



ŠOLSKI CENTER CELJE

Splošna in strokovna gimnazija Lava

SPLETNI SERVIS KNJIGE GOSTOV

(raziskovalna naloga)

Mentorja:

Mojmir KLOVAR, univ. dipl. inž.

Karmen KOTNIK, univ. dipl. inž.

Avtor:

Alen BRILEJ 4.e

Šentjur, marec 2005

1. KAZALO

1.1 Vsebinsko kazalo

1. KAZALO	2
1.1 Vsebinsko kazalo.....	2
1.2 Slikovno kazalo	2
2. POVZETEK	3
3. UVOD	4
3.1 Predstavitev raziskovalnega problema	4
3.2 Hipoteze	4
3.3 Opis raziskovalnih metod.....	4
3.4 Pripomočki pri izdelavi	4
4. OSREDNJI DEL NALOGE	6
4.1 Predstavitev rezultatov raziskovanja.....	6
4.1.1 Podatkovna baza.....	6
4.1.2 Programiranje spletne strani.....	8
4.1.2.1 Delo s podatkovno bazo	8
4.1.2.2 Nastavitve uporabniškega vmesnika	10
4.1.2.3 Knjiga gostov	12
4.1.3 Uporaba knjige gostov	14
4.1.3.1 Navadna povezava na knjigo gostov	14
4.1.3.2 Knjiga v okvirju	14
4.1.3.3 Dodajanje knjige gostov v IFRAME.....	15
4.2 Razprava.....	16
5. ZAKLJUČEK	17
6. VIRI IN LITERATURA	18
7. ZAHVALA	19

1.2 Slikovno kazalo

Slika 1: Izgled uporabniškega vmesnika (Nastavitve knjige gostov).....	10
Slika 2: Način prikaza statistike obiskov.....	11
Slika 3: Izgled knjige gostov	13

2. POVZETEK

V raziskovalni nalogi sem raziskoval izdelavo kompleksnejše spletne strani z uporabniškim sistemom. Konkretno sem izdeloval spletni servis, ki registriranim uporabnikom omogoča, da na svojo stran postavijo knjigo gostov.

Stran sem oblikoval s pomočjo CSS-a, popestril sem jo še z JavaScripti. Kot interaktivni spletni jezik, ki da strani dinamičnost, sem uporabil PHP v povezavi s podatkovno bazo MySQL. Oba sta v spletnem programiranju med najbolj uporabljenimi.

Ugotovil sem, da si je potrebno delo pri taki aplikaciji razdeliti na podprobleme (na več delov). Sam sem to razdelil na izdelavo oblike strani (designa), ustvarjanje normalizirane podatkovne baze in programiranje dinamičnega dela spletne strani.

Kot rezultat menim, da sem dobil izdelek, kakršnega sem pričakoval pred samo izdelavo. Ker pa se uporabnost in učinkovitost takšnega servisa pokažeta šele po množični uporabi, bo servis na voljo tudi za uporabo. Dosegljiv bo na spletnem naslovu: <http://www.spletni-servis.com>. Tako bom lahko po nekem časovnem obdobju ugotovil, kaj so pomanjkljivosti in prednosti aplikacije in to s pridom uporabil pri razvoju podobnih strani.

3. UVOD

3.1 Predstavitev raziskovalnega problema

Cilj moje raziskovalne naloge je bil izdelati spletno aplikacijo, ki omogoča, da si obiskovalec, ki ima svojo spletno stran, nanjo postavi knjigo gostov. Knjiga gostov je lahko tako za lastnika kot tudi za obiskovalca strani zelo uporabna in zanimiva popestritev. Omogoča možnost, da lastnik strani dobi mnenje o strani s strani obiskovalcev. Pri izdelavi take knjige gostov, začetniški izdelovalec strani (predvsem domačih), navadno naleti na nekaj težav. Te ga marsikdaj odvrnejo od vključitve knjige gostov na stran. Najpogostejše težave so: brezplačna gostovanja (katerih se marsikdo poslužuje) v večini ne omogočajo podpore za noben spletni interaktivni jezik, hkrati ne nudijo podatkovne baze in izdelava take aplikacija je nekoliko težja, kot izdelava same statične strani. Zato sem se odločil, da lastnikom spletnih strani samo z nekaj kliki omogočim postavitev take aplikacije na njihovo stran.

3.2 Hipoteze

Moja glavna hipoteza je izdelati spletno aplikacijo (spletni servis), ki bo za uporabnika enostavna in prijazna za uporabo ter dovolj funkcionalna za njegove potrebe. Ostale hipoteze so še: spletni servis bo izdelan v roku 14 dni (pri profesionalnem delu je izdelava do dogovorjenega roka zelo pomembna), stran bo izdelana po pravilih spletnih standardov HTML 4.01 in CSS (pravilnost se preveri preko spletnega servisa Markup Validation Service) ter stran bo izdelana poleg osnove vsake spletne strani HTML-ja tudi s CSS-som (slogi za oblikovanje strani), JavaScript-om (spletni skriptni jezik) ter PHP-jem (to je osrednji predmet mojega raziskovanja, predvsem njegova uporabnost in raznolikost) v povezavi s podatkovno bazo MySQL.

