

Web dizajn i programiranje

Prof. dr.sc. Dragutin Kermek
Sveučilište u Zagrebu
Fakultet organizacije i informatike
Pavlinska 2, Varaždin 42000
dkermek@foi.hr

17. dio

Programiranje na strani poslužitelja

Uvod u programiranje na strani poslužitelja/CGI.
Programski jezik PHP.
Izvršavanje obrade (pretraživanje, zapis podataka i sl.) na strani poslužitelja.
Slanje e-mail poruke. Generiranje HTML stranica.
Ponovno korištenje koda.
Rad s datumom i vremenom. Objektna orijentacija.
Upravljenje pogreškama.
Korištenje i administriranje baze podataka (MySQL).
Realizacija autentikacije i autorizacije korisnika.



Autentikacija

Korištenje svih ili samo određenih dijelova aplikacije treba biti dozvoljeno samo ograničenom i poznatom skupu korisnika.

Svaki korisnik identificira se svojim korisničkim imenom koje je zaštićeno pripadajućom lozinkom.

Web autentikacija može biti provedena:

- internom autentikacijom Web poslužitelja – Apache s datotekom
- vlastitom autentikacijom s formularom, bazom podataka i pohranom privremenih podataka u **session** (sjednica).



Interna autentikacija Apache poslužitelja s datotekom

Konfiguracijom Web poslužitelja može se dopustiti autentikacija od određenog dijela stabla direktorija. Cijeli sustav autentikacije temelji se na datoteci **.htaccess**

U njoj se određuju konfiguracija autentikacije.

Da bi se mogla primijeniti autentikacija metodom .htaccess potrebno je u datoteci konfiguracija **httpd.conf** postaviti

AllowOverride AuthConfig

Poželjno je da se postavi u datoteci konfiguracija **httpd.conf** da se uz **index.html** i **index.php** tretira kao indeksna datoteka

DirectoryIndex index.html index.php

<http://httpd.apache.org/docs/2.2/howto/htaccess.html>



.htaccess

```
AuthType Basic
AuthUserFile D:/Apache/htdocs/PzaWeb/PHP_Primjeri/autentikacija/-
admin1/korisnici.txt
AuthName "Administracija sustava"
Require valid-user
```

tip autentikacije

naziv datoteke
autentikacije

naziv
autentikacije

potrebno za
autentikaciju



Unos korisnika

Da bi autentikacija bila od koristi potrebno je kreirati datoteku korisnika i upisivati korisnike i njihove pripadajuće lozinke. Koristi se komanda **htpasswd**

Kreiranje datoteke provodi se opcijom **-c** i koristi se samo kod prvog korisnika.

```
htpasswd -c korisnici.txt dkermek
New password: ****
Re-type new password: ****
Adding password for user dkermek
```

Dodavanje korisnika provodi se u nastavku.

```
htpasswd korisnici.txt pero
New password: ****
Re-type new password: ****
Adding password for user pero
```



Primjer sadržaja datoteke s korisnicima

```
pero:$apr1$Hs4.....$9ZeGmdJt8/tGeDqcca/kU.  
mato:$apr1$2t4.....$YYYvgivFt2f4/xtCRYAVm/  
stef:$apr1$Y35.....$q2ApJHNxeghDhPL4C1MUV/  
ivo:$apr1$kObvPbf2$1G6IfmXhu9zYXrpwwsJTS.  
bartol:$apr1$RSj0yfbJ$fvess83s99EDw9RSA.2lgn/  
cico:$apr1$03Zx2aAD$riF9uUUqYszqEluzRKopA1  
ivek:$apr1$iJ3rN2Ix$C7XdKSWqvTyL8hK2j1lXn1  
mica:$apr1$Ab/c1YXj$e8cEfnlNf5qd4qdMWDYBJ1  
jaca:$apr1$zkz.zvAJt$BLvLbZvxhHFQf57Rp06vh0  
luka:$apr1$IdKWYeLI$xlUmNOWU5Ia/PmERgKtWe.  
modric:$apr1$Xq4HGa53$/D3La5ev6i5NwAmChTywf0  
kruno:$apr1$CX3rD.dT$w9TG4UDdtbBOPVQNX/h6v1  
miki:$apr1$YrWQkqYa$GpZUD8H4nqKotd6SVgVPr1
```



Autentikacija Apache poslužiteljem

Kada se zatraži prikaz bilo kojeg dokumenta s tog direktorija odnosno bilo kojeg direktorija koji ga slijedi tada Web poslužitelj pokrene postupak autentikacije korisnika i prikazuje se maska za unos podataka:

Enter Network Password

Please type your user name and password.

Site: localhost

Realm: Administracija sustava

User Name

Password

☐ Save this password in your password list

OK Cancel

Podaci uspješne autentikacije vrijede tako dugo dok se ne zatvori preglednik u kojem je obavljena autentikacija.



Autentikacija Apache poslužiteljem / 1.

