

Visoka škola strukovnih studija za informacione i komunikacione tehnologije



Dokumentacija

(<https://coffeestoreblend.000webhostapp.com>)

Predmet: Web programiranje PHP1

Student: David Carević 159/14

Beograd, Jun 2018.

Sadržaj :

1.	Uvod	4
•	Korišćeni programiski jezici	4
•	Opis funkcionalnosti.....	4
•	Template.....	5
•	Korišćeni CSS.....	6
2.	Organizacija	6
•	Organizaciona šema	6
•	Mapa sajta	6
•	Stranice	7
○	Home.....	7
○	About	8
○	Author	9
○	Admin.....	10
○	Contact.....	10
○	Shop	12
○	User	13
3.	Kodovi	13
•	PHP	13
○	about.php.....	13
○	admin.php	14
○	ajax_insert_poll.php.....	58
○	author.php	59
○	config.php	61
○	connection.php	61
○	contact.php	62
○	footer.php	64
○	funkcije.php (preuzeto sa lab vežbi)	64
○	header.php.....	65

○ home.php	69
○ index.php	69
○ logout.php	70
○ reg.php	70
○ shop.php	75
○ user.php	76
● CSS	79
○ stil.css	79
○ lightbox.css(preuzeto sa interneta)	81
● JavaScript(JQuery)	85
○ efekti.js	85
○ lightbox.js(preuzeto sa interneta)	90

1. Uvod

- Korišćeni programiski jezici

Za izradu sajta su korišćeni sledeći programski i markup jezici:

HTML, CSS, CSS3, JavaScript, XML, JQuery, AJAX. Sajt je urđen u Visual Studio Code.

- Opis funkcionalnosti

Sajt ima sledeće funkcionalnosti:

- **Drop down login** koji je urađen pomoću JQuery-ja nalazi se na svim stranicama radi lepšeg vizuelnog efekta. Omogućava prijavljivanje korisnika.
- **Anketa** odrađena u PHP i Ajax-u. Nalazi se na strani Author. Anketa ajaxom insertuje u bazu, a zatim se prikazuju rezultate korisniku. Korisnik koji je glasao više ne može da vidi formu za glasanje.
- **Forma za registraciju** odrađena pomoću JS-a, JQuery-ja, PHP-a. Validacija se vrši preko regularnih izraza. Nalazi se na stranici Register.
- **Forma za kontakt** odrađena pomoću JS-a i PHP-a . Validacija se vrši preko regularnih izraza. Nalazi se na stranici Contact. Njome se šalje email administartoru sajta.
- **Hover efekat** preko linkova u meniju uradjen pomoću JQuery-a. Takođe se efekat daje I dugmićima.
- **Hover efekat** preko slika izveden u css3.
- **Galerija na Home strani** urađena pomoću fotorama JQuery plugina koji se poziva putem interneta.
- **Galerija na About us strani** urađena pomoću PHP i lightbox plugina.

- **Galerija na Shop strani** urađena pomoću PHP i lightbox plugina. Shop se ceo učitava iz baze bez mogućnosti kupovine.
- **Forma na User strani** omogućava korisniku da promeni svoj password i email.
- **Forme na Admin strani** omogućavaju korisnicima sa administratorskim privilegijama da kompletno kontrolišu sav dinamički sadržaj sajta.
- **Sav sadržaj** sem logo-a i footera sajta se učitava dinamički iz baze podataka.
- **Zaštita stranica sesijma.**

- Template

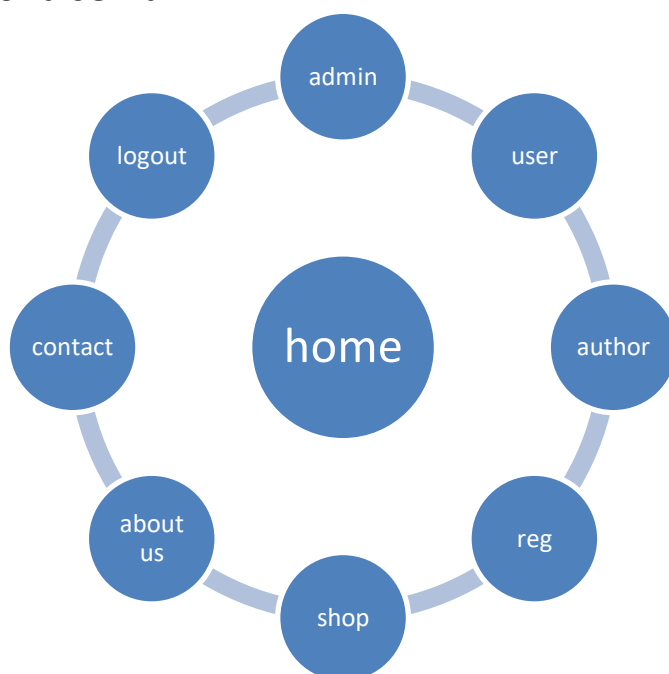


Sadržaj u centru se dinamički menja u zavisnosti od strane. Footer i header ostaju uvek.

- Korišćeni CSS
Nema tuđeg CSS-a sem za lightbox plugin.

2. Organizacija

- Organizaciona šema



Sve stranice mogu da pristupaju jedna drugoj pod uslovom da korisnik ima sesiju.

- Mapa sajta

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
    http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">
  <!-- created with Free Online Sitemap Generator www.xml-sitemaps.com -->

  <url>
    <loc>https://coffeestoreblend.000webhostapp.com/index.php</loc>
```

```

<lastmod>2018-06-18T10:44:47+00:00</lastmod>
<priority>1.00</priority>
</url>
<url>
  <loc>https://coffeestoreblend.000webhostapp.com/index.php?page=home</loc>
  <lastmod>2018-06-18T10:44:47+00:00</lastmod>
  <priority>0.80</priority>
</url>
<url>
  <loc>https://coffeestoreblend.000webhostapp.com/index.php?page=about</loc>
  <lastmod>2018-06-18T10:44:47+00:00</lastmod>
  <priority>0.80</priority>
</url>
<url>
  <loc>https://coffeestoreblend.000webhostapp.com/index.php?page=reg</loc>
  <lastmod>2018-06-18T10:44:47+00:00</lastmod>
  <priority>0.80</priority>
</url>
</urlset>

```

- Stranice
- Home



Coffee Store Blend

[Home](#) [About us](#) [Register](#) [Log In](#)

Welcome to the website of Coffee Shop Blend

Are you a fan of coffee? Like trying out different types of coffee?
Look no further, here we offer you the best blends of coffee you can find for a cheap price.
Please create your account and see what we have to offer.



Powered by  000webhost

Nalazi se glavni meni sa formom za logovanje koja se stvara na klik iz menija animacijom fadeIn-Out, tekst se učitava iz baze kao i galerija koja se zatim stavlja u fotorama plugin.

- About



Coffee Store Blend

[Home](#) [About us](#) [Register](#) [Log In](#)



Since the doors first opened in 1887 coffee and tea have dominated the shops atmosphere. Today, 130 years later, Coffe Store Blend has evolved into one of the worlds best known and leading suppliers of coffee. Offering a choice of over 80 coffees from around the world. The list continues to grow as we continue to source speciality coffee from small corners of the world. We also pride ourselves on sourcing original, delicious and fine confectionery.



There is also a large variety of domestic coffee makers. The shop still retains some original features, the wooden shelves along the walls, the original wooden counter and display case. If you are in the area please come and visit us. You may also like to join us for a coffee.



If ever there is something that you cannot find please contact us as we may have something in store that is not featured online.

Powered by 000webhost

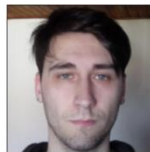
Sadržaj se učitava iz baze. Zbog narušavanja izgleda, ova stranica može samo da se update-uje u admin panelu. Slike se mogu uvećati sa lightboxom na klik.

- Author



Coffee Store Blend

[Home](#) [About us](#) [Shop](#) [David](#) [Contact](#) [Author](#) [Log Out](#) [Documentation](#)



My name is David Carev. I'm 4th year student at Viskoa ICT, Belgrade. This website was made as school project and is meant to showcase my current skillset when it comes to web programming. It was made using HTML, CSS, JS (and JQuery), PHP, AJAX. Most of the content of the website is being pulled from a database for easier manipulation and updating. The reason why I chose a coffee store as the theme for the website is simply because I'm a huge fan of different coffee blends and I tend to like trying out different and exotic brands of coffee. I hope you enjoy the website and please rate it via the poll.

Website ratings



Powered by  000webhost

Na author stranici se pojavljuje anketa inicijalno korisniku, kada glasa, njegov glas se čuva u veznoj tabeli i više ne može da vidi formu za glasanje, umesto toga prikazuje mu se statistika glasanja. Upis glasa vrši se AJAX-om.

- Admin



Coffee Store Blend

[Home](#) [About us](#) [Shop](#) [Admin](#) [Contact](#) [Author](#) [Log Out](#) [Documentation](#)

User

Nav

Home

About us

Shop

Author

Poll

Select a site page from the menu and edit it! To delete and update rows, use the checkboxes at the left of them.

User							
#	ID	Username	Password	Email	Name	Lastname	RoleId
<input type="checkbox"/>	1	admin	f949d7ba0bbd24214b186c	david.carevic.159.14@ict.e	David	Carevic	1
<input type="checkbox"/>	27	David	172522ec1028ab781d9dfc	david@david.com	David	David	2
<input type="checkbox"/>	30	Nex95	0d4e3eb97b434fce188ce8	nemanja@nemanja.com	Nemanja	Milic	2
<input type="checkbox"/>	31	Strahinja	4096fbf8691f74ce95fbc67	strahinja1@strahinja.com	Strahinja	Mikic	2
<input type="checkbox"/>	32	Marko	c28aa76990994587b0e907	marko1@marko.com	Marko	Marko	2
<input type="checkbox"/>	33	Milica	932e512d0da2821efe2b81	milica@milica.com	Milica	Milica	2
<input type="checkbox"/>	34	Marja	cb74c183402afe708a490f	maja@maja.com	Marja	Marja	2
<input type="checkbox"/>	38	Mirko	13592f2ca86af30572a825	mirko@mirko.co	Mirko	Mirko	2
<input type="checkbox"/>	39	Borka	1e0fa2162b09107c463a2a	borka@borka.com	Borka	Borka	2
<input type="checkbox"/>	47	Katarina	fe5648c1e094d163e68b8b	kata@kata.com	Katarina	Katarina	1
<input type="checkbox"/>	62	Dex56	8ffcafd621ae99733b9dec	dex565@gmail.com	Dex	Dexai	2
		+					
<div><div>Delete</div><div>Insert</div><div>Update</div></div>							

Powered by 000webhost

Admin strana inicijalno pokazuje samo side meni, koji kada se izabere stavka pokazuje tabelu koja omogućava modifikaciju stranice. Stranice author i about us nemaju funkcije za insert i delete zbog narušavanja izgleda. User i Poll se odnose na tabelu sa korisnicima i tabelu izbora za anketu.

- Contact



Coffee Store Blend

[Home](#) [About us](#) [Shop](#) [David](#) [Contact](#) [Author](#) [Log Out](#) [Documentation](#)

Contact

Subject:

Message:

Send

Powered by 000webhost

Kontakt stranica sadrži formu kojom se administratoru sajta šalje email.



Coffee Store Blend

[Home](#) [About us](#) [Register](#) [Log In](#)

New Account

Username:

Password:

E-mail:

Information

First name:

Last name:







[Create your account](#)

Reg stranica sadrži formu za kreiranje naloga. Provera se radi regularnim izrazima u javascriptu i php-u.



Coffee Store Blend

[Home](#) [About us](#) [Shop](#) [David](#) [Contact](#) [Author](#) [Log Out](#) [Documentation](#)

 <p>Affogato Price: \$2 Description: An affogato is a simple dessert coffee that is treat during summer and after dinner. It is made by placing one big scoope of vanilla ice cream within a single or double shot of espresso</p> <p>Add to cart</p>	 <p>Espresso Price: \$1 Description: The espresso (aka "short black") is the foundation and the most important part to every espresso based drink</p> <p>Add to cart</p>	 <p>Long Macchiato Price: \$3 Description: A long macchiato is the same as a short macchiato but with a double shot of espresso. The same rule of thirds applies in the traditionally made long macchiato</p> <p>Add to cart</p>
 <p>Americano Price: \$2 Description: A long black (aka "americano") is hot water with an espresso shot extracted on top of the hot water</p> <p>Add to cart</p>	 <p>Piccolo Latte Price: \$3 Description: A piccolo latte is a café latte made in an espresso cup. This means it has a very strong but mellowed down espresso taste thanks to the steamed milk and micro foam within it. There are two ways of making a piccolo latte, with either 1 espresso shot or 1 ristretto shot</p> <p>Add to cart</p>	 <p>Ristretto Price: \$2 Description: A ristretto is an espresso shot that is extracted with the same amount of coffee but half the amount of water. The end result is a more concentrated and darker espresso extraction</p> <p>Add to cart</p>

Powered by 000webhost

Shop strana iz baze učitava artikle i prikazuje ih. Slike su thumbnailovi koji se funkcijom posebno kreiraju pri insert u bazu. Lightbox prikauje pravu verziju slike. Dugmići ne rade ništa.

- User



Coffee Store Blend

[Home](#) [About us](#) [Shop](#) [David](#) [Contact](#) [Author](#) [Log Out](#) [Documentation](#)

Hello David, here you can update your data.

New password:
<input type="text"/>
New Email:
<input type="text" value="david@david.com"/>
<input type="button" value="Update your account"/>

Powered by 000webhost

User strana omogućava korisniku da promeni svoj password i email. Email mu se automatski postavlja na trenutni. Proverava se regularnim izrazima.

3. Kodovi

- PHP

- about.php

