

# PHP i MySQL

III. FAZA PROJEKTA

PROGRAMIRANJE WEB APLIKACIJA

U sklopu treće faze projekta potrebno je nadograditi web sjedište iz prošle faze na način da se omogućiti upravljanje sadržajem unesenog putem HTML forme kreirane u sklopu druge faze. Osim što se uneseni sadržaj forme ispisuje na web stranici nakon unosa pomoću PHP skripte skripta.php, isti je potrebno pohraniti u bazu. Osim toga, potrebno je modificirati stranicu index.html (kreirati umjesto toga index.php) koja će služiti za ispis svih unesenih vijesti pohranjenih u bazi na glavnoj stranici. Prilikom prikaza vijesti na stranici treba paziti da li je prilikom pohrane bila označena kućica za sakrivanje vijesti (arhiva) ili ne, jer ukoliko je kućica označena vijest se sprema u bazu, ali ne prikazuje na stranici. Osim toga, na stranici index.php vijesti se trebaju prikazivati u minimalno dvije kategorije (po vlastitom izboru; npr. Sport, kultura, politika, zabava ...). Iste te kategorije se trebaju nalaziti u navigaciji i voditi na stranicu kategorija.php, na kojoj se prikazuju sve vijesti koje nisu arhivirane, a pripadaju toj kategoriji.

Potrebno je napraviti i stranicu clanak.php, u kojoj će se prikazivati samo jedan članak i cijeli njegov sadržaj prema predlošku kojeg ste dobili u zadatku.

Izgled stranica index.php i clanak.php treba biti što sličniji predlošku kojeg ste dobili za zadatak, dok ostale stranice mogu biti proizvoljne. Pridržavajte se pravila da sve stranice imaju isti css, header i navigaciju, te footer.

U sklopu ove faze još je potrebno kreirati i administracijsku stranicu naziva administrator.php na kojoj će biti omogućeno brisanje zapisa (vijesti) iz baze, te promjena sadržaja.

### Rad s bazom podataka - MySQL

Baze podataka su organizirani način pohrane podataka. Bazom podataka upravlja server za bazu podataka. Na ovom kolegiju koristit će se MySQL server. Komunikacija sa serverom obavlja se u njemu razumljivom jeziku, a za MySQL je to jezik SQL. Baza podataka organizirana je u tablice u koje se podaci zapisuju ili dohvaćaju po potrebi.

Za potrebe kolegija potrebno je osmisлити i kreirati bazu podataka korištenjem alata phpMyAdmin iz paketa XAMPP. Upute za korištenje XAMPP-a nalaze se na stranicama kolegija na LMS sustavu.

U nastavku su opisane neke od osnovnih naredbi za rad s bazom podataka:

- **CREATE DATABASE**
  - služi za kreiranje baze
  - upotreba:
    - `CREATE naziv_baze;`
- **USE**
  - služi za odabir baze/tablice s kojom će se raditi
  - upotreba:
    - `USE naziv_baze;`
- **CREATE TABLE**
  - služi za kreiranje tablice u bazi
  - upotreba:
    - `CREATE TABLE naziv_tablice (ime varchar(30), prezime(30), ...);`
- **INSERT**
  - služi za unos podataka u bazu
  - upotreba:
    - `INSERT INTO naziv_tablice (naziv_stupca1, naziv_stupca2, ...) VALUES ('vrijednost1', 'vrijednost2', ...)`
- **DELETE**
  - služi za brisanje podataka iz baze
  - upotreba:
    - `DELETE FROM naziv_tablice`
    - npr. `DELETE FROM naziv_tablice WHERE email='neki_email@email.com'`
- **SELECT**
  - služi za dohvat podataka iz tablice
  - upotreba:
    - `SELECT popis_stupaca FROM naziv_tablice`
    - `SELECT * FROM naziv_tablice`
      - dohvat cijele tablice
- **DESCRIBE**
  - služi za prikaz strukture tablice

- upotreba:
  - DESCRIBE naziv\_tablice
- DROP TABLE
  - služi za brisanje tablice iz baze
  - upotreba:
    - DROP TABLE naziv\_tablice
- ALTER TABLE
  - služi za izmjenu tablice
  - upotreba:
    - ALTER TABLE naziv\_tablice ADD naziv\_stupca tip\_podatka

