
194         echo '</div>';
195     }
196 }
197 else echo '<p>Podľa zadáných kritérií nebola nájdená žiadna pon
198 uka.</p>';
```

5. Prehľadávanie databázy PHP algoritmom. (5/5)

```
189     }  
190     else die('Chyba pri dotazovaní.');
```

191

```
192     $vysledok->close();  
193     $spojenie->close();  
194     ?>
```

6. Šablóna na generovanie XHTML dokumentov. (1/5)

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://w
ww.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" lang="sk" xml:lang="sk">
3 <head>
4 <meta http-equiv="content-language" content="sk" />
5 <meta http-equiv="content-type" content="text/html; charset=windows-
1250" />
6 <meta http-equiv="cache-control" content="no-cache" />
7 <meta http-equiv="pragma" content="no-cache" />
8 <meta name="keywords" content="<?=$_PAGE['keywords'];?>" />
9 <meta name="description" content="<?=$_PAGE['description'];?>" />
10 <title><?=$_PAGE['title'];?> | Dobrovoľníci EU</title>
11 <meta name="author" content="Martin Miškovič" />
12 <meta name="googlebot" content="index, follow" />
13 <meta name="robots" content="index, follow" />
14 <meta name="copyright" content="Dobrovoľníci Ekonomickej univerzit
y" />
15 <link rel="shortcut icon" href="<?=$_PAGE['base_url'];?>images/favic
on_deu.ico" type="image/x-icon" />
16 <link rel="canonical" href="<?=$_PAGE['base_url'];?>" />
17
18 <link rel="stylesheet" href="<?=$_PAGE['base_url']. 'template/'. $_PAG
E['template']. '/style.css';?>" type="text/css" />
19 <link rel="stylesheet" href="<?=$_PAGE['base_url']. 'template/'. $_PAG
E['template']. '/admin.css';?>" type="text/css" />
20 <link rel="stylesheet" href="<?=$_PAGE['base_url']. 'template/'. $_PAG
E['template']. '/contact_form.css';?>" type="text/css" />
21 <link rel="stylesheet" href="<?=$_PAGE['base_url']. 'template/'. $_PAG
E['template']. '/filter.css';?>" type="text/css" />
22 <link rel="stylesheet" href="<?=$_PAGE['base_url']. 'template/'. $_PAG
E['template']. '/jquery-ui-1.10.4.custom.min.css';?>" type="text/css" />
23
24 <script type="text/javascript" src="<?=$_PAGE['base_url'];?>configs/
js/contact.form.script.js"></script>
25 <script type="text/javascript" src="<?=$_PAGE['base_url'];?>configs/
js/jquery-1.10.2.js"></script>
26 <script type="text/javascript" src="<?=$_PAGE['base_url'];?>configs/
js/jquery-ui-1.10.4.custom.min.js"></script>
27
28 <?php // zmena pozadia textoveho menu ?>
29 <script type="text/javascript">
30 function changeHoverIn(id) {$('#'+id).addClass('hover');}
31 function changeHoverOut(id) {$('#'+id).removeClass('hover');}
32 </script>
33
34 <script type="text/javascript" >
35 $(document).ready(function(){
36
37 <?php // zmena pozadia textoveho menu na zaklade aktualnej pozicie
38 if($_GET['list']=='vzdelavanie') $menu=1;
39 elseif($_GET['list']=='zivotne-prostredie') $menu=2;
40 elseif($_GET['list']=='ludske-prava') $menu=3;
41 elseif($_GET['list']=='socialna-pomoc') $menu=4;
42 elseif($_GET['list']=='volny-cas') $menu=5;
43 else $menu=0;
44
```

6. Šablóna na generovanie XHTML dokumentov. (2/5)

```
45     if($menu)
46         echo '$("#n_". $menu.'').addClass("hover");';
47
48     if(($_GET['list']=="administracia") OR ($_GET['list']=="admin") OR (
$_GET['list']=="registracia"))
49     {
50         echo '
51         $("#content").css("padding","0 10px");
52         $(".datepicker").datepicker({
53             showOtherMonths: true,
54             selectOtherMonths: true,
55             minDate: 0,
56             showButtonPanel: true,
57             dateFormat: "dd. mm. yy",
58             firstDay: 1
59         });';
60     }
61     elseif (isset($_GET['act1']))
62     {
63         echo '
64         $("#contact_form").css("display","none");
65         $("#contact_click").click(function(){$("#contact_form").slideTog
gle("slow");});';
66     }
67     if ($_GET['list']=="registracia")
68     {
69         echo '
70         $("#contact_click").click(function(){$("#contact_form").slideTog
gle("slow");});';
71     }
72
73     // zabezpecenie prepiania medzi ponukou a vyhľadavanim
74     if ( (!isset($_GET['act1']) OR $_GET['s']==1) AND ($_GET['list']=="v
zdelavanie" OR $_GET['list']=="zivotne-prostredie" OR $_GET['list']=="lu
dske-prava" OR $_GET['list']=="socialna-pomoc" OR $_GET['list']=="volny-
cas"))
75     {
76         echo '
77         $("#content").hide();
78         $("#filter").show().addClass("shown");
79         $("#filter_click").click(function(){
80             if($("#filter").hasClass("shown")){
81                 $("#filter").animate({opacity: 0}, "slow");
82                 setTimeout(function(){$("#filter").hide();$("#content").show
();
83                 $("#content").animate({opacity: 1}, "slow");},500);
84                 $("#filter").removeClass("shown");
85             }
86             else {
87                 $("#content").animate({opacity: 0}, "slow");
88                 setTimeout(function(){$("#filter").show();$("#content").hide
();
89                 $("#filter").animate({opacity: 1}, "slow");},500);
90                 $("#filter").addClass("shown");
91             }
92         });';
93     }
94     else
```