3.3 Opis raziskovalnih metod

Funkcionalnost moje aplikacije, ki deluje kot spletni servis, se pokaže zagotovo v uporabniškem sistemu. Ta omogoča, da ima vsak uporabnik svojo knjigo gostov in da so nastavitve odvisne od posameznega uporabnika in njegovih želja, potreb. Tak sistem pa je potrebno seveda programirati, torej bom uporabljal metodo programiranja. Vsi ti podatki (podatki o lastniku knjige, njegovih nastavitvah, vsebina knjige gostov, statistika,...) se hranijo v podatkovno bazo. Del teh podatkov vpisuje lastnik knjige gostov, del obiskovalec, statistika pa se vpisuje (generira) samodejno oz. avtomatsko. Podatki, ki jih želi uporabnik bilo kje na strani, se preko PHP-ja, ki pridobi želeno vsebino iz podatkovno baze, pošljejo uporabnikovemu brskalniku, ki le te prikaže na zaslonu.

3.4 Pripomočki pri izdelavi

Za izdelavo take aplikacije v celoti je potrebnih kar nekaj programov. Osnova je seveda spletni brskalnik, s katerim brskamo po spletnih straneh in nam omogoča, da spletno stran sploh vidimo. Danes se množično uporablja že kar nekaj brskalnikov, ki pa se tudi med seboj malce razlikujejo (se držijo različnih standardov, dodajajo svoje,...). Zato nekateri lahko prikažejo isto stran tudi malenkost drugače kot drugi. Sam sem zaradi prilagoditve uporabljal za pregled tri glavne: Internet Explorer 6, Mozilla Firefox 1.0.1 in Opero 7.54. Za samo

izdelavo, torej pisanje kode (tako HTML, kot tudi CSS, JS, PHP,...), načeloma ne potrebujemo nič drugega kot preprost urejevalnik besedila – Beležnico (Notepad). Ampak da sem si zadevo olajšal, sem v večini uporabljal meni zelo uporaben (tako funkcionalno kot tudi vizualno) program PHP Expert Editor 3.2. Da pa se lahko PHP koda sploh izvaja, sem potreboval spletni strežnik, kateremu sem dodal še PHP modul (uporabljal sem PHP verzije 4.3.10, to je zadnja in zelo stabilna verzija iz PHP 4). Uporabil sem danes najpogosteje uporabljen strežnik Apache, konkretno verzije 1.39. Za shranjevanje podatkov sem moral namestiti še podatkovni strežnik, izbral sem MySQL 3.23. Za lažje delo s podatkovno bazo MySQL sem uporabil spletno aplikacijo phpMyAdmin 2.6.0. Nekaj slikovnega gradiva, ki ga vsebuje stran, sem izdelal oz. uredil v grafičnem programu Paint Shop Pro 7.

4. OSREDNJI DEL NALOGE

4.1 Predstavitev rezultatov raziskovanja

Ko sem že natančno vedel, kakšna bo idejna zasnova moja aplikacija, sem si zamislil približen izgled (design) strani in ga narisal na papir. Po tem vzorcu sem izdelal samo obliko strani. Za oblikovanje sem uporabljal CSS oblikovanje s slogi, katere sem priredil elementom kode HTML. Izdelal sem glavo, glavno kazalo in nogo strani. Pri tem sem se striktno držal standardov HTML 4.01 in CSS. Osredotočil sem se predvsem na pozicioniranje s pomočjo tabel in oblikovanje s CSS, manj sem uporabljal grafične elemente (slike). To je bil tudi moj namen na samem začetku. Zaradi tega npr. glava strani ni nekaj posebnega, ampak je le preprosto oblikovan napis. V spodnji desni rob strani sem dal tudi povezavi, kjer se lahko samo s klikom preveri ali je stran narejena po teh dveh standardih. Nato sem mogel začeti razmišljati o strukturi podatkovne baze, kamor se bodo shranjevali vsi podatki. Ko je bila baza skonstruirana, sem lahko začel s programiranjem dinamičnosti spletne strani.

4.1.1 Podatkovna baza

Ker program moje raziskovalne naloge temelji na množici dinamičnih podatkov, sem moral izbrati način za shranjevanje le teh. Na izbiro sem imel shranjevanje v datoteko ali podatkovno bazo. Izbral sem podatkovno bazo, konkretno je to MySQL. Za delo s to bazo se uporablja strukturirani jezik SQL (Structured Query Language). MySQL podatkovno bazo sem izbral iz več razlogov. Ključni so: dobra prilagojenost s PHP-jem, omogoča zahtevnejša povpraševanja, veliko število vpisov (velikost baze je načeloma odvisna le od moči serverja) in je pod javno licenco GNU, kar pomeni, da je v mojem primeru brezplačna. Za še lažje delo s podatkovno bazo, sem uporabil spletni program phpMyAdmin. To je že narejena aplikacija, pod GPL licenco, ki olajša delo z bazo in nam jo predstavi na enostaven način (načeloma se ta podatkovna baza upravlja preko ukazne vrstice).

