

# Dokumentacija

## Web programiranje PHP 2, sajt 1

### Mladen Bošković 37/16

<https://luggage-online.000webhostapp.com/>

## 1. UVOD

**1.1) Korišćeni programski jezici:** HTML, CSS, JavaScript, XML i PHP

Dodatno – AJAX

**1.2) Opis funkcionalnosti:**

- Dinamičko učitavanje iz baze (sav sadržaj koji postoji u bazi)
- Dinamički meni (na svim stranicama)
- Anketa (svi posetioči sajta mogu videti pitanja i odgovore, ali samo prijavljeni korisnici mogu glasati i videti rezultate glasanja)
- Kontakt forma (kontaktiranje vlasnika sajta)
- Registracija i prijavljivanje
- Promena i resetovanje lozinke
- Straničenje
- Filtriranje i sortiranje proizvoda
- Admin Panel (dodavanje, brisanje i menjanje korisnika i proizvoda)
- Korpa (dodavanje proizvoda u korpu, izbacivanje iz korpe i kupovina proizvoda)

## 1.3) Template



### 1.3.1) HTML

```
<!DOCTYPE html>
<html lang="sr">
<head>
  <title>
    <?php
      switch ($page){
        case "home" : echo "Početna"; break;
        case "products" : echo "Proizvodi"; break;
        case "product" : echo "Proizvod"; break;
        case "contact" : echo "Kontakt"; break;
        case "login" : echo "Prijava"; break;
        case "register" : echo "Registracija"; break;
        case "admin" : echo "Admin Panel"; break;
        case "author" : echo "Autor"; break;
        case "shop" : echo "Korpa"; break;
        case "404" : echo "Stranica ne postoji"; break;
        case "userprofile" : echo "Profil"; break;
        case "resetpassword" : echo "Resetovanje lozinke"; break;
        default : echo "Početna"; break;
      }
    ?>
  </title>
  <link rel="shortcut icon" href="images/favicon.ico"/>
  <link rel="stylesheet" type="text/css" href="css/style.css"/>
  <meta charset="UTF-8">
  <meta name="author" content="Mladen Bošković"/>
  <meta name="description" content="Luggage"/>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"/>
  <link rel="stylesheet" type="text/css" href="css/simpleLightbox.min.css"/>
</head>
<body>

<?php
include "functions/nav.php";
?>

<div id="beforenav">
  <div id="beforenav_drzac">
    <div>
      <?php if(isset($_SESSION['korisnik'])): ?>
        <a href="index.php?page=userprofile" title="Profil">
          <span class="fa fa-user" aria-hidden="true"></span>
        </a>
      </?php if(isset($_SESSION['korisnik'])): ?>
    </div>
  </div>
</div>
```

```

        <?php else: ?>
            <a href="#" title="Profil" id="profilLink">
                <span class="fa fa-user" aria-hidden="true"></span>
            </a>
        <?php endif; ?>
    </div>
<div><a href="index.php"></a></div>
<div>
    <?php if(isset($_SESSION['korisnik'])): ?>
        <a href="index.php?page=shop" title="Korpa">
            <span class="fa fa-shopping-cart" aria-hidden="true"></span>
        </a>
    <?php else: ?>
        <a href="#" title="Korpa" id="korpaLink">
            <span class="fa fa-shopping-cart" aria-hidden="true"></span>
        </a>
    <?php endif; ?>
</div>
</div>
</div>

<div id="nav2">
    <div id="nav_drzac">
        <div id="nav">
            <?php nav(); ?>

        </div>
    </div>
</div>

<?php if(isset($_SESSION['korisnik'])): ?>
    <?php if($_SESSION['korisnik']->uloga == "Admin"): ?>
        <a href="index.php?page=admin" id="adminLink">Admin Panel</a>
    <?php endif; ?>
<?php endif; ?>

<?php
$footerObj = new Footer(Baza::instanca());
$rezultatFooter = $footerObj->dohvatiSve();
?>

<div id="footer">
    <ul>
        <?php foreach ($rezultatFooter as $footer): ?>
            <?php if($footer->href == "index.php?page=author"): ?>
                <li><a href="<?= $footer->href ?>"><span class="<?= $footer->class ?>" aria-
hidden="true"></span></a></li>
            <?php else: ?>
                <li><a href="<?= $footer->href ?>" target="_blank"><span class="<?= $footer->class ?>"
aria-hidden="true"></span></a>
            <?php endif; ?>
        <?php endforeach; ?>
    </ul>

    <p>Copyright &copy; Luggage</p>
</div>

<script src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/simpleLightbox.min.js"></script>
</body>
</html>