Prilikom stvaranja tablice bitno je znati koji tip podatka će biti pohranjen u svaki od stupaca. Neki od najčešće korištenih MySQL tipova podataka su:

- CHAR
  - služi za pohranu teksta od 1 do 255 znakova
  - fiksni tip
- DEC
  - služi za pohranu decimalnih brojeva
- VARCHAR
  - služi za pohranu tekstualnih podataka
  - varijabilni tip
- INT
  - služi za pohranu cijelih brojeva
- TEXT
  - služi za pohranu veće količine teksta
- DATETIME
  - služi za pohranu datuma i vremena
  - format YYYY-MM-DD HH:MM:SS
- DATE
  - služi za pohranu datuma

- format YYYY-MM-DD
- TIME
  - služi za pohranu vremena
  - format HH:MM:SS

Svaka tablica u bazi mora imati primarni ključ. Primarni ključ je stupac u tablici s različitim vrijednostima za svaki redak čime se uvjetuje njegova jedinstvenost. U nastavku su navedena pravila koja se odnose na upotrebu primarnog ključa:

- vrijednost primarnog ključa je jedinstvena za svaki redak (zapis)
- svaki primarni ključ mora imati definiranu vrijednost
- vrijednost primarnog ključa je nepromjenjiva

Prilikom kreiranja baze i tablica potrebno je paziti na upotrebu odgovarajuće kodne stranice. Za hrvatski jezik vrijede sljedeće kodne stranice:

- Character set: latin2, collate: latin2\_croatian\_ci
  - ISO—8859-2 standard
- Character set: cp1250, collate: cp1250\_croatian\_ci
  - Windows Central European standard.
- Character set: utf8, collate: utf8\_croatian\_ci
  - UTF 8 standard
  - Collate utf8\_croatian\_ci vrijediti tek od verzije 6.0, a do tada se može upotrebljavati utf8\_general\_ci ili aplicirati patch za verziju 5.1

Prilikom izrade baze potrebno je paziti na normalizaciju podataka. Normalizacija je pristup izradi baze podataka čiji će dizajn biti optimiziran za izvršavanje upita nad bazom podataka i koji će umanjivati mogućnost dobivanja anomalija u podacima baze.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
2	datum	varchar(32)	utf8_croatian_ci		No	None			Change  Drop  More
3	naslov	varchar(64)	latin2_croatian_ci		No	None			Change  Drop  More
4	sazetak	text	latin2_croatian_ci		No	None			Change  Drop  More
5	tekst	text	latin2_croatian_ci		No	None			Change  Drop  More
6	slika	varchar(64)	latin2_croatian_ci		No	None			Change  Drop  More
7	kategorija	varchar(64)	latin2_croatian_ci		No	None			Change  Drop  More
8	arhiva	tinyint(1)			No	None			Change  Drop  More

## Briga o bazi podataka – PHP

Uloga PHP-a u radu s bazom podataka je automatizacija procesa pohrane i dohvata podataka. Za rad s bazom podataka u PHP-u se koriste tri glavne funkcije: **mysqli\_connect()**, **mysqli\_query()** i **mysqli\_close()**.

Rad s MySQL bazom podataka iz PHP-a vrši se preko PHP ekstenzija, a najčešće korištene su:

- mysql
  - najstarija ekstenzija za rad s MySQL bazom podataka
- mysqli
  - mysql improved
  - napravljena zato da se omoguće nova svojstva MySQL baze podataka uvedena u verzijama 4.1.3 na više
- pdo
  - PHP Data Objects
  - najnovija ekstenzija koja pruža set objekata uz pomoć kojih se može raditi sa svim podržanim bazama podataka

U nastavku su navedeni i opisani osnovni koraci u radu s bazom podataka:

### 1) Spajanje na bazu podataka korištenjem funkcije mysqli\_connect()

- za spajanje su potrebni: adresa servera (ime domene, IP adresa ili localhost), korisničko ime i lozinka za rad s bazom, ime baze (opcionalno, ali uz korištenje funkcije mysqli\_select\_db())
- MySQL administrator (root) nema postavljenu lozinku

- u slučaju neuspjelog spajanja na bazu poželjno je PHP funkcijom `die()` prekinuti izvođenje PHP skripte
- defaultni username za XAMPP je 'root', a password je prazan

```
$dbc = mysqli_connect('localhost', 'username', 'password', 'base_name') or
die('Error connecting to MySQL server.'. mysqli_connect_error());
```