Moja aplikacija torej za shranjevanje podatkov uporablja podatkovno bazo. Najprej, še preden sem začel s programiranjem spletne strani, sem moral zbrati vse podatke, ki so potrebni za delovanje aplikacije. Po daljšem premisleku glede strukture baze, sem dobil več relacijskih shem. Ker pa je snovanje baze ključnega pomena za delovanje, sem se moral prepričati ali je moja podatkovna baza res normalizirana in da je v tretji normalni obliki. To pomeni, da nima ponavljajočih se skupin, tranzitivne odvisnosti in delne odvisnosti. Ponavljajočih se skupin nima, ker se noben atribut ne ponavlja. Tranzitivne odvisnosti prav tako ni, saj noben atribut ni odvisen od atributa, ki ni ključ. Ker pa ni noben atribut odvisen le od dela sestavljenega ključa, tudi delne funkcionalne odvisnosti ni. Tako sem dobil skupaj 5 relacij. Glavna relacija je relacija uporabniki, ki med seboj povezuje še 3 relacije (vpisi, stevci, aktivn_gostje). To pomeni, da je primarni ključ relacije uporabniki tuji ključ v teh treh relacijah. S ključi sem omogočil medsebojno povezovanje relacij. Kot 5 relacija je relacija vprasanja, ki mi je v pomoč pri urejanju vsebinskega dela rubrike Pogosta vprašanja. Ta relacija ima tudi primarni ključ (unikatno, avtomatsko naraščajoče število), ki pa ne opravlja svojega glavnega pomena (povezovanje relacij med seboj).

Relacijske sheme za zgoraj omenjene normalizirane relacije so:

- **uporabniki:**

uporabniki(id_uporabnika#, ime, geslo, email, datum_registracije, predzadnja_prijava, zadnja_prijava, aktivacija, auto_prijava, level_administracije, barva_strani, naslov_knjige, barva_knjige, jezik, st_na_stran);

Komentar o relaciji:

Relacija vsebuje vse podatke o registriranih uporabnikih (lastnikih knjige). To so primarni ključ (id_uporabnika), osnovni podatki (ime – uporabniško ime uporabnika, geslo – geslo za dostop, email – njegov delujoči email), statistični podatki (datum_registracije – datum, ko se je uporabnik registriral, predzadnja_prijava – čas predzadnje prijave, zadnja_prijava – čas zadnje prijave), podatki glede nastavitve strani (aktivacija – ali je uporabnik aktiviran, auto_prijava – ali uporablja avtomatsko prijavo v uporabniški sistem, level_administracije – ali je administrator, barva_strani – barvna shema strani za administracijo knjige gostov) in nastavitve same knjige gostov (naslov_knjige – naslov knjige gostov, barva_knjige – izgled knjige, jezik – slovenski ali angleški jezik knjige, st_na_stran – število vpisov na stran v knjigi gostov).

- **vpisi:**

vpisi(id#, id_uporabnika, cas_vpisa, naziv, email, vsebina, aktivnost);

Komentar o relaciji:

Struktura te relacije je sestavljena iz atributov, ki so potrebni pri posameznem vpisu v knjigo (id – primarni ključ, cas_vpisa – čas ko je bil vpis oddan, naziv – ime oz. naziv, ki si ga izbere avtor vpisa, email – avtorjev email, vsebina – vsebina vpisa, aktivnost – ali je vpis prikazan v knjigi). Pomemben je tuji ključ, ki to relacijo poveže z relacijo uporabniki.

- **stevci:**

stevci(id_uporabnika + datum#, st_obiskov, st_prikazov);

Komentar o relaciji:

V tej relaciji pa je primarni ključ sestavljen iz ključa, saj z enodelnim ključem nebi mogel ustrezno uporabiti relacije, saj se beleži v to relacijo statistika obiska knjige gostov za vse uporabnike vsak dan posebej (st_obiskov – število različnih obiskovalcev na dan, st_prikazov – število prikazov na dan).

- **aktivni gostje:**

aktivni_gostje(ip_obiskovalca#, id_uporabnika, cas_obiska);

Komentar o relaciji:

Ta relacija je namenjena števcem obiska uporabnikove knjige gostov. Le tako lahko ločim število obiskov (različnih obiskovalcev na strani) od števila prikazov strani. Tukaj je primarni ključ ip_obiskovalca, id_uporabnika je tuji ključ, atribut cas_obiska pa vsebuje podatek o zadnjem prikazu strani.

- **vprasanja:**

vprasanja(id#, zaporedna_st_izpisa, vprasanje, odgovor);

Komentar o relaciji:

Relacija ima primarni ključ. Atribut zaporedna_st_izpisa pa je število, ki določa vrstni red izpisa vprašanj (atribut vprasanje) in odgovorov (atribut odgovor).

Relacije kot tudi attribute relacij sem poimenoval glede na njihovo vsebino oz. funkcionalnost. Atributom sem določil tudi lastnosti navezujoč na podatke, ki jih bodo vsebovali.

Glede na zgornje relacijske sheme sem se nato lotil kreiranja podatkovne baze in tabel s programskega stališča. S tem je bilo poskrbljeno za shranjevanje podatkov. Sledilo je programiranje, kjer se je pokazala uporabnost in smiselnost relacij, katere sem strukturiral in opisal v tem poglavju.

4.1.2 Programiranje spletne strani

Ko sem imel obliko strani izdelano in tabele v podatkovni bazi ustvarjene, sem lahko začel programirati dinamični del strani. Dele strani (oblika, sistem prijave, slike, skripte, vsebina) sem si organiziral tako, da sem vsakemu delu določil svojo mapo. Tako sem v bistvu ločil obliko strani, sistem za prijavo in vsebinski del (ta temelji na dinamičnih podatkih iz podatkovne baze). S tem sem si olajšal odvečno delo, ki bi ga porabil npr. za iskanje nekega dela kode. Glavna datoteka pa je seveda prva, torej index.php, ki vse te dele med seboj povezuje, glede na to, kaj obiskovalec strani zahteva.