Nakon uspješne autentikacije Web poslužitelj upisuje podatke o korisniku u varijablama okoline:

Korisnik:

`$_SERVER["PHP_AUTH_USER"]`



Autentikacija Apache poslužiteljem / 2.

```
<html>
<body>
<?php
print "Dobrodošao: <b>" . $_SERVER["PHP_AUTH_USER"] . "</b>";
?>
</body>
</html>
```

[index.php – Autentikacija Apache poslužiteljem](#)



Pregled korisnika

```
<table border="1">
  <tr><td>Korisničko ime</td></tr>
  <?php
    $fn = "korisnici.txt";
    $fcontents = file($fn);
    while (list ($line_num, $line) = each($fcontents)) {
      list($pime) = explode(":", $line);
      $pime = htmlspecialchars(trim($pime));
      print "<td>$pime</td></tr>\n";
    }
  ?>
</table>
```

[listaKorisnika.php – Pregled korisnika](#)



Dodavanje korisnika putem Web aplikacije

Svaki dobar programmer će umjesto “spuštanja” na komandni mod operacijskog sustava napraviti Web aplikaciju koja će omogućiti jednostavnije i brže ažuriranje i dodavanje korisnika.

Zbog specifičnosti načina dodavanja korisnika tu se može raditi o potencijalnom **sigurnostnom propustu** jer se izvršava poziv funkcije sustava (**shell_exec**) iz PHP skripte za izvršavanje komande httpasswd.

Konfiguracijom PHP sustava može biti zabranjeno izvršavanje poziva funkcije sustava.



Autentikacija Apache poslužiteljem / 3.

```
<html>
<body>
<center>
<table border="0" cellspacing="5" cellpadding="5">
<form action="dodaj.php" method="POST">
<tr>
<td>Korisnik</td>
<td><input type="text" size="10" name="f_user"></td>
</tr>
<tr>
<td>Lozinka</td>
<td><input type="password" size="10" name="f_pass"></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit"
name="submit" value="Dodaj"></td> </tr> </form> </table>
</center>
</body>
</html>
```

[dodaj.html – Autentikacija Apache poslužiteljem](#)



Autentikacija Apache poslužiteljem / 4.

```
<?php
$f_user = $_POST["f_user"];
$f_pass = $_POST["f_pass"];

$sname = $_SERVER["SCRIPT_FILENAME"];
$pos = strrpos($sname, "/");
$dir = substr($sname, 0, $pos + 1);
$dat = $dir . "korisnici.txt";

$rez = shell_exec("D:/Apache/bin/htpasswd -b $dat $f_user $f_pass");

echo "<h1>Upis korisnika u datoteku autentikacije</h1>";
if(strlen($rez) != 0) {
    echo "<pre>$rez</pre>";
}
else {
    echo "Uspješan upis korisnika <b>$f_user</b>";
}
?>
```

[dodaj.php – Autentikacija Apache poslužiteljem](#)



Autentikacija korištenjem sesije

Session (sesija, sjednica) je privremeno spremište podataka koje se nalazi na poslužitelju i vezano je uz rad pojedinačnog korisnika. Spremljeni podaci u sesiji dostupni su kroz više različitih stranica (skripata) jednog korisnika aplikacije koji koristi određeni preglednik.

Sesija se kreira pozivom funkcije `session_start()`; koja se mora izvršiti prije bilo kojeg HTML sadržaja u skripti.

Podaci se upisuju i preuzimaju putem varijable `$_SESSION`, koja je asocijativni niz. U sesiju se mogu spremati jednostavni tipovi podataka (integer, double, string i sl.) kao i složeni (nizovi, objekti i sl.).

```
$_SESSION["aktivniKorisnik"] = "pero";  
$korisnik = $_SESSION["aktivniKorisnik"] ;
```

Brisanje podataka sesije provodi se funkcijom `session_unset()`;

Brisanje sesija provodi se funkcijom `session_destroy()`;

Korisnik nakon kreiranja sesije provodi svoje aktivnost s aplikacijom. Tijekom tog rada poslužitelj pamti vrijeme zadnje aktivnosti korisnika. Ukoliko je korisnik neaktivan dulje od određenog vremena (određuje se postavkama) poslužitelj će obrisati sesiju. Gašenjem i ponovnim pokretanjem poslužitelja brišu se sve sesije (uobičajeno).



Autorizacija

Provođenje pojedinih operacija u svim ili samo određenim dijelovima aplikacije treba biti dozvoljeno samo ograničenom i poznatom skupu autentičnih korisnika.

Autorizacija se treba provoditi za svaki zahtjev korisnika čime se usporava rad poslužitelja. Uobičajeno je da se podaci autorizacije upisuju u bazu podataka.

Zbog toga je praktično podijeliti web aplikaciju na javni i privatni dio, pri čemu se samo privatni dio štiti autorizacijskim mehanizmom.

Autorizacija se može koristiti, osim ograničavanja pristupa, i za filtriranje podataka/akcija u nekoj skripti. Npr. običan korisnik može vidjeti/mijenjati samo vlastite podatke dok "jaki korisnik" (npr. administrator) nema takva ograničenja



Autorizacija

Vrlo često korisnici se grupiraju u grupe prema organizacijskim, projektnim i sličnim shemama.

Poznati su jednostavni mehanizmi autorizacije kao što je npr. kod UNIX datotečnih sustava, koji se temelji na tri kategorije korisnika (vlasnik, član grupe, ostalo-svijet) i tri operacije (čitaj, piši, izvrši).

Složenija autorizacija može se provoditi na sličan način kao što je **ACL – Access Control Lists** (Novell NetWare, Windows). To znači da se svakom korisniku, grupi određuje koje operacije smije obavljati, odnosno kojim stranicima smije pristupiti.