## 2) Kreiranje SQL upita i njegova pohrana (kao znakovnog niza) u PHP varijablu

- za komunikaciju sa serverom koriste se SQL naredbe
- za ime PHP varijable koja sadrži upit obično se koristi ime varijable `$query`

Unos u bazu/tablicu možemo obaviti eksplicitnim navođenjem vrijednosti atributa:

```
$query = "INSERT INTO table_name (name, surname, age) VALUES ('Ivan', 'Grozni',
'25')";
```

ili dohvatom iz varijable `$_POST`:

```
<?php
$first_name = $_POST['firstname'];
$last_name = $_POST['lastname'];
$age = $_POST['age'];

$query = "INSERT INTO table_name (name, surname, age) VALUES ('$first_name',
'$last_name', " . "'$age')";
?>
```

## 3) Prosljeđivanje upita korištenjem funkcije `mysqli_query()`

- funkciji `mysqli_query()` potrebno je proslijediti dva argumenta: konekciju prema bazi (u našem primjeru varijablu `$dbc`) i sam upit (u našem primjeru `$query`)
- b. u varijablu `$result` sprema se rezultat uspješnosti upita prema bazi

```
$result = mysqli_query($dbc, $query) or die('Error querying database.');
```

#### 4) Zatvaranje konekcije prema serveru korištenjem funkcije *mysqli\_close()*

- označava kraj rada sa serverom i bazom podataka
- stvar dobre navike!
- funkciji *mysqli\_query()* potrebno je kao argument proslijediti konekcijsku varijablu (u našem primjeru *\$dbc*)

```
mysqli_close($dbc);
```

#### Ispis podataka iz baze

Nakon što smo podatke pohranili u bazu, potrebno ih je i dohvatiti iz iste. Sljedeći primjer prikazuje PHP kod za dohvat svih podataka iz tablice upotrebom funkcije *mysqli\_fetch\_array()* i njihov ispis uz korištenje funkcije *echo*. Funkcija *mysqli\_fetch\_array()* sprema zapise (jedan red tablice) u polje koji se zatim ispisuju.

```
$query = "SELECT * FROM vijesti WHERE arhiva=0 AND kategorija='sport' LIMIT 4";  
$result = mysqli_query($dbc, $query);  
while($row = mysqli_fetch_array($result)) {  
    echo $row['datum'] . ' ' . $row['naslov'] . ' ' . $row['tekst'] . '<br />';  
}
```

Primjer prikazuje ispisivanje vijesti na naslovnicu. Ispisuju se samo one vijesti koje nisu arhivirane (*arhiva=0*), kojima je kategorija sport (*kategorija='sport'*) i samo 4 zapisa.

Brisanje podataka iz tablice obavlja se slično primjeru u nastavku gdje se iz tablice brišu zapisi koji odgovaraju navedenom e-mailu:

```
$query = "DELETE FROM ime_tablice WHERE email = 'pwa@tvz.hr'";
```

U nastavku će još biti opisan postupak uploada i dodavanja slika u bazu. Ako preko HTML forme želimo u bazu spremiti neku datoteku, prvo je potrebno pripremiti HTML formu za upload slike slično primjeru:

```
<form enctype="multipart/form-data" method="post" action="skripta.php">  
    ...  
<input type="file" id="picture" name="picture" />    ... </form>
```



Atribut *enctype* služi za definiranje potrebnog načina kodiranja podataka kako bi se metodom POST do servera poslala uploadana datoteka iz forme. Nakon toga definiramo polje za odabir datoteke. U PHP-u informacije o datoteci poslanoj na server iz HTML forme spremaju se u superglobalnu varijablu naziva `$_FILES`. Neke od informacija pohranjenih u varijabli `$_FILES` su:

- `$_FILES['picture']['name']`
  - ime datoteke
- `$_FILES['picture']['type']`
  - MIME tip datoteke
- `$_FILES['picture']['size']`
  - veličina datoteke (u byteima)
- `$_FILES['picture']['tmp_name']`
  - privremena lokacija na kojoj je pohranjena uploadana datoteka

Ukoliko, npr. želimo naziv datoteke pospremiti u bazu to ćemo napraviti na sljedeći način:

```
<...>
$picture = $_FILES['picture']['name'];
<...>
$query = "INSERT INTO table_name VALUES('$first_col', '$second_col', ...,
'$picture')";
```

