

## Postavljanje softvera:

Instalirati MAMP

U opcijama odrediti www direktorij

## Instalacija

Nakon kopiranja softvera potrebno je ručno podesiti podatke za spajanje na bazu i url softvera.

### *Promijene u header.php*

Zatim je potrebno podesiti MySQL podatke, pa pronađi liniju:

```
$global_database = new mysqli("localhost", "root", "root", "webshop");
```

Prvo je ip od baze, to je obično localhost. Zatim db username i password. Zadnje je naziv baze.

### *Promijene u install.php*

Tu se također nalaze podaci za spajanje na bazu. Potrebno ih je jednako podesiti kao i u header.php s jednom malom promijenom.

```
$database = new mysqli("localhost", "root", "root", "");
```

Zadnji parametar je prazan jer za vrijeme instalacije još ne postoji tablica webshop.

### *Instalacija*

Nakon što je podešena putanja i baza pokrenite instalaciju preko browsera upisom slijedećeg urla:

<http://localhost/views/install.php>

Ukoliko su sve vrijednosti zelene baza je uspješno instalirana.

Najbolje je ovu PHP skritu zakomentirati nakon instalacije jer ona dropa sve tablice i ponovno ih kreira kako se ne bi slučajno ponovno pozvala. Tada bi se izbrisali svi podaci i baza bi opet bila prazna. Najlakše je otvoriti install.php datoteku koja se nalazi u views folderu i u prvi red ispod <?php dodati naredbu:

```
return;
```

Instalacija baze:

Uspiješno kreirana tablica users.

Uspiješno kreirana tablica categories.

Uspiješno kreirana tablica products.

Uspiješno kreirana tablica cart.

Uspiješno kreirana tablica cart\_user.

Uspiješno kreirana tablica comments.

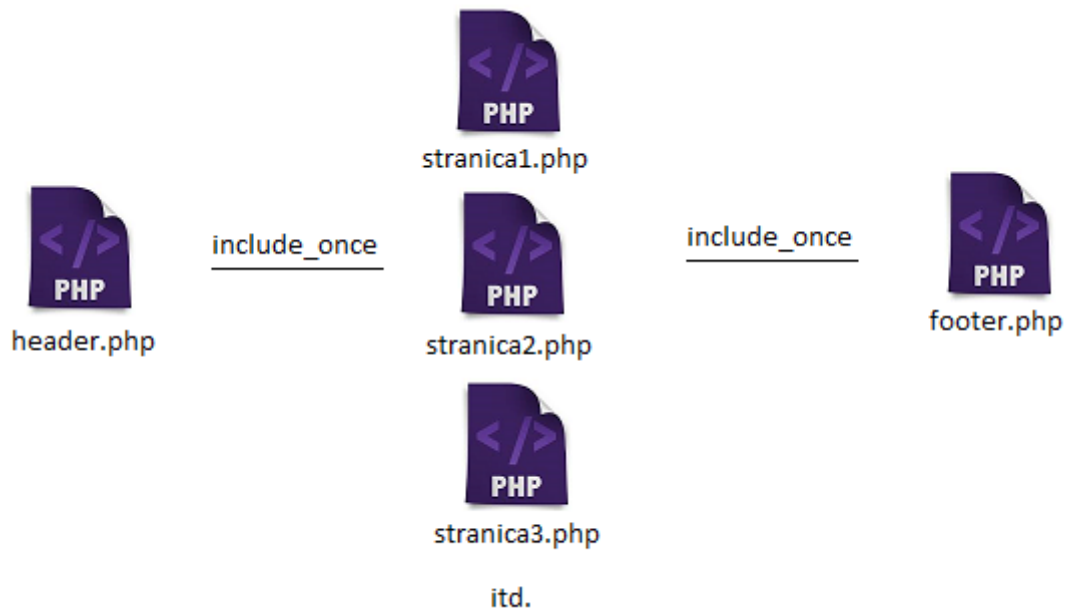
Uspiješno dodan administrator.

Instalacija baze završena.

*Slika 1 Prikaz rezultata instalacije*

## Princip rada aplikacije

Svakoj stranici se pristupa preko url-a. Npr. <http://localhost/views/stranica1.php> . Na svakoj stranici na početku je ubačen(include\_once) header.php a na kraju footer.php.



Slika 2 Princip rada aplikacije

Fileovi su posloženi u tri direktorija i root:

- Bez direktorija (u rootu direktorija) nalaze se globalni fileovi:
  - Header.php – početak svake stranice
  - Footer.php – kraj svake stranice
  - Index.php – početni file koji samo redirekta na homepage
  - Hrv.php – jezični file, tu su svi tekstovi programa zapisani
- Views
  - Sve stranice aplikacije
- Css
  - Style.css – css file
- Img
  - Tu se uploadaju slike porizvoda, nalazi se i nopic.jpg koja se prikazuje za sve proizvode bez slike

## HEADER.php

Poziva se na apsolutno svim stranicama. Najbitniji i najkompleksniji php file cijelog webshopa.

```
<?php
include_once 'hrv.php'; // dodajemo tekstove programa
global $global_txt;      // Polje za tekstove iz hrv.php
// - URL do web stranica, obavezno slash na kraju!
$global_url = "http://localhost/webshop/";
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/* POSTAVLJANJE SESSIONA */
// SESSION postavke - http://php.net/manual/en/function.session-name.php
session_name("webshop");
session_start();
// Postavljanje vremenske zone - http://php.net/manual/en/function.date-default-timezone-set.php
date_default_timezone_set("Europe/Zagreb");
```