6. Šablóna na generovanie XHTML dokumentov. (3/5)

```
95     echo '
96         $("#filter").hide();
97         $("#filter_click").click(function(){
98             if($("#filter").hasClass("shown")){
99                 $("#filter").animate({opacity: 0}, "slow");
100                setTimeout(function(){$("#filter").hide();$("#content").show
101                (});
102                $("#content").animate({opacity: 1}, "slow");};500);
103                $("#filter").removeClass("shown");
104            }
105            else {
106                $("#content").animate({opacity: 0}, "slow");
107                setTimeout(function(){$("#filter").show();$("#content").hide
108                (});
109                $("#filter").animate({opacity: 1}, "slow");};500);
110                $("#filter").addClass("shown");
111            }
112        });};
113    </script>
114    </head>
115    <body>
116        <?php // Google Analytics ?>
117        <script>
118            (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||f
119            unction(){
120                (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.cre
121            ateElement(o),
122                m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.in
123            sertBefore(a,m)
124            })(window,document,'script','//www.google-analytics.com/analytic
125            s.js','ga');
126
127            ga('create', 'UA-40768836-1', 'dobrovolnicieu.eu');
128            ga('send', 'pageview');
129        </script>
130
131        <div id="main">
132            <div id="header">
133                <a class="title" href="<?=$_PAGE['base_url'];?>homepage/" title=
134                "Dobrovolníci EU - Domov" >Dobrovolníci ekonomickej univerzity</a>
135                <a class="description" href="<?=$_PAGE['base_url'];?>homepage/"
136                title="Dobrovolníci EU - Domov" >informačný portál s aktuálnou pon
137                ukou pre dobrovoľníkov</a>
138            </div>
139            <div class="navigation" >
140                <a id="n_1" href="<?=$_PAGE['base_url'];?>vzdelavanie/" title="V
141                zdelavanie" >vzdelavanie</a>
142            </div>
143            <div class="navigation" >
144                <a id="n_2" href="<?=$_PAGE['base_url'];?>zivotne-prostredie/" t
145                itle="Životné prostredie" >životné prostredie</a>
146            </div>
147            <div class="navigation" >
148                <a id="n_3" href="<?=$_PAGE['base_url'];?>ludske-prava/" title="
149                Ľudské práva" >ľudské práva</a>
150            </div>
```

6. Šablóna na generovanie XHTML dokumentov. (4/5)

```
141     <div class="navigation" >
142         <a id="n_4" href="<?=$_PAGE['base_url'];?>socialna-pomoc/" title
="Sociálna pomoc" >socialna pomoc</a>
143     </div>
144     <div class="navigation" >
145         <a id="n_5" href="<?=$_PAGE['base_url'];?>volny-cas/" title="Voľný čas" >voľný čas</a>
146     </div>
147     <br class="clear_both" />
148     <div class="images">
149         <!-- Image author: AhaSoft, http://www.aha-soft.com -->
150         <a href="<?=$_PAGE['base_url'];?>vzdelavanie/" title="Vzdelávanie" onmouseover="changeHoverIn('n_1')" <?php if($menu!=1) echo 'onmouseout="changeHoverOut(\'n_1\')";?> ></a>
151     </div>
152     <div class="images">
153         <!-- Image author: Jasmina Stanojevic, http://noupe.com/author/jasmina-stanojevic -->
154         <a href="<?=$_PAGE['base_url'];?>zivotne-prostredie/" title="Životné prostredie" onmouseover="changeHoverIn('n_2')" <?php if($menu!=2) echo 'onmouseout="changeHoverOut(\'n_2\')";?> ></a>
155     </div>
156     <div class="images">
157         <!-- Image author: IconLeak, http://www.iconleak.com -->
158         <a href="<?=$_PAGE['base_url'];?>ludske-prava/" title="Ľudské práva" onmouseover="changeHoverIn('n_3')" <?php if($menu!=3) echo 'onmouseout="changeHoverOut(\'n_3\')";?> ></a>
159     </div>
160     <div class="images">
161         <!-- Image author: Klukeart, http://www.klukeart.com -->
162         <a href="<?=$_PAGE['base_url'];?>socialna-pomoc/" title="Sociálna pomoc" onmouseover="changeHoverIn('n_4')" <?php if($menu!=4) echo 'onmouseout="changeHoverOut(\'n_4\')";?> ></a>
163     </div>
164     <div class="images">
165         <!-- Image author: IconLeak, http://www.iconleak.com -->
166         <a href="<?=$_PAGE['base_url'];?>volny-cas/" title="Voľný čas" onmouseover="changeHoverIn('n_5')" <?php if($menu!=5) echo 'onmouseout="changeHoverOut(\'n_5\')";?> ></a>
167     </div>
168     <br class="clear_both" />
169     <?php if(($_GET['list']== "vzdelavanie") OR ($_GET['list']== "zivotne-prostredie") OR ($_GET['list']== "ludske-prava") OR ($_GET['list']== "socialna-pomoc") OR ($_GET['list']== "volny-cas"))
170     {
171         echo '<!-- Image author: Brsev, http://brsev.deviantart.com -->'
172         ;
173         echo '<div id="filter_click" ><img src=".'$_PAGE['base_url'].images/search.png /></div>';
174     }
```