U takvim modelima autorizacije efektivna prava korisnika proizlaze iz sinteze osobnih dozvola i grupnih svih grupa u kojima ima članstvo.



Autorizacija 1/3

Metode autorizacije u odnosu na skriptu mogu biti statičkog i dinamičkog tipa.

Kod **statičkog** tipa sama skripta sadrži podataka o zahtjevanom elementu autorizacije te ih provjerava u odnosu na korisnikove attribute koji su obično upisani u tablici/cama.

Kod **dinamičkog** tipa skripta provjerava dozvole izvođenja na temelju svog ID ili naziva u odnosu na podatke iz tablica.



Autorizacija 2/3

Metode autorizacije:

- autentikacijska**: temelji se na vrsti korisnika: anonimni i prijavljeni. Skripta provjerava zapis prijavljivanja u sesiji.
- na temelju uloge**: korisnicima se dodjeljuje uloga (u tablici "KORISNIK" postoji stupac "uloga" koji je vanjski ključ na tablicu "ULOGA"). Skripta provjerava zahtjevanu ulogu (ili jednu od više njih) za izvršavanje skripte u odnosu na ulogu korisnika.
- na temelju uloga**: korisnicima se dodjeljuju uloge (postoji tablica "KORISNIK_ULOGA" u kojoj su zapisi uloga pojedinog korisnika, vanjski ključevi prema tablicama "KORISNIK" i "ULOGA"). Skripta provjerava zahtjevanu ulogu (ili jednu od više njih ili nekoliko od više njih) za izvršavanje skripte u odnosu na uloge korisnika.
- na temelju ovlaštenja**: korisnicima se dodjeljuje izvršavanje određene skripte (postoji tablica "KORISNIK_SRIPTA" u kojoj su zapisi skripata pojedinog korisnika, vanjski ključ prema tablici "KORISNIK"). Skripta provjerava ima li korisnik dozvolu izvršavanja skipte tog naziva.



Autorizacija 3/3

- na temelju akcija**: korisnicima se dodjeljuju akcije koje smiju izvršavati (postoji tablica "KORISNIK_AKCIJA" u kojoj su zapisi akcija pojedinog korisnika, vanjski ključevi prema tablicama "KORISNIK" i "AKCIJA"). Skripta provjerava zahtjevanu akciju (ili jednu od više njih ili nekoliko od više njih) za izvršavanje skripte u odnosu na akcije korisnika.
- mješovite**: grupama i korisnicima dodjeljuje se izvršavanje određene skripte, grupama i korisnicima dodjeljuje se izvršavanje akcija i sl. Važeća dozvola je kumulativ dodjela kao pojedincu i na temelju grupa u kojima je član.



Dnevnik rada - log

Vrlo često postoji potreba za praćenjem rada korisnika, od prijavljivanja, preko svake zahtjevane stranice, do odjavljivanja.

Postoji dva načina praćenja rada korisnika:

- Apache log
- vlastita evidencija (datoteka ili baza podataka)



Apache log

Prema potrebama postavlja se u datoteci konfiguracija **httpd.conf** format zapisa podataka

```
ErrorLog logs/error.log  
LogLevel warn
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent
```

```
CustomLog logs/access.log common
```



Apache log / 1.

Primjer kod autentikacije metodom .htaccess

```
127.0.0.1 - - [08/Jan/2004:09:28:24 +0100] "GET
/PzaWeb/PHP_Primjeri/autentikacija/admin1/ HTTP/1.1" 401 475

127.0.0.1 - dkermek [08/Jan/2004:09:28:30 +0100] "GET
/PzaWeb/PHP_Primjeri/autentikacija/admin1/ HTTP/1.1" 200 70

127.0.0.1 - dkermek [08/Jan/2004:09:30:26 +0100] "GET
/PzaWeb/PHP_Primjeri/autentikacija/admin1/dodaj.html HTTP/1.1" 304 -

127.0.0.1 - dkermek [08/Jan/2004:09:30:46 +0100] "POST
/PzaWeb/PHP_Primjeri/autentikacija/admin1/dodaj.php HTTP/1.1" 200 95
```



Apache log / 2.

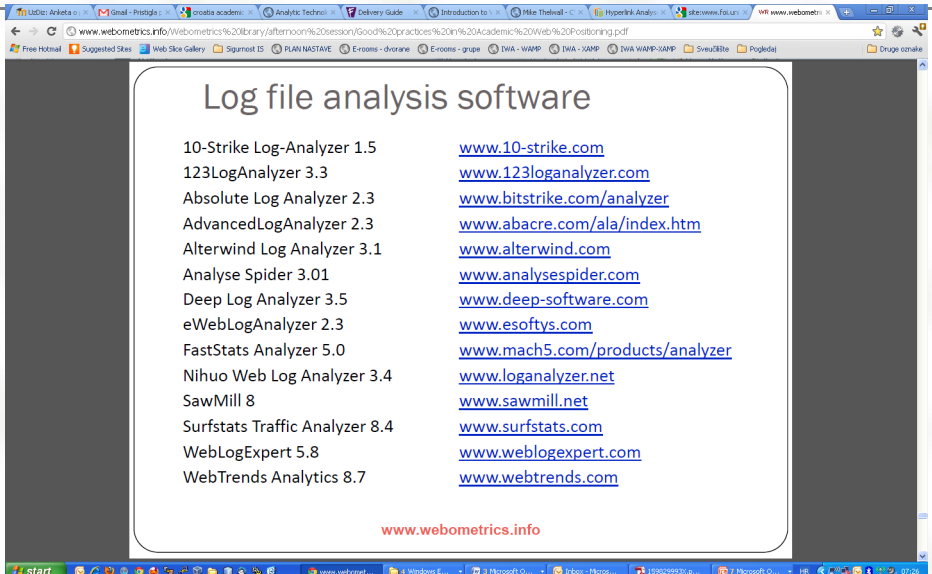
Za pretraživanja podataka mogu se koristiti razni alati koji su razvijeni za tu namjenu, ili se može razviti vlastiti alat.