Potrebno je uočiti da se prilikom pohrane datoteke u bazu sprema samo njezino ime, ali ne i sama datoteka. Ona će se pohraniti u privremeni direktorij na serveru koji se stvara automatski. Postavlja se pitanje kako se dohvaća uploadana datoteka i da li je moguće kontrolirati mjesto njezine pohrane na serveru? U pravilu sve slike poželjno je pospremiti u direktorij naziva *img* ili *images*. Izbor direktorija u koji će se spremati slike poslane iz forme moguće je napraviti na sljedeći način korištenjem PHP funkcije *move\_uploaded\_file()*:

```
<...>
$target = 'images/' . $picture;
move_uploaded_file($_FILES['pictures']['tmp_name'], '$target');
<...>
```

Pregled izrade potrebnih stranica (unos.php, index.php, administracija.php, clanak.php, kategorija.php)

### Unos.php

Ova stranica je zapravo nadogradnja stranice unos.html u kojoj se nalazi forma za unos podataka u bazu. Sljedeći kod može se nalaziti u istoj stranici ili se može pozivati kao vanjski dokument, npr. Insert.php.

```
<?php
include 'connect.php';

$picture = $_FILES['pphoto']['name'];
$title=$_POST['title'];
$about=$_POST['about'];
$content=$_POST['content'];
$category=$_POST['category'];
$date=date('d.m.Y. ');
if(isset($_POST['archive'])){
    $archive=1;
}else{
    $archive=0;
}

$target_dir = 'img/'.$picture;
move_uploaded_file($_FILES["pphoto"]["tmp_name"], $target_dir);

$query = "INSERT INTO Vijesti (datum, naslov, sazetak, tekst, slika, kategorija,
arhiva ) VALUES ('$date', '$title', '$about', '$content', '$picture',
'$category', '$archive')";

$result = mysqli_query($dbc, $query) or die('Error querying database. ');
mysqli_close($dbc);
?>
```

## Connect.php

Na svim stranicama koje zapisuju ili ispisuju podatke u tablice, odnosno u bazu, potrebno je uspostaviti konekciju na bazu korištenjem funkcije `mysqli_connect()`. Kako ne bi svaki put, na svakoj novoj stranici, pisali ili kopirali konekciju, možemo napraviti novi dokument `connect.php`. Ovaj dokument će u sebi imati samo konekciju na bazu i ništa više, a uključiti će mo ga na svaki dokument koji traži konekciju na bazu sa naredbom `include`.

Primjer `connect.php` dokumenta

```
<?php
header('Content-Type: text/html; charset=utf-8');

$servername = "localhost";
$username = "root";
$password = "";
$basename = "jutarnji";

// Create connection
$dbc = mysqli_connect($servername, $username, $password, $basename) or die('Error
connecting to MySQL server.'.mysqli_error());
mysqli_set_charset($dbc, "utf8");
// Check connection
if ($dbc) {
    echo "Connected successfully";
}??>
```

Primjer uključivanja u druge dokumente:

```
<?php
include 'connect.php';
?>
```

## Index.php

Izgled početne stranice, `index.php`, ovisiti će o predlošku kojeg ste dobili za zadatak. Osim toga, na stranici `index.php`, vijesti se trebaju prikazivati u minimalno dvije kategorije (po vlastitom izboru; npr. Sport, kultura, politika, zabava ...). Iste te kategorije se trebaju nalaziti u navigaciji i voditi na stranicu `kategorija.php`, na kojoj se prikazuju sve vijesti koje nisu arhivirane, a pripadaju

toj kategoriji. Ovdje će se pokazati primjer kojeg lako možete izmijeniti da odgovara vašem zadatku.

Slijedi primjer u kojemu se ispisuje jedan red koji se sastoji od 4 članka. Uvjet za ispisivanje članaka je da pripadaju određenoj kategoriji (u ovom primjeru to je sport) i da nisu arhivirani.