Slika 3 Header.php početak

- <php - početni php tag

*NAPOMENA. Svi php fileovi počinju ovim tagom. Nijedan php file nema završni tag jer je nepotreban, umjesto toga na kraju svakog filea piše naziv filea unutar komentara npr.*

*/\* header.php \*/*

*To je za slučaj da se poremeti nešto sa php tagovima, u browseru će ispisati taj komentar pa onda znamo točno unutar kojeg filea je greška.*

- Include\_once 'hrv.php';

Učitava hrv.php odmah na početku. Unutra je samo definirano jedno polje \$global\_txt u kojem su zapisani svi tekstovi koji se vide u programu. To omogućava da na jednom mjestu uređuješ tekstove ili npr. da samo taj file prevedeš i cijeli program će biti na tome jeziku. Vidi ispod detaljnije u dijelu za Jezik programa.

- global \$global\_txt;

Samo pozivamo globalno polje definirano u hrv.php.

*NAPOMENA. U PHP-u svim varijablama(i poljima) koji se pozivaju unutar neke druge funkcije ili unutar nekog drugog php filea može se pristupiti nakon što ih pozovemo sa ključnom riječi global. Takve varijable nazivam globalnima.*

*Svim varijablama koje sam koristio kao globalne dao sam prefiks "global\_", npr. \$global\_database;*

- \$global\_url = <http://localhost/webshop/> ;

Postavljamo tekstualnu varijablu \$global\_url tako da na jednom mjestu imamo url aplikacije (objašnjeno u instalaciji).

U idućim linijama samo se postavlja naziv session objekta i pokreće te se postavlja vremenska zona. Ona je bitna ako radimo sa date() funkcijom php-a(a radimo jer preko nje uzimamo vrijeme kada je kupnja izvršena ili kada je komentar napisan itd.) inače nam PHP prijavi upozorenja.

Više o tome na linkovima u komentaru koji stoji u kodu.

```

////////////////////////////////////
/* SPAJANJE NA BAZU */
$global_database = new mysqli("localhost", "root", "root", "webshop");
if ($global_database -> connect_errno) {
    echo "Failed to connect to MySQL: " . $global_database ->connect_error;
    $global_database ->close();
    return;
}

// Postavlja se encoding
$global_database -> set_charset('utf8');
// AUTOCOMMIT - automatsko pisanje u bazu nakon svake naredbe postavljamo na
// Postavlja se na FALSE npr. kada se želi niz naredbi upisati u bazu ali ka
$global_database -> autocommit(TRUE);
// Konekciji se govori koju bazu treba koristiti
$global_database -> query ("USE `webshop`");

```

Slika 4 header.php spajanje na bazu

Na slici iznad prikazan je kod za spajanje na bazu. On je ubačen u header.php da se ne moramo na svakoj stranici ponovno spajati i podešavati osnovne stvari. Tu se također kreira globalna varijabla `$global_database()` koju ćemo koristiti na svakoj stranici za rad sa bazom, to je zapravo objekt klase `mysqli`. <http://php.net/manual/en/class.mysqli.php>

```

echo "<!doctype html>";
echo "<html lang='hr'>";
echo "<head>";
    echo "<meta charset='utf-8'>";
    echo "<meta name='viewport' content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=yes'>";
    echo "<link rel='stylesheet' href='\".$global_url.\"css/style.css' type='text/css' media='Screen'>";
    echo "<title>\".$global_txt['title'].\"</title>";
    echo "<meta name='description' content='Webshop, online kupnja'>";
    echo "<meta name='author' content='\">";
echo "</head>";

echo "<body>";
    echo "<div id='main'>";

```

Slika 5 header.php html početak

Nakon uvodnog kreiranja varijabli slijedi dio ispisa čistog html-a. Sve što se prikazuje korisniku je prikazano preko php echo naredbe. <http://php.net/manual/en/function.echo.php>

Osim html head-a, tu je startan i body te dodan main div unutar kojeg je zapravo cijela aplikacija. U footer.php se zatvaraju ovi otvoreni tagovi i konekcija na bazu.

Ostatak header.php filea su funkcije koje možemo zvati globalnim funkcijama. One se kao i globalne varijable pozivaju na ostalim stranicama. Npr. funkcija za prikaz glavnog izbornika programa. Nema smisla je kreirati na svakoj stranici. Tu se kreira a onda je pozivamo na svakoj stranici gdje nam treba.

```

////////////////////////////////////
// KORISNIČKI DIO - prijava, registracija ako nije prijavljen korisnik inače username i logout button
/* Prikaz podataka o korisniku ako je spojen ili ponuda za registraciju/prijavu u sustav */
function show_user_part()
{
    global $global_url;
    global $global_txt;
    if(isset($_SESSION['user']))
    {
        echo "<div id='user_part'>";
        echo $_SESSION['user']['username'];
        echo " ";
        echo "<a href='". $global_url. "views/logout.php'>";
        echo $global_txt['logout'];
        echo "</a>";
        echo "</div>";
    }
    /* Inače prikaži login i register opcije*/
    else
    {
        echo "<div id='user_part'>";
        echo "<a href='". $global_url. "views/login.php'>";
        echo $global_txt['login'];
        echo "</a>";
        echo " ";
        echo "<a href='". $global_url. "views/register.php'>";
        echo $global_txt['register'];
        echo "</a>";
        echo "</div>";
    }
}

```

Slika 6 show\_user\_part()

### Show\_user\_part()

Ako je korisnik prijavljen prikazuje username i logout link, ako nije onda prikazuje linkove za prijavu i registraciju korisnika.