4.1.2.1 Delo s podatkovno bazo

Celotna aplikacija v večini temelji na dinamičnih podatkih, ki se shranjujejo v podatkovno bazo. V tem razdelku bom na nekaterih primerih predstavil, opisal in ponazoril s programsko kodo, kako moja aplikacija shranjuje, ureja ali briše podatke iz MySQL baze.

Kot glavni funkcionalni del knjige gostov je seveda vpisovanje novih vpisov vanjo. Koda za vpis izgleda takole:

```
<?php
if(isset($_POST['vpisi_post']))
{ $email = filter($_POST['email_posiljatelja']);
  $naziv = filter($_POST['ime_posiljatelja']);
  $vsebina = filter($_POST['komentar']);
  if(strlen($naziv)>1 && strlen($vsebina)>19 && eregi("^([0-9a-z_\.]{1,}@([0-9a-z_\.]{1,})\.[0-9a-z_\.]{2,5})$", $email))
  { $cas = time();
    mysql_query("INSERT INTO vpisi (id_uporabnika,cas_vpisa,naziv,email,vsebina,aktivnost)
VALUES ($id,$cas,'$naziv','$email','$vsebina','1')");
  }
}
?>
```

Koda najprej preveri, če je obiskovalec pritisnil gumb za oddajo novega vpisa v knjigo. V primeru da je, se preko filtra, ki prečisti nedovoljene znake, priredi vsebina spremenljivkam.

Nad temi spremenljivkami so podani pogoji. V kolikor so vsi resnični, se podatki vpišejo v podatkovno bazo in vnos je zaključen.

Zanimiv način dela s podatkovno bazo je delovanje števecv. Ti beležijo število obiskov in prikazov uporabnikove knjige gostov. Število prikazov se poveča ob vsakem prikazu knjige, medtem ko se število obiskov poveča samo enkrat, če je znotraj časovnega območja, ki velja kot en obisk. Koda za generiranje števila prikazov in obiskov knjige gostov:

```
<?php
// spremenljivka $id je id lastnika te knjige gostov, za katerega se beleži obisk
$ip = $_SERVER['REMOTE_ADDR']; // obiskovalčev ip
$trajanje = 900; // trajanje obiska v sekundah
$datum = date("Y-m-d");
$time_datum = time();
$prej_datumi = mysql_query("SELECT * FROM stevci WHERE id_uporabnika='$id' ORDER BY
datum DESC LIMIT 1");
$prej_datum = mysql_fetch_assoc($prej_datumi);
if($prej_datum['datum'] != $datum)
{ mysql_query("DELETE FROM aktivni_gostje WHERE cas_obiska<$time_datum-$trajanje AND
id_uporabnika='$id'");
mysql_query("INSERT INTO stevci (id_uporabnika,datum,st_obiskov,st_prikazov) VALUES
('$id','$datum',0,0)");
}
$ze_al_ne = mysql_query("SELECT * FROM aktivni_gostje WHERE ip_obiskovalca='$ip' AND
id_uporabnika='$id'");
if(mysql_num_rows($ze_al_ne)==0)
{ mysql_query("INSERT INTO aktivni_gostje (ip_obiskovalca,cas_obiska,id_uporabnika) VALUES
('$ip','$time_datum','$id'");
mysql_query("UPDATE stevci SET st_obiskov=st_obiskov+1,st_prikazov=st_prikazov+1 WHERE
datum='$datum' AND id_uporabnika='$id'");
}
else
{ $neki = mysql_fetch_assoc($ze_al_ne);
if($neki['cas_obiska']<$time_datum-$trajanje)
{ mysql_query("UPDATE stevci SET st_obiskov=st_obiskov+1,st_prikazov=st_prikazov+1
WHERE datum='$datum' AND id_uporabnika='$id'");
}
else mysql_query("UPDATE stevci SET st_prikazov=st_prikazov+1 WHERE datum='$datum' AND
id_uporabnika='$id'");
mysql_query("UPDATE aktivni_gostje SET cas_obiska='$time_datum' WHERE
ip_obiskovalca='$ip' AND id_uporabnika='$id'");
}
?>
```

Skripta potrebuje za učinkovito ločevanje med obiski in prikazi še dodatno tabelo, katero sem poimenoval `aktivni_gostje`. Vanjo se zapisujejo vsi aktivni uporabniki. Na podlagi obiskovalčevega IP-ja se ugotovi ali je že bil na strani in če je bil v roku trajanja obiska (obisk sem definiral za 900 sekund, 15 minut). Da pa število aktivnih uporabnikov nebi zavzemalo prevelik del podatkovne baze, se enkrat dnevno brišejo za konkretnega uporabnika. Brišejo se vnosi, ki so izven roka trajanja obiska.

4.1.2.2 Nastavitve uporabniškega vmesnika

Najpomembnejši in osrednji del spletne strani (administracijskega vmesnika) je zagotovo uporabniški sistem. Deluje na preprostem principu: ob vpisu pravih podatkov je uporabnik prijavljen. Prijavljenost uporabnika se ohranja s pomočjo sej (sessions). Uporabnik ima tudi v nastavitvah možnost stalne prijave. Ta temelji na piškotkih (cookies), kar pomeni da mora imeti uporabnik omogočene. Ob vzpostavitvi prijave, se v sejo shrani tudi število vseh vpisov v knjigi gostov za konkretnega uporabnika in število novih vpisov. S shranjevanjem teh dveh podatkov v sejo, sem se izognil nepotrebnih povpraševanj (odvečnih) iz podatkovne baze.