Kod:

```
<?php
include 'connect.php';
define('UPLPATH', 'img/');
?>
<section class="sport">
<?php
$query = "SELECT * FROM vijesti WHERE arhiva=0 AND kategorija='sport' LIMIT 4";
$result = mysqli_query($dbc, $query);
$i=0;
while($row = mysqli_fetch_array($result)) {
    echo '<article>';
    echo '<div class="article">';
    echo '<div class="sport_img">';
    echo '<img src="" . UPLPATH . $row['slika'] . ''';
    echo '</div>';
    echo '<div class="media_body">';
    echo '<h4 class="title">';
    echo '<a href="clanak.php?id='.$row['id'].'">';
    echo $row['naslov'];
    echo '</a></h4>';
    echo '</div></div>';
    echo '</article>';
}
?>
</section>
```

Slijedi primjer izgleda navigacije

```
<nav class="navbar main_nav " role="navigation">
    <ul class="main nav navbar-nav">
    <li>
        <a href="index.php" class="">Početna</a>
    </li>
    <li>
        <a href="kategorija.php?id=sport" class="">Sport</a>
    </li>
    <li>
```

```
        <a href="kategorija.php?id=kultura" class="">Kultura</a>
    </li>
    <li>
        <a href="administracija.php" class="">Administracija</a>
    </li>
</ul>
</nav>
```

Slanje varijabli preko linka (adrese)

Naslovnica (index.php) i stranica kategorija (kategorija.php) prikazuju samo skraćeni zapis vijesti, odnosno, ovisno o zadatku, naslov, podnaslov, slika i datum nastanka vijesti. Ukoliko posjetitelj želi vidjeti cijelu vijest, potrebno je kliknuti na naslov vijesti i tada se otvara nova stranica koja prikazuje cjeloviti izgled vijesti sa svim njezinim elementima. Da bi to bilo moguće, svaki naslov treba se prikazivati kao link koji poziva stranicu clanak.php. zajedno sa id vrijednost preko koje možemo dohvatiti zapis iz baze.

```
echo '<h4 class="title">';
echo '<a href="clanak.php?id='.$row['id'].'">';
echo $row['naslov'];
echo '</a></h4>';
```

U ovom primjeru id je \$row['id'] koji se dohvaća iz baze prilikom ispisivanja naslovnice ili stranice kategorija. Id se šalje stranici clanak.php GET metodom, pa gas toga i treba prihvatiti superglobalnom varijablom \$\_GET[].

## Administracija.php

Stranica administracija.php, kao i ostale stranice, treba imati iste header i footer elemente kao i sve ostale stranice, dok je izgled glavnog sadržaja i css proizvoljan.

Primjer:

Prilikom izrade primjera, prvo je napravljena obična HTML forma, koja je zatim stavljena u while() petlju kako bi se ispisala za svaki zapis. Za atribut value u elementima su postavljene vrijednosti koje se nalaze u bazi (npr. \$row['naslov']), dok su te vrijednosti kod textarea

elementa stavljene između njegovih tagova `<textarea>${row['sazetak']} </textarea>`. Na taj način smo dobili defaultne vrijednosti elemenata, ali i mogućnost editiranja postojećeg sadržaja.

Budući da se pojedinačnom zapisu može pristupiti samo pomoću njegovog id-a, u formu je ubačen sljedeći element:

```
<input type="hidden" name="id" class="form-field-textual" value="'.${row['id']}.'">
```

Type="hidden" skriva element od prikazivanja na stranici, iako je prisutan u kodu. Vrijednost mu je postavljena na id koji je dohvaćen iz baze. Na taj način smo dobili jedinstveni id za svaki ispis, kojeg možemo koristiti kod promjene podataka ili brisanja zapisa.

```
$query = "SELECT * FROM vijesti";
$result = mysqli_query($dbc, $query);
while($row = mysqli_fetch_array($result)) {

    echo '<form enctype="multipart/form-data" action="" method="POST">
        <div class="form-item">
            <label for="title">Naslov vjesti:</label>
            <div class="form-field">
                <input type="text" name="title" class="form-field-textual"
value="'.${row['naslov']}.'">
            </div>
        </div>
        <div class="form-item">
            <label for="about">Kratki sadržaj vijesti (do 50
znakova):</label>
            <div class="form-field">
                <textarea name="about" id="" cols="30" rows="10" class="form-
field-textual">'.${row['sazetak']}.'</textarea>
            </div>
        </div>
        <div class="form-item">
            <label for="content">Sadržaj vijesti:</label>
            <div class="form-field">
                <textarea name="content" id="" cols="30" rows="10" class="form-
field-textual">'.${row['tekst']}.'</textarea>
            </div>
        </div>
        <div class="form-item">
            <label for="pphoto">Slika:</label>
            <div class="form-field">
```

```

        <input type="file" class="input-text" id="pphoto"
value="'. $row['slika'] . '" name="pphoto"/> <br>
// pokraj gumba za odabir slike pojavljuje se umanjeni prikaz postojeće slike

    </div>
</div>
<div class="form-item">
    <label for="category">Kategorija vijesti:</label>
    <div class="form-field">
        <select name="category" id="" class="form-field-textual"
value="'. $row['kategorija'] . '">
            <option value="sport">Sport</option>
            <option value="kultura">Kultura</option>
        </select>
    </div>
</div>
<div class="form-item">
    <label>Spremiti u arhivu:
    <div class="form-field">';
        if($row['arhiva'] == 0) {
            echo '<input type="checkbox" name="archive" id="archive"/>
Arhiviraj?';
        } else {
            echo '<input type="checkbox" name="archive" id="archive"
checked/> Arhiviraj?';
        }
        echo '</div>
    </label>
</div>
</div>
<div class="form-item">
    <input type="hidden" name="id" class="form-field-textual"
value="'. $row['id'] . '">
        <button type="reset" value="Poništi">Poništi</button>
        <button type="submit" name="update" value="Prihvati">
Izmjeni</button>
        <button type="submit" name="delete" value="Izbriši">
Izbriši</button>
    </div>
</form>';
}