Zatim se provjerava da li je user spojen u sustav. To znamo preko \$\_SESSION objekta, on će biti postavljen samo ako je user logiran u sustav(to smo odradili u login.php odnosno logout.php) inače isset vraća FALSE.

Da bi se kod izveo mora se pozvati funkcija. Ona se poziva na većini stranica(odmah na početku stranice).

### Show\_menu\_user()

```
// GLAVNI IZBORNIK
function show_menu_user($current_page)
{
    global $global_txt;

    //- provjeravamo je li admin da znamo treba li admin menu prikazati
    $isAdmin = FALSE;
    if(isset($_SESSION['user']))
    {
        if($_SESSION['user']['rights'] == '999')
        {
            $isAdmin = TRUE;
        }
    }

    echo "<div id='menu'>";
    show_menu_item("homepage.php", $global_txt['menu_home'], $current_page);
    show_menu_item("products.php", $global_txt['menu_products'], $current_page);
    //- ako je korisnik registriran prikaži i izbornik za korisničke stranice
    if(isset($_SESSION['user']))
    {
        show_cart("cart.php", $global_txt['menu_cart'], $current_page);
        show_menu_item("history.php", $global_txt['menu_user_history'], $current_page);
    }
    // adminu prikaži izbornik za administraciju
    if($isAdmin === TRUE)
    {
        show_menu_item("admin_products.php", $global_txt['menu_administration'], $current_page);
    }
    else // korisnik ima izbornik za kontakt, nema smisla ga prikazati adminu
    {
        show_menu_item("contacts.php", $global_txt['menu_contact'], $current_page);
    }

    echo "</div>"; //- kraj #menu diva
}

```

Slika 7 show\_menu\_user()

Ispis glavnog izbornika programa. Poziva se na svim stranicama osim administratorskim.

Izbornik ima slijedeće elemente:

- Početna(homepage.php)
- Proizvodi(products.php)
- Košarica(cart.php) – prikazuje se samo ako je korisnik ili admin logiran
- Povijest kupnje(history.php) – prikazuje se samo ako je korisnik ili admin logiran
- Administracija(admin\_products.php) – prikazuje se samo ako je admin logiran
- Kontaktirajte nas(constacts.php) – prikazuje se svima osim adminu

Kao parametar funkcije šalje se trenutna stranica. Npr. na products.php ovu funkciju zovemo show\_menu\_user("products.php"); Tada nam \$current\_page ima naziv stranice. To nam služi samo da bi na izborniku označili gdje se nalazimo.

Prvo provjeravamo je li korisnik admin. Admin se u instalaciji dodaje sa pravima(rights) 999 I jedino taj račun ima takva prava. Ako je korisnik koji se spojio u sustav admin on će imati rights 999 svi ostali 1. To nam je bitno da možemo prikazati adminu u izborniku link na administraciju koji nema smisla da vidi neregistrirani korisnik ili onaj koji nije admin. Njima se prikazuje contacts.php(kontaktna forma). Ona pak ne treba adminu jer nema smisla da sam sebe kontaktira. ☺

Unutar div-a #menu poziva se funkcija `show_menu_item` koja iscrtava menu. To sam prebacio u funkciju da skratim kod, općenito sve što je isto a poziva se više puta samo sa različitim ulaznim parametrima se nalazi unutar funkcija.

Ako je korisnik prijavljen(`if(isset($_SESSION['user']))`) onda prikazujemo i stanje košarice i prikaz povijesti kupnje.

#### *Show\_menu\_admin()*

```
// ADMINISTRATORSKI IZBORNIK
function show_menu_admin($current_page)
{
    global $global_txt;
    echo "<div id='menu'>";
        show_menu_item("homepage.php",          $global_txt['menu_home'],          $current_page);
        show_menu_item("admin_products.php",      $global_txt['menu_admin_products'],      $current_page);
        show_menu_item("admin_categories.php",    $global_txt['menu_admin_categories'],    $current_page);
        show_menu_item("admin_users.php",        $global_txt['menu_admin_users'],        $current_page);
    echo "</div>"; //- kraj #menu diva
}
```

Slika 8 *show\_menu\_admin()*

Ovaj izbornik je alternativa prethodnom. On se poziva na stranicama administratora.

Izbornik ima slijedeće elemente:

- Početna(homepage.php)
- Administracija proizvoda(admin\_products.php)
- Administracija kategorija(admin\_categories.php)
- Administracija korisnika(admin\_users.php)

#### *Show\_menu\_item()*

```
/* Pomoćna funkcija za iscrtavanje menu itema */
function show_menu_item($link, $item_name, $page_name)
{
    global $global_url;
    $cls = '';
    if($page_name == $link)
        $cls = "sel";

    echo "<div class='menu_item ".$cls."'>";
        echo "<a href='".$global_url."views/".$link."'>";
            echo $item_name;
        echo "</a>";
    echo "</div>";
}
```

Slika 9 *show\_menu\_item()*

Pomoćna funkcija koju koriste prethodne dvije, služi za generiranje html koda itema izbornika.



Uspoređuje naziv trenutne stranice(`$page_name`) i link(`$link`). Ako su isti dodaje klasu "sel" na taj item, to služi samo da bi označili na izborniku gdje se nalazimo. Npr. ako smo na proizvodima da na izborniku link na proizvode je označen da znamo da smo na toj stranici.

