

WEBLOG

SADRŽAJ

SADRŽAJ	1
UVOD	2
DIZAJN BAZE PODATAKA	3
IMPLEMENTACIJA BAZE PODATAKA	4
PROBNI UNOSI	5
PROGRAMIRANJE BLOGA	6
KONFIGURACIJA	6
KORISNIČKI INTERFEJS	6
header.php	7
footer.php	8
PRIKAZ UNOSA	9
index.php	10
login.php	12
logout.php	13
viewentry.php	14
viewcat.php	18
addcat.php	19
addentry.php	21
updateentry.php	22

UVOD

Weblog, ili skraćeno Blog, predstavlja ličnu internet prezentaciju koju možete dopunjavati bilo kakvim sadržajem koji želite da podijelite sa svijetom. Pored mogućnosti objavljivanja ličnih sadržaja, blogovi često omogućavaju: Komentarisane od strane posjetilaca i autora bloga, kategorizaciju tema i arhiviranje i pristup starim unosima. Ovaj projekat će sadržati sve nabrojane mogućnosti.

Koristićemo besplatne alate:

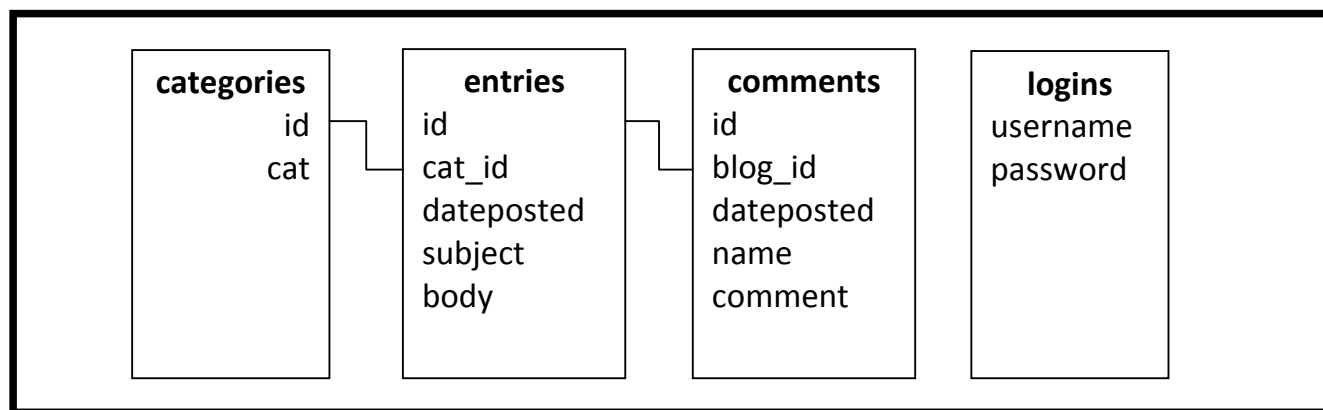
- **Notepad++**
Tekstualni editor sa raznim pomagalima za pisanje svih vrsta kodova.
Web sajt: <http://notepad-plus.sourceforge.net/uk/site.htm>
- **Wamp Server**
Instalacija prekonfigurisanih Apache servera, MySQL servera i PHP interpretera
Web adresa: <http://www.wampserver.com/en/>

DIZAJN BAZE PODATAKA

Prvi korak kod izrade ovog projekta je izrada baze podataka koja će predstavljati temelj našeg Blog-a. Za početak, napravićemo novu bazu podataka pomoću web aplikacije phpMyAdmin i nazvaćemo je *weblog*. Unutar te baze ćemo kreirati četiri tabele:

IME TABELE	SADRŽAJ TABELE
<i>Categories</i>	Različite kategorije bloga
<i>Entries</i>	Naši unosi
<i>Comments</i>	Komentari na unose
<i>logins</i>	Korisnička imena i lozinke

TABELA 1-1 – Spisak tabela i njihov sadržaj.



NACRT 1 – Čak i za jednostavnu aplikaciju kao što je ova za 'blogovanje', gledano na duže staze, pažljivo dizajnirana baza podataka može itekako skratiti posao.

Nacrt 1 prikazuje relacije između tabela.

Prva relacija je između kategorije bloga (tabela *categories*) i unosa na blogu (tabela *entries*). Da bi povezali unos sa kategorijom, vrijednost *id* odgovarajuće kategorije iz tabele *categories* ćemo sačuvati u polje *cat_id* pripadajućeg unosa iz tabele *entries*. *id* predstavlja *primarni ključ* dok *cat_id* predstavlja *spoljni ključ*. Ako se te vrijednosti poklapaju, one predstavljaju vezu između te dvije tabele.

Druga relacija je između unosa (tabela *entries*) i komentara na njih (tabela *comments*). Ova relacija je uspostavljena po istom principu. Polje *id* iz tebele *entries* je povezano sa poljem *blog_id* iz tabele *comments*. Ova relacija povezuje svaki komentar sa odgovarajućim unosom.

Poslednja tabela *logins* je tabela koja sadrži podatke za pristup (korisnička imena i odgovarajuće lozinke) i nema relacija sa drugim tabelama.

IMPLEMENTACIJA BAZE PODATAKA

Sada kada smo dizajnirali bazu, možemo da napravimo gore pomenute tabele takođe pomoću web aplikacije phpMyAdmin.

Polja *categories* table:

- *id* – tipa *TINYINT* (ovaj tip ćemo koristiti jer neće biti veliki broj kategorija), odabraćemo opciju *auto_increment* u *Extras* koloni. Štrikiraćemo i opciju *primary key*
- *cat* – tipa *VARCHAR*. Postavićemo veličinu na 20 jer ime kategorije neće prelaziti 20 karaktera.

Polja *entries* table:

- *id* – tipa *INT* (ovdje može biti veliki broj unosa), odabraćemo opciju *auto_increment* i štrikiraćemo opciju *primary key*.
- *cat_id* – tipa *TINYINT* (isti tip podatka kao i polje *id* u tabeli *categories* na koje upućuje).
- *dateposted* – tipa *DATETIME*. Ovaj tip podatka čuva trenutni datum i vrijeme u internacionalnom formatu po ISO standardu, koji je prilično neshvatljiv, ali ga je moguće kasnije formatirati pomoću PHP-a.
- *subject* – tipa *VARCHAR*. Postavićemo dužinu polja na 100 karaktera.
- *body* – tipa *TEXT*. Tip podatka *TEXT* je najbolji izbor za čuvanje vrlo velikih tekstualnih zapisa. Ne moramo da postavljamo veličinu polja.