Jedan od poznatih alata za analizu dnevnika rada je webalizer

<http://www.mrunix.net/webalizer/>.



Apache log / 2.



The screenshot shows a web browser window with the address bar displaying www.webometrics.info. The main content area is titled "Log file analysis software" and lists various tools with their corresponding websites. The list includes:

Software Name	Website
10-Strike Log-Analyzer 1.5	www.10-strike.com
123LogAnalyzer 3.3	www.123loganalyzer.com
Absolute Log Analyzer 2.3	www.bitstrike.com/analyzer
AdvancedLogAnalyzer 2.3	www.abacre.com/ala/index.htm
Alterwind Log Analyzer 3.1	www.alterwind.com
Analyse Spider 3.01	www.analysespider.com
Deep Log Analyzer 3.5	www.deep-software.com
eWebLogAnalyzer 2.3	www.esoftys.com
FastStats Analyzer 5.0	www.mach5.com/products/analyzer
Nihuo Web Log Analyzer 3.4	www.loganalyzer.net
SawMill 8	www.sawmill.net
Surfstats Traffic Analyzer 8.4	www.surfstats.com
WebLogExpert 5.8	www.weblogexpert.com
WebTrends Analytics 8.7	www.webtrends.com

At the bottom of the list, the URL www.webometrics.info is displayed. The browser's taskbar at the bottom shows several open applications, including a start button and various icons.

25

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin

Vlastiti dnevnik rad

Uobičajeno je da se podaci za dnevnik rada upisuju u bazu podataka, čime se otvara mogućnost jednostavnijih upita, odnosno pretraživanja.

Obično se upisuju sljedeći podaci:

- vrijeme
- url
- adresa računala korisnika
- korisnik
- status

Dnevnički okvir (Log framework)

Zbog različitih analiza (ispitivanje funkcionalnosti, testiranje, praćene rada korisnika, praćenje rada sustava i sl) korisno je ako se mogu koristiti različiti sustavi za spremanje, prikaz i pretraživanje podataka (konzola, sistemski dnevnik, vlastita tekstualna datoteka, baza podataka, mail i sl.).

Postoje gotovi dnevnički sustavi (klase) koji sadrže traženo rješenje npr. Log (<http://pear.php.net/package/Log>)



Aplikacijski okvir

Za ovu priliku izgradit će se mali aplikacijski okvir koji će integrirati određene temelje funkcionalnosti web aplikacije:

- postavljanje razine izvještaja o pogreškama
- obradu pogrešaka
- rad s bazom podataka
- rad sa sesijom
- autentikaciju
- autorizaciju
- dnevnik
- rad s košaricom i sl.



Aplikacijski okvir

Aplikacijski okvir sadrži nekoliko datoteka u kojima su ugrađeni njegovi dijelovi.

```
<?php
include_once('obradaPogresaka.php');
include_once('korisnik.php');
include_once('bazaPodataka.php');
include_once('autentikacija.php');
include_once('provjeraKorisnika.php');
include_once('dnevnik.php');
?>
```

[aplikacijskiOkvir.php – Aplikacijski okvir za autentikaciju sessionom i klasom](#)



Obrada pogrešaka

```
<?php
include_once('postavke.php');

set_error_handler('obradaPogresaka');

function obradaPogresaka($errno, $errstr, $errfile, $errline,
    $errcontext) {
    echo "Desila se pogreška kod izvršavanja!<br>";
    echo "Datoteka: $errfile<br>";
    echo "Liniya: $errline<br>";
    echo "Opis: $errstr<br>";
    echo "Kod: $errno<br>";
    exit;
}
?>
```

[obradaPogresaka.php – Obrada pogrešaka uz postavljanje razine izv. o pogreškama](#)



Obrada pogrešaka

```
<?php
$razina = 30719;
error_reporting($razina);
?>
```

[postavke.php](#) – Postavljenje razine izvještavanja - generira se



Klasa Korisnik/ 1

```
<?php
class Korisnik {
    private $kor_ime;
    private $ime;
    private $prezime;
    private $lozinka;
    private $vrsta = 9;
    private $prijavljen_od;
    private $status = 0;
    private $adresa;

    public function __construct() {
    }
}
```



Klasa Korisnik/ 1

```
public function set_podaci($p_kor_ime, $p_ime, $p_prezime,
    $p_lozinka, $vrsta) {
    $this->kor_ime = $p_kor_ime;
    $this->ime = $p_ime;
    $this->prezime = $p_prezime;
    $this->lozinka = $p_lozinka;
    $this->vrsta = $vrsta;
    $this->prijavljen_od = time();
    $this->status = 0;
    $this->adresa = $_SERVER["REMOTE_ADDR"];
}
```