6. Šablóna na generovanie XHTML dokumentov. (5/5)

```
174     ?>
175     <div id="container">
176         <?php
177             if(($_GET['list']=="administracia") OR ($_GET['list']=="admin"
) OR ($_GET['list']=="registracia"))
178             {
179                 echo '<div id="sidebar">';
180                 if (file_exists('./sections/side/side_'.$_GET['list'].'.php'
) ) require './sections/side/side_'.$_GET['list'].'.php';
181                 echo '</div><div id="block"></div>';
182             }
183             elseif($_GET['list']!="homepage")
184             {
185                 echo '<div id="filter">';
186                 require './sections/body/filter.php';
187                 echo '</div>';
188             }
189         ?>
190     <div id="content">
191         <?php
192             if (($_GET['list']=="administracia") OR ($_GET['list']=="adm
in") OR ($_GET['list']=="registracia") OR ($_GET['list']=="homepage"))
193             {
194                 if (file_exists('./sections/body/'.$_GET['list'].'/body_'
$_GET['act1'].'.php') ) require './sections/body/'.$_GET['list'].'/body_'
.$_GET['act1'].'.php';
195                 elseif (file_exists('./sections/body/body_'.$_GET['list']
'.php') ) require './sections/body/body_'.$_GET['list'].'.php';
196             }
197             elseif (isset($_GET['act1']))
198             {
199                 if (file_exists('./sections/body/vypis_ponuk.php') ) requi
re './sections/body/vypis_ponuk.php';
200             }
201             else
202             {
203                 if (file_exists('./sections/body/body_'.$_GET['list'].'.ph
p') ) require './sections/body/body_'.$_GET['list'].'.php';
204                 else require './sections/body/body_uvitanie.php';
205             }
206
207             if (file_exists('./template/'.$_PAGE['template'].'/footer.php'
) ) require './template/'.$_PAGE['template'].'/footer.php';
208         ?>
```

7. Definície kaskádových štýlov (CSS). (1/5)

```
1  /*
2
3      Designed by: Martin Miškovič
4      Version    : 2.0
5      Released   : 20140313
6
7      Background Image
8      Author     : Free CSS Templates, http://www.freecsstemplates.org
9
10 */
11
12 * {
13     margin: 0;
14     padding: 0;
15 }
16
17 body {
18     line-height: 1.5em;
19     font-family: Georgia, sans-serif;
20     background: #73AAE0 url('/images/bg.jpg') no-repeat;
21     background-size: 100%;
22     color: #e1eaf0;
23 }
24
25 h1, h2, h3, h4, h5 {
26     margin-bottom: 1em;
27     font-variant: small-caps;
28 }
29
30 h1 {
31     font-size: 1.5em;
32 }
33
34 h2, h3, h4, h5 {
35     font-size: 1.2em;
36 }
37
38 a {
39     text-decoration: underline;
40     color: #666666;
41 }
42
43 a:hover {
44     text-decoration: none;
45 }
46
47 p {
48     padding-bottom: 1.5em;
49 }
50
51 .center {
52     text-align: center;
53 }
54
55 .clear_both {
```

7. Definície kaskádových štýlov (CSS). (2/5)

```
56         clear: both;
57     }
58
59     .paragraph {
60         padding: 0 0 0 150px;
61     }
62
63     .paragraph_2 {
64         padding: 0 0 0 350px;
65     }
66
67     .narrow {
68         padding-bottom: 0;
69         float: left;
70         width: 150px;
71     }
72
73     .narrow_2 {
74         padding-bottom: 0;
75         float: left;
76         width: 350px;
77     }
78
79     .navigation {
80         font-variant: small-caps;
81         font-size: 1.2em;
82         float: left;
83         width: 20%;
84         max-width: 252px;
85         text-align: center;
86     }
87
88     .navigation img {
89         height: 100px;
90         padding: 10px 0 10px 0;
91     }
92
93     .navigation a {
94         text-decoration: none;
95         color: #666666;
96         background-color: #ffffff;
97         padding: 0 15px 2px 15px;
98         border-radius: 1.875em;
99     }
100
101     .navigation a:hover {
102         background-color: transparent;
103         color: #ffffff;
104         text-shadow: 0px 2px 1px #4786b6;
105     }
106
107     .navigation a.hover {
108         background-color: transparent;
109         color: #ffffff;
110         text-shadow: 0px 2px 1px #4786b6;
111     }
112
113     .images {
```

7. Definície kaskádových štýlov (CSS). (3/5)

```
114         float: left;
115         height: 100px;
116         width: 20%;
117         max-width: 252px;
118         padding: 10px 0 10px 0;
119         text-align: center;
120     }
121
122     .images a img {
123         border: none;
124         outline: none;
125     }
126
127     #main {
128         width: 92%;
129         min-width: 890px;
130         max-width: 1260px;
131         margin: 0 auto;
132     }
133
134     #header {
135         margin: 30px 0 0 20px;
136         padding-bottom: 30px;
137     }
138
139     #header a {
140         text-decoration: none;
141         color: #ffffff;
142         text-shadow: 0px 2px 1px #4786b6;
143     }
144
145     #header a.title {
146         display: block;
147         font-variant: small-caps;
148         word-spacing: 10px;
149         letter-spacing: 2px;
150         font-size: 2.5em;
151         padding-bottom: 10px;
152     }
153
154     #header a.description {
155         word-spacing: 2px;
156         letter-spacing: 1px;
157         font-size: 1.25em;
158     }
159
160     #container {
161         overflow: hidden;
162         float: left;
163         padding: 20px 0 0 0;
164         width: 100%;
165         background-color: #ffffff;
166         border-radius: 1em;
167         color: #666666;
168     }
169
170     #filter_click {
171         position: absolute;
```