Polja *comments* table:

- *id* – tipa *INT* (ovdje može biti veliki broj unosa), odabraćemo opciju *auto_increment* i štrikiraćemo opciju *primary key*.
- *blog_id* – tipa *INT* (isti tip podatka kao i polje *id* u tabeli *entries* na koje upućuje).
- *dateposted* – tipa *DATETIME*.
- *name* – tipa *VARCHAR*. Postavićemo dužinu polja na 50 karaktera.
- *comment* – tipa *TEXT*.

Polja *logins* table:

- *id* – tipa *TINYINT* (biće vrlo malo korisnika, možda čak samo jedan), odabraćemo opciju *auto_increment* i štrikiraćemo opciju *primary key*.
- *username* – tipa *VARCHAR*. Postavićemo dužinu polja na 10 karaktera.
- *password* – tipa *VARCHAR*. Postavićemo dužinu polja na 10 karaktera.

PROBNI UNOSI

Da bi mogli da testiramo našu buduću web aplikaciju moramo napraviti neke unose u bazu podataka. U tabelama sa vrijednostima koje treba da unesemo u bazu nećemo prikazivati *id* polje jer ono ima uključenu opciju *auto_increment* što znači da će MySQL server sam unositi vrijednosti u to polje (počevši od 1 pa do broja unosa).

cat
Posao
Sport
Ljubav

Tabela *categories*

cat_id	dateposted	subject	body
1	<i>Izaberite NOW funkciju</i>	Dobrodošli na moj blog!	Ovo je moj prvi unos u moj blog.
1	<i>Izaberite NOW funkciju</i>	Drugi unos!	Ovo je drugi unos u moj blog.

Tabela *entries*.

cat_id	dateposted	name	comment
1	<i>*Izaberite NOW funkciju</i>	Marko	Bravo!
1	<i>*Izaberite NOW funkciju</i>	Janko	Nadam se da ćeš se lijepo zabaviti!

Tabela *comments*

username	password
admin	admin

Tabela *logins*

* Datum i vrijeme sačuvane pomoću funkcije *NOW()* izvlačimo pomoću PHP funkcije *date(„format“, \$promjenljiva)*. Format koji nam odgovara izgleda ovako 20.12.2009. 12:52 a to bi u znakovima za formatiranje izgledalo ovako: "j.n.Y. G:i"

- j = dan (20)
- n = mjesec (12)
- Y = godina (2009)
- G = sad (15)
- i = minut (52)

PROGRAMIRANJE BLOGA

KONFIGURACIJA

Počnemo izradom konfiguracionog PHP fajla koji će nam olakšati buduće izmjene. Kreiramo novi fajl *config.php* i u njega smještamo sledeći kod:

```
<?php

$dbhost = "localhost";
$dbuser = "root";
$dbpassword = "";
$dbdatabase = "weblog";

$config_blogname = "MOJ PRVI WEBLOG";

$config_author = "Branimir Jelic";

$config_basedir = "http://localhost/weblog";

error_reporting(0);

?>
```

PRIMJER 1

Najveći dio ovog koda predstavlja inicijalizaciju promjenljivih koje ćemo kasnije koristiti. Prve četiri promjenljive predstavljaju pristupne podatke za bazu podataka. Zatim imamo naziv bloga (*\$config_blogname*), ime autora (*\$config_author*) i lokaciju početnog direktorijuma (*\$config_basedir*). Vrlo je bitno da lokacija početnog direktorijuma bude upisana tačno jer će se neke komande oslanjati na ovu promjenljivu. Poslednja komanda gasi prijavljivanje upozorenja i grešaka od strane PHP interpretera.

KORISNIČKI INTERFEJS

Prvo ćemo napraviti fajl sa CSS stilovima, koji će oblikovati izgled ovog web sajta. Fajl ćemo nazvati *stylesheet.css* i u njega iskopirati kod koji se nalazi na poslednjoj strani.

header.php

Zatim pravimo gornje zaglavlje web sajta koje će se, naravno, ponavljati na svim stranama. Kreiramo novi PHP fajl, nazivamo ga *header.php* i u njega smještamo sledeći kod:

```

<?php
require("config.php");
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title><?php echo $config_blogname; ?></title>
<link rel="stylesheet" href="stylesheet.css" type="text/css" />
</head>
<body>
<div id="header">
<h1><?php echo $config_blogname; ?></h1>
<div id="menu">
  <a href="index.php">početna strana</a>-
  <a href="viewcat.php">kategorije</a>-
  <a href="login.php">uloguj se</a>
</div>
<div id="main">

```

PRIMJER 2

U ovom kodu imamo nekoliko bitnih tačaka. Na početku se otvara PHP blok i uključuje se config.php u stranicu. Koristili smo require() funkciju umjesto include(), jer je config.php obavezan fajl zbog podataka koji se nalaze u njemu a neophodni su nam za bilo kakav rad sa bazom.

HTML kod je standardan kao i svugdje, jedino što u <title> bloku ponovo imamo jedan PHP blok koji daje naziv strani i to onaj naziv koji je smješten u promjenljivu *\$config_blogname* iz *config.php* (PRIMJER 1) i koji će biti prikazan u naslovu pretraživača. Taj naziv je takođe iskorišćen i u prvom <div> bloku unutar <h1> bloka. Ovo nam omogućava jedan najosnovniji vid brendiranja.

Na kraju imamo navigaciju sajta ispod <h1> bloka koja će sadržati linkove ka početnoj strani, kategorijama i strani za prijavljivajne. Zatim otvaramo glavni <div> blok kojeg ćemo zatvoriti u donjem zaglavlju.

footer.php

Sada kada smo napravili gornje zaglavlje napravićemo i donje zaglavlje. Novi fajl ćemo nazvati footer.php i unijeti sledeći kod.

```

</div>
<div id="footer">
&copy; <?php echo $config_author; ?>
</div>
</body>
</html>