Klasa Korisnik / 2

```
public function set_status ($p_status) {
    $this->status = $p_status;
}
public function get_status () {
    return $this->status;
}
public function get_kor_ime() {
    return $this->kor_ime;
}
public function get_ime_prezime() {
    return $this->ime . " " . $this->prezime;
}
public function get_prezime() {
    return $this->prezime;
}
public function get_ime() {
    return $this->ime;
}
public function get_vrsta() {
    return $this->vrsta;
}
```



Klasa Korisnik / 3

```
public function get_prijavljen_od() {  
    return date("d.m.Y H:i:s", $this->prijavljen_od);  
}  
public function get_aktivan() {  
    return time() - $this->prijavljen_od;  
}  
public function get_adresa() {  
    return $this->adresa;  
}  
}  
?>
```

[korisnik.php – Klasa korisnik za autentikaciju sessionom](#)



Rad s bazom podataka

```
<?php  
define('ADMINISTRATOR', '0');  
define('KORISNIK', '1');  
  
$bp_server = 'localhost';  
$bp_korisnik = 'WebDiP';  
$bp_lozinka = 'FOI';  
$bp_baza = 'PzaWeb';  
$bp_znakovi = 'utf8';  
if (!isset($dbc)) {  
    $dbc = false;  
}
```



Rad s bazom podataka / 1

```
function pripremiBazuPodataka() {  
    global $bp_server, $bp_korisnik, $bp_lozinka, $bp_baza,  
           $bp_znakovi, $dbc;  
  
    if ($dbc === false) {  
        $dbc = new mysqli($bp_server, $bp_korisnik,  
                          $bp_lozinka);  
        if(! $dbc) {  
            trigger_error("Problem kod povezivanja na bazu  
                           podataka!", E_USER_ERROR);  
        }  
        $dbc->set_charset($znakovi);  
    }  
  
    return $dbc;  
}
```



Rad s bazom podataka / 1

```
function zatvoriBazuPodataka() {  
    global $dbc;  
  
    if (isset($dbc) && $dbc instanceof mysqli) {  
        $dbc->close();  
        $dbc = false;  
    }  
}  
?>
```

[bazaPodataka.php – Rad s bazom podataka](#)



Autentikacija sessionom s klasom

```
<?
function autentikacija($user, $pass)
{
    $result = -1;
    $dbc = pripremiBazuPodataka();
    $sql = "select prezime, ime, lozinka, vrsta FROM POLAZNICI " .
        " where maticni_broj = '$user'";
    $rs = $dbc->query($sql);
    if(! $rs) {
        trigger_error("Problem kod upita na bazu podataka!",
            E_USER_ERROR);
    }
}
```



Autentikacija sessionom s klasom / 1

```
$korisnik = new Korisnik();
$broj = $rs->num_rows;
if($broj == 1) {
    list($prezime, $ime, $lozinka, $vrsta) =
        $rs->fetch_array();
    if($lozinka == $pass) {
        $korisnik->set_podaci($user, $ime, $prezime,
            $lozinka, $vrsta);
        $result = 1;
    } else {
        $result = 0;
    }
} else {
    $result = -1;
}
$korisnik->set_status($result);
return $korisnik;
}
?>
```

[autentikacija.php –Autentikacija sessionom](#)



Provjera korisnika i uloge

```
<?php
include_once('korisnik.php');

function provjeraKorisnika() {
    $korisnik = null;
    session_start();
    if (!isset($_SESSION["PzaWeb"])) {
        header("Location: error.php?e=2");
        exit();
    } else {
        $korisnik = $_SESSION["PzaWeb"];
        if($korisnik->get_status() != 1) {
            header("Location: error.php?e=2");
            exit();
        }
        if($korisnik->get_adresa() != $_SERVER["REMOTE_ADDR"]) {
            header("Location: error.php?e=2");
            exit();
        }
    }
    return $korisnik;
}
```

41

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



Provjera korisnika i uloge / 1.

```
function provjeraUloge($uloga) {
    session_start();
    $korisnik = isset($_SESSION["PzaWeb"]) ? $_SESSION["PzaWeb"] : "";
    if ($korisnik == "" || $korisnik->get_status() != 1 ||
        $korisnik->get_vrsta() != $uloga) {
        header("Location: error.php?e=2");
        exit();
    }
}
}??>
```

[provjeraKorisnika.php – Provjera korisnika \(da li je prijavljen\) i ima li ulogu](#)

42

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



Dnevnik

```
<?php
function dnevnik_zapis($tekst) {
    $korisnik = isset($_SESSION["PzaWeb"]) ?
        $_SESSION["PzaWeb"]->get_kor_ime() : "";
    $adresa = $_SERVER["REMOTE_ADDR"];
    $skripta = $_SERVER["REQUEST_URI"];
    $preglednik = $_SERVER["HTTP_USER_AGENT"];
    $dbc = pripremiBazuPodataka();
    $sql = "insert into DNEVNIK (korisnik, adresa, skripta, tekst," .
        " preglednik) values ('$korisnik', '$adresa', '$skripta', " .
        " '$tekst', '$preglednik')";

    $rs = $dbc->query($sql);
    if (!$rs) {
        trigger_error("Problem kod upisa u bazu podataka!" .
            $dbc->error, E_USER_ERROR);
    }
}
?>
```