Da se uporabnik sploh lahko prijavi, se mora na strani najprej registrirati. Ko se registrira, dobi na vpisani email (ta mora biti pravilen in delujoč) osnovne podatke o registraciji in povezavo za aktivacijo. Neaktiviran uporabnik se v sistem ne more prijaviti, zato je klik na povezavo, ob kateri se račun avtomatsko aktivira, obvezen.

V administraciji je definirano tudi polje, ki vsebuje rubrike, katere se lahko prikažejo le prijavljenemu uporabniku. Neprijavljen uporabnik dobi obvestilo o zavrnitvi dostopa .

Nato pa se glede na parameter stran, ki je poslan z metodo GET, na sredini pokaže ustrezna vsebina. S stavkom switch se omogoči, da se prikazujejo samo strani, ki so navadene v kodi. Če obiskovalec vtipka nedefinirano vsebino parametra, se mu prikaže prva stran, saj se v switch-u izvrši pogoj default. Tako sem poskrbel kako, kje in katere strani se naj prikažejo.

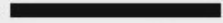
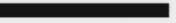
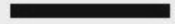
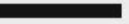


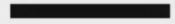
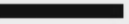


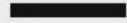
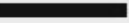
Registriran uporabnik (lastnik knjige gostov) ima administracijo za: Uporabnikove nastavitve, Nastavitve knjige, Urejanje vpisov knjige gostov in Statistiko obiska knjige. V primeru, da uporabnik ni registriran, ima vseeno na voljo nekaj funkcij: Registracijo, Kontakt z administratorji servisa, Pogosta vprašanja, Statistiko servisa in Pozabljeno geslo. Uporabniški vmesnik (na sliki uporabnik ureja nastavitve knjige) izgleda takole:

The screenshot shows the administration interface for 'Spletni-servis.com'. At the top, the user is identified as 'Spletni-servis' with 'Ni novih vpisov' and 'Št. vseh vpisov: 16'. A navigation menu on the left includes options like 'Prva stran', 'O spletnem servisu', 'Moje nastavitve', 'Nastavitve knjige', 'Urejanje vpisov', 'Moja statistika', 'Statistika servisa', 'Pogosta vprašanja', and 'Kontakt'. The main content area is titled 'Nastavitve knjige:' and contains several settings: 'Naslov knjige:' set to 'Knjiga gostov Spletni-servis.com', 'Barvna shema:' with radio buttons for 'privzeta' and 'vaša barva:' (with a color picker), 'Št. sporočil na stran:' set to '5', and 'Jezik knjige:' with radio buttons for 'SLO' and 'ENG'. Below these are buttons for 'Spremeni nastavitve' and 'Obnovi polja'. A message states: 'Vaša knjiga gostov se nahaja na spletnem naslovu: http://www.spletni-servis.com/knjiga_gostov/?uporabnik=Spletni-servis'. A code snippet is provided: 'Knjiga gostov'. The footer contains copyright information for 2005 and validation icons for W3C HTML 4.01 and CSS.

Slika 1: Izgled uporabniškega vmesnika (Nastavitve knjige gostov)

Vsaka od zgoraj naštetih funkcij administracije je izdelana uporabniku prijazno in funkcionalno zmogljivo. Pomembno je, da uporabniku ne dovoli vnosa, ki nebi bil ustrezen. Lep primer je Nastavitve knjige (slika 1). V primeru, da uporabnik izbere lastno barvno shemo, mora biti zapis kode barve ustrezen. Vsebovati mora 7 znakov, od tega mora biti prvi obvezno #.

Lastnik knjige gostov ima možnost nadzora nad obiskom njegove knjige gostov. Podatki se mu posredujejo tako v numerični kot tudi grafični obliki. Okrepljena podatka sta največja obiska in prikaza knjige gostov za določen dan.

Moja statistika:			
Datum	Število obiskov		Število prikazov
24.03.2005	133		291 
23.03.2005	101		210 
22.03.2005	126		351 
21.03.2005	101		214 
20.03.2005	69		194 
19.03.2005	73		219 

Slika 2: Način prikaza statistike obiskov

Na zanimiv način se generira tudi glavno kazalo administracije. Temelji na dveh poljih, katerih izbira enega od njiju prikazuje spodnja koda:

```
<?php
if(isset($_SESSION['id_uporabnika'])&&!isset($_POST['odjava']))
{ if($_SESSION['id_uporabnika'] > 0) $meni = array("Prva stran","O spletnem servisu","Moje nastavitve","Nastavitve knjige","Urejanje vpisov","Moja statistika","Statistika servisa","Pogosta vprašanja","Kontakt");
  else $meni = array("Prva stran","O spletnem servisu","Registracija","Statistika servisa","Pogosta vprašanja","Kontakt");
}
?>
```

In sicer, glede na to ali je uporabnik prijavljen izbere primerno polje in ga v zanki ponavlja, dokler ne pride do konca polja. S takšnim pristopom so spremembe zelo enostavne. V primeru da bilo kazalo statičeno, bi ob morebitnih popravkih imel veliko več dela.