7. Definície kaskádových štýlov (CSS). (4/5)

```
172         left: 30px;
173         cursor: pointer;
174     }
175
176     #filter_click img {
177         display: block;
178         float: left;
179         height: 32px;
180         width: 32px;
181         padding: 10px;
182         background-color: #ffffff;
183         border-radius: 1em;
184         border-top-left-radius: 1em;
185     }
186
187     #sidebar {
188         float: left;
189         width: 270px;
190         margin: 2px;
191     }
192
193     #sidebar .nadpis {
194         font-size: 1.1em;
195         font-weight: bold;
196     }
197
198     #sidebar a {
199         text-decoration: none;
200     }
201
202     #sidebar a:hover {
203         text-decoration: underline;
204     }
205
206     #block {
207         overflow: hidden;
208         width: 1px;
209         height: 1px;
210         background-color: #666666;
211         padding: 20px 0 10000px 0;
212         margin: -20px 0 -10000px 0;
213     }
214
215     #content {
216         overflow: hidden;
217         padding: 0 40px;
218     }
219
220     #footer {
221         padding-left: 20px;
222         color: #B4D6E8;
223         margin: 20px auto 60px auto;
224         letter-spacing: 0.1em;
225         text-transform: lowercase;
226     }
227
228     #footer a {
229         color: #B4D6E8;
```

7. Definície kaskádových štýlov (CSS). (5/5)

```
230 }
231
232 #footer .right {
233     float: right;
234     padding-right: 20px;
235 }
236
237 #admin p {
238     text-align: center;
239 }
240
241 #admin a {
242     text-decoration: underline;
243 }
244
245 #admin a:hover {
246     text-decoration: none;
247 }
```

8. Kontrola formulárov na strane klienta cez JavaScript. (1/11)

```
1  function input_focus(id)
2  {
3      $("#contact_"+id).addClass('current');
4  }
5
6  function input_blur(id)
7  {
8      $("#contact_"+id).removeClass('current');
9  }
10
11 function show_info(id)
12 {
13     if($("#info-"+id).hasClass('test'))
14     {
15         $("#info-"+id).slideUp(500);
16         $("#info-"+id).removeClass('test');
17     }
18     else
19     {
20         $("#info-"+id).slideDown(500);
21         $("#info-"+id).addClass('test');
22     }
23 }
24
25 function hide_info(id)
26 {
27     $("#info-"+id).slideUp(500);
28 }
29
30 function PonukyForm()
31 {
32     // check nazov
33     var name = $("#name").val();
34     if(name.length == 0)
35     {
36         $("#name-error").slideDown(500);
37         $("#name").focus();
38         return false;
39     }
40     else
41         $("#name-error").slideUp(500);
42
43     // check email
44     var email = $("#email").val();
45     var filter = /^[a-zA-Z0-9_-.-]+@((([a-zA-Z0-9-])+.)+)([a-zA-Z0-9]
{2,4})+$/;
46     if(email.length != 0)
47     {
48         if(!filter.test(email))
49         {
50             $("#email-error").slideDown(500);
51             $("#email").focus();
52             return false;
53         }
54         else
55             $("#email-error").slideUp(500);
```

8. Kontrola formulárov na strane klienta cez JavaScript. (2/11)

```
56     }
57     else
58     {
59         $("#email-error").slideDown(500);
60         $("#email").focus();
61         return false;
62     }
63
64     // check subject
65     var subject = $("#subject").val();
66     if(subject.length == 0)
67     {
68         $("#subject-error").slideDown(500);
69         $("#subject").focus();
70         return false;
71     }
72     else
73         $("#subject-error").slideUp(500);
74
75     // check message
76     var message = $("#message").val();
77     if(message.length == 0)
78     {
79         $("#message-error").slideDown(500);
80         $("#message").focus();
81         return false;
82     }
83     else
84         $("#message-error").slideUp(500);
85
86     // check for captcha
87     var captcha = $("#captcha").val();
88     if(captcha != captcha_c)
89     {
90         $("#captcha-error").slideDown(500);
91         $("#captcha").focus();
92         return false;
93     }
94     else
95         $("#captcha-error").slideUp(500);
96
97     return true;
98 }
99
100 function contactform()
101 {
102     // check nazov
103     var nazov = $("#nazov").val();
104     if(nazov.length == 0)
105     {
106         $("#nazov-error").slideDown(500);
107         $("#nazov").focus();
108         return false;
109     }
110     else
111         $("#nazov-error").slideUp(500);
112
113     // check sidlo
```