```

PRIMJER 3

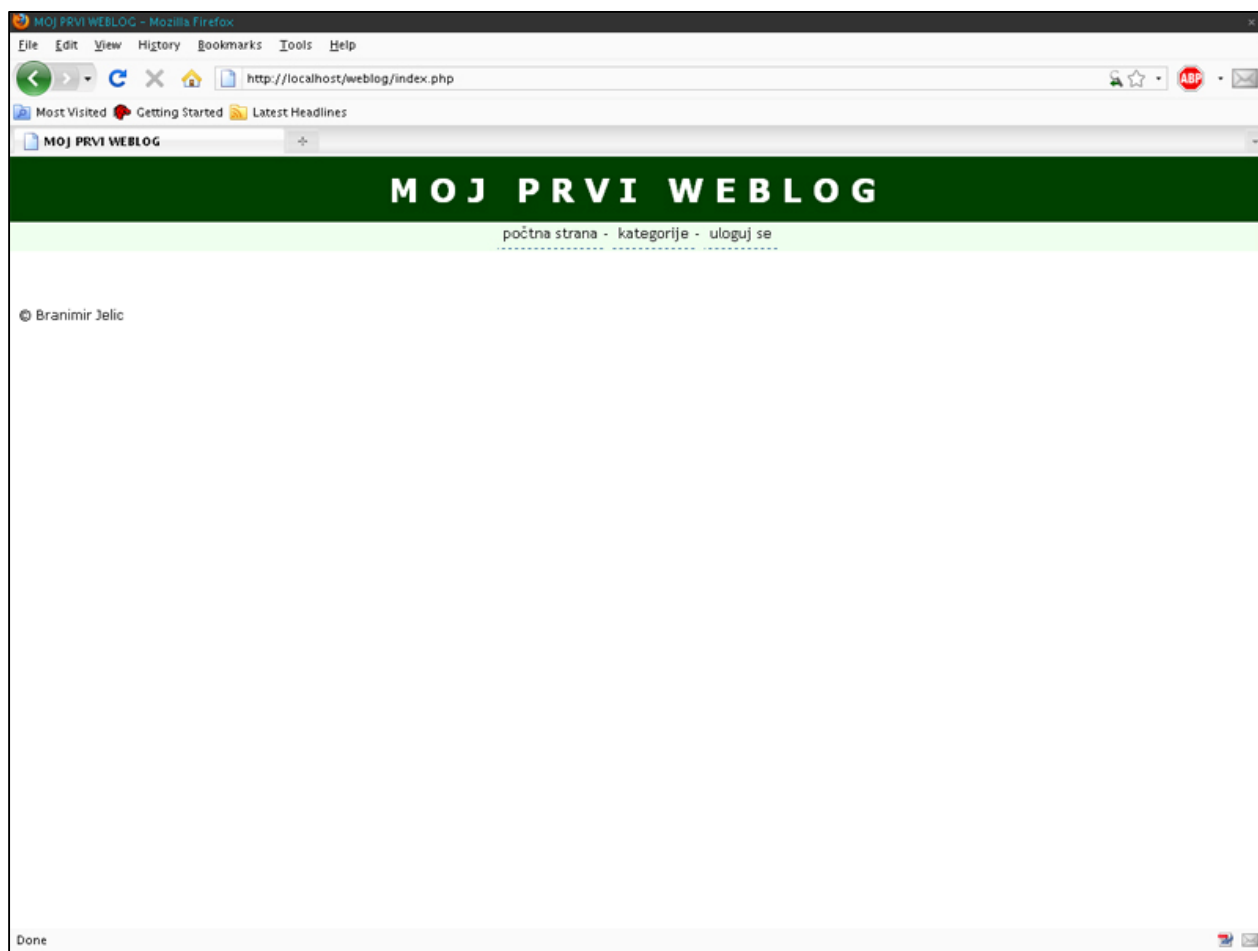
U primjeru 2-2 prva linija koda zatvara <div> blok sa id-om "main" iz gornjeg zaglavlja (na dnu primjera 4-1). Zatim pravimo <div> blok sa id-om "footer" i u njemu smještamo znak © (©) i PHP blok koji ispisuje ime autora predefinisano u *config.php*. Potom zatvaramo ostatak HTML tagova otvorenih u *header.php*.

Sada kada smo pripremili zaglavlja, fajlovi sa sadržajem se mnogo jednostavnije realizuju. Da bi vidjeli osnovni prikaz, kreiraćemo novi fajl, nazvati ga *index.php* i upisati kod iz sledećeg primjera:

```
<?php  
require("header.php");  
  
// ovdje ide ostatak koda  
  
require("footer.php");  
  
?>
```

PRIMJER 4

Kada u pretraživaču pozovemo stranicu *index.php* dobićemo okvirni sadržaj koji će se ponavljati na svakoj stranici.



SLIKA 1 – Gornje i donje zaglavlje web sajta.

PRIKAZ UNOSA

Da bi omogućili konekciju sa bazom podataka na svakoj strani, u header.php ćemo dodati, odmah ispod `require("header.php");` sledeći kod:

```
$db = mysql_connect($dbhost, $dbuser, $dbpassword);  
mysql_select_db($dbdatabase, $db);
```

PRIMJER 5

index.php

Glavna stranica koja će dočekivati goste će, pored okvirnog sadržaja, prikazivati poslednji postavljeni unos kao i hronološki poređane naslove prošlih unosa. U već napravljeni `index.php` ćemo dodati sledeći kod:

```
$sql = "SELECT entries.*, categories.cat  
      FROM entries, categories  
      WHERE entries.cat_id = categories.id  
      ORDER BY dateposted DESC  
      LIMIT 1;";  
  
$result = mysql_query($sql);  
$row = mysql_fetch_assoc($result);  
if(isset($row['id'])) {  
    echo "<h2><a href='viewentry.php?id=" . $row['id'] . "'> . $row['subject'] .  
    "</a></h2>";  
    echo "<i>Kategorija <a href='viewcat.php?id=" . $row['cat_id'] . "'> . $row['cat'] . "</a> -  
    Postavljeno: " . date("j.n.Y. G:i", strtotime($row['dateposted'])) . "</i>";  
  
    echo "<p>";  
    echo nl2br($row['body']);  
    echo "</p>";  
} else { echo "Nema unosa."; }  
echo "<hr />";
```

PRIMJER 6

Gore navedenim kodom na početku definišemo upit nad bazom. Taj upit izdvaja sva polja *entries* tabele i *cat* polje *categories* tabele kod kojih važi da je vrijednost polja *cat_id* iz *entries* tabele jednaka vrijednosti polja *id* iz *categories* tabele, sortirane po polju *dateposted* u redosledu od najvećeg ka najmanjem i uzima samo jedan, tj. prvi iz tabele redova koji zadovoljavaju navedene kriterijume.

Zatim u promjenljivu *\$result* smještamo rezultat upita pomoću PHP komande *mysql_query()*. Iz tog rezultata komandom *mysql_fetch_assoc()* izvlačimo taj red koji smo tražili. Ako postoji taj red predajemo pretaraivaču HTML strukturu sa izvučenim naslovom unosa, kategorijom, datumom postavljanja i tekstom. Ako ne postoji traženi red (tj. ne postoji nikakav unos) pretraživač će ispisati „Nema unosa”.