```
<?php
$rez=executeQuery("SELECT * FROM about;");
//var_dump($rez);
?>

<div class="about-holder">

    <div class="about-image"><a href="<?php echo $rez[0]->src;?>" data-
lightbox="aboutus">alt?>" /></a></div>

    <div class="about-text"><p><h4><?php echo $rez[0]->text;?></h4></p></div>
</div>
```

```

<div class="about-holder">
<div class="about-text"><p><h4><?php echo $rez[1]->text;?></h4></p></div>
<div class="about-image"><a href="<?php echo $rez[1]->src;?>" data-
lightbox="aboutus">alt?>" /></a></div>
</div>

<div class="about-holder">
<div class="about-image"><a href="<?php echo $rez[2]->src;?>" data-
lightbox="aboutus">alt?>" /></a></div>
<div class="about-text"><p><h4><?php echo $rez[2]->text;?></h4></p></div>

</div>

```

○ admin.php

```

<?php
if(isset($_SESSION['user']) && $_SESSION['user']->role=='admin'):
?>
<div id="omotAdmin">
<div id="desni">
<h3>Select a site page from the menu and edit it! To delete and update rows, use the
checkboxes at the left of them.</h3>

<!--DELETE-->
<?php
if(isset($_POST['delete-user'])){

    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $upit=$konekcija->prepare("DELETE FROM user WHERE id=:c");
            $upit->bindParam(':c',$c);
            $rez=$upit->execute();
            if($rez){
                echo"<h4 style='color:green;'>Users have been deleted! </h4>";
            }
            else{
                echo"<h4 style='color:red;'>Error! </h4>";
            }
        }
    }
}
else{
    echo"<h4 style='color:red;'>Error you must check something! </h4>";
}
}

```

```

}

if(isset($_POST['delete-nav'])){
    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $upit=$konekcija->prepare("DELETE FROM nav WHERE id=:c");
            $upit->bindParam(':c',$c);
            $rez=$upit->execute();
            if($rez){
                echo"<h4 style='color:green;'>Link have been deleted! </h4>";
            }
            else{
                echo"<h4 style='color:red;'>Error! </h4>";
            }
        }
    }
}
else{
    echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

if(isset($_POST['delete-home-text'])){
    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $upit=$konekcija->prepare("DELETE FROM home_text WHERE id=:c");
            $upit->bindParam(':c',$c);
            $rez=$upit->execute();
            if($rez){
                echo"<h4 style='color:green;'>Text has been deleted! </h4>";
            }
            else{
                echo"<h4 style='color:red;'>Error! </h4>";
            }
        }
    }
}
else{
    echo"<h4 style='color:red;'>Error you must check something! </h4>";
}
}

```

```

}

if(isset($_POST['delete-home-image'])){
    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $upit=$koneksi->prepare("DELETE FROM home_image WHERE id=:c");
            $upit->bindParam(':c',$c);
            $rez=$upit->execute();
            if($rez){
                echo"<h4 style='color:green;'>Images have been deleted! </h4>";
            }
            else{
                echo"<h4 style='color:red;'>Error! </h4>";
            }
        }
    }
}
else{
    echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

if(isset($_POST['delete-shop'])){
    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $upit=$koneksi->prepare("DELETE FROM shop WHERE id=:c");
            $upit->bindParam(':c',$c);
            $rez=$upit->execute();
            if($rez){
                echo"<h4 style='color:green;'>Items have been deleted! </h4>";
            }
            else{
                echo"<h4 style='color:red;'>Error! </h4>";
            }
        }
    }
}
else{
    echo"<h4 style='color:red;'>Error you must check something! </h4>";
}
}

```



```

}

if(isset($_POST['delete-poll'])){
    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $upit=$konekcija->prepare("DELETE FROM poll WHERE id=:c");
            $upit->bindParam(':c',$c);
            $rez=$upit->execute();
            if($rez){
                echo"<h4 style='color:green;'>Votes have been deleted! </h4>";
            }
            else{
                echo"<h4 style='color:red;'>Error! </h4>";
            }
        }
    }
}
else{
    echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

?>

<!--DELETE-->

<!--INSERT-->
<?php

if(isset($_POST['insert-user'])){

    $user=$_POST['newusername'];
    $pass1=$_POST['newpassword'];
    $email=$_POST['newemail'];

```

```

$name=$_POST['newname'];
$lastname=$_POST['newlastname'];
$roleId=$_POST['newrole'];

$errors=[];

$reRoleId="/^[1-2]{1}$/";
if(!preg_match($reRoleId,$roleId)){ $errors[]="Role Id is not okay.";}

$reUser="/^[A-Za-z0-9]{5,20}$/";
if(!preg_match($reUser,$user)){

    $errors[]="Username is not good.";
}
$rePass1="/^[A-Za-z0-9]{5,15}$/";
if(!preg_match($rePass1,$pass1)){
    $errors[]="Pass is not good.";
}

if(!filter_var($email, FILTER_VALIDATE_EMAIL)){

    $errors[]="Email is not good.";

}

$reName="/(^([A-Z]{1}[a-z]{2,19})(\s[A-Z]{1}[a-z]{2,19})*$/";
if(!preg_match($reName,$name)){

    $errors[]="Name is not good.";
}

$reLastName="/(^([A-Z]{1}[a-z]{4,39})(\s[A-Z]{1}[a-z]{4,39})*$/";
if(!preg_match($reLastName,$lastname)){

    $errors[]="Lastname is not good.";
}
$rezUser=$koneksi->prepare("SELECT username FROM user WHERE username=:user");
$rezUser->bindParam(':user',$user);
$rezUser->execute();
$rezU=$rezUser->fetch();
if($rezU){
    $errors[]="Username is taken !";
}
$rezEmail=$koneksi->prepare("SELECT email FROM user WHERE email=:email");
$rezEmail->bindParam(':email',$email);
$rezEmail->execute();
$rezE=$rezEmail->fetch();
if($rezE){
    $errors[]="Email is taken !";
}

```

```

if(count($errors)>0){

    echo"<ul style='margin:auto;color:red;text-align:center;'>";
    foreach($errors as $error){
        echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
    }
    echo"</ul>";

}
else{

    $pass1=md5($pass1);

    $upit="INSERT INTO user VALUES(NULL,:user,:pass1,:email,:name,:lastname,:roleid)";

    $priprema_upit=$konekcija->prepare($upit);
    $priprema_upit->bindParam(":user",$user);
    $priprema_upit->bindParam(":pass1",$pass1);
    $priprema_upit->bindParam(":email",$email);
    $priprema_upit->bindParam(":name",$name);

    $priprema_upit->bindParam(":lastname",$lastname);
    $priprema_upit->bindParam(":roleid",$roleId);
    try{
    $rezultat= $priprema_upit->execute();
    }
    catch(PDOException $ex){
    echo $ex->getMessage();
    }

    if($rezultat){

```

```

echo"<p><h4 style='margin:0auto;text-align:center;color:green'>A new account has been
inserted !</h4></p>";

}
else{ echo "Error with the database.";}

}

}

if(isset($_POST['insert-nav'])){

    $errors=[];

    $hrefP=$_POST['newhref'];
    $href="";
    if($hrefP=="#"){ $href="#";}
    else{

        $reHref="/^[A-Za-z0-9\.\?\=\:\\/\-\]{1,50}$/";

        if(!preg_match($reHref,$hrefP)){ $errors[]="You can eather type in a link or '#.';"}
        $href=$hrefP;
    }
    $name=$_POST['newname'];
    $reName="/^[A-z]{1,50}$/";
    if(!preg_match($reName,$name)){ $errors[]="Wrong format of the link name.";}
    $type=$_POST['newtype'];
    $reType="/^[1-5]{1}$/";
    if(!preg_match($reType,$type)){ $errors[]="Type can only be from 1 to 5";}

    if(count($errors)>0){

        echo"<ul style='margin:auto;color:red;text-align:center;'>";
        foreach($errors as $error){
            echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
        }
        echo"</ul>";

    }
    else{

        $upit=$koneksi->prepare("INSERT INTO nav VALUES(NULL,:href,:name,:type)");
        $upit->bindParam(':href',$href);
        $upit->bindParam(':name',$name);
        $upit->bindParam('type',$type);
    }
}

```

```

try{
    $rez=$upit->execute();
}
catch(PDOException $ex){
    echo $ex->getMessage();
}

if($rez){

    echo"<p><h4 style='margin:0auto;text-align:center;color:green'>A new navigation item
has been created !</h4></p>";

}
else{ echo "Error with the database.";}

}

}

if(isset($_POST['insert-home-text'])){
    $errors=[];
    $text="";

    if($_POST['newtext']!=""){
        $text=$_POST['newtext'];
    }

    else{$errors[]="The textfield must not be empty !";}

    //echo $text;

    if(count($errors)>0){

        echo"<ul style='margin:auto;color:red;text-align:center;'>";
        foreach($errors as $error){
            echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
        }
        echo"</ul>";
    }
}

```

```

}
else{
    $upit=$konekcija->prepare("INSERT INTO home_text VALUES(NULL,:text)");
    $upit->bindParam(':text',$text);

    try{
        $rez=$upit->execute();
    }
    catch(PDOException $ex){
        echo $ex->getMessage();
    }

    if($rez){

        echo"<p><h4 style='margin:0auto;text-align:center;color:green'>New homepage text has
been added !</h4></p>";

    }
    else{ echo "Error with the database.";}
}

}

}

if(isset($_POST['insert-home-image'])){

    $errors=[];

    $slika=$_FILES['newsrc'];
    $tip=$slika['type'];
    $name=$slika['name'];
    $tmp=$slika['tmp_name'];
    $size=$slika['size'];

    $filter=array('image/jpg','image/jpeg','image/png');

    $src="slike/".time().$name;

    $alt=$_POST['newalt'];

    // echo $alt."<br/>";
    //var_dump($slika);

    $reAlt="/^[A-Za-z0-9\s]{1,50}$/";

```

```

if(!preg_match($reAlt,$alt)){ $errors[]="Alt is in the wrong format";}

if(!in_array($tip,$filter)){ $errors[]='Image type is not okay.';}
if($size>3500000){ $errors[]='The image must not be larger than 3.5MB';}


if(count($errors)>0){

    echo"<ul style='margin:auto;color:red;text-align:center;'>";
foreach($errors as $error){
echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
}
echo"</ul>";

}
else{

    $upload=move_uploaded_file($tmp,$src);
    if($upload){

        echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Image uploaded
!</h4></p>";

        $upit=$koneksi->prepare("INSERT INTO home_image VALUES(NULL,:src,:alt)");
        $upit->bindParam(":src",$src);
        $upit->bindParam(':alt',$alt);

        try{
            $rez=$upit->execute();
        }
        catch(PDOException $ex){
            echo $ex->getMessage();
        }

        if($rez){

            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>New image has been
added to the homepage !</h4></p>";

        }
        else{ echo "Error with the database.";}
    }
    else{
        echo"<p><h4 style='margin:0auto;text-align:center;color:red'>Image was not uploaded
.</h4></p>";
    }
}

```

```

    }
}

}

if(isset($_POST['insert-shop'])){
    //335x250
    //$putanja_mala="slike/mala_".time().$ime_fajla;
    //malaSlika($putanja,$putanja_mala,100,100);

    $errors=[];

    $slika=$_FILES['newsrc'];

    $tip=$slika['type'];
    $name=$slika['name'];
    $tmp=$slika['tmp_name'];
    $size=$slika['size'];

    $filter=array('image/jpg','image/jpeg','image/png');

    $src="slike/shop/".time().$name;
    $src_mala="slike/thumbnails/".time().$name;

    //echo $src."<br/>".$src_mala."<br/>";

    $title=$_POST['newtitle'];
    $reTitle="/^[A-Z]{1}[a-z]{2,19})(\s[A-Z]{1}[a-z]{2,19})*$/";

    if(!preg_match($reTitle,$title)){ $errors[]="Title is in the wrong format.";}

    $price=$_POST['newprice'];
    $rePrice="/^[0-9]{1,10}$/";
    if(!preg_match($rePrice,$price)){ $errors[]="Price is in the wrong format.";}

    $description="";
    if($_POST['newdescription']!=""){
        $description=$_POST['newdescription'];}

    else{ $errors[]="The textfield must not be empty !";}

    //var_dump($slika);

```



```

if(!in_array($tip,$filter)){ $errors[]='Image type is not okay.';}
if($size>3500000){ $errors[]='The image must not be larger than 3.5MB';}

if(count($errors)>0){

    echo"<ul style='margin:auto;color:red;text-align:center;'>";
foreach($errors as $error){
echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
}
echo"</ul>";