### *Show\_cart()*

```
function show_cart($link, $item_name, $page_name)
{
    $count = 0;
    global $global_url;
    global $global_database;
    // zamijeni COUNT sa SUM ako želiš količinu proizvoda, npr. ako i
    $stmt = $global_database->prepare("SELECT COUNT(cu_product Quanti

    $stmt->bind_param('i', $_SESSION['user']['id']);
    $stmt->execute();
    $stmt->bind_result($count);
    $stmt->fetch();
    $stmt->close();

    $cls = '';
    if($page_name == $link)
        $cls = "sel";

    echo "<div class='menu_item ".$cls."'>";
        echo "<a href='".$global_url."views/".$link."'>";
            echo $item_name;
            echo " ( ".$count." ) ";
        echo "</a>";
    echo "</div>";
}
```

*Slika 10 show\_cart()*

Koristi se samo u funkciji za prikaz glavnog izbornika da uz naziv i link prikaže broj artikala u košarici. Tako da se kod od prethodne funkcije(`show_menu_item()`) razlikuje samo u dijelu za pristup bazi i što je dodana linija za prikaz broja proizvoda:

```
echo " ( ".$count." ) ";
```

SQL upit se ne vidi na slici a izgleda ovako:

```
"SELECT COUNT(cu_product_quantity) FROM cart JOIN cart_user ON cart_user.FK_cu_cart_id =
cart.cart_id WHERE FK_cart_user_id=? AND cart_is_done = 0"
```

Znači vraćamo samo broj proizvoda koje imamo u košarici(ne i količinu, npr. ako imamo 2 ISTE majice vratiti će 1 kao broj proizvoda a količina je 2, ako se želi prikazati količinu treba zamijeniti COUNT sa SUM). Tablicu proizvoda košarice(`cart_user`) JOIN-amo sa tablicom košarica korisnika (`cart`) preko id-a košarice. Da nam da podatke za konkretnog usera u WHERE dodajemo `FK_cart_user_id`. Kako korisnik

može imati više košarica(jer svaka košarica predstavlja jednu kupnju a može imati neograničeno kupnji), ispisujemo samo onu košaricu kojoj je `cart_is_done=0`. Jer to je košarica koja nije kupljena.

### STRANIČENJE

Posljednje dvije funkcije `header.php` su vezane za straničenje(pagination). One su malo kompleksnije jer se straničenje odvija preko GET-a.

To se radi preko funkcije `pagination()` iz `header.php`. Dok funkcija `show_pagination()` samo iscrtava ispod linkove za pomicanje po stranicama.

To sve znači da dva puta pozivamo SQL upit. Prvi puta samo sa `COUNT` dijelom da nam vrati ukupan broj redova iz baze a drugi put, nakon što sve proračunamo, sa `LIMIT` dijelom na kraju upita. Npr.

```
$sql = "SELECT COUNT(*) AS num_of_rec FROM products WHERE FK_product_category_id=?";
```

A nakon proračuna:

```
$sql = "SELECT * FROM products WHERE FK_product_category_id=? LIMIT ".$start.", ".$end;
```

*pagination()*

```

////////////////////////////////////
/**
 * STRANIČENJE
 * Izračuni za straničenje
 *
 * @param int $per_page      - koliko redova se prikazuje po stranici
 * @param int $num_of_rec    - broj redova vraćen iz baze
 * @param int $cur_page      - trenutna stranica
 * @param int $pages         - vraćamo broj stranica
 * @param int $start         - početno za sql limit
 * @param int $end           - završno za sql limit
 */
function pagination($per_page, $num_of_rec, &$cur_page, &$number_of_pages, &$start, &$end)
{
    $number_of_pages = 1;
    if ($num_of_rec > 0)
    {
        $number_of_pages = (int)($num_of_rec / (float)($per_page));
        // Svaki put kad nije modulo jednak 0 dodaj jednu stranicu
        // jer znači ili da smo na prvoj stranici pa je $pages = 0
        // (svejedno treba tu stranicu prikazati pa moramo povećati za 1)
        // ili npr. imamo više 15 recorda a prikazuje se po 10,
        // dobiti ćemo 1 i moramo uvećati za još 1 da korisnik može doći na stranicu 2
        if (($num_of_rec % (float)$per_page) != 0)
            $number_of_pages = $number_of_pages + 1;
    }

    /* Računamo je li stranica koju trebamo prikazati unutar broja stranica */
    if($cur_page <= 0)
        $cur_page = 1;
    if($cur_page > $number_of_pages)
        $cur_page = $number_of_pages;

    /* kada znamo stranicu, onda znamo i od kuda do kuda prikazujemo podatke */
    /* Početna, npr. ako vidimo 50str za 1 str je 0*50+0 =0, za 2str je 1*50 = 50, itd.*/
    $start = (($cur_page-1) * $per_page);
    /* Zadnja je jednaka broju rekorda po stranici */
    $end    = $per_page;
}

```

Slika 11 pagination

Ulazni parametri su opisani iznad funkcije pa samo dodatak:

- \$per\_page je defaultno na svakoj stranici 10. To je koliko se proizvoda(redova iz baze) ispisuje po stranici paginationa.
- \$num\_of\_rec – je broj redova u bazi(npr. broj proizvoda)
- \$cur\_page je trenutna stranica paginationa
- \$number\_of\_pages – vraća broj pagination stranica
- \$start I \$end- njih šaljem u LIMIT dio SQL-a (\$end je uvijek jednak broju proizvoda po stranici, nisam ga trebao ni slati u funkciju ali eto, neka bude cijeli izračun na istom mjestu)