8. Kontrola formulárov na strane klienta cez JavaScript. (3/11)

```
114     var sidlo = $("#sidlo").val();
115     var filter = /^(.+)\s(.+)\s,\s+((([0-9]{5})|([0-9]{3}\s[0-9]{2}))\s+
s+(.+)$)/;
116     if(sidlo.length != 0)
117     {
118         $("#sidlo-error").slideUp(500);
119         if(!filter.test(sidlo))
120         {
121             $("#sidlo-error-2").slideDown(500);
122             $("#sidlo").focus();
123             return false;
124         }
125         else
126             $("#sidlo-error-2").slideUp(500);
127     }
128     else
129     {
130         $("#sidlo-error").slideDown(500);
131         $("#sidlo").focus();
132         return false;
133     }
134
135     // check email
136     var email = $("#email").val();
137     var filter = /^[a-zA-Z0-9_-]+@((([a-zA-Z0-9]{2,4})+$.)+([a-zA-Z0-9]
{2,4})+)$)/;
138     if(email.length != 0)
139     {
140         if(!filter.test(email))
141         {
142             $("#email-error").slideDown(500);
143             $("#email").focus();
144             return false;
145         }
146         else
147             $("#email-error").slideUp(500);
148     }
149     else
150     {
151         $("#email-error").slideDown(500);
152         $("#email").focus();
153         return false;
154     }
155
156     // check web
157     var web = $("#web").val();
158     var filter = /^[a-zA-Z0-9-]+\.\((([a-zA-Z0-9]{2,4})+)|((([a-zA-Z
0-9]{2,4})+.*))$)/;
159     if(web.length != 0)
160     {
161         if(!filter.test(web))
162         {
163             $("#web-error").slideDown(500);
164             $("#web").focus();
165             return false;
166         }
167         else
168             $("#web-error").slideUp(500);
```

8. Kontrola formulárov na strane klienta cez JavaScript. (4/11)

```
169     }
170
171     // check meno
172     var meno = $("#meno").val();
173     var filter = /^[a-zA-Z0-9_-]+$/;
174     if(meno.length < 6)
175     {
176         $("#meno-error").slideDown(500);
177         $("#meno").focus();
178         return false;
179     }
180     else
181     {
182         $("#meno-error").slideUp(500);
183         if(!filter.test(meno))
184         {
185             $("#meno-error-2").slideDown(500);
186             $("#meno").focus();
187             return false;
188         }
189         else
190             $("#meno-error-2").slideUp(500);
191     }
192
193     // check heslo
194     var heslo = $("#heslo").val();
195     var filter = /^[a-zA-Z0-9_-]+$/;
196     if(heslo.length < 6)
197     {
198         $("#heslo-error").slideDown(500);
199         $("#heslo").focus();
200         return false;
201     }
202     else
203     {
204         $("#heslo-error").slideUp(500);
205         if(!filter.test(heslo))
206         {
207             $("#heslo-error-2").slideDown(500);
208             $("#heslo").focus();
209             return false;
210         }
211         else
212             $("#heslo-error-2").slideUp(500);
213     }
214
215     // check heslo-2
216     var heslo_2 = $("#heslo-2").val();
217     if(heslo_2 != heslo)
218     {
219         $("#heslo-2-error").slideDown(500);
220         $("#heslo-2").focus();
221         return false;
222     }
223     else
224         $("#heslo-2-error").slideUp(500);
225
226     // check popis
```

8. Kontrola formulárov na strane klienta cez JavaScript. (5/11)

```
227     var msg = $("#popis").val();
228     if(msg.length == 0)
229     {
230         $("#popis-error").slideDown(500);
231         $("#popis").focus();
232         return false;
233     }
234     else
235         $("#popis-error").slideUp(500);
236
237     // check for captcha
238     var captcha = $("#captcha").val();
239     if(captcha != captcha_c)
240     {
241         $("#captcha-error").slideDown(500);
242         $("#captcha").focus();
243         return false;
244     }
245     else
246         $("#captcha-error").slideUp(500);
247
248     return true;
249 }
250
251 function ZmenaUdajov()
252 {
253     // check nazov
254     var nazov = $("#nazov").val();
255     if(nazov.length == 0)
256     {
257         $("#nazov-error").slideDown(500);
258         $("#nazov").focus();
259         return false;
260     }
261     else
262         $("#nazov-error").slideUp(500);
263
264     // check sidlo
265     var sidlo = $("#sidlo").val();
266     var filter = /^(.+)\s(.+)\s,\s+((([0-9]{5})|([0-9]{3}\s[0-9]{2}))\s+
s+(.+)$)/;
267     if(sidlo.length != 0)
268     {
269         $("#sidlo-error").slideUp(500);
270         if(!filter.test(sidlo))
271         {
272             $("#sidlo-error-2").slideDown(500);
273             $("#sidlo").focus();
274             return false;
275         }
276         else
277             $("#sidlo-error-2").slideUp(500);
278     }
279     else
280     {
281         $("#sidlo-error").slideDown(500);
282         $("#sidlo").focus();
283         return false;

```

8. Kontrola formulárov na strane klienta cez JavaScript. (6/11)

```
284     }
285
286     // check email
287     var email = $("#email").val();
288     var filter = /^[a-zA-Z0-9_-]+@((([a-zA-Z0-9-]{2,4})+)|([a-zA-Z
289     if(email.length != 0)
290     {
291         if(!filter.test(email))
292         {
293             $("#email-error").slideDown(500);
294             $("#email").focus();
295             return false;
296         }
297         else
298             $("#email-error").slideUp(500);
299     }
300     else
301     {
302         $("#email-error").slideDown(500);
303         $("#email").focus();
304         return false;
305     }
306
307     // check web
308     var web = $("#web").val();
309     var filter = /^[a-zA-Z0-9-]+\.\.((([a-zA-Z0-9]{2,4})+)|((([a-zA-Z
310     if(web.length != 0)
311     {
312         if(!filter.test(web))
313         {
314             $("#web-error").slideDown(500);
315             $("#web").focus();
316             return false;
317         }
318         else
319             $("#web-error").slideUp(500);
320     }
321     else
322     {
323         $("#web-error").slideDown(500);
324         $("#web").focus();
325         return false;
326     }
327
328     // check popis
329     var msg = $("#popis").val();
330     if(msg.length == 0)
331     {
332         $("#popis-error").slideDown(500);
333         $("#popis").focus();
334         return false;
335     }
336     else
337         $("#popis-error").slideUp(500);
338
339     // check for captcha
```