```

## Brisanje

Dio skripte koji izvršava brisanje zapisa iz baze poziva se tek nakon što je napravljena provjera da li je stisnut gumb „Izbriši“

```
if(isset($_POST['delete'])){\n    $id=$_POST['id'];\n    $query = "DELETE FROM vijesti WHERE id=$id ";\n    $result = mysqli_query($dbc, $query);\n}
```

## Update

Isto kao brisanje, UPDATE se poziva tek nakon provjere da li je stisnut gumb „Izmjeni“

```
define('UPLPATH', 'img/');\n\nif(isset($_POST['update'])){\n    $picture = $_FILES['pphoto']['name'];\n    $title=$_POST['title'];\n    $about=$_POST['about'];\n    $content=$_POST['content'];\n    $category=$_POST['category'];\n    if(isset($_POST['archive'])){\n        $archive=1;\n    }else{\n        $archive=0;\n    }\n    $target_dir = 'img/'.$picture;\n    move_uploaded_file($_FILES["pphoto"]["tmp_name"], $target_dir);\n\n    $id=$_POST['id'];\n    $query = "UPDATE vijesti SET naslov='$title', sazetak='$about', tekst='$content',\n    slika='$picture', kategorija='$category', arhiva='$archive' WHERE id=$id ";\n    $result = mysqli_query($dbc, $query);\n}
```



## Clanak.php

U nastavku je dan primjer jednostavne PHP skripte koja dohvaća podatke iz baze (naslov, tekst vijesti i sliku za pojedinu vijest) i ispisuje ih u clanak.php dokumentu:

```
<section role="main">
    <div class="row">
        <h2 class="category"><?php
            echo "<span>".$row['kategorija']."</span>";
        ?></h2>
        <h1 class="title"><?php
            echo $row['naslov'];
        ?></h1>
        <p>AUTOR:</p>
        <p>OBJAVLJENO: <?php
            echo "<span>".$row['datum']."</span>";
        ?></p>
    </div>
    <section class="slika">
        <?php
            echo '<img src="" . UPLPATH . $row['slika'] . "">';
        ?>
    </section>
    <section class="about">
        <p>
            <?php
                echo "<i>".$row['sazetak']."</i>";
            ?>
        </p>
    </section>
    <section class="sadrzaj">
        <p>
            <?php
                echo $row['tekst'];
            ?>
        </p>
    </section>
</section>
```

## Kategorija.php

Stranica kategorija, kategorija.php, koristi isti kod za ispisivanje kao i index.php, samo što nije limitirana brojem prikaza vijesti (3 ili 4), već ispisuje sve vijesti iz baze za tu kategoriju pod uvjetom da nisu arhivirane.

Primjer:

```
$kategorija=$_GET[kategorija];  
$query = "SELECT * FROM vijesti WHERE kategorija=$kategorija";
```

Vrijednost varijable kategorija, također se šalje GET metodom preko linka u navigaciji. npr:

```
<li>  
    <a href="kategorija.php?id=sport " class="">Sport</a>  
</li>
```