```

### 1.3.2) CSS

```
/*-----NAV-----*/
#beforenav{
  width: 100%;
  height: 125px;
  display: flex;
  align-items: center;
  justify-content: center;
}

#beforenav_drzac{
  margin: 0px auto;
  width: 1300px;
  display: flex;
  align-items: center;
  justify-content: space-around;
}

#beforenav a span{
  font-size: 25px;
  color: #25262a;
}

#beforenav a span:hover{
  color: #7971ea;
}

#nav_drzac{
  width: 1300px;
  margin: 0px auto;
  display: flex;
  justify-content: center;
  align-items: center;
  padding-bottom: 5px;
  padding-top: 5px;
}

#nav2{
  width: 100%;
  height: 65px;

  border-top: 1px solid #F5F5F5;
  border-bottom: 1px solid #F5F5F5;
}

#nav{
  margin-left: 10px;
}

#nav ul{
  font-size: 0;
  margin: 0;
  padding: 0;
}

#nav ul li{
  display: inline-block;
  position: relative;
}

#nav ul li a{
  color: #25262a;
  display: block;
  font-size: 15px;
  padding: 15px 14px;
  transition: 0.3s linear;
  letter-spacing: 2px;
}
```

```

#nav ul li a:hover{
    background: #F5F5F5;
}

#nav ul li ul{
    border-top: 3px solid #7971ea;
    display: none;
    position: absolute;
    width: 175px;
}

#nav ul li ul li:last-child{
    border-bottom: 3px solid #7971ea;
}

#nav ul li ul li{
    border-top: 1px solid #F5F5F5;
    display: block;
}

#nav ul li ul li:first-child{
    border-top: none;
}

#nav ul li ul li a{
    background: white;
    display: block;
    padding: 10px 14px;
    color: #7971ea;
}

#nav ul li ul li a:hover{
    background: #F5F5F5;
    color: black;
}

#adminLink{
    position: fixed;
    right: 0px;
    top: 0px;
    font-weight: bold;
    font-size: 16px;
    //color: #25262a;
    color: white;
    padding: 5px 10px;
    //background-color: palegreen;
    background-color: #25262a;
    border-bottom-left-radius: 5px;
    letter-spacing: 1px;
}

#adminLink:hover{
    background-color: #7971ea;
}

/*-----NAV-----*/

/*-----FOOTER-----*/
#footer{
    width: 100%;
    height: 200px;
    background-color: #020B13;
    text-align: center;
}

#footer ul li{
    display: inline-block;
    font-size: 25px;
    padding: 10px;
}