Zatim želimo da ispišemo imena ljudi koji su ostavljali komentare na poslednji unos, koji smo i izvukli prošlim upitom:

```
$commsql = "SELECT name
            FROM comments
            WHERE blog_id = " . $row['id'] . "
            ORDER BY dateposted;";

$commresult = mysql_query($commsql);
$numrows_comm = mysql_num_rows($commresult);

if($numrows_comm == 0) {
    echo "<p>Nema komentara.</p>";
} else {
    echo "<strong>" . $numrows_comm . "</strong> komentari : ";
    $i = 1;

    while($commrow = mysql_fetch_assoc($commresult)) {
        echo "<a href='viewentry.php?id=" . $row['id'] . "#comment" . $i . "'>" .
            $commrow['name'] . "</a> ";
        $i++;
    }
    echo "</p>";
}
```

PRIMJER 7

U primjeru 6 pravimo novi upit koji izdvaja sva imena (polje *name*) iz tabele *comments* kojima je *blog_id* jednak polju *id* iz reda izvučenog upitom iz primjera 6. Izvršavamo ga. Izvlačimo broj redova (funkcija *mysql_num_row()*) koje smo dobili poslednjim upitom. Ako je broj redova jednak nuli onda ispisujemo "Nema komentara", u suprotnom izlistavamo imena ljudi koji su ostavili komentare.

Sledeći korak je izlistavanje pet zadnjih unosa iz bilo koje kategorije:

```
$prevsq = "SELECT entries.*, categories.cat
            FROM entries, categories
            WHERE entries.cat_id = categories.id
            ORDER BY dateposted DESC
            LIMIT 1, 5;";

$prevresult = mysql_query($prevsq);
$numrows_prev = mysql_num_rows($prevresult);

if($numrows_prev == 0) { echo "<p>Nema prošlih unosa.</p>";
} else {
    echo "<ul>";
}
```

```

        while($prevrow = mysql_fetch_assoc($prevresult)) { echo "<li><a
        href='viewentry.php?id=". $prevrow['id'] . "'>" . $prevrow ['subject'] .
        "</a></li>";
    }
}

echo "</ul>";

```

PRIMJER 8

U prethodnom primjeru novim upitom izvlačimo nazive poslednjih pet (ili manje) unosa iz bilo koje kategorije. I provjeravamo koliko ih ima pa ih prikazujemo kao listu.

login.php

Sada ćemo napraviti stranu za identifikovanje administratora bloga. Kreiraćemo novi fajl *login.php* i u njega upisati sledeći kod.

```

<?php
require("config.php");
$db = mysql_connect($dbhost, $dbuser, $dbpassword);
mysql_select_db($dbdatabase, $db);

if($_POST['submit']) {
    $sql = "SELECT * FROM logins
    WHERE username = '" . $_POST['username'] . "' AND password = '" .
        $_POST['password'] . "'";
    $result = mysql_query($sql);
    $numrows = mysql_num_rows($result);

    if($numrows == 1) {
        $row = mysql_fetch_assoc($result);
        session_register("USERNAME");
        session_register("USERID");
        $_SESSION['USERNAME'] = $row['username'];
        $_SESSION['USERID'] = $row['id'];
        header("Location: " . $config_basedir);
    } else {
        header("Location: " . $config_basedir . "/login.php?error=1");
    }
}
else {
    require("header.php");
    if($_GET['error']) {
        echo "Netacno, probajte ponovo!";
    }
}
?>

```

PRIMJER 9

Ovog puta ćemo za početak učitati *config.php* da bi se spojili na bazu i provjerili da li je fajlu proslijeđena forma i ako nije da ne učitavamo ostatak sajta. U slučaju da jeste napraviti ćemo MySQL upit koji će nam vratiti korisnika i njegovu lozinku koji se poklapaju sa istima iz forme koja je proslijeđena, postavimo u niz `$_SESSION[]` promjenljive *USERNAME* sa vrijednošću korisničkog imena i *USERID* sa *id* koje smo izvukli iz baze. U slučaju da forma nije proslijeđena učitaćemo *header.php* i ako je promjenljiva `$_GET('error')` predata ispisati "Netačno probajte ponovo!".

```
<h2>Uloguj se</h2>
<form action="login.php" method="post">
  <table>
    <tr>
      <td>Korisnicko ime</td>
      <td><input type="text" name="username"></td>
    </tr>
    <tr>
      <td>Lozinka</td>
      <td><input type="password" name="password"></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" name="submit" value="Uloguj se!"></td>
    </tr>
  </table>
</form>

<?php
}
require("footer.php");
?>
```

PRIMJER 10

Zatim pravimo formu sa poljima za upis korisničkog imena i lozinke i dugme za proslijeđivanje. Forma će se proslijeđivati na ovaj isti fajl kako bi pokrenula kod iz primjera 2-8. Na dnu koda ćemo pozvati *footer.php*.



SLIKA 2 – Strana za identifikovanje

logout.php

Pošto sada imamo stranu za prijavljivanje napravićemo i stranu za odjavljivanje.

```
<?php
session_start();
session_destroy();
require("config.php");
header("Location: " . $config_basedir);
?>
```

PRIMJER 11

Ova strana se sastoji samo od PHP koda se neće prikazivati, samo će pri pozivu odratiti to što je potrebno za odjavljivanje tj *session_destroy()* i prebaciti korisnika na početnu stranu.

Moramo izmijeniti i header.php kako bi omogućili korisniku da, kada je ulogovan, ima i link da se izloguje, i da mu se dodaju linkovi za novu kategoriju i novi zapis. Novi kod se nalazi ispod `kategorije`- i izgleda ovako:

```
<a href="viewcat.php">kategorije</a>-

<?php
    if(isset($_SESSION['USERNAME']) == TRUE) {
        echo "<a href='logout.php'>izloguj se</a>";
        echo " - ";
        echo "<a href='addentry.php'>novi unos</a>";
        echo " - ";
        echo "<a href='addcat.php'>nova kategorija</a>";
    }
    else {
        echo "<a href='login.php'>uloguj se</a>";
    }
?>

</div>
</div>
<div id="main">
```

PRIMJER 12

Provjeravamo da li je postavljena promjenljiva *USERNAME* u globalnom nizu *\$_SESSION[]* i ako jeste znamo da je korisnik koji trenutno pristupa stranici u stvari vlasnik bloga i dodajemo linkove za novu kategoriju i novi unos.