NAPOMENA - & - Call by reference. Funkcija može vratiti samo jednu vrijednost (ili polje, ili objekt itd.) međutim ako želimo neke parametre poslati u funkciju, unutar funkcije im promijeniti vrijednost I raditi sa tim izmjenjenim vrijednostima onda moramo koristiti ovaj način rada. Lijepo objašnjeno ovdje:

<http://php.net/manual/en/language.references.pass.php>

Prvo računamo broj stranica paginationa. Računamo prvo broj proizvoda u bazi / broj proizvoda koji će biti prikazan po stranici. Za gornji primjer:

```
$number_of_pages = (int)(27 / 10) = 2
```

Međutim nama trebaju tri stranice, prva od 0-10, druga 11-21, treća 22 – 27. Zbog toga radimo provjeru ostatka dijeljenja(modulo). Svaki put kad NIJE modulo jednak 0 dodati ćemo jednu stranicu(objašnjeno u komentaru). U našem slučaju ostatak dijeljenja je 7, znači veći je od 0 I dodati ćemo jednu stranicu. Tako na kraju imamo:

```
$number_of_pages = 3
```

Sada radimo samo provjeru stranice. Kako korisnik preko GET metode šalje trenutnu stranicu onda može zlonamjerno upisati neku preveliku ili malu. Da nam ne bi “poludio” program radim provjeru trenutne stranice. Ako je manja od 1 postavim je na 1 ako je veća od maksimalne, postavim je na maksimalnu.

To sprečava I slučajeve kada npr. obrišemo 7 stranica, nama u GET-u ostane stranica 3 kao trenutna a u ovom primjeru se broj stranica dinamički smanjio na 2. Ostali bi pred praznim ekranom I pitali se u čemu je problem. Ova provjera će forsirati trenutnu stranicu na maksimalnu, u tome slučaju to je stranica 2.

Posljednja stvar koja se odvija je računanje LIMIT vrijednosti. \$end je uvijek jednak broju stranica koje se prikazuju dok \$start raste pa je data formula za njegov izračun:

```
$start = (($cur_page-1) * $per_page);
```

Izračuni za primjer:

Kada smo na 1 stranici:  $\$start = ((1-1)*10) = 0;$

Kada smo na 2 stranici:  $\$start = ((2-1)*10) = 10;$

Kada smo na 3 stranici:  $\$start = ((3-1)*10) = 20;$

*Show\_pagination()*

```
// Prikazuje pagination ispod tablice, slika i sl.
function show_pagination($url, $per_page, $cur_page, $number_of_pages, $num_of_rec, $get1="", $get2="", $get3="")
{
    // Ako je samo jedna stranica nema smisla prikazivati stranicenje
    if($number_of_pages <= 1)
        return;

    // ako imamo dodatne parametre za get
    $url_extension = "";
    if(!empty($get1))
        $url_extension .= "&".$get1;
    if(!empty($get2))
        $url_extension .= "&".$get2;
    if(!empty($get3))
        $url_extension .= "&".$get3;

    global $global_txt;
    echo "<div id='pagination'>";
    echo "<a class='pagination_page' href='". $url. "?p=1&pp=".$per_page.$url_extension.">first</a>";
    $pages_before = ($cur_page - 1);
    if($pages_before > 0)
    {
        $counter = 0;
        for($i = 1; $i <= $pages_before; $i++)
        {
            echo "<a class='pagination_page' href='". $url. "?p=".$i."&pp=".$per_page.$url_extension.">".$i."</a>";
            $counter++;
            if($counter > 2)
                break;
        }
    }
    echo "<a class='pagination_page sel' href='". $url. "?p=".$cur_page."&pp=".$per_page.$url_extension.">".$cur_page.">";
    $pages_after = ($number_of_pages - $cur_page);
    if($pages_after > 0)
    {
        $counter = 0;
        for($i = $cur_page+1; $i <= $cur_page+$pages_after; $i++)
        {
            echo "<a class='pagination_page' href='". $url. "?p=".$i."&pp=".$per_page.$url_extension.">".$i."</a>";
            $counter++;
            if($counter > 2)
                break;
        }
    }
    echo "<a class='pagination_page' href='". $url. "?p=".$number_of_pages."&pp=".$per_page.$url_extension.">last</a>";
    echo "<div id='pagination_message'>";
    echo $global_txt['total_records']." ". $num_of_rec;
    echo "</div>";
    echo "</div>";
}
}
```

Slika 12 show\_pagination()

Služi za iscrtavanje paginationa. Poziva se obično ispod tablice ili liste proizvoda. Pagination se izvodi preko GET metode tako da šaljemo u url-u trenutnu stranicu i broj prikaza po stranici, moguće je da neke stranice šalju još stvari u getu (npr. filter proizvoda) pa ih prosljedimo ovdje preko varijable \$url\_extension. Znači ako želimo prikazati 2 stranicu proizvoda sa 10 proizvoda po stranici get nam izgleda ovako:

<http://localhost/views/products.php?p=2&pp=10>

p – broj trenutne stranice

pp – broj proizvoda po stranici (per page)

Parametri funkcije su:

- \$url – osnovni dio url-a stranice, onaj koji ide ispred GET dijela
  - npr. sa proizvoda šaljem
    - <http://localhost/views/products.php>
  - Sa homepage-a
    - <http://localhost/views/homepage.php>
  - Itd.
- \$per\_page – broj proizvoda po stranici (šalje se u getu pod pp=)
- \$cur\_page – trenutna stranica (šalje se u getu pod p=)
- \$number\_of\_pages – broj stranica u paginationu, služi za izračun koliko imamo stranica prije i poslije trenutne
- \$num\_of\_rec – služi samo za ispis na kraju koliko imamo podataka u bazi
- \$get1,\$get2,\$get3 – dodatni za get uvjet, to koristimo samo na proizvodima kad imamo filter

Ukoliko je samo jedna stranica pagination se neće prikazati, nema smisla.