8. Kontrola formulárov na strane klienta cez JavaScript. (7/11)

```
340     var captcha = $("#captcha").val();
341     if(captcha != captcha_c)
342     {
343         $("#captcha-error").slideDown(500);
344         $("#captcha").focus();
345         return false;
346     }
347     else
348         $("#captcha-error").slideUp(500);
349
350     return true;
351 }
352
353 function ZmenaHesla()
354 {
355     // check meno
356     var meno = $("#meno").val();
357     var filter = /^[a-zA-Z0-9_.-]+$/;
358     if(meno.length < 6)
359     {
360         $("#meno-error").slideDown(500);
361         $("#meno").focus();
362         return false;
363     }
364     else
365     {
366         $("#meno-error").slideUp(500);
367         if(!filter.test(meno))
368         {
369             $("#meno-error-2").slideDown(500);
370             $("#meno").focus();
371             return false;
372         }
373         else
374             $("#meno-error-2").slideUp(500);
375     }
376
377     // check heslo
378     var heslo = $("#heslo").val();
379     var filter = /^[a-zA-Z0-9_.-]+$/;
380     if(heslo.length < 6)
381     {
382         $("#heslo-error").slideDown(500);
383         $("#heslo").focus();
384         return false;
385     }
386     else
387     {
388         $("#heslo-error").slideUp(500);
389         if(!filter.test(heslo))
390         {
391             $("#heslo-error-2").slideDown(500);
392             $("#heslo").focus();
393             return false;
394         }
395         else
396             $("#heslo-error-2").slideUp(500);
397     }
```

8. Kontrola formulárov na strane klienta cez JavaScript. (8/11)

```
398
399 // check heslo-2
400 var heslo_2 = $("#heslo-2").val();
401 if(heslo_2 != heslo)
402 {
403     $("#heslo-2-error").slideDown(500);
404     $("#heslo-2").focus();
405     return false;
406 }
407 else
408     $("#heslo-2-error").slideUp(500);
409
410 // check for captcha
411 var captcha = $("#captcha").val();
412 if(captcha != captcha_c)
413 {
414     $("#captcha-error").slideDown(500);
415     $("#captcha").focus();
416     return false;
417 }
418 else
419     $("#captcha-error").slideUp(500);
420
421 return true;
422 }
423
424 function novapponuka()
425 {
426     // check zameranie
427     var zameranie = $("#zameranie").val();
428     if(zameranie == 0)
429     {
430         $("#zameranie-error").slideDown(500);
431         $("#zameranie").focus();
432         return false;
433     }
434     else
435         $("#zameranie-error").slideUp(500);
436
437     // check osoba
438     var osoba = $("#osoba").val();
439     if(osoba.length == 0)
440     {
441         $("#osoba-error").slideDown(500);
442         $("#osoba").focus();
443         return false;
444     }
445     else
446         $("#osoba-error").slideUp(500);
447
448     // check tel
449     var tel = $("#tel").val();
450     var filter = /^(?([0-9]{9}))|(?([0-9]{3})\s+(?([0-9]{3})\s+(?([0-9]{3}))|(?([0-9]{2,3})\s+\s+(?([0-9]{3})\s+(?([0-9]{2})\s+(?([0-9]{2})))\s+$/);
451     if(tel.length != 0)
452     {
453         if(!filter.test(tel))
454         {
```

8. Kontrola formulárov na strane klienta cez JavaScript. (9/11)

```
455         $("#tel-error").slideDown(500);
456         $("#tel").focus();
457         return false;
458     }
459     else
460         $("#tel-error").slideUp(500);
461 }
462 else
463 {
464     $("#tel-error").slideDown(500);
465     $("#tel").focus();
466     return false;
467 }
468
469 // check email
470 var email = $("#email").val();
471 var filter = /^[a-zA-Z0-9_-]+@((([a-zA-Z0-9-])+(.[a-zA-Z0-9-
{2,4})+)$)/;
472 if(email.length != 0)
473 {
474     if(!filter.test(email))
475     {
476         $("#email-error").slideDown(500);
477         $("#email").focus();
478         return false;
479     }
480     else
481         $("#email-error").slideUp(500);
482 }
483 else
484 {
485     $("#email-error").slideDown(500);
486     $("#email").focus();
487     return false;
488 }
489
490 // check nazov
491 var nazov = $("#nazov").val();
492 if(nazov.length == 0)
493 {
494     $("#nazov-error").slideDown(500);
495     $("#nazov").focus();
496     return false;
497 }
498 else
499     $("#nazov-error").slideUp(500);
500
501 // check miesto
502 var miesto = $("#miesto").val();
503 if(miesto.length == 0)
504 {
505     $("#miesto-error").slideDown(500);
506     $("#miesto").focus();
507     return false;
508 }
509 else
510     $("#miesto-error").slideUp(500);
511
```