}

else{

    $upload=move_uploaded_file($tmp,$src);
    malaSlika($src,$src_mala,335,250);
    if($upload){

        echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Image uploaded
!</h4></p>";

        $upit=$konekcija->prepare("INSERT INTO shop
VALUES(NULL,:src,:src_mala,:title,:price,:description)");
        $upit->bindParam(":src",$src);
        $upit->bindParam(':src_mala',$src_mala);
        $upit->bindParam(':title',$title);
        $upit->bindParam(':price',$price);
        $upit->bindParam(':description',$description);

        try{
            $rez=$upit->execute();
        }
        catch(PDOException $ex){
            echo $ex->getMessage();
        }

        if($rez){

            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>New item has been
added to the shop !</h4></p>";

        }
        else{ echo "Error with the database.";}
    }
}
else{

```

```

        echo"<p><h4 style='margin:0auto;text-align:center;color:red'>Image was not uploaded
.</h4></p>";
    }
}

}

if(isset($_POST['insert-poll'])){

    $errors=[];

    $voteId=$_POST['newvoteid'];
    $vote=$_POST['newvote'];

    $reVoteId="/^[0-9]{1,10}$/";
    $reVote="/^([A-Z]{1}[a-z]{2,39})(\s[A-Z]{1}[a-z]{2,39})*$/";

    //echo $voteId." ".$vote;

    if(!preg_match($reVoteId,$voteId)){ $errors[]="Id is not in the correct format.";}
    if(!preg_match($reVote,$vote)){ $errors[]="Vote name is not in the correct format";}

    $upitId=$koneksi->prepare("SELECT * FROM poll WHERE id=:id");
    $upitId->bindParam(':id',$voteId);
    $rezultat=$upitId->execute();
    $rezultat=$upitId->fetch();
    if($rezultat){
        $errors[]="You must choose a differnt ID for the vote.";
    }

    if(count($errors)>0){

        echo"<ul style='margin:auto;color:red;text-align:center;'>";
        foreach($errors as $error){
            echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
        }
        echo"</ul>";

    }
    else{
        $upit=$koneksi->prepare("INSERT INTO poll VALUES(:id,:vote_name)");
        $upit->bindParam(':id',$voteId);
        $upit->bindParam(':vote_name',$vote);

        try{
            $rez=$upit->execute();
        }
        catch(PDOException $ex){

```

```
echo $ex->getMessage();
}

if($rez){

    echo"<p><h4 style='margin:0auto;text-align:center;color:green'>New vote option has
been added!</h4></p>";

}
else{ echo "Error with the database.";}

}

}

?>

<!--INSERT-->
```

```

<!--UPDATE-->
<?php
if(isset($_POST['update-user'])){

    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $id=$c;
            $user=$_POST['username'].$c;
            $pass1=$_POST['pass'].$c;
            $email=$_POST['email'].$c;
            $name=$_POST['name'].$c;
            $lastname=$_POST['lastname'].$c;
            $roleId=$_POST['idRole'].$c;

            //echo
            $id."<br/>".$user."<br/>".$pass."<br/>".$email."<br/>".$name."<br/>".$lastname."<br/>".$id
            Role."<br/>";

            $errors=[];

            $reRoleId="/^[1-2]{1}$/";
            if(!preg_match($reRoleId,$roleId)){ $errors[]="Role Id is not okay.";}

            $reUser="/^[A-Za-z0-9]{5,20}$/";
            if(!preg_match($reUser,$user)){

                $errors[]="Username is not good.";
            }
            $rePass1="/^[A-Za-z0-9]{5,15}$/";
            if(!preg_match($rePass1,$pass1)){
                $errors[]="Pass is not good.";
            }

            if(!filter_var($email, FILTER_VALIDATE_EMAIL)){

                $errors[]="Email is not good.";

            }

            $reName="/(^[A-Z]{1}[a-z]{2,19})(\s[A-Z]{1}[a-z]{2,19})*$/";
            if(!preg_match($reName,$name)){

                $errors[]="Name is not good.";
            }

            $reLastName="/(^[A-Z]{1}[a-z]{4,39})(\s[A-Z]{1}[a-z]{4,39})*$/";
            if(!preg_match($reLastName,$lastname)){

```

```

$errors[]="Lastname is not good.";
}

$rezUser=$konekcija->prepare("SELECT id,username FROM user WHERE username=:user");
$rezUser->bindParam(':user',$user);
$rezUser->execute();
$rezU=$rezUser->fetch();
if($rezU){
    if($rezU->id!=$id){
        $errors[]="Username is taken !";
    }
}
$rezEmail=$konekcija->prepare("SELECT id,email FROM user WHERE email=:email");
$rezEmail->bindParam(':email',$email);
$rezEmail->execute();
$rezE=$rezEmail->fetch();
if($rezE){
    if($rezE->id!=$id){
        $errors[]="Email is taken !";
    }
}
}

if(count($errors)>0){

    echo"<ul style='margin:auto;color:red;text-align:center;'>";
    foreach($errors as $error){
        echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
    }
    echo"</ul>";

}
else{

$pass1=md5($pass1);

```

```

$upit="UPDATE user SET
username=:user,password=:pass1,email=:email,name=:name,lastname=:lastname,idRole=:roleid
WHERE id=:id;";

$priprema_upit=$konekcija->prepare($upit);
$priprema_upit->bindParam(":id",$id);
$priprema_upit->bindParam(":user",$user);
$priprema_upit->bindParam(":pass1",$pass1);
$priprema_upit->bindParam(":email",$email);
$priprema_upit->bindParam(":name",$name);

$priprema_upit->bindParam(":lastname",$lastname);
$priprema_upit->bindParam(":roleid",$roleId);
    try{
$rezultat= $priprema_upit->execute();
}
catch(PDOException $ex){
echo $ex->getMessage();
}

if($rezultat){

echo"<p><h4 style='margin:0auto;text-align:center;color:green'>User has been updated
!</h4></p>";

}
else{ echo "Error with the database.";}

}

}

}

else{
echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

if(isset($_POST['update-nav'])){

```

```

if(isset($_POST['cb'])){
    $cb=$_POST['cb'];

    foreach($cb as $c){
        $id=$c;
        $hrefP=$_POST['href'].$c;
        $name=$_POST['name'].$c;
        $type=$_POST['type'].$c;

        $errors=[];

        $href="";
        if($hrefP=="#"){ $href="#"; }
        else{

            $reHref="/^[A-Za-z0-9\.\?\=\:\\/\-\]{1,50}$/";

            if(!preg_match($reHref,$hrefP)){ $errors[]="You can eather type in a link or '#.'; }
            $href=$hrefP;
        }

        $reName="/^[A-z]{1,50}$/";
        if(!preg_match($reName,$name)){ $errors[]="Wrong format of the link name."; }

        $reType="/^[1-5]{1}$/";
        if(!preg_match($reType,$type)){ $errors[]="Type can only be from 1 to 5"; }


        if(count($errors)>0){

            echo"<ul style='margin:auto;color:red;text-align:center;'>";
            foreach($errors as $error){
                echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
            }
            echo"</ul>";

        }
        else{

```

```

$upit="UPDATE nav SET href=:href,name=:name,type=:type WHERE id=:id;";

$priprema_upit=$konekcija->prepare($upit);
$priprema_upit->bindParam(":id",$id);
$priprema_upit->bindParam(":href",$href);
$priprema_upit->bindParam(":name",$name);
$priprema_upit->bindParam(":type",$type);

    try{
$rezultat= $priprema_upit->execute();
    }
    catch(PDOException $ex){
echo $ex->getMessage();
    }

    if($rezultat){

        echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Navigation element has
been updated !</h4></p>";

    }
    else{ echo "Error with the database.";}

}

}

}

}
else{

```



```

echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

if(isset($_POST['update-home-text'])){

    $errors=[];
    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $id=$c;
            $text="";

            if($_POST['text'].$c!=""){
                $text=$_POST['text'].$c;
            }

            else{$errors[]="The textfield must not be empty !";}

        }

    }

    if(count($errors)>0){

        echo"<ul style='margin:auto;color:red;text-align:center;'>";
        foreach($errors as $error){
            echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
        }
        echo"</ul>";

    }
    else{

```

```

$upit="UPDATE home_text SET text=:text WHERE id=:id;";

$priprema_upit=$konekcija->prepare($upit);
$priprema_upit->bindParam(":id",$id);
$priprema_upit->bindParam(":text",$text);

    try{
$rezultat= $priprema_upit->execute();
}
catch(PDOException $ex){
echo $ex->getMessage();
}

if($rezultat){

echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Home text has been updated
!</h4></p>";

}
else{ echo "Error with the database.";}

    }

}
    }

```

```

else{
echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

if(isset($_POST['update-home-image'])){

    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $id=$c;
            $errors=[];

            $slika=$_FILES['src'].$c;
            $tip=$slika['type'];
            $name=$slika['name'];
            $tmp=$slika['tmp_name'];
            $size=$slika['size'];

            $filter=array('image/jpg','image/jpeg','image/png');

            $src="slike/".time().$name;

            $alt=$_POST['alt'].$c;

            // echo $alt."<br/>";
            //var_dump($slika);

            $reAlt="/^[A-Za-z0-9\s]{1,50}$/";

            if(!preg_match($reAlt,$alt)){ $errors[]="Alt is in the wrong format";}

            if(!in_array($tip,$filter)){ $errors[]='Image type is not okay.';}
            if($size>3500000){ $errors[]='The image must not be larger than 3.5MB';}


            if(count($errors)>0){

                echo"<ul style='margin:auto;color:red;text-align:center;'>";
                foreach($errors as $error){
echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
}

```

```

echo"</ul>";

}
else{

    $upload=move_uploaded_file($tmp,$src);
    if($upload){

        echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Image uploaded
!</h4></p>";

        $upit=$koneksi->prepare("UPDATE home_image SET src=:src,alt=:alt WHERE id=:id");
        $upit->bindParam(":id",$id);
        $upit->bindParam(":src",$src);
        $upit->bindParam(':alt',$alt);

        try{
            $rez=$upit->execute();
        }
        catch(PDOException $ex){
            echo $ex->getMessage();
        }

        if($rez){

            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Home image has been
updated !</h4></p>";

        }
        else{ echo "Error with the database.";}
    }
    else{
        echo"<p><h4 style='margin:0auto;text-align:center;color:red'>Image was not uploaded
.</h4></p>";
    }
}

}

}

}
else{
echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

```

```

}

if(isset($_POST['update-about'])){

    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $id=$c;
            $errors=[];

            $slika=$_FILES['src'].$c;
            $tip=$slika['type'];
            $name=$slika['name'];
            $tmp=$slika['tmp_name'];
            $size=$slika['size'];

            $filter=array('image/jpg','image/jpeg','image/png');

            $src="slike/".time().$name;

            $alt=$_POST['alt'].$c;

            if($_POST['text'].$c!=""){
                $text=$_POST['text'].$c;
            }

            else{$errors[]="The textfield must not be empty !";}

            // echo $alt."<br/>";
            //var_dump($slika);

            $reAlt="/^[A-Za-z0-9\s]{1,50}$/";

            if(!preg_match($reAlt,$alt)){ $errors[]="Alt is in the wrong format";}

            if(!in_array($tip,$filter)){ $errors[]='Image type is not okay.';}
            if($size>3500000){ $errors[]='The image must not be larger than 3.5MB';}


            if(count($errors)>0){

                echo"<ul style='margin:auto;color:red;text-align:center;'>";
                foreach($errors as $error){
                    echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
                }
                echo"</ul>";
            }
        }
    }
}

```

```

}
else{

    $upload=move_uploaded_file($tmp,$src);
    if($upload){

        echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Image uploaded
!</h4></p>";

        $upit=$koneksi->prepare("UPDATE about SET src=:src,alt=:alt,text=:text WHERE
id=:id");
        $upit->bindParam(":id",$id);
        $upit->bindParam(":src",$src);
        $upit->bindParam(':alt',$alt);
        $upit->bindParam(':text',$text);

        try{
            $rez=$upit->execute();
        }
        catch(PDOException $ex){
            echo $ex->getMessage();
        }

        if($rez){

            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>About item has been
updated !</h4></p>";

        }
        else{ echo "Error with the database.";}
    }
    else{
        echo"<p><h4 style='margin:0auto;text-align:center;color:red'>Image was not uploaded
.</h4></p>";
    }
}

}

}

}
else{
    echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

```

```

}

}

if(isset($_POST['update-author'])){

    if(isset($_POST['cb'])){
        $cb=$_POST['cb'];

        foreach($cb as $c){
            $id=$c;
            $errors=[];

            $slika=$_FILES['src'].$c;
            $tip=$slika['type'];
            $name=$slika['name'];
            $tmp=$slika['tmp_name'];
            $size=$slika['size'];

            $filter=array('image/jpg','image/jpeg','image/png');

            $src="slike/".time().$name;

            $alt=$_POST['alt'].$c;

            if($_POST['text'].$c!=""){
                $text=$_POST['text'].$c;}}

            else{$errors[]="The textfield must not be empty !";}

            // echo $alt."<br/>";
            //var_dump($slika);