[dnevnik.php – Upis u dnevni rada](#)



Pogreška

```
<?php
include_once('aplikacijskiOkvir.php');
dnevnik_zapis("Neuspješna autorizacija");
zatvoriBazuPodataka();
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
```



Pogreška

```
$e = $_GET["e"];
$message = "";
switch($e) {
    case -1:      $message = "Korisnik ne postoji.";
                  break;
    case 0:      $message = "Neispravno korisničko ime/lozinka.";
                  break;
    case 2:      $message = "Neautorizirani pristup.";
                  break;
    default:     $message = "Nepoznata pogreška.";
                  break;
}
print $message;
?>
</body>
</html>
```

[error.php – Pogreška](#)



Prijavljivanje korisnika - početak aplikacije

```
<?php
if(! isset($_SERVER["HTTPS"])) ||
    strtolower($_SERVER["HTTPS"]) != "on" {
    $adresa = 'https://' . $_SERVER["SERVER_NAME"] .
        $_SERVER["REQUEST_URI"];
    header("Location: $adresa");
    exit();
}
include_once('aplikacijskiOkvir.php');
dnevnik_zapis("Početak aplikacije");
zatvoriBazuPodataka();
?>
```



Prijavljivanje korisnika

```
<html>
<body>
<table border="0">
<form action="login.php" method="POST">
<tr><td>Korisnik</td>
<td><input type="text" size="10" name="f_user"></td></tr>
<tr><td>Lozinka</td>
<td><input type="password" size="10" name="f_pass"></td></tr>
<tr><td colspan="2" align="center"><input type="submit"
name="submit" value="Log In"></td> </tr> </form> </table>
</center>
</body>
</html>
```

[index.php – Prijavljivanje korisnika](#)



Provjera prijavljivanja korisnika

```
<?
$uri = $_SERVER["REQUEST_URI"];
$pos = strrpos($uri, "/");
$dir = $_SERVER["SERVER_NAME"] . substr($uri, 0, $pos + 1);

if (!isset($_SERVER["HTTPS"]) ||
    strtolower($_SERVER["HTTPS"]) != "on") {
    $adresa = 'https://' . $dir . 'index.php';
    header("Location: $adresa");
    exit();
}

include_once('aplikacijskiOkvir.php');
dnevnik_zapis("Prijava korisnika");
```



Provjera prijavljivanja korisnika

```
$f_user = $_POST["f_user"];
$f_pass = $_POST["f_pass"];

$korisnik = autentikacija($f_user, $f_pass);
zatvoriBazuPodataka();
if ($korisnik->get_status() == 1) {
    session_start();
    $_SESSION["PzaWeb"] = $korisnik;
    $adresa = 'http://' . $dir . 'aplikacija.php';
    header("Location: $adresa");
    exit();
}
else {
    $adresa = 'http://' . $dir . 'error.php?e=p';
    header("Location: $adresa" . $korisnik->get_status());
    exit();
}
?>
```

[login.php – Autentikacija cookieom](#)



Rad korisnika u zaštićenom dijelu aplikacije

```
<?php
include_once('aplikacijskiOkvir.php');
$korisnik = provjeraKorisnika();
dnevnik_zapis("Uspješna autorizacija");
?>
<html>
<body>
<center>
Pozdrav korisniku: <b>
<?php
    echo $korisnik->get_ime_prezime() . "</b> koji je prijavljen od: " .
        $korisnik->get_prijavljen_od() . " i aktivan : " .
        $korisnik->get_aktivan() . " sek";
?>
</center>
```



Rad korisnika u zaštićenom dijelu aplikacije / 1

```
<p align="right"><a href="kosarica.php">Košarica</a></p>
<p align="right"><a href="postaviPostavke.php">Postavi postavke</a></p>
<p align="right"><a href="dnevnik_priprema.php">
    Pregled dnevnika</a></p>
<p align="right"><a href="logout.php">Odjava</a></p>
<?php
$dbc = pripremiBazuPodataka();

$sql = "select prezime, ime, maticni_broj FROM POLAZNICI " .
        "order by prezime, ime";
$rs = $dbc->query($sql);
if(! $rs) {
    trigger_error("Problem kod upita na bazu podataka!",
        E_USER_ERROR);
}
?>
<table border=1><tr><td>Prezime</td><td>Ime</td>
    <td>Maticni_broj</td><td>Akcija</td></tr>
```



Rad korisnika u zaštićenom dijelu aplikacije / 2

```
<?php
while( list($prezime, $ime, $maticni_broj) = $rs->fetch_array()) {
    print "<tr><td>$prezime</td><td>$ime</td><td>$maticni_broj</td>";
    print "<td><a ref='kosarica.php?akcija=dodaj&korisnik=$maticni_broj'>
        Dodaj u košaricu</a></td></tr>\n";
}

zatvoriBazuPodataka();
?>
</table>
</body>
</html>
```

[aplikacija.php – Aplikacija](#)