\$url\_extension – dodaje parametre u get koji nemaju veze sa paginationom. Npr. filter na proizvodima (npr. filtriraj po veličini, boji itd.) nam je izveden preko get-a pa moramo te vrijednosti proslijediti linkovima stranica inače bi nam se filter resetirao svaki put kad bi stisli na neku stranicu.

Unutar div #pagination prvi i zadnji <a> su linkovi na prvu odnosno zadnju stranicu. Zatim računamo ima li stranica ispred (\$page\_before) i ako ima prikazujemo ih (for petlja) s time da prikazujemo maksimalno dvije prethodne stranice. Ako imamo 5 stranica i nalazimo se na 4 toj, biti će prikazani linkovi samo na 2 i 3. Na isti način računamo i stranice nakon (\$pages\_after). Između toga je dodan link na trenutnu stranicu. Pagination će izgledati ovako za primjer sa 5 stranica kada smo na 4-toj:

first 2, 3, **4**, 5 last

Ako smo na prvoj:

first **1**, 2, 3 last

itd.

## HRV.php - JEZIK PROGRAMA

Svi tekstovi koji se ispisuju nalaze su hrv.php datoteci.

```

k?php
$global_txt = array();
$global_txt['title']      = "Web trgovina";
$global_txt['email']      = "E-mail:";
$global_txt['username']   = "Korisničko ime:";
$global_txt['password']   = "Lozinka:";
$global_txt['name']       = "Ime:";
$global_txt['lastname']   = "Prezime:";
$global_txt['login']      = "Prijava";
$global_txt['register']   = "Registracija";
$global_txt['homepage']   = "Dobrodošli u web trgovinu!";
$global_txt['repassword'] = "Ponovite lozinku:";
$global_txt['submit']     = "Potvrdi";
$global_txt['logout']     = "Odjava";
$global_txt['add']        = "Dodaj";
$global_txt['edit']       = "Izmjeni";
$global_txt['cancel']     = "Odustani";
$global_txt['empty_error'] = "Potrebno je upisati obavezne vrijednosti.";
$global_txt['total_records'] = "Ukupno zapisa";
$global_txt['all']        = "Svi proizvodi";
$global_txt['add_to_cart'] = "Dodaj u košaricu";
$global_txt['details']    = "Više o proizvodu";
$global_txt['no_product'] = "Nema podataka o proizvodu.";

```

Slika 13 Jezik programa hrv.php

Na bilo kojoj stranici kada želimo ispisati neki tekst samo zovemo polje:

```
global $global_txt;
```

```
echo $global_txt['title'];
```

### *Dodavanje drugog jezika*

Ako želimo prevesti site na engleski možemo kopirati hrv.php, promijeniti mu naziv u eng.php te onda u header.php ili zamijeniti include\_once 'hrv.php' sa include\_once 'eng.php' ili dodati neko dinamičko inkludanje npr. sa zastavicama i slično.

## DATABASE

Znači imamo mysqli objekt `$global_database` koji se kreira u `header.php`. Njega koristimo na svim stranicama.

`$global_database->query()`; se koristi samo kada izvršavamo čisti upit, npr. u instalaciji i sl. gdje je 100% sigurno ili gdje nema nekih uvjeta koji se šalju u SQL-u. Inače da bi se zaštitili od SQL injection napada koristimo tkzv. "pripremljene upite". Za to koristimo `stmt`.

<http://php.net/manual/en/class.mysqli-stmt.php>

Primjer SELECT-a na bazu:

```
//- Potraži da li je već registrirana email adresa
$stmt = $global_database-> prepare("SELECT id FROM users WHERE email=?");
$stmt->bind_param('s', $_POST['email']);
$stmt->execute();
$stmt->bind_result($id);
$stmt->fetch();
$stmt->close();
```

*Slika 14 Prikaz SELECT-a na bazu*

Prvo pripremamo SQL upit. Zatim bindamo parametre za svaku vrijednost koju šaljemo u upitu (prikazano znakom ? u upitu, na gornjoj slici „email=?“).