Definirana je tudi ena konstanta. Vsebuje kodo dodatne barve na strani. Prijavljen uporabnik jo lahko spreminja (shrani se v podatkovno bazo) in se uporablja po celi strani, zato je v tem primeru hranjenje te barve kot konstanta pametna rešitev. Koda za izbiro barve, glede na to ali je uporabnik prijavljen ali ne, je:

```
<?php
if(isset($_SESSION['id_uporabnika'])&&!isset($_POST['odjava']))
{ if($_SESSION['id_uporabnika'] > 0) define("DODATNA_BARVA",$barva_strani);
}
else define("DODATNA_BARVA","#990000");
?>
```

V uporabniškem vmesniku in v uporabnikovi knjigi gostov imam dve zelo uporabni PHP funkciji. Ena je za povezavo s podatkovno bazo MySQL, druga pa je nekakšen filter za vse podatke, ki se preko kateregakoli obrazca na strani vnašajo s strani uporabnika. Ta prestreže in uredi nedovoljene znake (tudi HTML kodo), odvečne presledke na začetku in koncu. Preveri tudi, če ima PHP vklopljeno funkcijo `mysql_escape_string`. Če je nima, jo izvrši nad prejetim podatkom, ki ga potem vrne nazaj. Funkcija za filtriranje podatkov poslanih z metodo POST preko obrazca:

```
<?php
function filter($niz)
{ $niz = htmlspecialchars(trim($niz));
  if(!get_magic_quotes_gpc()) $niz = mysql_escape_string($niz);
  return $niz;
}
?>
```

Zanimiva je tudi funkcija Pozabljeno geslo. V kolikor je registriran uporabnik pozabil geslo za dostop, lahko dobi novo. Dobi ga na email ob vpisu pravilnega uporabniškega imena in emaila. Koda za naključno izbiranje 8 mestnega gesla iz nabora 33-tih znakov:

```
<?php
function geslo_random()
{ $salt = "abchefghjkmnpqrstuvwxyz0123456789";
  srand((double)microtime()*1000000);
  $i = 0;
  while ($i <= 7)
  { $stevilo = rand() % 33;
    $trenutna = substr($salt, $stevilo, 1);
    $geslo = $geslo . $trenutna;
    $i++;
  }
  return $geslo;
}
?>
```

Opisati velja še funkcijo Urejanje vpisov iz knjige gostov. Lastnik knjige gostov lahko spremeni vsebino vpisa, lahko ga onemogoči / omogoči za prikaz ali izbriše.

4.1.2.3 Knjiga gostov

Knjiga gostov je pravzaprav glavni del moje aplikacije. Za tako knjigo gostov je potrebna dovolj kvalitetna administracija, ki je opisana zgoraj.

Vsak registriran uporabnik ima svojo knjigo gostov. Ločijo se glede na URL naslov. V URL naslovu knjige gostov je navedeno uporabniško ime, tako da se ločijo knjige med seboj prav po tem atributu. Sprva sem imel namen narediti ločevanje kar po id-u registriranega uporabnika. Kasneje sem ugotovil, da je bolje, da je uporabniško ime, saj lahko v nasprotnem primeru nek obiskovalec pogleda večina knjig na servisu, kar pa ni namen servisa. Pri načinu, ki pa je uporabljen sedaj, mora poznati uporabniško ime uporabnika.

Skripta ob vpisu zahteva za knjigo gostov preveri, če se uporabnik, katerega uporabniško ime je bilo vneseno, nahaja v bazi in je aktiven. Če tega uporabnika ni, se obiskovalca napoti na stran o obvestilu, da knjiga gostov ne obstaja (od tukaj se obiskovalca avtomatsko po 5 sekundah premesti na osrednjo stran servisa).

V primeru da uporabnik obstaja, se obiskovalcu odpre knjiga gostov zahtevanega uporabnika.

Knjiga gostov Spletni-servis.com	
Strani vpisov: 1 2 3 4 > >>	
----> Avtor: Miha Datum vpisa: 23.03.2005 Čas vpisa: 19:50:18	Besedilo v knjigi gostov... Besedilo v knjigi gostov...
----> Avtor: Manca Datum vpisa: 23.03.2005 Čas vpisa: 19:50:16	Besedilo v knjigi gostov... Besedilo v knjigi gostov...
----> Avtor: David Datum vpisa: 23.03.2005 Čas vpisa: 19:49:58	Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov...
----> Avtor: Denis Datum vpisa: 23.03.2005 Čas vpisa: 19:47:12	Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov...
----> Avtor: Matej Datum vpisa: 23.03.2005 Čas vpisa: 19:46:45	Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov... Besedilo v knjigi gostov...
Strani vpisov: 1 2 3 4 > >>	
Vpišite se v knjigo gostov:	
----> Vaše ime:	<input type="text"/>
----> Vaš email naslov:	<input type="text"/>
----> Sporočilo:	<input type="text"/>
<input type="button" value="Vpis v knjigo gostov"/>	

Slika 3: Izgled knjige gostov

Obiskovalec knjige lahko pregleduje vpise v knjigi gostov (ki so odstranjeni). Knjiga gostov je tudi glede na nastavitve v ustreznih barvah in jeziku ter ima naslov določen s strani lastnika knjige, kot tudi vse prej omenjeno.

Nastavitve glede jezika sem uredil tako, da sem vse napise, ki se pojavijo v knjigi gostov, dal v dve datoteki. In sicer, v eni so napisi slovenski, v drugi pa angleški. Skripta na začetku glede na podatek o jeziku iz baze le še »include« (pokliče) ustrežno datoteko z napisi in jo uporabi.