Sada ćemo napraviti stranu za pojedinačni prikaz unosa i njihovih komentara. Napravićemo novi fajl i nazvati ga *viewentry.php*. Na početku ćemo upisati sledeći kod

```
<?php
require("config.php");
if(isset($_GET['id'])) {
    if(is_numeric($_GET['id']) == FALSE) {
        $validentry = $_GET['id'];
    }
    else {
        header("Location: " . $config_basedir);
    }
}
else {
    $validentry = 0;
}

if($_POST['submit']) {
    $db = mysql_connect($dbhost, $dbuser, $dbpassword);
    mysql_select_db($dbdatabase, $db);
    $sql = "INSERT INTO comments(blog_id, dateposted, name, comment)
          VALUES(" . $validentry . ", NOW(), '" . $_POST['name'] . "', '" .
              $_POST['comment'] . "')";
    mysql_query($sql);
    header("Location: viewentry.php?id=" . $validentry);
}
else {
    require("header.php");
    if($validentry == 0) {
        $sql = "SELECT entries.*, categories.cat
              FROM entries, categories " . "
              WHERE entries.cat_id = categories.id " . "
              ORDER BY dateposted DESC " . "
              LIMIT 1;";
    }
}
```

PRIMJER 13

U ovom primjeru smo ponovo, uključili *config.php* u stranicu da bi prvo odradili provjeru. Provjeravamo da li je predat *id* parameter preko globalnog niza *\$_GET*[1. Ako jeste provjeravamo da li je *id* parameter broj i ako jeste postavljamo u promjenljivu *\$validentry* vrijednost parametra *id*, u suprotnom vraćemo korisnika na početnu stranu 2. U ostalim slučajevima postavljamo *\$validentry* na "0".

Zatim provjeravamo da li je metodom POST proslijeđena forma sa komentarom tako što provjeravamo da li postoji parameter *submit* u globalnom nizu *\$_POST*[]. Ako postoji povezujemo se na bazu i pravimo upit za unošenje novog reda u tabelu *comments* koji će sadržati ime i komentar iz proslijeđene forme. Izvršavamo taj upit i prebačamo korisnika na prikaz pojedinačnih unosa koji će prikazati onaj unos na koji je i postavljen komentar.

U slučaju da nije proslijeđen komentar treba da prikazemo traženu stranicu. Učit ćemo *header.php*. Zatim vidjeti da li je *\$validentry* postavljen na 0. Ako jeste u *\$sql* postavljamo upit koji vraće poslednji unos..

Zatim nastavljamo dalje:

```
else {
    $sql = "SELECT entries.*, categories.cat
    FROM entries, categories " . "
    WHERE entries.cat_id = categories.id AND entries.id = " . $validentry . "
    ORDER BY dateposted DESC LIMIT 1;";
}
$result = mysql_query($sql);
$row = mysql_fetch_assoc($result);
echo "<h2>" . $row['subject'] . "</h2>";
echo "<i>Kategorija: <a href='viewcat.php?id=" . $row['cat_id'] . "'>" .
$row['cat'] . "</a> - Postavljeno " . date("j.n.Y. G:i", strtotime($row['dateposted'])) . "</i>";
if(isset($_SESSION['USERNAME']) == TRUE) {
    echo " [<a href='updateentry.php?id=" . $row['id'] . "'>Izmijeni</a>]";
}
echo "<p>";
echo nl2br($row['body']); echo "</p>";

echo "</p>";
$commsql = "SELECT * FROM comments WHERE blog_id = " . $validentry . " ORDER BY dateposted
DESC;";
$commresult = mysql_query($commsql); $numrows_comm = mysql_num_rows($commresult);
echo "<hr />";
if($numrows_comm == 0) {
    echo "<p>Nema komentara.</p>";
} else {
    $i = 1;
    while($commrow = mysql_fetch_assoc($commresult)) { echo "<a name='comment' " . $i . "'>"; echo
        "<h3>Postavio: " . $commrow['name'] . " - " .
        date("j.n.Y. G:i", strtotime($commrow['dateposted'])) . "</h3>";
        echo $commrow['comment'];
        $i++;
    }
}
?>
```

PRIMJER 14

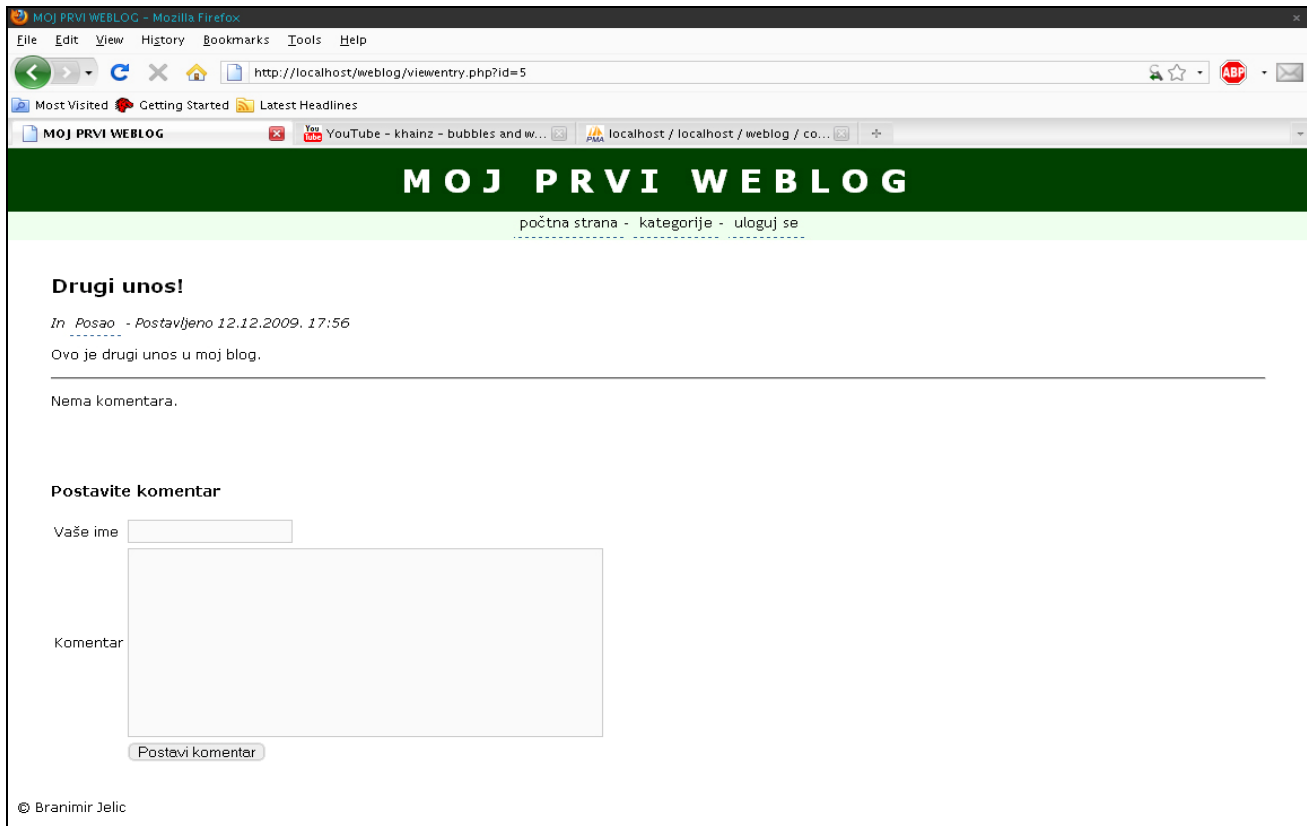
Ako *\$validentry* ima neku drugu vrijednost u *\$sql* smještamo upit koji izvlači red čija je vrijednost polja *id* jednaka vrijednosti iz promjenljive *\$validentry*. Izvršavamo *\$sql* upit. Izvučeni red smještamo u promjenljivu *\$row*. Zatim štamujemo vrijednosti koje sadrži red, tj. naslov, kategoriju, vrijeme i tekst uslova. Pored naslova unosa postavimo link sa vrijednošću *\$validentry* promjenljive kao parametar GET metode koji će da se proslijedi strani za izmjenu postojećeg unosa, tj. ka *updateentry.php*.