            $reAlt="/^[A-Za-z0-9\s]{1,50}$/";

            if(!preg_match($reAlt,$alt)){ $errors[]="Alt is in the wrong format";}

            if(!in_array($tip,$filter)){ $errors[]='Image type is not okay.';}
            if($size>3500000){ $errors[]='The image must not be larger than 3.5MB';}


            if(count($errors)>0){

                echo"<ul style='margin:auto;color:red;text-align:center;'>";
                foreach($errors as $error){
                    echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";

```

```

}
echo"</ul>";

}
else{

    $upload=move_uploaded_file($tmp,$src);
    if($upload){

        echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Image uploaded
!</h4></p>";

        $upit=$koneksi->prepare("UPDATE author SET src=:src,alt=:alt,text=:text WHERE
id=:id");
        $upit->bindParam(":id",$id);
        $upit->bindParam(":src",$src);
        $upit->bindParam(':alt',$alt);
        $upit->bindParam(':text',$text);

        try{
            $rez=$upit->execute();
        }
        catch(PDOException $ex){
            echo $ex->getMessage();
        }

        if($rez){

            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Author information
has been updated !</h4></p>";

        }
        else{ echo "Error with the database.";}
    }
    else{
        echo"<p><h4 style='margin:0auto;text-align:center;color:red'>Image was not uploaded
.</h4></p>";
    }
}

}

}
}

```



```

else{
echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

if(isset($_POST['update-shop'])){

if(isset($_POST['cb'])){
$cb=$_POST['cb'];

foreach($cb as $c){
$id=$c;
$errors=[];

$sluka=$_FILES['src'].$c;
$tip=$sluka['type'];
$name=$sluka['name'];
$tmp=$sluka['tmp_name'];
$size=$sluka['size'];

$filter=array('image/jpg','image/jpeg','image/png');

$src="slike/shop/".time().$name;
$src_mala="slike/thumbnails/".time().$name;

$price=$_POST['price'].$c;

$alt=$_POST['alt'].$c;

if($_POST['text'].$c!=""){
$text=$_POST['text'].$c;}}

else{$errors[]="The textfield must not be empty !";}

// echo $alt."<br/>";
//var_dump($sluka);

$reAlt="/^([A-Z]{1}[a-z]{2,19})(\s[A-Z]{1}[a-z]{2,19})*$$/";

if(!preg_match($reAlt,$alt)){ $errors[]="Title is in the wrong format";}

$rePrice="/^[0-9]{1,10}$/";
if(!preg_match($rePrice,$price)){ $errors[]="Price is in the wrong format.";}

if(!in_array($tip,$filter)){ $errors[]='Image type is not okay.';}
if($size>3500000){ $errors[]='The image must not be larger than 3.5MB';}

```

```

    if(count($errors)>0){

        echo"<ul style='margin:auto;color:red;text-align:center;'>";
        foreach($errors as $error){
            echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
        }
        echo"</ul>";

    }
    else{

        $upload=move_uploaded_file($tmp,$src);
        malaSlika($src,$src_mala,335,250);
        if($upload){

            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Image uploaded
!</h4></p>";

            $upit=$konekcija->prepare("UPDATE shop SET
src=:src,thumb=:src_mala,title=:alt,price=:price,description=:text WHERE id=:id");
            $upit->bindParam(":id",$id);
            $upit->bindParam(":src",$src);
            $upit->bindParam(":src_mala",$src_mala);
            $upit->bindParam(':alt',$alt);
            $upit->bindParam(':price',$price);
            $upit->bindParam(':text',$text);

            try{
                $rez=$upit->execute();
            }
            catch(PDOException $ex){
                echo $ex->getMessage();
            }

            if($rez){

                echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Shop item has been
updated !</h4></p>";

            }
            else{ echo "Error with the database.";}
        }
        else{
            echo"<p><h4 style='margin:0auto;text-align:center;color:red'>Image was not uploaded
.</h4></p>";
        }
    }
}

```

```

    }
}

}

}
else{
echo"<h4 style='color:red;'>Error you must check something! </h4>";
}

}

if(isset($_POST['update-poll'])){

$errors=[];
    if(isset($_POST['cb'])){
$cb=$_POST['cb'];

foreach($cb as $c){
    $id=$c;
    $vote=$_POST['vote'.'$c'];

    $reVote="/^([A-Z]{1}[a-z]{2,39})(\s[A-Z]{1}[a-z]{2,39})*$/";

    if(!preg_match($reVote,$vote)){ $errors[]="Vote name is not in the correct format";}
}
}
}

```

```

if(count($errors)>0){

    echo"<ul style='margin:auto;color:red;text-align:center;'>";
    foreach($errors as $error){
        echo "<li style='color:red;list-style:none;'><h4>".$error."</h4></li>";
    }
    echo"</ul>";

}
else{

$upit="UPDATE poll SET vote_name=:vote WHERE id=:id;";

$priprema_upit=$konekcija->prepare($upit);
$priprema_upit->bindParam(":id",$id);
$priprema_upit->bindParam(":vote",$vote);

    try{
$rezultat= $priprema_upit->execute();
    }
    catch(PDOException $ex){
echo $ex->getMessage();
    }

if($rezultat){

echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Vote name has been updated
!</h4></p>";

```



```

<!--USER !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<?php
    if(isset($_POST['user'])):

?>

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'].'?page=admin';?>">
<table style="border:2px solid #222;display:inline-block;" class="adminSelect">
    <th colspan=8 style="background-color:#222;color:white;">User</th>
<tr>
    <td>#</td>
    <td>ID</td>
    <td>Username</td>
    <td>Password</td>
    <td>Email</td>
    <td>Name</td>
    <td>Lastname</td>
    <td>RoleId</td>

    </tr>

<?php

    $rez=executeQuery('SELECT * FROM user');

    foreach($rez as $red): ?>

        <tr>
            <td><input type="checkbox" name="cb[]" value="<?php echo $red->id;?>"></td>
            <td><input type="text" disabled value="<?php echo $red->id;?>" /></td>
            <td><input type="text" name="username<?php echo $red->id;?>" value="<?php echo $red->username;?>" /></td>
            <td><input type="text" name="pass<?php echo $red->id;?>" value="<?php echo $red->password;?>" /></td>
            <td><input type="text" name="email<?php echo $red->id;?>" value="<?php echo $red->email;?>" /></td>
            <td><input type="text" name="name<?php echo $red->id;?>" value="<?php echo $red->name;?>" /></td>
            <td><input type="text" name="lastname<?php echo $red->id;?>" value="<?php echo $red->lastname;?>" /></td>
            <td><input type="text" name="idRole<?php echo $red->id;?>" value="<?php echo $red->idRole;?>" /></td>

        </tr>

```



```

<td>Name</td>
<td>Type</td>

</tr>

<?php

$rez=executeQuery('SELECT * FROM nav');

foreach($rez as $red): ?>

<tr>
<td><input type="checkbox" name="cb[]" value="<?php echo $red->id;?>"></td>
<td><input type="text" disabled value="<?php echo $red->id;?>" /></td>
<td><input type="text" name="href<?php echo $red->id;?>" value="<?php echo $red-
>href;?>" /></td>
<td><input type="text" name="name<?php echo $red->id;?>" value="<?php echo $red-
>name;?>" /></td>
<td><input type="text" name="type<?php echo $red->id;?>" value="<?php echo $red-
>type;?>" /></td>

</tr>
<?php endforeach;?>

<tr>
<td></td>
<td>+</td>
<td><input type="text" name="newhref" /></td>
<td><input type="text" name="newname" /></td>
<td><input type="text" name="newtype" /></td>

</tr>

<tr>

<td colspan="8">
<input type="submit" class="Btn" name="delete-nav" value="Delete"/>
<input type="submit" class="Btn" name="insert-nav" value="Insert"/>
<input type="submit" class="Btn" name="update-nav" value="Update"/>
</td>

</tr>

</table>

```



```

</form>

<?php endif;?>

<!--NAV !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<!--HOME !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->
<?php
    if(isset($_POST['home'])):

?>

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'].'?page=admin';?>">
<table style="border:2px solid #222;display:inline-block;" class="adminSelect">
    <th colspan=7 style="background-color:#222;color:white;">Home_Text</th>
<tr>
    <td>#</td>
    <td>ID</td>
    <td>Text</td>

    </tr>

<?php

    $rez=executeQuery('SELECT * FROM home_text');

    foreach($rez as $red): ?>

<tr>
<td><input type="checkbox" name="cb[]" value="<?php echo $red->id;?>"></td>
    <td><input type="text" disabled value="<?php echo $red->id;?>" /></td>
    <td><textarea name="text<?php echo $red->id;?>"><?php echo $red-
>text;?></textarea></td>

    </tr>
<?php endforeach;?>

    <tr>
<td></td>
<td>+</td>
<td><textarea name="newtext"></textarea></td>

```

```

</tr>

<tr>

<td colspan="8">
<input type="submit" class="Btn" name="delete-home-text" value="Delete"/>
<input type="submit" class="Btn" name="insert-home-text" value="Insert"/>
<input type="submit" class="Btn" name="update-home-text" value="Update"/>
</td>

</tr>

</table>
</form>

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'].'?page=admin';?>"
enctype="multipart/form-data">

<table style="border:2px solid #222;display:inline-block;" class="adminSelect">
<th colspan=7 style="background-color:#222;color:white;">Home_Image</th>
<tr>
<td>#</td>
<td>ID</td>
<td>Src</td>
<td>Alt</td>

</tr>

<?php

$rez=executeQuery('SELECT * FROM home_image');

foreach($rez as $red): ?>

<tr>
<td><input type="checkbox" name="cb[]" value="<?php echo $red->id;?>"></td>
<td><input type="text" disabled value="<?php echo $red->id;?>" /></td>
<td><input type="file" name="src<?php echo $red->id;?>" /></td>
<td><input type="text" name="alt<?php echo $red->id;?>" value="<?php echo $red-
>alt;?>" /></td>

</tr>
<?php endforeach;?>

<tr>
<td></td>
<td>+</td>

```

```
<td><input type="file" name="newsrsrc"/></td>
<td><input type="text" name="newalt"/></td>

</tr>

<tr>

<td colspan="8">
<input type="submit" class="Btn" name="delete-home-image" value="Delete"/>
<input type="submit" class="Btn" name="insert-home-image" value="Insert"/>
<input type="submit" class="Btn" name="update-home-image" value="Update"/>
</td>

</tr>

</table>
</form>

<?php endif;?>

<!--HOME !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<!--ABOUT US !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<?php
    if(isset($_POST['about'])):

?>

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'].'?page=admin';?>"
enctype="multipart/form-data">
<table style="border:2px solid #222;display:inline-block;" class="adminSelect">
    <th colspan=7 style="background-color:#222;color:white;">About</th>
<tr>
    <td>#</td>
    <td>ID</td>
    <td>Src</td>
    <td>Alt</td>
    <td>Text</td>

</tr>

<?php

$rez=executeQuery('SELECT * FROM about');
```



```

        <td>Title</td>
        <td>Price</td>
        <td>Description</td>

    </tr>

<?php

    $rez=executeQuery('SELECT * FROM shop');

    foreach($rez as $red): ?>

        <tr>

            <td><input type="checkbox" name="cb[]" value="<?php echo $red->id;?>"></td>
            <td><input type="text" disabled value="<?php echo $red->id;?>" /></td>
            <td><input type="file" name="src<?php echo $red->id;?>" /></td>

            <td><input type="text" name="alt<?php echo $red->id;?>" value="<?php echo $red->title;?>" /></td>
            <td><input type="text" name="price<?php echo $red->id;?>" value="<?php echo $red->price;?>" /></td>
            <td><textarea name="text<?php echo $red->id;?>"><?php echo $red->description;?></textarea></td>

        </tr>
    <?php endforeach;?>

    <tr>
    <td></td>
    <td>+</td>

    <td>
        <input type='file' name="newsrc">
    </td>
    <td><input type="text" name="newtitle" /></td>
    <td><input type="text" name="newprice" /></td>
    <td><textarea name="newdescription" ></textarea></td>

    </tr>

    <tr>

<td colspan="8">

```

```
<input type="submit" class="Btn" name="delete-shop" value="Delete"/>
<input type="submit" class="Btn" name="insert-shop" value="Insert"/>
<input type="submit" class="Btn" name="update-shop" value="Update"/>
</td>

</tr>

</table>
</form>

<?php endif;?>

<!--SHOP !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<!--AUTHOR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<?php
    if(isset($_POST['author'])):

?>

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'].'?page=admin';?>"
enctype="multipart/form-data">
<table style="border:2px solid #222;display:inline-block;" class="adminSelect">
    <th colspan=7 style="background-color:#222;color:white;">Author</th>
<tr>
    <td>#</td>
    <td>ID</td>
    <td>Src</td>
    <td>Alt</td>
    <td>Text</td>

    </tr>

<?php

$rez=executeQuery('SELECT * FROM author');

foreach($rez as $red): ?>

<tr>
<td><input type="checkbox" name="cb[]" value="<?php echo $red->id;?>"></td>
<td><input type="text" disabled value="<?php echo $red->id;?>" /></td>
<td><input type="file" name="src<?php echo $red->id;?>" /></td>
<td><input type="text" name="alt<?php echo $red->id;?>" value="<?php echo $red-
>alt;?>" /></td>
```