Obiskovalec se lahko tudi vpiše v knjigo gostov (to je zaželeno oz. namen knjige gostov). To stori tako, da vpiše zahtevane podatke. Vpisati mora tudi svoj email naslov. Ker pa to danes ljudje ne počnejo radi zaradi strahu pred nezaželeno pošto (spamom), sem prikaz emailov zaščitil. Tako do emailov razni »spamboti« in robotki ne morejo, saj se geslo zakodira v heksadecimalno obliko. Funkcija, ki zakodira email naslov:

```
<?php
function kodiraj_email($email,$text=null)
{ $emailHex = bin2hex($email);
  $emailHex = chunk_split($emailHex,2,"%");
  $emailHex = "%" . substr($emailHex, 0, strlen($emailHex) - 1);
  $out = "<a class=\"povezava\" href=\"mailto:$emailHex\">";
  $out .= ($text? $text:str_replace("@", "@", $email));
  $out .= "</a>";
  return $out;
}
?>
```

V knjigi gostov deluje števec, katerega koda je predstavljena v rubriki Delo s podatkovno bazo. Beleži dnevni obisk in število prikazov.

4.1.3 Uporaba knjige gostov

Ko se uporabnik registrira na servisu, ima možnost, da uporabi knjigo gostov. Teorija delovanja takega servisa je, da se vsi podatki in datoteke nahajajo na strežniku servisa in ne na strežniku registriranega uporabnika (lastnika knjige gostov). Lastnik strani mora knjigo gostov vgraditi v svojo stran. Načeloma to ne drži, da jo vgradi, ampak samo na svoji strani pokliče povezavo (povezavo za njegovo knjigo gostov najde v servisu pod rubriko Nastavitve knjige), na kateri se nahaja njegova knjiga gostov. To lahko stori na 3 načine, našel bi se še kakšen. Vsak od njih ima določene prednosti in slabosti.

4.1.3.1 Navadna povezava na knjigo gostov

Ta način je najenostavnejši. Lastnik strani v svojo stran vgradi samo povezavo, ki kaže na knjigo gostov. HTML koda, ki jo vgradi uporabnik v svojo stran:

```
<a href=http://www.spletniservis.com/knjiga_gostov/?uporabnik=ime
target="_blank">Knjiga gostov</a>
```

4.1.3.2 Knjiga v okvirju

Če je spletna stran zgrajena s pomočjo okvirjev (framov), lahko lastnik strani v kodo, kjer definira okvirje, doda le ustrezno povezavo do knjige gostov. Ta način osebno ne priporočam, saj so danes tabele ali samo CSS oblikovanje primernejše za izdelovanje oblike strani. Poleg tega imajo okvirji nekaj slabosti (težave z iskalniki, pri dodajanju strani med priljubljene, nekateri starejši brskalniki okvirjev ne podpirajo,...).

4.1.3.3 Dodajanje knjige gostov v IFRAME

Ta način se mi zdi najbolj ustrezen, saj se ga da lepo vključiti v stran, da zglada le ta kot celota. V IFRAME značko (nekakšen »podokvirček«) dodamo samo ustrezno povezavo do knjige gostov. Temu okvirčku lahko določamo razne attribute: širino, višino, rob,... HTML koda, ki jo vgradi uporabnik v svojo stran:

```
<iframe src="http http://www.spletni-servis.com/knjiga_gostov/?uporabnik=ime"
scrolling="auto" marginwidth="0" marginheight="0" frameborder="0" vspace="0" hspace="0"
width="100%" height="100%"></iframe>
```

Kater način bo lastnik strani izbral, je odvisno od zgradbe strani in od želja lastnika samega. V vsakem primeru vsi načini delujejo. Seveda pri zgoraj napisanih kodah mora uporabnik namesto ime vpisati svoje uporabniško ime.

4.2 Razprava

Po koncu razvoja in testiranja spletnega servisa sem ugotovil, da sem pred pričetkom izdelave postavil hipoteze, katere lahko v celoti potrdim. Aplikacija je do uporabnika prijazna, dovolj funkcionalna ter vsebuje vse potrebne informacije kaj mora storiti za pridobitev in postavitev knjige gostov. Izdelal sem jo v točno določenem roku, v 14-tih dneh. V tem roku je bila končana aplikacija s programerske strani. Vendar ideje o tem, kakšna bo aplikacija, kaj bo vsebovala in kaj ponujala, sem snoval že kar nekaj časa prej. Pomembno je tudi, da sem se držal standardov HTML 4.01 in CSS, saj sem s tem lažje omogočil enak prikaz v večini spletnih brskalnikov. Spletna stran vsebuje poleg HTML-ja in CSS-a tudi nekaj JavaScript skript in seveda PHP kode v povezavi s podatkovno bazo MySQL.

Pri tej aplikaciji je glavni in najbolj kompleksen del prav uporabniški sistem. Zagotoviti je potrebno dovolj varnosti. V nasprotnem primeru lahko pride do kakšnih zlorab. Ampak v mojem primeru te zlorabe nebi mogle biti usodne (velikega pomena), ker nikjer ne operiram z zaupnimi podatki (npr. zelo tvegano bi bilo pri kakšni spletni trgovini, banki,...). Mislim, da sem za osnovno raven varnosti poskrbel dobro in ne pričakujem nepričakovanih zapletov glede varnosti v fazi delovanja servisa.