Ako nema komentara ispisujemo poruku, ako ima izlistavamo ih.

Na kraju fajla ćemo postaviti HTML kod forme za unos komentara koja će se prosleđivati na *viewentry.php*.

```
<br /><br /> <h3>Postavite komentar</h3>
<form action="<?php echo "viewentry.php?id=" . $validentry; ?>" method="post"><table>
  <tr>
    <td>Vaše ime</td>
    <td><input type="text" name="name"></td>
  </tr>
  <tr>
    <td>Komentar</td>
    <td><textarea name="comment" rows="10" cols="50"></textarea></td>
  </tr>
  <tr>
    <td></td>
    <td><input type="submit" name="submit" value="Postavi komentar"></td>
  </tr>
</form></table>
<?php
}
require("footer.php");
?>
```

PRIMJER 15



SLIKA 3 – Prikaz pojedinačnih unosa.

Sada ćemo napraviti i stranu za pregled kategorija koja će izgledati ovako:



SLIKA 4 – Prikaz stranice kategorija

Kod je sličan kao i kod prikaza unosa samo što je jednostavniji jer je potrebno samo da izlistamo kategorije koje imamo i po potrebi unose jedne od kategorija.

```
if(isset($_GET['id']) == TRUE) {
    if(is_numeric($_GET['id']) == FALSE) {
        $error = 1;
    }
    if($error == 1) {
        header("Location: " . $config_basedir . "/viewcat.php");
    } else {
        $validcat = $_GET['id'];
    }
} else {
    $validcat = 0;
}

$sql = "SELECT * FROM categories";
$result = mysql_query($sql);
echo "<h2>Kategorije</h2>";
while($row = mysql_fetch_assoc($result)) {
    if($validcat == $row['id']) {
        echo "<strong>" . $row['cat'] . "</strong><br />";
        $entriessql = "SELECT * FROM entries WHERE cat_id = " . $validcat . " ORDER BY
                        dateposted DESC;";
        $entriesres = mysql_query($entriessql);
        $numrows_entries = mysql_num_rows($entriesres);
        echo "<ul>";
        if($numrows_entries == 0) {
```

```

        echo "<li>Nema unosa!</li>";
    } else {
        while($entriesrow = mysql_fetch_assoc($entriesres)) {
            echo "<li>" . date("j.n.Y. G:i", strtotime($entriesrow['dateposted'])) . "

                    - <a href='viewentry.php?id=" . $entriesrow['id'] . "'>" .
                      $entriesrow['subject'] . "</a></li>";

        }
    }
    echo "</ul>";
} else {
    echo "<a href='viewcat.php?id=" . $row['id'] . "'>" . $row['cat'] . "</a><br />";
}
}
}

```

PRIMJER 16

Kada kliknemo na neku kategoriju, stranica se ponovo učitava i prenosi *id* vrijednost kategorij (koja je sačuvana u linku) putem *GET* metode, na stranici treba da se prikaže lista unosa ispod kategorije na koju smo kliknuli.

addcat.php

Napravićemo stranicu za unos kategorija. Novi fajl ćemo nazvati addcat.php. I upisaćemo sledeći kod

```

$db = mysql_connect($dbhost, $dbuser, $dbpassword);
mysql_select_db($dbdatabase, $db);

if(isset($_SESSION['USERNAME']) == FALSE) {
    header("Location: " . $config_basedir);
}

if($_POST['submit']) {
    $sql = "INSERT INTO categories(cat) VALUES('" . $_POST['cat'] . "')";
    mysql_query($sql);
    header("Location: " . $config_basedir . "/viewcat.php");
}
else {
    require("header.php");
?>
<h2>Nova kategorija</h2>

<form action="addcat.php" method="post">
<table>
<tr>
<td>Kategorija: </td>
<td><input type="text" name="cat"></td>