8. Kontrola formulárov na strane klienta cez JavaScript. (10/11)

```
512         // check termin
513         var termin = $("#termin").val();
514         if(termin.length == 0)
515         {
516             $("#termin-error").slideDown(500);
517             $("#termin").focus();
518             return false;
519         }
520         else
521             $("#termin-error").slideUp(500);
522
523         // check cas
524         var cas = $("#cas").val();
525         if(cas.length == 0)
526         {
527             $("#cas-error").slideDown(500);
528             $("#cas").focus();
529             return false;
530         }
531         else
532             $("#cas-error").slideUp(500);
533
534         // check exp
535         var exp = $("#exp").val();
536         var filter = /^[0-9]{2}+.\s+([0-9]{2})+.\s+2{([0-9]{3})+$/;
537         if(exp.length != 0)
538         {
539             $("#exp-error").slideUp(500);
540             if(!filter.test(exp))
541             {
542                 $("#exp-error-2").slideDown(500);
543                 $("#exp").focus();
544                 return false;
545             }
546             else
547                 $("#exp-error-2").slideUp(500);
548         }
549         else
550         {
551             $("#exp-error").slideDown(500);
552             $("#exp").focus();
553             return false;
554         }
555
556         // check poziadavky
557         var poziadavky = $("#poziadavky").val();
558         if(poziadavky.length == 0)
559         {
560             $("#poziadavky-error").slideDown(500);
561             $("#poziadavky").focus();
562             return false;
563         }
564         else
565             $("#poziadavky-error").slideUp(500);
566
567         // check vyhody
568         var vyhody = $("#vyhody").val();
569         if(vyhody.length == 0)
```


8. Kontrola formulárov na strane klienta cez JavaScript. (11/11)

```
570     {
571         $("#vyhody-error").slideDown(500);
572         $("#vyhody").focus();
573         return false;
574     }
575     else
576         $("#vyhody-error").slideUp(500);
577
578     // check popis
579     var msg = $("#popis").val();
580     if(msg.length == 0)
581     {
582         $("#popis-error").slideDown(500);
583         $("#popis").focus();
584         return false;
585     }
586     else
587         $("#popis-error").slideUp(500);
588
589     // check for captcha
590     var captcha = $("#captcha").val();
591     if(captcha != captcha_c)
592     {
593         $("#captcha-error").slideDown(500);
594         $("#captcha").focus();
595         return false;
596     }
597     else
598         $("#captcha-error").slideUp(500);
599
600     return true;
601 }
602
603 var captcha_a = Math.ceil(Math.random() * 10);
604 var captcha_b = Math.ceil(Math.random() * 10);
605 var captcha_c = captcha_a + captcha_b;
606 function generate_captcha(id)
607 {
608     var id = (id) ? id : 'contact_captcha';
609     $("#"+id).html(captcha_a + " + " + captcha_b + " = ");
610 }
```

9. Ochrana prístupu k administrácii webovej aplikácie pomocou PHP. (1/3)

```
1  <?php
2
3  // Set logout page
4  define('LOGOUT_URL', 'http://www.dobrovolnicieu.eu/');
5
6  // Set possible inactivity [minutes]
7  define('TIMEOUT_MINUTES', 15);
8
9  // Convert timeout to seconds
10 $timeout = (TIMEOUT_MINUTES == 0 ? 0 : time() + TIMEOUT_MINUTES * 60);
11
12 // Logout finalization
13 if(isset($_GET['logout']))
14 {
15     setcookie("verify", null, -1, '/');
16     $_SESSION['name'] = "";
17     session_destroy();
18     header('Location: ' . LOGOUT_URL);
19     exit();
20 }
21
22 if(!function_exists('showLoginPasswordProtect'))
23 {
24     // show login form
25     function showLoginPasswordProtect($error_msg) {?>
26         <h3 style="text-align:center;">Zadajte prosím svoje prihlasovacie m
27         eno a heslo</h3>
28         <div id="admin">
29             <form method="post">
30                 <p style="color:red" ><?php echo $error_msg; ?></p>
31                 <p><label>Meno:</label><input tabindex="1" class="text" type="in
32                 put" name="access_login" /></p>
33                 <p><label>Heslo:</label><input tabindex="2" class="text" type="p
34                 assword" name="access_password" /></p>
35                 <p><input tabindex="3" type="submit" class="submit" name="Submit
36                 " value="Odoslať" /></p>
37             </form>
38
39             <p>Ešte nemáte svoje meno a heslo?<br /><a href="http://www.dobr
40             volnicieu.eu/registracia/" >Registrujte sa u nás.</a></p>
41         </div>
42     </div>
43 </div>
44 <br class="clear_both" />
45 <div id="footer" >
46 <a href="http://www.dobrovolnicieu.eu/kto-sme/" title="Kto sme" >Kto sme
47 </a> | <a href="http://www.dobrovolnicieu.eu/cnas/" title="CNAS" >CNAS</
48 <a> | <a href="http://www.dobrovolnicieu.eu/kontakt/" title="Kontakt" >Ko
49 ntakt</a> | <a href="http://www.dobrovolnicieu.eu/vseobecne-podmienky/"
50 title="Všeobecné podmienky" >Všeobecné podmienky</a> | <a href="http
51 ://www.dobrovolnicieu.eu/administracia/" title="Administrácia" >Adminis
52 trácia</a>
53 <span class="right">
54 2013 &copy; dobrovolnicieu.eu
55 </span>
56 </div>
```