```

        <td><textarea name="text"><?php echo $red->id;?>"><?php echo $red-
>text;?></textarea></td>

    </tr>
    <?php endforeach;?>

    <tr>

<td colspan="8">

<input type="submit" class="Btn" name="update-author" value="Update"/>
</td>

</tr>

    </table>
</form>

<?php endif;?>

<!--AUTHOR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<!--POLL !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-->

<?php
    if(isset($_POST['poll'])):

?>

<form method="POST" action="<?php echo $_SERVER['PHP_SELF'];?>">
<table style="border:2px solid #222;display:inline-block;" class="adminSelect">
    <th colspan=7 style="background-color:#222;color:white;">Poll</th>
<tr>
    <td>#</td>
    <td>ID</td>
    <td>Vote_Name</td>

    </tr>

<?php

    $rez=executeQuery('SELECT * FROM poll');

    foreach($rez as $red): ?>

```



```
<li>
  <input type="submit" class="Btn" name="user" value="User"/>
<li>
<li>
  <input type="submit" class="Btn" name="nav" value="Nav"/>
<li>
<li>
<input type="submit" class="Btn" name="home" value="Home"/>
<li>
<li>
<input type="submit" class="Btn" name="about" value="About us"/>
<li>
<li>
<input type="submit" class="Btn" name="shop" value="Shop"/>
<li>
<li>
<input type="submit" class="Btn" name="author" value="Author"/>
<li>
<li>
<input type="submit" class="Btn" name="poll" value="Poll"/>
<li>

</ul>
</form>
</div>
```

```
<div class="cistac"></div>
</div>
```

```
<?php else:echo "<p><h2 style='color:red;text-align:center;'>Error 403. You cannot access
this page.</h2></p>"; header("Location:index.php?page=home");
endif;
?>
```

- o ajax_insert_poll.php

```
<?php

$statusCode = 404;

if($_SERVER['REQUEST_METHOD'] != "POST"){
    echo "Can't access this page.";
}

if(isset($_POST['id'])){

    $id = $_POST['id'];
    $vote=$_POST['vote'];

    include "connection.php";

    $upit = $konekcija->prepare("INSERT INTO user_poll VALUES(:id,:vote)");
    $upit->bindParam(':id', $id);
    $upit->bindParam(':vote', $vote);

    try{
        $rezultat = $upit->execute();

        if($rezultat){
            $statusCode = 204;
        } else {
            $statusCode = 500;
        }
    }
    catch(PDOException $e){
        $statusCode = 500;
    }
}

http_response_code($statusCode);
```

- author.php

```
<?php if(isset($_SESSION['user'])):?>

    <?php
        $upit="SELECT * FROM author";
        $rez=executeQuery($upit);

        foreach($rez as $red):

            ?>
            <div id="author-image"><a href="<?php echo $red->src;?>" data-lightbox="author">alt;?>" /></a></div>

            <div id="author">
                <h4><?php echo $red->text;?></h4>

            </div>

            <?php endforeach;?>

        <?php

            $idUser=$_SESSION['user']->idUser;
            $upit=$konekcija->prepare('SELECT * FROM user_poll WHERE id_user=:id');
            $upit->bindParam(':id',$idUser);
            $upit->execute();
            $rez=$upit->fetch();

            if(!$rez):
                ?>

                <table>
                    <th colspan="5"><h2>Poll</h2></th>
                    <tr>

                        <th colspan="5"><h4>How would you rate this website?</h4></th>

                    </tr>
                    <tr>

                <?php
```

```

        $anketa=executeQuery('SELECT * FROM poll');

        foreach($anketa as $red):
            ?>
            <td><input type="radio" name="rbPoll" value=<?php echo $red->id;?> /><?php
echo $red->vote_name?></td>

            <?php endforeach;?>

        </tr>
        <tr>
            <th colspan="5"><input type="button" class="Btn" id="btnVote" name='btnVote'
value="Vote"/></th>

            <input type="hidden" id="userId" value='<?php echo $_SESSION['user']-
>idUser;?>'>
        </tr>

    </table>

    <?php else:

        $upit=$konekcija->query('SELECT vote_name, count(id_vote) as count FROM user_poll
up RIGHT OUTER JOIN poll p on up.id_vote=p.id GROUP BY vote_name ORDER BY id');
        $rez=$upit->execute();
        $rez=$upit->fetchAll();

        ?>

    <script>
window.onload = function () {

var options = {
    title: {
        text: "Website ratings"
    },
    data: [
        {

```

```

        type: "column",
        dataPoints: [

            <?php foreach($rez as $red):?>
            { label: "<?php echo $red->vote_name;?>", y: <?php echo $red->count;?> },
            <?php endforeach;?>

        ]
    }
]
};

$("#chartContainer").CanvasJSChart(options);
}
</script>
<div id="chartContainer" style="height: 370px; width: 500px;"></div>

    <script type="text/javascript"
src="https://canvasjs.com/assets/script/jquery.canvasjs.min.js"></script>

    <?php endif;?>

    <?php else:echo "<p><h2 style='color:red;text-align:center;'>Error 403. You cannot
access this page.</h2></p>";header('Location:index.php?page=home');
    endif;
?>

```

○ config.php

```

<?php

define("MYSQL_HOST", "localhost");
define("MYSQL_DBNAME", "blend");
define("MYSQL_USERNAME", "root");
define("MYSQL_PASSWORD", "");

```

○ connection.php

```

<?php

```

```

require_once "config.php";

try {

    $konekcija = new PDO("mysql:host=".MYSQL_HOST.";dbname=".MYSQL_DBNAME.";charset=utf8",
MYSQL_USERNAME, MYSQL_PASSWORD);

    $konekcija->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $konekcija->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);

}
catch(PDOException $e){
    echo $e->getMessage();
}

function executeQuery($upit){
    global $konekcija;

    $rezultat = $konekcija->query($upit)->fetchAll();

    return $rezultat;
}

```

○ contact.php

```

<?php if(isset($_SESSION['user'])):?>

<?php
    if(isset($_POST['btnContact'])){
        $errors=[];
        $subject=$_POST['subject'];
        $text=$_POST['textArea'];
        $emailUser=$_SESSION['user']->email;

        $reSubject="/^[\\w\\s0-9\\-\\_\\@\\!]{1,255}$/";

        if(!preg_match($reSubject,$subject)){ $errors[]="Subject must contain something.";}
        if($text==null){ $errors[]="The message cannot be empty.";}

        $text=$text." From:$emailUser";

        //echo $subject."<br/>".$text."<br/>".$emailUser."<br/>";

        if(count($errors)>0){

```

```

        echo"<ul style='margin:auto;color:red;text-align:center;'>";
        foreach($errors as $error){
            echo "<li style='color:red;'><h4>".$error."</h4></li>";
        }
        echo"</ul>";

    }

    else{

        $sendMail=mail("david.carevic.159.14@ict.edu.rs",$subject,$text,"From:
coffeestoreblend.000webhostapp.com");

        if($sendMail){
            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Your
email has been send!</h4></p>";
        }

    }

}

?>
<div class="forme">

<form action="<?php echo $_SERVER['PHP_SELF'].'?page=contact';?>" method="POST"
name="forma">
    <table>
        <th colspan="1"><h2>Contact</h2></th>

        <tr>
            <td><h4>Subject:</h4>
            <input type="text" name="subject"/></td>

        </tr>
        <tr>
            <td><h4>Message:</h4>
            <textarea name="textArea" cols="40" rows="10"></textarea></td>

        </tr>

        <tr>

            <td colspan="1"><input class="Btn" type="submit" value="Send"
name="btnContact"/></td>
        </tr>
    </table>
</form>
</div>
<?php else:echo "<p><h2 style='color:red;text-align:center;'>Error 403. You cannot access
this page.</h2></p>";header("Location:index.php?page=home");
endif;

```

?>

- footer.php

```
</div>
<div id="footer"></div>
</body>
</html>
```

- funkcije.php (preuzeto sa lab vežbi)

```
<?php

function malaSlika($putanja_velika, $putanja_mala, $nova_sirina, $nova_visina){

    // primer: image/slika1.jpg

    $podaciSlika=explode('.', $putanja_velika);

    $ekstenzija = $podaciSlika[1]; // "jpg" itd

    // Provera da li je ekstenzija jpg/jpeg
    // "/i" - ne gleda da li je veliko/malo slovo - case INsensitive

    if (preg_match('/jpg|jpeg/i', $ekstenzija)){

        // "imagecreatefromjpeg" - ugradjena PHP funkcija, kreira NOVU SLIKU od originalne
        // koja se nalazi na "putanja_velika"
        // Nova slika je za sada ista kao originalna

        $nova_slika = imagecreatefromjpeg($putanja_velika);
    }

    // Provera da li je ekstenzija png
    if (preg_match('/png/i', $ekstenzija)){
        $nova_slika = imagecreatefrompng($putanja_velika);
    }

    // Preuzimanje sirine i visine nove slike - isto kao na originalnoj

    $sirina_original = imageSX($nova_slika);
    $visina_original = imageSY($nova_slika);

    // Skaliranje slike

    if ($sirina_original > $visina_original) {
```



```

        $tmp_sirina = $nova_sirina;
        $tmp_visina = $visina_original * ($nova_visina / $visina_original);
        // $nova_visina/$visina - vraća procentualno razliku nove i stare visine - za
toliko smanjujemo sliku
    }
    if ($sirina_original < $visina_original) {
        $tmp_sirina = $sirina_original * ($nova_sirina / $sirina_original);
        $tmp_visina = $nova_visina;
    }
    if ($sirina_original == $visina_original) {
        $tmp_sirina=$nova_sirina;
        $tmp_visina=$nova_visina;
    }

    // Slika koja se dobija kao rezultat metode
    $izlazna_slika = imagecreatetruecolor($tmp_sirina, $tmp_visina);

    imagecopyresampled($izlazna_slika, $nova_slika,
                        0, 0, 0, 0,
                        $tmp_sirina, $tmp_visina,
                        $sirina_original, $visina_original);

    if (preg_match("/png/i",$ekstenzija)){
        imagepng($izlazna_slika, $putanja_mala);
    } else {
        imagejpeg($izlazna_slika, $putanja_mala);
    }
    imagedestroy($izlazna_slika);
    imagedestroy($nova_slika);
}

```

- header.php

```

<?php

if(isset($_POST['btnLogIn'])){

    $username=trim($_POST['username']);
    $password=$_POST['password'];
    $password=md5($password);

    $upit="SELECT *,u.id AS idUser FROM user u INNER JOIN role r ON u.idRole=r.id WHERE
username=:username AND password=:password";

```

```

$priprema=$konekcija->prepare($upit);
$priprema->bindParam(":username",$username);
$priprema->bindParam(":password",$password);

$rez=$priprema->execute();
$red=$priprema->fetch();

if($red){

    $_SESSION['user']=$red;
    //var_dump($_SESSION['user']);

}
else{
    echo"<script>
        alert('Wrong username or password!');

    </script>";
}

}
?>
<!DOCTYPE html>
<html>

<head>
<title>Coffee Store Blend</title>
    <meta charset='UTF-8' />
    <meta name="description" content="Coffee Store Blend website dedicated to all lovers
of coffee." />
<meta name="keyword" content="coffee,store,blend"/>
<meta name="viewport" content="width=device-width">
    <link rel="stylesheet" type="text/css" href="stil.css"/>
    <link href="https://fonts.googleapis.com/css?family=Noto+Sans" rel="stylesheet">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <link href="lightbox/src/css/lightbox.css" rel="stylesheet">
    <script src="lightbox/src/js/lightbox.js"></script>

```

```

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script type="text/javascript" src="efekti.js"></script>

<link rel="shortcut icon" href="slike/coffeeicon.png" type="image/x-icon"/>

<!-- fotorama -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/fotorama/4.6.4/fotorama.css"
rel="stylesheet">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/fotorama/4.6.4/fotorama.js"></script>

<!-- fotorama -->

</head>
<body>
<div id="header">
<div id="logo">
<a href="index.php">
<h1>Coffee Store Blend<h1>
</a>

</div>
<div id="nav">
<ul>
<?php
    $upit=executeQuery('SELECT * FROM nav WHERE type=1');
    foreach ($upit as $red){
        echo "<li><a href=\"".$red->href.">".$red->name."</a></li>";
    }

    if(isset($_SESSION['user'])&& $_SESSION['user']->role=='admin'){
        $upit=executeQuery('SELECT * FROM nav WHERE type=2 OR type=5');
        foreach($upit as $red){
            echo "<li><a href=\"".$red->href.">".$red->name."</a></li>";
        }
    }
    elseif(isset($_SESSION['user'])&& $_SESSION['user']->role=='user'){
        $upit=executeQuery('SELECT * FROM nav WHERE type=2 OR type=4');
        foreach($upit as $red){
            if($red->name=='User'){echo "<li><a href=\"".$_SESSION['user']->username."</a></li>";}
            else{
                echo "<li><a href=\"".$red->href.">".$red->name."</a></li>";
            }
        }
    }
    }
    else{

```