```

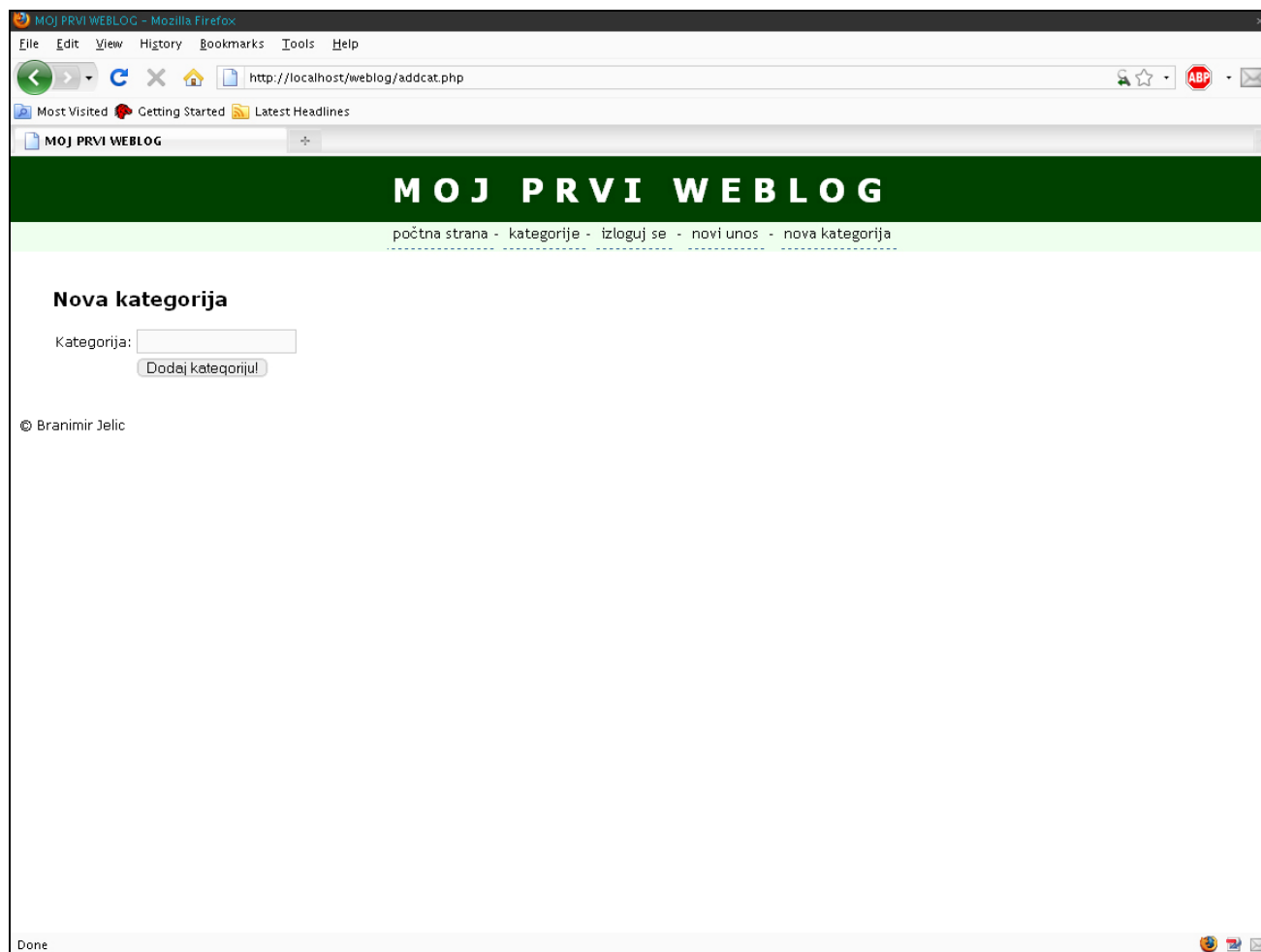
```

</tr>
<tr>
  <td></td>
  <td><input type="submit" name="submit" value="Dodaj kategoriju!"></td>
</tr>
</table>
</form>
<?php
}

```

PRIMJER 17

Ova strana takođe počinje sa spajanjem na bazu i provjerom prava korisnika. Naravno da ne želimo da bilo ko dodaje nove kategorije ili unose. Zatim imamo upit za unos nove kategorije u bazu. Ako je sve kako treba iscrtavamo stranu i prikazujemo formu za unošenje nove kategorije.



SLIKA 6 – Prikaz strane za novu kategoriju.

addentry.php

Sada pravimo stranu za unos Tema. Prvo smo pravili stranu za unos kategorija da bi na strani za unos tema mogli da omogućimo vlasniku bloga da izabere kategoriju iz list box-a i time smanjujemo mogućnost za grešku u bazi. Jer novu temu neće moći da veže za nekategorijom koja ne postoji. NApravimo *addentry.php* i u njega upišimo sledeći kod:

```
$db = mysql_connect($dbhost, $dbuser, $dbpassword);
mysql_select_db($dbdatabase, $db);
if(isset($_SESSION['USERNAME']) == FALSE) {
    header("Location: " . $config_basedir);
}
if($_POST['submit']) {
    $sql = "INSERT INTO entries(cat_id, dateposted, subject, body)
          VALUES(" . $_POST['cat'] . ", NOW(), '" . $_POST['subject'] . "', '" . $_POST['body'] .
    "');";
    mysql_query($sql);
    header("Location: " . $config_basedir);
}
else {
    require("header.php");
?>

<h2>Novi unos</h2>
<form action="addentry.php" method="post">
  <table>
    <tr>
      <td>Kategorija</td>
      <td><select name="cat">
<?php
$catsql = "SELECT * FROM categories;";
$catres = mysql_query($catsql);
while($catrow= mysql_fetch_assoc($catres)) {
    echo "<option value='" . $catrow['id']. "'>" . $catrow['cat'] . "</option>";
}
?>
      </select></td>
    </tr>
    <tr>
      <td>Tema</td>
      <td><input type="text" name="subject"></td>
    </tr>
    <tr>
      <td>Text</td>
      <td><textarea name="body" rows="10" cols="50"></textarea></td>
    </tr>
    <tr>
      <td></td>
      <td><input type="submit" name="submit" value="Dodaj unos!"></td>
    </tr>
  </table>
</form>
```