Rad s košaricom

```
<?php
include_once('aplikacijskiOkvir.php');
$korisnik = provjeraKorisnika();
dnevnik_zapis("Uspješna autorizacija");
?>
<html>
<body>
<center>
Pozdrav korisniku: <b>
<?php
echo $korisnik->get_ime_prezime() . "</b> koji je prijavljen od: " .
      $korisnik->get_prijavljen_od() . " i aktivan : " .
      $korisnik->get_aktivan() . " sek";
?>
</center>
<p align="right"><a href="logout.php">Odjava</a></p>
<?php
if (isset($_SESSION["korisnici"])) {
    $korisnici = $_SESSION["korisnici"];
} else {
    $korisnici = array();
}
```



Rad s košaricom / 1

```
if (isset($_GET["akcija"]) && $_GET["akcija"] == "dodaj" &&
    isset($_GET["korisnik"])) {
    $noviKorisnik = $_GET["korisnik"];
    if(isset($korisnici[$noviKorisnik])) {
        $stariKorisnik = $korisnici[$noviKorisnik];
        print "Korisnik: $noviKorisnik: " .
              $stariKorisnik->get_ime_prezime() .
              " već postoji u košarici!";
    } else {
        $dbc = pripremiBazuPodataka();

        $sql = "select prezime, ime, lozinka, vrsta " .
              " FROM POLAZNICI where maticni_broj = '$noviKorisnik'";
        $rs = $dbc->query($sql);
        if(! $rs) {
            trigger_error("Problem kod upita na bazu
                          podataka!", E_USER_ERROR);
        }

        $broj = $rs->num_rows;
        $dodaniKorisnik = new Korisnik();
    }
}
```



Rad s košaricom / 2

```
if($broj == 1) {  
    list($prezime, $ime, $lozinka, $vrsta) =  
        $rs->fetch_array();  
    $dodaniKorisnik->set_podaci($noviKorisnik, $ime,  
        $prezime, $lozinka, $vrsta);  
    $korisnici[$noviKorisnik] = $dodaniKorisnik;  
    $_SESSION["korisnici"] = $korisnici;  
} else {  
    print "Korisnik: $noviKorisnik: ne postoji u  
        bazi podataka!<br>";  
}  
}  
}  
zatvoriBazuPodataka();
```



Rad s košaricom / 2

```
if (isset($_GET["akcija"]) && $_GET["akcija"] == "brisi" &&  
    isset($_GET["korisnik"])) {  
    $noviKorisnik = $_GET["korisnik"];  
    if(isset($korisnici[$noviKorisnik])) {  
        unset($korisnici[$noviKorisnik]);  
        $_SESSION["korisnici"] = $korisnici;  
    } else {  
        print "Korisnik: $noviKorisnik: NE postoji u  
            košarici!<br>";  
    }  
}  
?>
```



Rad s košaricom / 3

```
<br><a href="aplikacija.php">Pocetak aplikacije</a><br>
Korisnici u košarici<br>
<table border=1><tr><td>Prezime</td><td>Ime</td>
    <td>Maticni_broj</td><td>Akcija</td></tr>

<?php
foreach($korisnici as $k => $v) {
    $prezime = $v->get_prezime();
    $ime = $v->get_ime();
    $maticni_broj = $k;
    print "<tr><td>$prezime</td><td>$ime</td>
        <td>$maticni_broj</td>";
    print "<td>
        <a href='kosarica.php?akcija=brisi&korisnik=$maticni_broj'>
            Obrisi iz košarice</a></td></tr>\n";
    }
?>
</body>
</html>
```

[kosarica.php – Ras s košaricom u aplikaciji](#)



Odjava sa sustava

```
<?
include_once('aplikacijskiOkvir.php');
dnevnik_zapis("Odjava korisnika");
zatvoriBazuPodataka();
session_start();
unset($_SESSION["PzaWeb"]);
session_destroy();
header("Location: index.html");
?>
```

[logout.php – Odjavljivanje iz aplikacije](#)



Odabir razine izvještavanja o pogreškama

```
<?php
include_once('aplikacijskiOkvir.php');
provjeraUloge(ADMINISTRATOR);
dnevnik_zapis("Uspješna autorizacija");
zatvoriBazuPodataka();
?>
```



Odabir razine izvještavanja o pogreškama

```
<html>
<head>
<title>Odabir razine izvještavanja o pogreškama</title>
</head>
<body>
<h1>Odabir razine izvještavanja o pogreškama</h1>
<form method="POST" action="upisiPostavke.php">
Odaberi elemente:<br>
<input type="checkbox" name="E_ALL" value="1">E_ALL<br>
<input type="checkbox" name="E_ERROR" value="1">E_ERROR<br>
<input type="checkbox" name="E_WARNING" value="1">E_WARNING<br>
<input type="checkbox" name="E_PARSE" value="1">E_PARSE<br>
<input type="checkbox" name="E_NOTICE" value="1">E_NOTICE<br>
<input type="submit" value=" Pošalji ">
</form>
</body>
</html>
```

[postaviPostavke.php](#) – Postavljanje razine izvještavanja o pogreškama



Upis razine izvještavanja o pogreškama

```
<?php
include_once('aplikacijskiOkvir.php');
provjeraUloge(ADMINISTRATOR);
dnevnik_zapis("Uspješna autorizacija");
zatvoriBazuPodataka();
?>
<html>
<head>
<title>Upis razine izvještavanja o pogreškama</title>
</head>
<body>
<h1>Upis razine izvještavanja o pogreškama</h1>
<?php
if($_SERVER["REQUEST_METHOD"] != "POST") {
    echo "Samo za POST metodu!";
    exit;
}

$vrste = array("E_ALL", "E_ERROR", "E_WARNING", "E_PARSE", "E_NOTICE");
$razina = 0;
```