```

        $upit=executeQuery('SELECT * FROM nav WHERE type=3');
        foreach($upit as $red){
            if($red->name=="Log In"){echo "<li id='showSrc'><a href=\".$red->href.\">\".$red->name.\"</a></li>";}
            else{
                echo "<li><a href=\".$red->href.\">\".$red->name.\"</a></li>";
            }
        }
    }
    //var_dump($upit);

?>

</ul>

<div class="dropdown">
    <form action="<?php echo $_SERVER['PHP_SELF'];?>" method="POST">

        <input type="text" placeholder="Username" name="username"
id="userLog"/><input type="password" name="password" placeholder="Password"
id="passLog"/><input type="submit" value="Log In" class='Btn' name="btnLogIn"
id="searchBtn"/>

    </form>
</div>
</div>
</div>
<div id="center">

```

- home.php

```
<div id="home">
<p><h2>Welcome to the website of Coffee Shop Blend</h2></p><br/>
<?php

    $upit=executeQuery("SELECT * FROM home_text");

    foreach($upit as $red){

        echo "<p><h3>".$red->text."</h3></p>";
    }

?>
<br/>
<div class="fotorama">
    <?php
        $upit=executeQuery("SELECT * FROM home_image");
        foreach($upit as $red){

            echo "alt."/>";
        }
    ?>
</div>

</div>
```

- index.php

```
<?php
session_start();
include("connection.php");
include "funkcije.php";
include("header.php");

$page="";

if(isset($_GET['page'])){
    $page = $_GET['page'];
}

switch($page){
```

```

    case "reg":
        include "reg.php";
        break;
    case "author":
        include "author.php";
        break;
    case "contact":
        include "contact.php";
        break;
    case "about":
        include "about.php";
        break;
    case "shop":
        include "shop.php";
        break;

    case "user":
        include "user.php";
        break;
    case "admin":
        include "admin.php";
        break;
    default:
        include "home.php";
        break;
}

include("footer.php");

?>

```

○ logout.php

```

<?php
session_start();
unset($_SESSION['user']);
header('Location:index.php');

?>

```

○ reg.php

```

<?php

    if(isset($_POST['btnSubmit'])){
        //echo "asdasdasd";
    }

```

```

// var_dump($_POST);

$user=trim($_POST['user']);
//var_dump($user);
$pass1=$_POST['pass1'];

$email=trim($_POST['email']);

$name=trim($_POST['name']);
$lastname=trim($_POST['lastname']);

$errors=[];

$reUser="/^[A-Za-z0-9]{5,20}$/";
if(!preg_match($reUser,$user)){

    $errors[]="Username is not good.";
}
$rePass1="/^[A-Za-z0-9]{5,15}$/";
if(!preg_match($rePass1,$pass1)){
    $errors[]="Pass is not good.";
}

if(!filter_var($email, FILTER_VALIDATE_EMAIL)){

    $errors[]="Email is not good.";

}

$reName="/^([A-Z]{1}[a-z]{2,19})(\s[A-Z]{1}[a-z]{2,19})*$/";
if(!preg_match($reName,$name)){

    $errors[]="Name is not good.";
}

$reLastName="/(^([A-Z]{1}[a-z]{4,39})(\s[A-Z]{1}[a-z]{4,39})*$/";
if(!preg_match($reLastName,$lastname)){

    $errors[]="Lastname is not good.";
}
$rezUser=$koneksi->prepare("SELECT username FROM user WHERE
username=:user");
$rezUser->bindParam(':user',$user);
$rezUser->execute();
$rezU=$rezUser->fetch();
if($rezU){

```

```

        $errors[]= "Username is taken !";
    }
    $rezEmail=$konekcija->prepare("SELECT email FROM user WHERE
email=:email");
    $rezEmail->bindParam(':email',$email);
    $rezEmail->execute();
    $rezE=$rezEmail->fetch();
    if($rezE){
        $errors[]= "Email is taken !";
    }

}

if(count($errors)>0){

    echo"<ul style='margin:auto;color:red;text-align:center;'>";
    foreach($errors as $error){
        echo "<li style='color:red;'><h4>".$error."</h4></li>";
    }
    echo"</ul>";

}

else{

    $pass1=md5($pass1);

    $roleid=2;

    $upit="INSERT INTO user
VALUES(NULL,:user,:pass1,:email,:name,:lastname,:roleid)";

    $priprema_upit=$konekcija->prepare($upit);
    $priprema_upit->bindParam(":user",$user);
    $priprema_upit->bindParam(":pass1",$pass1);
    $priprema_upit->bindParam(":email",$email);
    $priprema_upit->bindParam(":name",$name);

```



```

        $priprema_upit->bindParam(":lastname",$lastname);
        $priprema_upit->bindParam(":roleid",$roleid);
        try{
            $rezultat= $priprema_upit->execute();
        }
        catch(PDOException $ex){
            echo $ex->getMessage();
        }

        if($rezultat){

            echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Your
account has been created !</h4></p>";

        }
        else{ echo "Error with the database.";}

    }

}

?>
<div class="forme">
    <form action="<?php echo $_SERVER['PHP_SELF'].'?page=reg';?>" method="POST"
name="newAcc">
    <table id="reg">
        <th colspan="2"><h2>New Account</h2></th>

        <tr>

            <td><h4>Username:</h4>
                <input class="spi" name="user" type="text" id="user"/>
            </td>

        </tr>

        <tr id="hiddenuser" style="display:none"><td colspan="2"><h4>Username must contain
5-20 characters, no special characters !</h4></td></tr>
        <tr>
            <td><h4>Password:</h4>
                <input class="spi" name="pass1" type="password" id="pass1"/></td>

        </tr>

```

```

        <tr id="hiddenpass1" style="display:none"><td colspan="2"><h4>Password must contain
5-15 characters, no special characters !</h4></td></tr>

        <tr>
            <td><h4>E-mail:</h4>
                <input class="spi" name="email" type="text" id="email"/></td>

        </tr>
        <tr id="hiddenemail" style="display:none"><td colspan="2"><h4>E-mail must contain "@"
and at least 1 "." !</h4></td></tr>
        <th colspan="2"><h2>Information</h2></th>

        <tr>
            <td><h4>First name:</h4>
                <input class="spi" name="name" type="text" id="name"/></td>

        </tr>
        <tr id="hiddenname" style="display:none"><td colspan="2"><h4>Name must contain 3-20
characters, the first letter must be capital, you can type in multiple names
!</h4></td></tr>
        <tr>
            <td><h4>Last name:</h4>
                <input class="spi" name="lastname" type="text" id="lastname"/></td>

        </tr>
        <tr id="hiddenlastname" style="display:none"><td colspan="2"><h4>Lastname must
contain 5-40 characters, the first letter must be capital, you can type in mutiple
lastnames !</h4></td></tr>

        <tr>
            <td colspan="2"><input class="Btn" type="submit" id="btnSubmit" name="btnSubmit"
value="Create your account"/></td>

        </tr>

    </table>
</form>

</div>

```

- shop.php

```
<?php if(isset($_SESSION['user'])){?>
<div id='shop-holder'>
<?php
$rez=executeQuery("SELECT * FROM shop;");

//var_dump($rez);

foreach($rez as $red):
?>

<div class="item">

    <div class="shop-slika"><a href="<?php echo $red->src;?>" data-lightbox="shop">title;?>" /></a></div>

    <table>
    <th colspan="2"><h2><?php echo $red->title;?></h2></th>

    <tr>
    <td><h3>Price:</h3> </td>
    <td><h2>$<?php echo $red->price?></h2></td>

</tr>

    <tr>

    <td><h3>Description: </h3></td>

    <td><h4 style><?php echo $red->description?></h4></td>

</tr>

    <tr>
    <td colspan="2"> <input type="button" class="Btn" value=" Add to cart " /> <input
type="hidden" value="<?php echo $red->id?>" /> </td>
    </tr>
</table>

</div>

<?php endforeach;?>
```

```

</div>

<?php
}else{
    echo "<p><h2 style='color:red;text-align:center;'>Error 403. You cannot access this
page.</h2></p>";
    header("Location:index.php?page=home");
}

?>

```

○ user.php

```

<?php
    if(isset($_SESSION['user'])&& $_SESSION['user']->role=='user'):
?>

<h2 style="margin:10px auto;text-align:center;">Hello <?php echo $_SESSION['user']-
>username;?>, here you can update your data.</h2>

<?php

    if(isset($_POST['update']))){

        $errors=[];

        $pass1=$_POST['pass1'];
        $email=$_POST['email'];
        $id=$_SESSION['user']->idUser;

        $rePass1="/^[A-Za-z0-9]{5,15}$/";
        if(!preg_match($rePass1,$pass1)){
            $errors[]="Pass is not good.";
        }

        if(!filter_var($email, FILTER_VALIDATE_EMAIL)){

            $errors[]="Email is not good.";

        }

        $upit=$konekcija->prepare("SELECT * FROM user WHERE email=:email");
        $upit->bindParam(':email',$email);

```

```

$upit->execute();
$rez=$upit->fetch();
//echo $rez->email;
if($rez && $rez->email!=$_SESSION['user']->email){
    $errors[]="The email you entered is taken.";
}

if(count($errors)>0){

    echo"<ul style='margin:0 auto;color:red;text-align:center;'>";
foreach($errors as $error){
    echo "<li style='color:red;list-style:none'><h4>".$error."</h4></li>";
}
echo"</ul>";

}

else{

$pass1=md5($pass1);

$upit="UPDATE user SET password=:pass1,email=:email WHERE id=:id";

$priprema_upit=$konekcija->prepare($upit);

$priprema_upit->bindParam(":pass1",$pass1);
$priprema_upit->bindParam(":email",$email);
$priprema_upit->bindParam(":id",$id);

    try{
$rezultat= $priprema_upit->execute();
    }
catch(PDOException $ex){
echo $ex->getMessage();
}

if($rezultat){

```

```

echo"<p><h4 style='margin:0auto;text-align:center;color:green'>Your account has been
updated !</h4></p>";
$_SESSION['user']->email=$email;

}
else{ echo "Error with the database.;"}}

}

}
?>

<div class="forme">
  <form name="forma" method="POST" action="<?php echo $_SERVER['PHP_SELF'].'?page=user'?>">

    <table>

      <tr>
        <td><h4>New password:</h4> <input name="pass1" class="spi" id="pass1"
type="password"/><td></tr>
      </tr>
      <tr id="hiddenpass1" style="display:none"><td colspan="2"><h4>Password must contain
5-15 characters, no special characters !</h4></td></tr>
      <tr>
        <td><h4>New Email:</h4>
        <input type="email" name="email" class="spi" id="email" value="<?php echo
$_SESSION['user']->email;?>"/><td>
      </tr>
      <tr id="hiddenemail" style="display:none"><td colspan="2"><h4>E-mail must contain "@"
and at least 1 "." !</h4></td></tr>

      <tr>
        <td><input type="submit" class="Btn" name="update" value="Update your account"><td>
      </tr>
    </table>

  </form>
</div>

```

```
<?php else: echo "<p><h2 style='color:red;text-align:center;'>Error 403. You cannot access
this page.</h2></p>";
header('Location:index.php');
endif;
?>
```

- CSS

- stil.css

```
*{font-family: 'Noto Sans', sans-serif;}
*{margin:0;padding:0;}
#header{display:block;}

#logo{display:block;text-align:center;font-size:40px}

#logo img{width:200px;height:200px;display:inline-block;}
#logo h1{margin:0px;}
#logo a{text-decoration:none;color:#222;}

#nav{display:block;text-align:center;}
#nav ul{display:inline-block;text-align:center;}
#nav li{font-size:40px;list-style:none;;float:left;}
#nav a{text-decoration:none;color:#aaa;padding:10px;}

#nav input{border:2px solid #222;margin:10px;font-size:20px;padding:5px;background-
color:#fff;color:#aaa;}
.Btn{border:2px solid #222;margin:10px;font-size:20px;padding:5px;background-
color:#fff;color:#aaa;}

#showSrc{text-align:left;}

.dropdown{display:none;}

#center{background-color:white;margin:20px auto;display:block;min-height:500px;}
#footer{display:block;height:100px;background-color:#222;margin-top:10px;}
```

```

#center table{margin:20px auto;text-align:center;}
#center img:hover{opacity:0.8;}

.about-holder{display:block;width:1000px;margin:0 auto;}
.about-image{display:inline-block;height:300px;width:300px;border:1px solid
#000;margin:10px;}
.about-image img{width:300px;height:300px;}
.about-text{display:inline-block;width:600px;word-wrap:break-word;}

#author-image{border:1px solid #222;display:block;margin:10px
auto;height:200px;width:200px;}
#author-image img{height:200px;width:199px;}
#author{display:block;width:600px;margin:10px auto;}
#author h4{width: 600px;word-wrap:break-word;text-align:center;}

.forme{margin:20px;}
.forme h4{word-wrap:break-word;width:400px;text-align:center;}

.forme input{border:2px solid #222;}

.item{display:inline-block;margin:10px;border:2px solid #000;padding:10px;text-
align:center;height:570px;}
.item h4{display:inline-block;word-wrap:break-word;width:400px;height:140px;}
.shop-slika{display:inline-block;margin:0 auto;}
.shop-slika img{border:1px solid #222;min-height:250px;}
.item table{text-align:left;}
#shop-holder{margin:auto;text-align:center;}