`bind_param()`

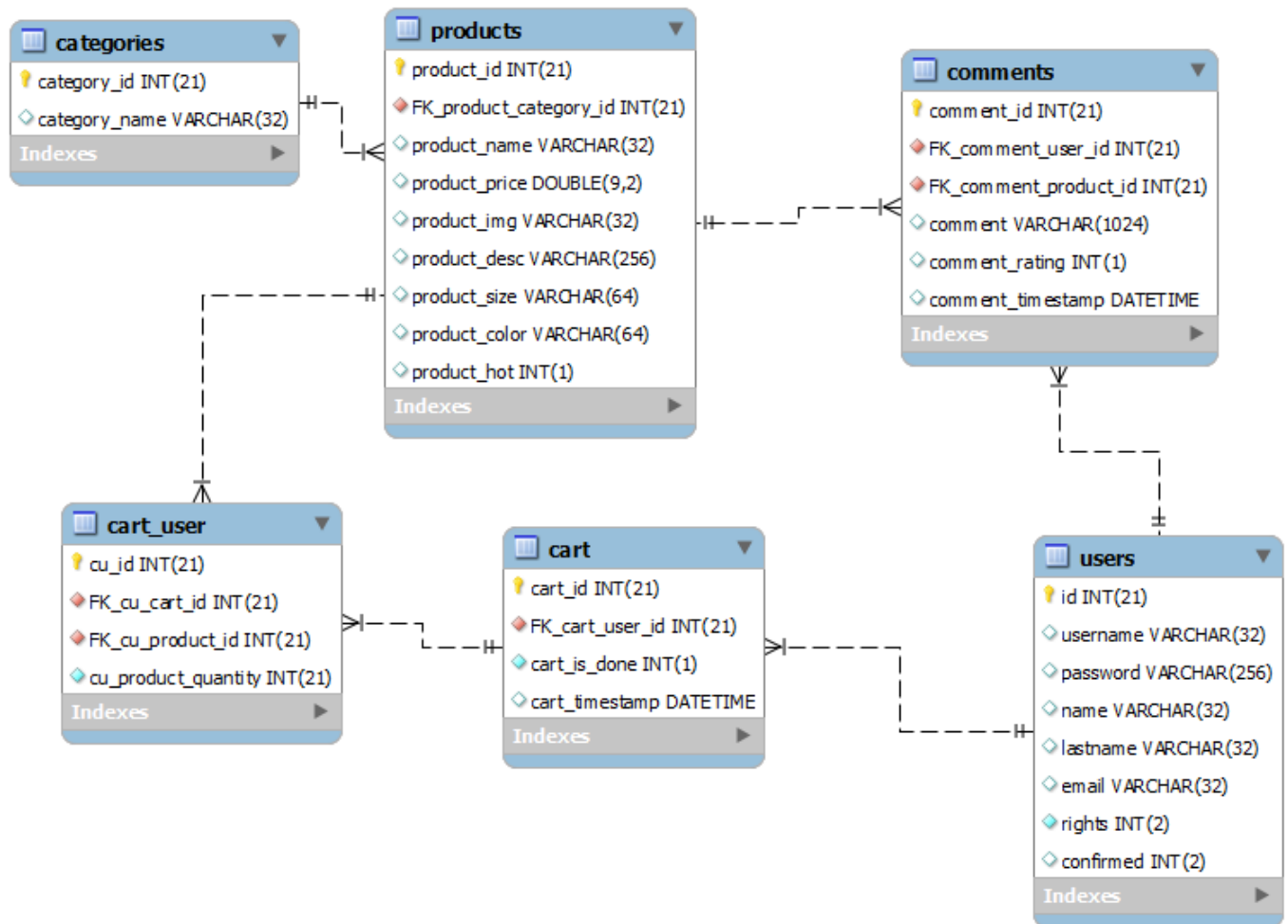
- prvi parametar funkcije 's' govori da šaljemo jedan string
- drugi parametar je sama vrijednost koja se šalje

Zatim izvršavamo sa `execute()` te vraćamo rezultat u varijabku `$id` preko `fetch()` funkcije. Na kraju zatvaramo statement.

Ako imamo više rezultata onda vrtimo `while($stmt->fetch())`.



## OPIS TABLICA BAZE



Slika 15 Model podataka

Tablice u bazi:

- categories
- products
- comments
- cart\_user
- cart
- users

## CATEGORIES

Tablica kategorija. Sastoji se od id-a i naziva kateogrija. Kategorije se dodaju preko admin panela.

Npr. Majice, Hlaće, Čarape. Ili što već želimo da bude kategorija.

category_id	category_name
1	Hlaće

Slika 16 tablica kategorija

## PRODUCTS

Tablica proizvoda. Npr. Lacosta majica, XL, crvena.

Proizvodi se dodaju u admin panelu. Potrebno je prvo dodati kategorije da bi mogli dodavati proizvode. Proizvod je moguće označiti kao hot, to znači da će se prikazati na početnom ekranu. Moguće je dodati sliku za svaki proizvod.

product_id	FK_product_category_id	product_name	product_price	product_img	product_desc	product_size	product_color	product_hot
1	1	Lacoste hlaćice	450.00		Dobre hlaće	XL	plava	1
2	1	Nike trenerka	340.00	slika.jpg	Dugi rukavi	L	Crvena	0

Slika 17 tablica proizvoda

## COMMENTS

Tablica komentara i ocijena. Svaki korisnik može dodati jedan komentar za svaki proizvod i jednom ga ocijeniti. Nije moguće mijenjati komentar ili ocijenu. Zapisuje se vrijeme kada je komentiran i tko ga je komentirao.

comment_id	FK_comment_user_id	FK_comment_product_id	comment	comment_rating	comment_timestamp
1	1	1	Odlične hlaće, dobro mi i stoje.	5	2015-01-19 15:03:10
2	2	1	Odvratni krokodil!	2	2015-01-19 15:04:17
3	2	2	Lijepa crvena majica.	3	2015-01-19 15:04:38

Slika 18 tablica komentara

## CART

Tablica kupovina korisnika. Kada korisnik doda prvi item u košaricu kreira se zapis. Taj zapis ima `cart_is_done = 0`. Kako korisnik dodaje proizvode oni se dodaju u tablicu `cart_user` i vezani su za ovu tablicu preko `FK_cu_cart_id`.

Kada korisnik obavi kupnju `cart_is_done` se postavlja na 1 i upisuje se `cart_timestamp` kao vrijeme kupnje.

Ako korisnik želi novu kupnju, čim doda proizvod u košaricu kreira se novi zapis u ovoj tablici sa `cart_us_done = 0`. I tako u krug.