Upis razine izvještavanja o pogreškama / 1

```
foreach($vrste as $i => $v) {
    if(isset($_POST[$v])) {
        echo "i: " . $i . " v: " . $v . "<br>";
        switch ($i) {
            case 0: $razina |= E_ALL;
                        break;
            case 1: $razina |= E_ERROR;
                        break;
            case 2: $razina |= E_WARNING;
                        break;
            case 3: $razina |= E_PARSE;
                        break;
            case 4: $razina |= E_NOTICE;
                        break;
        }
    }
}
```



Odabir razine izvještavanja o pogreškama /2

```
$zapis = '<?php $razina = ' . $razina . ' . ' . "\n";  
$zapis .= 'error_reporting($razina);' . "\n";  
$zapis .= '?>' . "\n";  
$fp = fopen("postavke.php", "w");  
fwrite($fp, $zapis);  
fclose($fp);  
?>  
  
</body>  
</html>
```

[upisiPostavke.php – Upis razine izvještavanja o pogreškama](#)



Priprema za pregled dnevnika

```
<?php  
include_once('aplikacijskiOkvir.php');  
$korisnik = provjeraKorisnika();  
dnevnik_zapis("Uspješna autorizacija");  
zatvoriBazuPodataka();  
?>  
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Pregled dnevnika</title>  
    <meta charset="utf-8">  
  </head>  
  <body>  
    <form id=form1 method=post name="form1"  
      action="dnevnik_pregled.php">  
      <p>  
        <?php  
        if ($korisnik->get_vrsta() == ADMINISTRATOR) {  
          ?>  
          Korisnik:<input name="korisnik"><br>  
          <?php  
        }  
      </p>  
    </form>  
  </body>  
</html>
```



Priprema za pregled dnevnika

```
$poc = "01." . date('m.Y');  
$zav = date('d.m.Y');  
?>  
Od datuma: <input name = "odDatuma" value=  
"<?php echo $poc; ?>"><br>  
Do datuma: <input name = "doDatuma" value=  
"<?php echo $zav; ?>"><br>  
<input type = "submit" value = " Šalji ">  
  
</form>  
</body>  
</html>
```

[dnevnik_priprema.php – Priprema za pregled dnevnika](#)



Pregled dnevnika

```
<?php  
include_once('aplikacijskiOkvir.php');  
$korisnik = provjeraKorisnika();  
dnevnik_zapis("Uspješna autorizacija");  
?>  
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Pregled dnevnika</title>  
    <meta charset="utf-8">  
  </head>  
  <body>  
    <?php  
    if ($korisnik->get_vrsta() != ADMINISTRATOR) {  
      $kor_ime = $korisnik->get_kor_ime();  
    } else {  
      if (!isset($_POST["korisnik"]) || $_POST["korisnik"] == "") {  
        $kor_ime = "%";  
      } else {  
        $kor_ime = $_POST["korisnik"];  
      }  
    }  
  }  
>
```



Pregled dnevnika

```
if (!isset($_POST["odDatuma"])) {
    echo "Nije upisan Od datuma!";
    exit();
} else {
    $pd = date_create_from_format("d.m.Y", $_POST["odDatuma"]);
    $odDatuma = date_format($pd, "Y.m.d 00:00:00");
}
if (!isset($_POST["doDatuma"]) || empty($_POST["doDatuma"])) {
    $doDatuma = date("Y.m.d 23:55:55");
} else {
    $pd = date_create_from_format("d.m.Y", $_POST["doDatuma"]);
    $doDatuma = date_format($pd, "Y.m.d 23:55:55");
}

$dbc = pripremiBazuPodataka();

$sql = "select korisnik, adresa, skripta, tekst, preglednik, " .
    " vrijeme FROM DNEVNIK WHERE korisnik like '" .
    $korime . "' and " . "vrijeme between '" .
    $odDatuma . "' and '" . $doDatuma . "'";
```



Pregled dnevnika

```
$rs = $dbc->query($sql);
if (!$rs) {
    echo "Problem kod upita na bazu podataka!";
    exit;
}
print "<table border=1><tr><td>Korisnik</td><td>Adresa</td>".
"<td>Skripta</td><td>Tekst</td><td>Vrijeme</td></tr>\n";

while (list($korime, $adresa, $skripta, $tekst, $preglednik,
    $vrijeme) = $rs->fetch_row()) {
    $d = date("d.m.Y H:i:s", strtotime($vrijeme));
    print "<tr><td>$korime</td><td>$adresa</td>" .
        "<td>$skripta</td><td>$tekst</td><td>$d</td></tr>\n";
}
print "</table>\n";
zatvoriBazuPodataka();
?>
<br><a href='aplikacija.php'>Početak aplikacije</a><br>
</table>
</body>
</html>
```

[dnevnik_pregled.php – Pregled dnevnika](#)



Korištena i dodatna literatura

- <http://httpd.apache.org/docs-2.0/howto/auth.html>
- <http://httpd.apache.org/docs-2.0/howto/htaccess.html>
- <http://www.oreilly.com/catalog/webdbapps/chapter/ch08.html>
- <http://www.gknw.de/development/apache/>
- <http://pear.php.net/package/Log>