#home{margin:10px auto;text-align: center;}
.fotorama{width:1280px;height:720px;margin:0 auto;}

#chartContainer{margin:0 auto;}

.adminSelect{margin:10px auto; border:2px solid #222;text-align:center;display:inline-
block;float:left;}
#adminNav{margin:10px auto; border:2px solid #222;text-align:center;display:inline-
block;float:left;}
#adminNav li{list-style:none;}
#adminFormSelect{display:inline-block;}
#desni{float:right;display:inline;width:1000px;text-align:center;}
#desni table{float:left;}
#levi{float:left;display:block;width:130px;}

```



```
#omotAdmin{display:block;width:1300px;margin:0 auto;}

.cistac{clear:both;}
```

- lightbox.css(preuzeto sa interneta)

```
html.lb-disable-scrolling {
  overflow: hidden;
  /* Position fixed required for iOS. Just putting overflow: hidden; on the body is not
  enough. */
  position: fixed;
  height: 100vh;
  width: 100vw;
}

.lightboxOverlay {
  position: absolute;
  top: 0;
  left: 0;
  z-index: 9999;
  background-color: black;
  filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=80);
  opacity: 0.8;
  display: none;
}

.lightbox {
  position: absolute;
  left: 0;
  width: 100%;
  z-index: 10000;
  text-align: center;
  line-height: 0;
  font-weight: normal;
}

.lightbox .lb-image {
  display: block;
  height: auto;
  max-width: inherit;
```

```

max-height: none;
border-radius: 3px;

/* Image border */
border: 4px solid white;
}

.lightbox a img {
border: none;
}

.lb-outerContainer {
position: relative;
*zoom: 1;
width: 250px;
height: 250px;
margin: 0 auto;
border-radius: 4px;

/* Background color behind image.
   This is visible during transitions. */
background-color: white;
}

.lb-outerContainer:after {
content: "";
display: table;
clear: both;
}

.lb-loader {
position: absolute;
top: 43%;
left: 0;
height: 25%;
width: 100%;
text-align: center;
line-height: 0;
}

.lb-cancel {
display: block;
width: 32px;
height: 32px;
margin: 0 auto;
background: url(../images/loading.gif) no-repeat;
}

.lb-nav {
position: absolute;
top: 0;
left: 0;
height: 100%;

```

```

width: 100%;
z-index: 10;
}

.lb-container > .nav {
  left: 0;
}

.lb-nav a {
  outline: none;
  background-image:
url('data:image/gif;base64,R0lGODlhAQABAPAAAP///wAAACH5BAEAAAAALAAAAABAAEAAAIICRAEAOw==');
}

.lb-prev, .lb-next {
  height: 100%;
  cursor: pointer;
  display: block;
}

.lb-nav a.lb-prev {
  width: 34%;
  left: 0;
  float: left;
  background: url(../images/prev.png) left 48% no-repeat;
  filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=0);
  opacity: 0;
  -webkit-transition: opacity 0.6s;
  -moz-transition: opacity 0.6s;
  -o-transition: opacity 0.6s;
  transition: opacity 0.6s;
}

.lb-nav a.lb-prev:hover {
  filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=100);
  opacity: 1;
}

.lb-nav a.lb-next {
  width: 64%;
  right: 0;
  float: right;
  background: url(../images/next.png) right 48% no-repeat;
  filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=0);
  opacity: 0;
  -webkit-transition: opacity 0.6s;
  -moz-transition: opacity 0.6s;
  -o-transition: opacity 0.6s;
  transition: opacity 0.6s;
}

.lb-nav a.lb-next:hover {
  filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=100);

```

```

    opacity: 1;
}

.lb-dataContainer {
    margin: 0 auto;
    padding-top: 5px;
    *zoom: 1;
    width: 100%;
    border-bottom-left-radius: 4px;
    border-bottom-right-radius: 4px;
}

.lb-dataContainer:after {
    content: "";
    display: table;
    clear: both;
}

.lb-data {
    padding: 0 4px;
    color: #ccc;
}

.lb-data .lb-details {
    width: 85%;
    float: left;
    text-align: left;
    line-height: 1.1em;
}

.lb-data .lb-caption {
    font-size: 13px;
    font-weight: bold;
    line-height: 1em;
}

.lb-data .lb-caption a {
    color: #4ae;
}

.lb-data .lb-number {
    display: block;
    clear: left;
    padding-bottom: 1em;
    font-size: 12px;
    color: #999999;
}

.lb-data .lb-close {
    display: block;
    float: right;
    width: 30px;
    height: 30px;

```

```

background: url(../images/close.png) top right no-repeat;
text-align: right;
outline: none;
filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=70);
opacity: 0.7;
-webkit-transition: opacity 0.2s;
-moz-transition: opacity 0.2s;
-o-transition: opacity 0.2s;
transition: opacity 0.2s;
}

.lb-data .lb-close:hover {
  cursor: pointer;
  filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=100);
  opacity: 1;
}

```

- JavaScript(jQuery)

- efekti.js

```

$(document).ready(function(){

    //menu color hover animation //////////////////////////////////
    $("#nav li a,.Btn").hover(function(){
        $(this).stop(true,true).animate({
            backgroundColor: "#222",
            color: "white",

        }, 350);
        //console.log("hover");

    },function(){
        $(this).stop(true,true).animate({
            backgroundColor: "#fff",
            color: "#aaa",

        }, 350);

    });

    // $(".dropdown").hide();

```

```

//menu color hover animation //////////////////////////////////

//search //////////////////////////////////
var counter=0;
$("#showSrc a").click(function(){
    if(counter%2==0){

        $(".dropdown").fadeIn();

    }
    else{$(".dropdown").fadeOut();}
    counter++;
});
//search //////////////////////////////////

//reg check //////////////////////////////////

var reNiz={user:/^[A-Za-z0-9]{5,20}$/ ,
pass1:/^[A-Za-z0-9]{5,15}$/ ,
email:/^\S+@\w+\.\w{2,3}$/ ,
name:/^(^[A-Z]{1}[a-z]{2,19}) (\s[A-Z]{1}[a-z]{2,19}) *$/ ,
lastname:/^(^[A-Z]{1}[a-z]{4,39}) (\s[A-Z]{1}[a-z]{4,39}) *$/};

var podaci={
user:"",
pass1:"",
email:"",
name:"",
lastname:""};

$(".spi").focus(function(){

    $("#hidden"+this.id).fadeIn();
    $(this).css('border-color','red');

```

```

$(this).blur(function(){

    var regHolder=reNiz[this.id];
    console.log(regHolder);
    if($(this).val().match(regHolder)){
        podaci[this.id]=$ (this).val();
        $(this).css('border-color','green');
        $("#hidden"+this.id).fadeOut();

    }

    else{podaci[this.id]="";$("#btnSubmit").css("border","2px solid red");}
    for(var podatak in podaci){
        if(podaci[podatak]==""){$("#btnSubmit").css("border","2px solid red");}
        else{$("#btnSubmit").css("border","2px solid #222");}
    }
});

});

```

```

$("#btnSubmit").click(function(e){

    console.log(podaci);
    var greske=0;

    for(podatak in podaci){
        if(podaci[podatak]=="") greske++;
    }

    console.log(greske);

```

```

    if(greske>0){
        e.preventDefault();
        console.log("nije poslao");
        document.querySelector("#btnSubmit").style="border:2px solid red";
    }

    else{

        console.log("poslao je");

    }

});

// reg check //////////////////////////////////////

// poll //////////////////////////////////////

$('#btnVote').click(function(){

    var id=document.querySelector('#userId').value;
    console.log("id korisnika= "+id);

    var rbVal="";
    var rb=document.getElementsByName('rbPoll');

    for(var i=0;i<rb.length;i++){

        if(rb[i].checked){
            rbVal=rb[i].value;
            break;
        }
    }

```



```

    }
    console.log("vrednost glasa= "+rbVal);

    $.ajax({

        method:'POST',
        url:'ajax_insert_poll.php',
        dataType:'json',
        data:{
            id:id,vote:rbVal
        },
        success:function(status){console.log(status);location.reload();},
        error: function(xhr,statusText,error){
            var status=xhr.status;
            switch(status){
                case 500:
                    alert("Greska na serveru. Trenutno nije moguće promeniti korisnika.");
                    break;
                case 404:
                    alert("Stranica nije pronadjena.");
                    break;
                default:
                    alert("Greska: " + status + " - " + statusTxt);
                    break;
            }
        }
    });

});

```

```
// poll //////////////////////////////////////
```

```
});
```

- lightbox.js(preuzeto sa interneta)

```
/*!
 * Lightbox v2.10.0
 * by Lokesh Dhakar
 *
 * More info:
 * http://lokeshdhakar.com/projects/lightbox2/
 *
 * Copyright 2007, 2018 Lokesh Dhakar
 * Released under the MIT license
 * https://github.com/lokesh/lightbox2/blob/master/LICENSE
 *
 * @preserve
 */

// Uses Node, AMD or browser globals to create a module.
(function (root, factory) {
  if (typeof define === 'function' && define.amd) {
    // AMD. Register as an anonymous module.
    define(['jquery'], factory);
  } else if (typeof exports === 'object') {
    // Node. Does not work with strict CommonJS, but
    // only CommonJS-like environments that support module.exports,
```

```

    // like Node.
    module.exports = factory(require('jquery'));
} else {
    // Browser globals (root is window)
    root.lightbox = factory(root.jQuery);
}
})(this, function ($) {

function Lightbox(options) {
    this.album = [];
    this.currentImageIndex = void 0;
    this.init();

    // options
    this.options = $.extend({}, this.constructor.defaults);
    this.option(options);
}

// Descriptions of all options available on the demo site:
// http://lokeshdhakar.com/projects/lightbox2/index.html#options
Lightbox.defaults = {
    albumLabel: 'Image %1 of %2',
    alwaysShowNavOnTouchDevices: false,
    fadeDuration: 600,
    fitImagesInViewport: true,
    imageFadeDuration: 600,
    // maxWidth: 800,
    // maxHeight: 600,
    positionFromTop: 50,
    resizeDuration: 700,
    showImageNumberLabel: true,
    wrapAround: false,
    disableScrolling: false,
    /*
    Sanitize Title
    If the caption data is trusted, for example you are hardcoding it in, then leave this
to false.
    This will free you to add html tags, such as links, in the caption.

    If the caption data is user submitted or from some other untrusted source, then set
this to true
    to prevent xss and other injection attacks.
    */
    sanitizeTitle: false
};

Lightbox.prototype.option = function(options) {
    $.extend(this.options, options);
};

Lightbox.prototype.imageCountLabel = function(currentImageNum, totalImages) {
    return this.options.albumLabel.replace(/%1/g, currentImageNum).replace(/%2/g,
totalImages);

```

```

};

Lightbox.prototype.init = function() {
    var self = this;
    // Both enable and build methods require the body tag to be in the DOM.
    $(document).ready(function() {
        self.enable();
        self.build();
    });
};

// Loop through anchors and areamaps looking for either data-lightbox attributes or rel
// attributes
// that contain 'lightbox'. When these are clicked, start lightbox.
Lightbox.prototype.enable = function() {
    var self = this;
    $('body').on('click', 'a[rel^=lightbox], area[rel^=lightbox], a[data-lightbox],
area[data-lightbox]', function(event) {
        self.start($(event.currentTarget));
        return false;
    });
};

// Build html for the lightbox and the overlay.
// Attach event handlers to the new DOM elements. click click click
Lightbox.prototype.build = function() {
    if ($('#lightbox').length > 0) {
        return;
    }

    var self = this;
    $('<div id="lightboxOverlay" class="lightboxOverlay"></div><div id="lightbox"
class="lightbox"><div class="lb-outerContainer"><div class="lb-container"><div class="lb-nav"><a class="lb-prev" href="" ></a><a class="lb-next" href=""
></a></div><div class="lb-loader"><a class="lb-cancel"></a></div></div></div><div
class="lb-dataContainer"><div class="lb-data"><div class="lb-details"><span class="lb-
caption"></span><span class="lb-number"></span></div><div class="lb-closeContainer"><a
class="lb-close"></a></div></div></div></div>').appendTo($('body'));

    // Cache jQuery objects
    this.$lightbox = $('#lightbox');
    this.$overlay = $('#lightboxOverlay');
    this.$outerContainer = this.$lightbox.find('.lb-outerContainer');
    this.$container = this.$lightbox.find('.lb-container');
    this.$image = this.$lightbox.find('.lb-image');
    this.$nav = this.$lightbox.find('.lb-nav');

    // Store css values for future lookup
    this.containerPadding = {
        top: parseInt(this.$container.css('padding-top'), 10),
        right: parseInt(this.$container.css('padding-right'), 10),

```

```

        bottom: parseInt(this.$container.css('padding-bottom'), 10),
        left: parseInt(this.$container.css('padding-left'), 10)
    });

    this.imageBorderWidth = {
        top: parseInt(this.$image.css('border-top-width'), 10),
        right: parseInt(this.$image.css('border-right-width'), 10),
        bottom: parseInt(this.$image.css('border-bottom-width'), 10),
        left: parseInt(this.$image.css('border-left-width'), 10)
    };