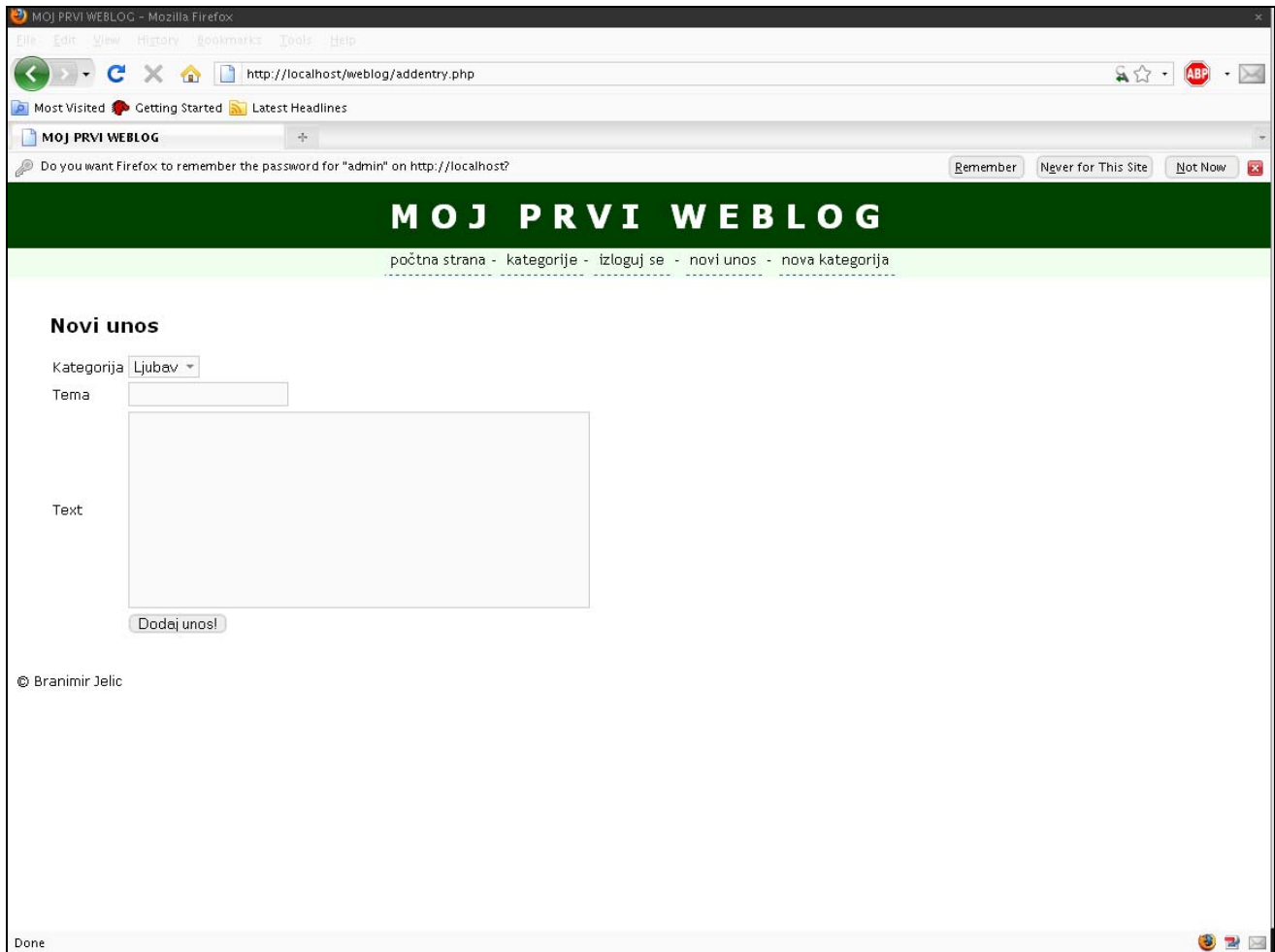
```

</tr>
</table>
</form>
<?php
}

```

PRIMJER 17

Skoro identičan kod kao kod unosa kategorije, samo što ovdje imamo malo drugačiji upit jer ima više argumenata i samim tim se razlikuje i forma koja prosljeđuje više polja, od kojih je prvo listbox koji iz baze vuče listu kategorija.



SLIKA 7 – Prikaz strane za unos nove Teme

updateentry.php

```

if(isset($_SESSION['USERNAME']) == FALSE) {
    header("Location: " . $config_basedir);
}
$db = mysql_connect($dbhost, $dbuser, $dbpassword);
mysql_select_db($dbdatabase, $db);

```

```

if(isset($_GET['id']) == TRUE) {
    if(is_numeric($_GET['id']) == FALSE) {
        $error = 1;
    }
    if($error == 1) {
        header("Location: " . $config_basedir);
    }
    else {
        $validentry = $_GET['id'];
    }
}
else {
    $validentry = 0;
}

if($_POST['submit']) {
    $sql = "UPDATE entries
    SET cat_id = " . $_POST['cat'] . ", subject = " . $_POST['subject'] . ", body = " . $_POST['body'] . "
    WHERE id = " . $validentry . ";";
    mysql_query($sql);
    header("Location: " . $config_basedir . "/viewentry.php?id=" . $validentry);
} else {
    require("header.php");
    $fillsql = "SELECT * FROM entries WHERE id = " . $validentry . ";";
    $fillres = mysql_query($fillsql);
    $fillrow = mysql_fetch_assoc($fillres);
?>

```

<h1>Izmijeni unos</h1>

<form action="<?php echo "updateentry.php?id=" . \$validentry; ?>" method="post">

<table>

<tr>

<td>Kategorija</td>

<td><select name="cat">

<?php

\$catsql = "SELECT * FROM categories;";

\$catres = mysql_query(\$catsql);

while(\$catrow= mysql_fetch_assoc(\$catres)) {

echo "<option value=" . \$catrow['id'] . """;

if(\$catrow['id'] == \$fillrow['cat_id']) {

echo " selected";

}

echo ">" . \$catrow['cat'] . "</option>";

}

?>

</select></td>

</tr>

<tr>

<td>Tema</td>

<td><input type="text" name="subject"

```

        value="<?php echo $fillrow['subject']; ?>"></td>
    </tr>
    <tr>
        <td>Tekst</td>
        <td><textarea name="body" rows="10" cols="50">
            <?php echo $fillrow['body']; ?></textarea></td>
    </tr>
    <tr>
        <td></td>
        <td><input type="submit" name="submit" value="Update Entry!"></td>
    </tr>
</table>
</form>
<?php
}

```

PRIMJER 18

Ovaj kod je skoro identičan kao kod *addentry.php* jedino što pri pokretanju strane polja forme vuku vrijednosti iz baze podataka i prikazuju ih kako bi mogli da ih izmijenimo. Kad se proslijedi forma, umjesto INSERT kao kod *addentry.php*, ovdje upit ima komandu UPDATE koja već postojeće vrijednosti u bazi mijenja, i to ona polja koja se poklapaju sa promjenljivom *\$validentry* koja ima istu funkciju kao i u svim ostalim stranama do sad.



SLIKA 8 – Prikaz

stylesheet.css

Ovaj kod je potrebno ubaciti u fajl stylesheet.css koji će predstavljati stil formatiranja stranica web sajta.

```
body{
    font-family:Verdana, Geneva, sans-serif;
    font-size:12px;
    line-height:20px;
    padding: 0px;
    color: #000000;
    background-color: #EEFEEE;
}

a{
    padding:5px
}

a:link {
    text-decoration: none;
    color: #000;
}

a:visited {
    text-decoration: none;
    border-bottom: 1px dashed #369;
    color: #000;
}

a:hover, a:active {
    text-decoration: none;
    border-bottom: 1px solid #036;
    color: #000;
}

#container {
    position: absolute;
    top: 85px;
    left: 0px;
    background: #ffffff;
    margin: 0 auto 0 auto;
    text-align: left;
    width: 100%;
    height: 100%;
}

#menu {
    font-family: "trebuchet ms", verdana,
                sans-serif;
    font-size: 14px;
    position: absolute;
    height: 27px;
    top: 60px;
    left: 0px;
    width: 100%;
}

#header {
    position: absolute;
    top: 0px;
    left: 0px;
    height: 80px;
    width: 100%;
    background:#004000;
    padding-top: 0px;
    text-align:center;
}

h1 {
    font-size: 30px;
    text-transform: uppercase;
    letter-spacing: 0.3em;
    color: #fff;
}

#main {
    margin: 90px 15px 15px 15px;
    padding: 15px 15px 15px 15px;
    background: #FFFFFF;
}

#bar {
    float: left;
    width: 200px;
    background: #eee;
    z-index: 1;
    padding: 10px;
    margin-right: 30px;
    height: 100%;
}

#bar h1 {
    font-size: 12px;
    text-transform: uppercase;
    letter-spacing: 0.3em;
}
```