Kada prikazujemo koje kupnje je korisnik obavio čitamo iz ove tablice sve one koje imaju id korisnika (`FK_cart_user_id`) za kojeg gledamo kupnje i `cart_is_done = 1`. JOIN-amo sa tablicom `cart_user` da bi ispisali točno proizvode koje je kupio kao i cijenu.

Košarica funkcionira isto kao i to samo gledamo `cart_is_done = 0` (uvijek samo jedan po korisniku ima `cart_is_done = 0`).

### *CART\_USER*

Tablica koja sadrži proizvode i količinu kupljenog proizvoda. Povezana je preko ključa sa tablicom `products` i sa tablicom `cart`.

Ona sadrži sve kupljene proizvode i sve proizvode trenutno u košaricama. Preko tablice `cart` znamo da li je kupljen proizvod i čija je košarica (koji user).

### *Primjer: CART – CART\_USER*

Zamislimo da za proizvode imamo (id proizvoda, naziv):

- 1 , jabuka
- 2 , banana
- 3 , jagoda
- 4 - šljiva

<code>cart_id</code>	<code>FK_cart_user_id</code>	<code>cart_is_done</code>	<code>cart_timestamp</code>
1	1	1	2015-01-21 14:44:40
2	1	0	NULL

*Slika 19 tablica kupnji korisnika*

U primjeru tablice iznad korisnik sa id-em 1 (to je admin) je napravio jednu kupnju 21.01.2015 u 14:44h. Vidimo da je upisano vrijeme kupnje (`cart_timestamp`) i označeno je da je kupljeno (`cart_is_done`). Nakon toga dodao je još neke proizvode u košaricu ali nije završio kupnju. To se vidi po tome što je isti user id a `cart_is_done` je 0.

Pogledajmo tablicu `cart_user`:

<code>cu_id</code>	<code>FK_cu_cart_id</code>	<code>FK_cu_product_id</code>	<code>cu_product_quantity</code>
1	1	1	2
2	1	2	1
3	1	3	1
4	2	4	2
5	2	2	3

*Slika 20 tablica proizvoda po košaricama*

Vidimo da je korisnik kupio dvije (`cu_product_quantity`) jabuke, te po jednu bananu i jagodu. To su kupljeni proizvodi jer su oni iz košarice sa id-em 1 (`FK_cu_cart_id`) za koju znamo da je kupljena (`cart_is_done` je 1). Ovo će se ispisati na povijesti kupnje (`history.php`) za korisnika.

Također vidimo da trenutno u košarici trenutno ima dvije šljive i tri banane. No tu kupnju nije obavio. Ti proizvodi će se pokazati na košarici (`cart.php`)

Vrijednost(cijena) se računa tako da ovim tablicama JOIN-amo product\_price iz tablice products te je pomnožimo sa količinom(cu\_product\_quantity).

## USERS

Tablica korisnika. Nakon registracije admin potvrđuje korisnika. Dok to ne napravi korisnik se ne može spojiti u sustav. To je polje confirmed u ovoj tablici a može biti 0 ili 1. Admin to obavlja preko administratorskog panela kod pregleda korisnika.

Rights je polje sa pravima korisnika, svi imaju 1 osim admina koji ima 999.

Password se zapisuje kao sha256 vrijednost u bazu. Na taj način štiti se lozinka korisnika(nikad se ne smije lozinka upisati u bazu kao običan tekst, time se narušava privatnost korisnika i omogućava krađa podataka).

## FOOTER.php

```
<?php
    echo "</div>"; // kraj main diva
    echo "<div class='footer'>Ovo je footer!</div>";
    echo "<body>"; // kraj body
echo "</html>"; // kraj html

/* Zatvaramo konekciju na bazu */
global $global_database;
// Uništava se thread korisnika da ne ostane nepotrebno u memoriji
$thread_id = $global_database->thread_id;
$global_database -> kill($thread_id);
// Zatvaranje konekcije
$global_database -> close();
/* footer.php */
```

Slika 21 Footer.php

Na kraju svake stranice se poziva footer.php. On završava tagove:

- div #main
- body
- html

Zatvara konekciju na bazu. Tu forsiramo kill threada baze iako će MySQL sam nakon određenog vremena napraviti timeout za usera ovako forsiramo skidanje usera sa baze.

## STRANICE

### INDEX.php

Index.php se poziva kada upišemo url <http://localhost>. Sve što radi je redirekcija na homepage.php jer je to početna stranica webshopa. Sve stranice webshopa nalaze se unutar views foldera.

```
<?php
// PRIKAZ STRANICE
include_once 'header.php';
global $global_url;
header('Location: ' . $global_url."views/homepage.php");

/* index.php */
```

Slika 22 index.php kod

Svaka stranica programa za prvu liniju ima include\_once 'header.php'.

Zadnja linija radi redirekciju.

<http://php.net/manual/en/function.header.php>

### LOGIN.php

Stranica prijave u sustav.

```
<?php
include_once "../header.php";
show_menu_user("login.php");

global $global_url;
global $global_txt;
global $global_database;

$error_msg = "";    //- varijabla za poruku o pogrešci, inicijalno nema pogreš
$username  = "";    //- varijabla za korisničko ime
```

Slika 23 login.php početak

Prikazujemo show\_menu\_user("login.php"); Ovdje ne prikazujemo linkove za registraciju I login inače bi prije ove linije imali show\_user\_part().

Pozivamo globalne varijable, imamo ih samo tri u programu i skoro svugdje se pozivaju sve tri.