Mislim da administracija, preko katere uporabnik ureja knjigo gostov (tako vsebinsko kot tudi funkcijsko), postreže z več ali manj vsem, kar se potrebuje pri urejanju take knjige. Je tudi zelo prijazna do samega uporabnika, omogoča mu nastavitve dodatne barve, v kazalu je ves čas med sprehajanjem po administraciji zaznamek, ki prikazuje v katerem tematskem sklopu administracije se trenutno uporabnik nahaja. Za še večjo prilagodljivost uporabniku pri sami knjigi gostov bi lahko dodal še kakšno dodatno možnost (funkcijo), kot je npr.: vklop uporabe smeškov, možnost uporabe nekaterih HTML značk (npr. krepko, podčrtano, ležeče, povezava,...), še večjo barvno prilagojenost (ne samo ozadje knjige), še kakšen dodaten jezik,...

Preden bo spletna stran na voljo za javno uporabo, pa imam namen še narediti dodatno administracijo za administratorja celotnega servisa. Ideje, kaj potrebuje administrator servisa, sem zbiral skozi celoten razvoj same aplikacije. Na koncu sem prišel na sledeče funkcije: nadzor nad uporabniki (tudi onemogočanje le teh), spreminjanje njihovih nastavitvev knjige gostov (če je to potrebno), vmesnik za urejanje / dodajanje / vrstni red pogostih vprašanja, prejemanje sporočil iz rubrike Kontakt v administracijo in ne na email naslov ter končno tudi statistiko obiska same strani, torej števec.

Težave, na katere sem naletel skozi razvoj aplikacije sem po kratkem premisleku ponavadi kmalu in uspešno rešil ali pa sem jih obšel in našel drug način, da se jim izognem.

5. ZAKLJUČEK

Kot sem že navedel v razpravi, lahko na začetku postavljene hipoteze v celoti potrdim. Za končno oceno lahko rečem, da sem z izdelanim zadovoljen, čeprav bi se dalo nekaj stvari še dodati, verjetno tudi kaj izboljšati v tehnološkem smislu. Takšen izdelek sem tudi pred pričetkom izdelave pričakoval. Mogoče sem pričakoval, da bo dela manj, ampak nekatere navidezne malenkosti lahko vzamejo dosti časa in povzročijo kak večji problem.

Pred javno objavo strani, bom izdelal še administracijo za administratorja servisa, ki pa ni bila načrtovana za izdelavo v sklopu te raziskovalne naloge, saj sem šele kasneje ugotovil, da bo le ta zelo zaželen, če že ne obvezna.

Za izdelavo take aplikacije kot je ta servis menim, da je potrebno odlično poznavanje HTML-ja in CSS-a. Glede dinamičnosti take strani pa seveda morajo biti osnove PHP-ja že rutina, isto velja tudi za delo s podatkovno bazo. Seveda pri vsem tem je treba poudariti, da pri izdelavi take aplikacije dosti štejejo tudi izkušnje s programerskega področja in seveda možnost naprednega razmišljanja. Vsak del kode je potrebno razumeti, vedeti je potrebno zakaj je tako in ne drugače. Marsikateri programerski problem je treba pred realizacijo podrobno miselno in idejno preučiti.

Ker bo stran dosegljiva tudi na spletnem naslovu (<http://www.spletni-servis.com>), imam namen, v kolikor bo odziv na to knjigo gostov pozitiven, dodati še kakšno novo funkcijo servisa, ne samo knjigo gostov. Kot glavno idejo imam servis za števec obiska, saj so ti tudi pomembni za lastnika spletne strani. Posredujejo mu informacijo o obiskanosti strani. Kot drugo pa morda funkcijo Povej prijatelju ali sistem za izmenjavo pasic (bannerev). Vse to bi vključil v ta pravkar izdelani servis.

Upam, da sem s to aplikacijo kakšnemu izdelovalcu spletnih strani omogočil, da malce poživi svojo stran. To je bil tudi moj glavni namen, hkrati pa sem se želel preizkusiti na področju internetnih strani, kot so razni spletni servisi.

6. VIRI IN LITERATURA

- **BILKE, P.:** Spoznajmo PHP in MySQL. 1. izdaja. Nova Gorica : Flamingo, 2002.
- **ŠTRANCER, M. in S. KLEMEN:** PHP in MySQL: na spletnem strežniku Apache. 1. izdaja. Ljubljana : Pasadena, 2002.
- **ZANDSTRA, M.:** Naučite se PHP v 24 urah. 3. izdaja. Ljubljana : Pasadena, 2004.
- **Welling, L. in L. Thomson:** PHP and MySQL Web Development. 2. izdaja. Indianapolis (Indiana) : Sams, cop. 2003.
- URL naslov: <http://www.php.net>

7. ZAHVALA

To raziskovalno nalogo nisem izdelal v celoti sam, nekaj pomoči sem potreboval skozi celoten čas izdelave. Vse od ideje, njene realizacije, predstavitve le te v pisni obliki in do izdaje.

Zato se za pomoč zahvaljujem predvsem mojima mentorjema. Profesorju Mojmirju Klovarju, ki mi je stal ob strani glede idejne in programerske zasnove projekta ter profesorici Karmen Kotnik, ki mi je pomagala s svojimi nasveti pri delu s podatkovno bazo. Oba sta delež prispevala tudi k oblikovnemu in vsebinskemu delu moje raziskovalne naloge.

Hvala obema!