    // Attach event handlers to the newly minted DOM elements
    this.$overlay.hide().on('click', function() {
        self.end();
        return false;
    });

    this.$lightbox.hide().on('click', function(event) {
        if ($(event.target).attr('id') === 'lightbox') {
            self.end();
        }
        return false;
    });

    this.$outerContainer.on('click', function(event) {
        if ($(event.target).attr('id') === 'lightbox') {
            self.end();
        }
        return false;
    });

    this.$lightbox.find('.lb-prev').on('click', function() {
        if (self.currentImageIndex === 0) {
            self.changeImage(self.album.length - 1);
        } else {
            self.changeImage(self.currentImageIndex - 1);
        }
        return false;
    });

    this.$lightbox.find('.lb-next').on('click', function() {
        if (self.currentImageIndex === self.album.length - 1) {
            self.changeImage(0);
        } else {
            self.changeImage(self.currentImageIndex + 1);
        }
        return false;
    });

    /*
    Show context menu for image on right-click

```

There is a div containing the navigation that spans the entire image and lives above of it. If

you right-click, you are right clicking this div and not the image. This prevents users from saving the image or using other context menu actions with the image.

To fix this, when we detect the right mouse button is pressed down, but not yet clicked, we

set pointer-events to none on the nav div. This is so that the upcoming right-click event on

the next mouseup will bubble down to the image. Once the right-click/contextmenu event occurs

we set the pointer events back to auto for the nav div so it can capture hover and left-click

events as usual.

*/

```
this.$nav.on('mousedown', function(event) {  
  if (event.which === 3) {  
    self.$nav.css('pointer-events', 'none');  
  
    self.$lightbox.one('contextmenu', function() {  
      setTimeout(function() {  
        this.$nav.css('pointer-events', 'auto');  
      }.bind(self), 0);  
    });  
  }  
});
```

```
this.$lightbox.find('.lb-loader, .lb-close').on('click', function() {  
  self.end();  
  return false;  
});  
};
```

// Show overlay and lightbox. If the image is part of a set, add siblings to album array.

```
Lightbox.prototype.start = function($link) {  
  var self = this;  
  var $window = $(window);  
  
  $window.on('resize', $.proxy(this.sizeOverlay, this));  
  
  $('select, object, embed').css({  
    visibility: 'hidden'  
  });  
  
  this.sizeOverlay();  
  
  this.album = [];  
  var imageNumber = 0;  
  
  function addToAlbum($link) {
```

```

        self.album.push({
            alt: $link.attr('data-alt'),
            link: $link.attr('href'),
            title: $link.attr('data-title') || $link.attr('title')
        });
    }

    // Support both data-lightbox attribute and rel attribute implementations
    var dataLightboxValue = $link.attr('data-lightbox');
    var $links;

    if (dataLightboxValue) {
        $links = $($link.prop('tagName') + '[data-lightbox="' + dataLightboxValue + '"']');
        for (var i = 0; i < $links.length; i = ++i) {
            addToAlbum($($links[i]));
            if ($links[i] === $link[0]) {
                imageNumber = i;
            }
        }
    } else {
        if ($link.attr('rel') === 'lightbox') {
            // If image is not part of a set
            addToAlbum($link);
        } else {
            // If image is part of a set
            $links = $($link.prop('tagName') + '[rel="' + $link.attr('rel') + '"']');
            for (var j = 0; j < $links.length; j = ++j) {
                addToAlbum($($links[j]));
                if ($links[j] === $link[0]) {
                    imageNumber = j;
                }
            }
        }
    }
}

// Position Lightbox
var top = $window.scrollTop() + this.options.positionFromTop;
var left = $window.scrollLeft();
this.$lightbox.css({
    top: top + 'px',
    left: left + 'px'
}).fadeIn(this.options.fadeDuration);

// Disable scrolling of the page while open
if (this.options.disableScrolling) {
    $('html').addClass('lb-disable-scrolling');
}

this.changeImage(imageNumber);
};

// Hide most UI elements in preparation for the animated resizing of the lightbox.
Lightbox.prototype.changeImage = function(imageNumber) {

```

```

var self = this;

this.disableKeyboardNav();
var $image = this.$lightbox.find('.lb-image');

this.$overlay.fadeIn(this.options.fadeDuration);

$('.lb-loader').fadeIn('slow');
this.$lightbox.find('.lb-image, .lb-nav, .lb-prev, .lb-next, .lb-dataContainer, .lb-
numbers, .lb-caption').hide();

this.$outerContainer.addClass('animating');

// When image to show is preloaded, we send the width and height to sizeContainer()
var preloader = new Image();
preloader.onload = function() {
    var $preloader;
    var imageHeight;
    var imageWidth;
    var maxImageHeight;
    var maxImageWidth;
    var windowHeight;
    var windowWidth;

    $image.attr({
        'alt': self.album[imageNumber].alt,
        'src': self.album[imageNumber].link
    });

    $preloader = $(preloader);

    $image.width(preloader.width);
    $image.height(preloader.height);

    if (self.options.fitImagesInViewport) {
        // Fit image inside the viewport.
        // Take into account the border around the image and an additional 10px gutter on
each side.

        windowWidth    = $(window).width();
        windowHeight   = $(window).height();
        maxImageWidth  = windowWidth - self.containerPadding.left -
self.containerPadding.right - self.imageBorderWidth.left - self.imageBorderWidth.right -
20;
        maxImageHeight = windowHeight - self.containerPadding.top -
self.containerPadding.bottom - self.imageBorderWidth.top - self.imageBorderWidth.bottom -
120;

        // Check if image size is larger then maxWidth|maxHeight in settings
        if (self.options.maxWidth && self.options.maxWidth < maxImageWidth) {
            maxImageWidth = self.options.maxWidth;
        }
        if (self.options.maxHeight && self.options.maxHeight < maxImageHeight) {

```



```

        maxImageHeight = self.options.maxHeight;
    }

    // Is the current image's width or height is greater than the maxImageWidth or
maxImageHeight
    // option than we need to size down while maintaining the aspect ratio.
    if ((preloader.width > maxImageWidth) || (preloader.height > maxImageHeight)) {
        if ((preloader.width / maxImageWidth) > (preloader.height / maxImageHeight)) {
            imageWidth = maxImageWidth;
            imageHeight = parseInt(preloader.height / (preloader.width / imageWidth), 10);
            $image.width(imageWidth);
            $image.height(imageHeight);
        } else {
            imageHeight = maxImageHeight;
            imageWidth = parseInt(preloader.width / (preloader.height / imageHeight), 10);
            $image.width(imageWidth);
            $image.height(imageHeight);
        }
    }
}
self.sizeContainer($image.width(), $image.height());
};

preloader.src          = this.album[imageNumber].link;
this.currentImageIndex = imageNumber;
};

// Stretch overlay to fit the viewport
Lightbox.prototype.sizeOverlay = function() {
    this.$overlay
        .width($(document).width())
        .height($(document).height());
};

// Animate the size of the lightbox to fit the image we are showing
Lightbox.prototype.sizeContainer = function(imageWidth, imageHeight) {
    var self = this;

    var oldWidth  = this.$outerContainer.outerWidth();
    var oldHeight = this.$outerContainer.outerHeight();
    var newWidth  = imageWidth + this.containerPadding.left + this.containerPadding.right
+ this.imageBorderWidth.left + this.imageBorderWidth.right;
    var newHeight = imageHeight + this.containerPadding.top + this.containerPadding.bottom
+ this.imageBorderWidth.top + this.imageBorderWidth.bottom;

    function postResize() {
        self.$lightbox.find('.lb-dataContainer').width(newWidth);
        self.$lightbox.find('.lb-prevLink').height(newHeight);
        self.$lightbox.find('.lb-nextLink').height(newHeight);
        self.showImage();
    }

    if (oldWidth !== newWidth || oldHeight !== newHeight) {

```

```

        this.$outerContainer.animate({
            width: newWidth,
            height: newHeight
        }, this.options.resizeDuration, 'swing', function() {
            postResize();
        });
    } else {
        postResize();
    }
};

// Display the image and its details and begin preload neighboring images.
Lightbox.prototype.showImage = function() {
    this.$lightbox.find('.lb-loader').stop(true).hide();
    this.$lightbox.find('.lb-image').fadeIn(this.options.imageFadeDuration);

    this.updateNav();
    this.updateDetails();
    this.preloadNeighboringImages();
    this.enableKeyboardNav();
};

// Display previous and next navigation if appropriate.
Lightbox.prototype.updateNav = function() {
    // Check to see if the browser supports touch events. If so, we take the conservative
    approach
    // and assume that mouse hover events are not supported and always show prev/next
    navigation
    // arrows in image sets.
    var alwaysShowNav = false;
    try {
        document.createEvent('TouchEvent');
        alwaysShowNav = (this.options.alwaysShowNavOnTouchDevices) ? true : false;
    } catch (e) {}

    this.$lightbox.find('.lb-nav').show();

    if (this.album.length > 1) {
        if (this.options.wrapAround) {
            if (alwaysShowNav) {
                this.$lightbox.find('.lb-prev, .lb-next').css('opacity', '1');
            }
            this.$lightbox.find('.lb-prev, .lb-next').show();
        } else {
            if (this.currentImageIndex > 0) {
                this.$lightbox.find('.lb-prev').show();
                if (alwaysShowNav) {
                    this.$lightbox.find('.lb-prev').css('opacity', '1');
                }
            }
            if (this.currentImageIndex < this.album.length - 1) {
                this.$lightbox.find('.lb-next').show();
                if (alwaysShowNav) {

```

```

        this.$lightbox.find('.lb-next').css('opacity', '1');
    }
}
}
};

// Display caption, image number, and closing button.
Lightbox.prototype.updateDetails = function() {
    var self = this;

    // Enable anchor clicks in the injected caption html.
    // Thanks Nate Wright for the fix. @https://github.com/NateWr
    if (typeof this.album[this.currentImageIndex].title !== 'undefined' &&
        this.album[this.currentImageIndex].title !== '') {
        var $caption = this.$lightbox.find('.lb-caption');
        if (this.options.sanitizeTitle) {
            $caption.text(this.album[this.currentImageIndex].title);
        } else {
            $caption.html(this.album[this.currentImageIndex].title);
        }
        $caption.fadeIn('fast')
            .find('a').on('click', function(event) {
                if ($(this).attr('target') !== undefined) {
                    window.open($(this).attr('href'), $(this).attr('target'));
                } else {
                    location.href = $(this).attr('href');
                }
            });
    }

    if (this.album.length > 1 && this.options.showImageNumberLabel) {
        var labelText = this.imageCountLabel(this.currentImageIndex + 1, this.album.length);
        this.$lightbox.find('.lb-number').text(labelText).fadeIn('fast');
    } else {
        this.$lightbox.find('.lb-number').hide();
    }

    this.$outerContainer.removeClass('animating');

    this.$lightbox.find('.lb-dataContainer').fadeIn(this.options.resizeDuration,
function() {
    return self.sizeOverlay();
});
};

// Preload previous and next images in set.
Lightbox.prototype.preloadNeighboringImages = function() {
    if (this.album.length > this.currentImageIndex + 1) {
        var preloadNext = new Image();
        preloadNext.src = this.album[this.currentImageIndex + 1].link;
    }
    if (this.currentImageIndex > 0) {

```

```

    var preloadPrev = new Image();
    preloadPrev.src = this.album[this.currentImageIndex - 1].link;
  }
};

Lightbox.prototype.enableKeyboardNav = function() {
  $(document).on('keyup.keyboard', $.proxy(this.keyboardAction, this));
};

Lightbox.prototype.disableKeyboardNav = function() {
  $(document).off('.keyboard');
};

Lightbox.prototype.keyboardAction = function(event) {
  var KEYCODE_ESC      = 27;
  var KEYCODE_LEFTARROW = 37;
  var KEYCODE_RIGHTARROW = 39;

  var keycode = event.keyCode;
  var key      = String.fromCharCode(keycode).toLowerCase();
  if (keycode === KEYCODE_ESC || key.match(/x|o|c/)) {
    this.end();
  } else if (key === 'p' || keycode === KEYCODE_LEFTARROW) {
    if (this.currentImageIndex !== 0) {
      this.changeImage(this.currentImageIndex - 1);
    } else if (this.options.wrapAround && this.album.length > 1) {
      this.changeImage(this.album.length - 1);
    }
  } else if (key === 'n' || keycode === KEYCODE_RIGHTARROW) {
    if (this.currentImageIndex !== this.album.length - 1) {
      this.changeImage(this.currentImageIndex + 1);
    } else if (this.options.wrapAround && this.album.length > 1) {
      this.changeImage(0);
    }
  }
};

// Closing time. :-(
Lightbox.prototype.end = function() {
  this.disableKeyboardNav();
  $(window).off('resize', this.sizeOverlay);
  this.$lightbox.fadeOut(this.options.fadeDuration);
  this.$overlay.fadeOut(this.options.fadeDuration);
  $('select, object, embed').css({
    visibility: 'visible'
  });
  if (this.options.disableScrolling) {
    $('html').removeClass('lb-disable-scrolling');
  }
};

return new Lightbox();
}));

```