9. Ochrana prístupu k administrácii webovej aplikácie pomocou PHP. (2/3)

```
46 </div>
47 </body>
48 </html>
49 <?php
50 die();
51 }
52 }
53
54 // Get user provided password
55 if (isset($_POST['access_password']))
56 {
57     $connection = mysql_connection();
58     $login = $connection->real_escape_string($_POST['access_login']);
59     $pass = $connection->real_escape_string($_POST['access_password']);
60
61     // Retrieve hash from DB
62     $query = "SELECT ORG_HESLO, ORG_STAV FROM ORGANIZACIE WHERE `ORG_MENO`
63 = '".$_.$login."'";
64 $result = $connection->query($query);
65 if(!$result) die('Chyba pri dotazovaní.');
```

66 if(\$result->num_rows==1)

```
67 {
68     $item = $result->fetch_array(MYSQLI_BOTH);
69     $pass_hash = $item['ORG_HESLO'];
70     $is_active = $item['ORG_STAV'];
71 }
72 else
73 {
74     showLoginPasswordProtect("Zadané prihlasovacie meno neexistuje.");
75 }
76 $result->close();
77 $connection->close();
78
79 // Validate password
80 if(crypt($pass, $pass_hash) == $pass_hash)
81 {
82     // Show error when user is set as inactive
83     if($is_active <= 1)
84     {
85         // Set cookie
86         setcookie("verify", $pass_hash, $timeout, '/');
87         $_SESSION['name'] = $login;
88         unset($redirect, $login, $pass, $pass_hash, $item);
89     }
90     else
91     {
92         showLoginPasswordProtect("Váš účet je deaktivovaný. Kontaktuj
93 te prosím administrátora.");
94     }
95
96     // Clear $_POST array password variables
97     unset($_POST['access_login']);
98     unset($_POST['access_password']);
99     unset($_POST['Submit']);
100 }
101 else
102 {
103     showLoginPasswordProtect("Nesprávne heslo.");
```

9. Ochrana prístupu k administrácii webovej aplikácie pomocou PHP. (3/3)

```
102     }
103
104     }
105
106     else
107     {
108         // Check if password cookie is set
109         if (!isset($_COOKIE['verify']))
110         {
111             showLoginPasswordProtect("");
112         }
113
114         // Retrieve hash from DB
115         $connection = mysql_connection();
116         $query = "SELECT ORG_HESLO, ORG_STAV FROM ORGANIZACIE WHERE `ORG_MENO`
= '".$_SESSION['name']."'";
117         $result = $connection->query($query);
118         if (!$result) die('Chyba pri dotazovaní. ');
119         if ($result->num_rows==1)
120         {
121             $item = $result->fetch_array(MYSQLI_BOTH);
122             $pass_hash = $item['ORG_HESLO'];
123             $is_active = $item['ORG_STAV'];
124         }
125         $result->close();
126         $connection->close();
127
128         // Validate cookie
129         if ($_COOKIE['verify'] == $pass_hash AND $is_active <= 1)
130         {
131             // Prolong timeout
132             setcookie("verify", $pass_hash, $timeout, '/');
133         }
134         else {
135             showLoginPasswordProtect("");
136         }
137         unset($pass_hash, $item);
138     }
139 }
140
141 if(!empty($redirect)) header('Location: ' . $redirect);
142 ?>
```

10. Ukážka internetového dotazníka. (1/3)

Dobrovoľnícky portál pre študentov

Dobrý deň,

Dotazník slúži na zistenie spôsobu využívania dobrovoľníckych portálov študentami Ekonomickej univerzity v Bratislave, prípadne na zistenie záujmu o využívanie špecifického univerzitného portálu.

Získané údaje budú spracované a použité výlučne pre účely záverečnej bakalárskej práce s názvom "Vývoj webovej aplikácie v jazyku PHP" v odbore Hospodárska informatika na Fakulte hospodárskej informatiky.

Ďakujem Vám za zúčastnenie sa nasledovného prieskumu.

Martin Miškovič

* Required

Využívaš internetové portály, ktoré inzerujú ponuky pre dobrovoľníkov? *

- Áno
 Nie

Continue »

Dobrovoľnícky portál pre študentov

Charakterizoval by si portál, ktorý využívaš, ako prehľadný?

Vieš v ňom jednoducho a rýchlo vyhľadať zaujímavé ponuky, o ktoré máš záujem?

- Áno
 Nie

Poskytuje portál, ktorý využívaš, dostatočné množstvo informácií o dobrovoľníctve?

- Áno
 Nie

Vieš kontaktovať priamo konkrétnu neziskovú organizáciu cez webový formulár?

- Áno
 Nie

Je Ti z popisu ponúk očividné, čo sa od dobrovoľníka očakáva?

- Áno
 Nie

« Back

Continue »

10. Ukážka internetového dotazníka. (3/3)

Dobrovoľnícky portál pre študentov

Oslovil by ťa jednoduchý portál s ponukami pre študentov EU, kde by si jednotlivé ponuky vedel rýchlo vyhľadať?

- Áno
 Nie

Chcel by si mať na portáli uverejnené ponuky iba z Bratislavy a okolia alebo z celého územia SR?

- iba z Bratislavy a okolia
 z celého územia SR

Označ kritériá, podľa ktorých by si rád vyhľadával ponuky:

- kategória ponuky
 aktuálnosť ponuky
 názov ponuky
 popis činnosti
 miesto činnosti
 názov neziskovej organizácie
 výhody pre dobrovoľníka
 kontaktná osoba
 sídlo neziskovej organizácie

Uveď, aké dodatočné informácie by si sa o ponuke rád dozvedel:

Okrem povinných údajov, akými sú kontaktné údaje, názov, miesto a čas dobrovoľníckych aktivít

« Back

Submit