```

```
}

#footer ul li a:link{
    color: #A9A9A9;
}

#footer ul li a:visited{
    color: #A9A9A9;
}

#footer ul li a:hover{
    color: #F5F5F5;
}

#footer p{
    color: #A9A9A9;
    font-size: 15px;
}

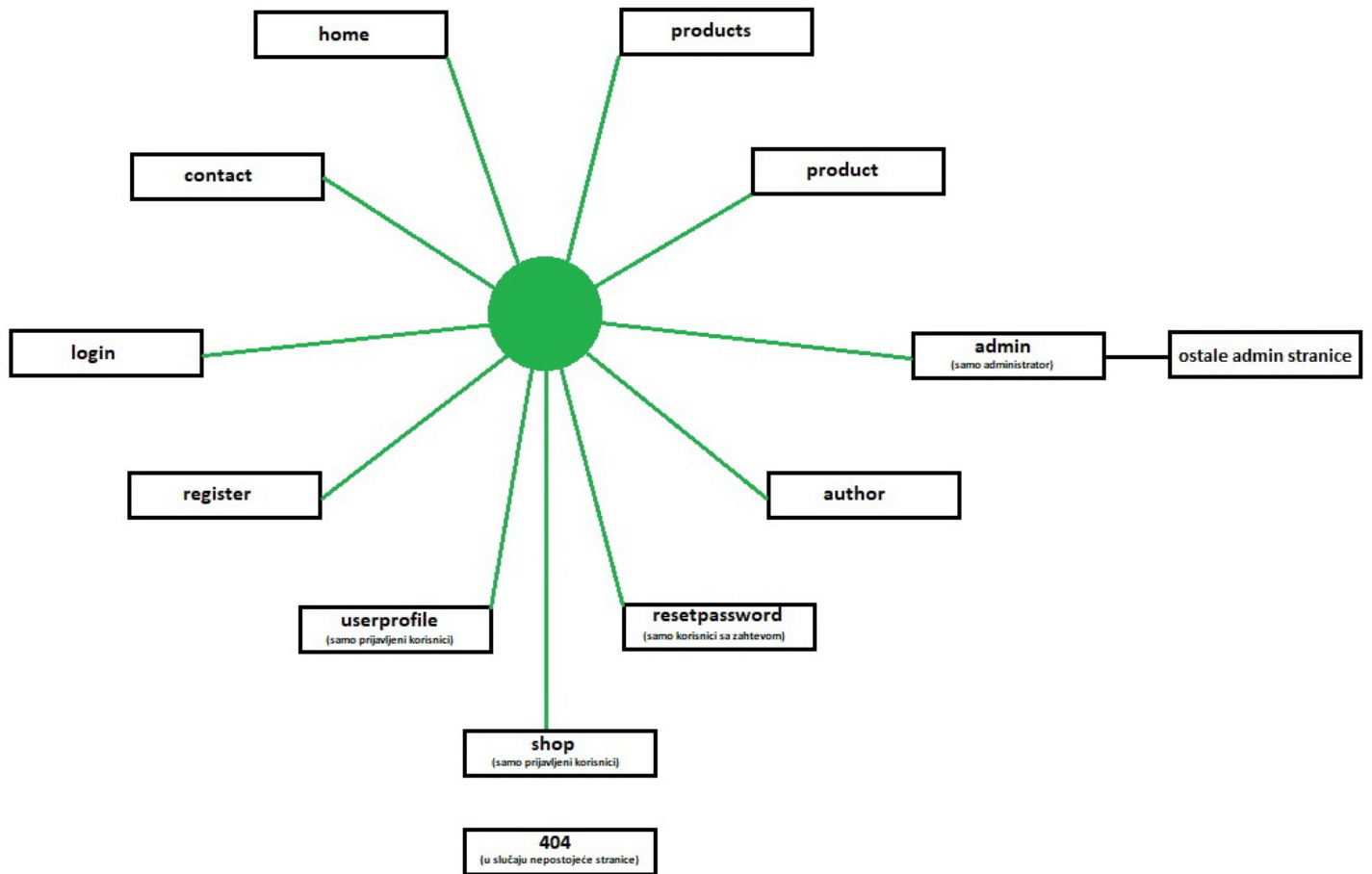
#footer ul{
    padding-top: 50px;
    padding-bottom: 25px;
}

/*-----FOOTER-----*/
```

#### 1.4) Korišćeni CSS – nije korišćen “tuđi” CSS kod (osim `font-awesome.min.css` i `simpleLightbox.min.css`)

## 2. ORGANIZACIJA

### 2.1) Organizaciona šema



### 2.2) Mapa sajta

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>
      https://luggage-online.000webhostapp.com/index.php?page=home
    </loc>
    <lastmod>2019-02-15</lastmod>
    <changefreq>weekly</changefreq>
    <priority>1</priority>
  </url>
  <url>
    <loc>
      https://luggage-online.000webhostapp.com/index.php?page=products
    </loc>
    <lastmod>2019-02-15</lastmod>
    <changefreq>daily</changefreq>
    <priority>1</priority>
  </url>
  <url>
    <loc>
      https://luggage-online.000webhostapp.com/index.php?page=contact
    </loc>
    <lastmod>2019-02-15</lastmod>
```

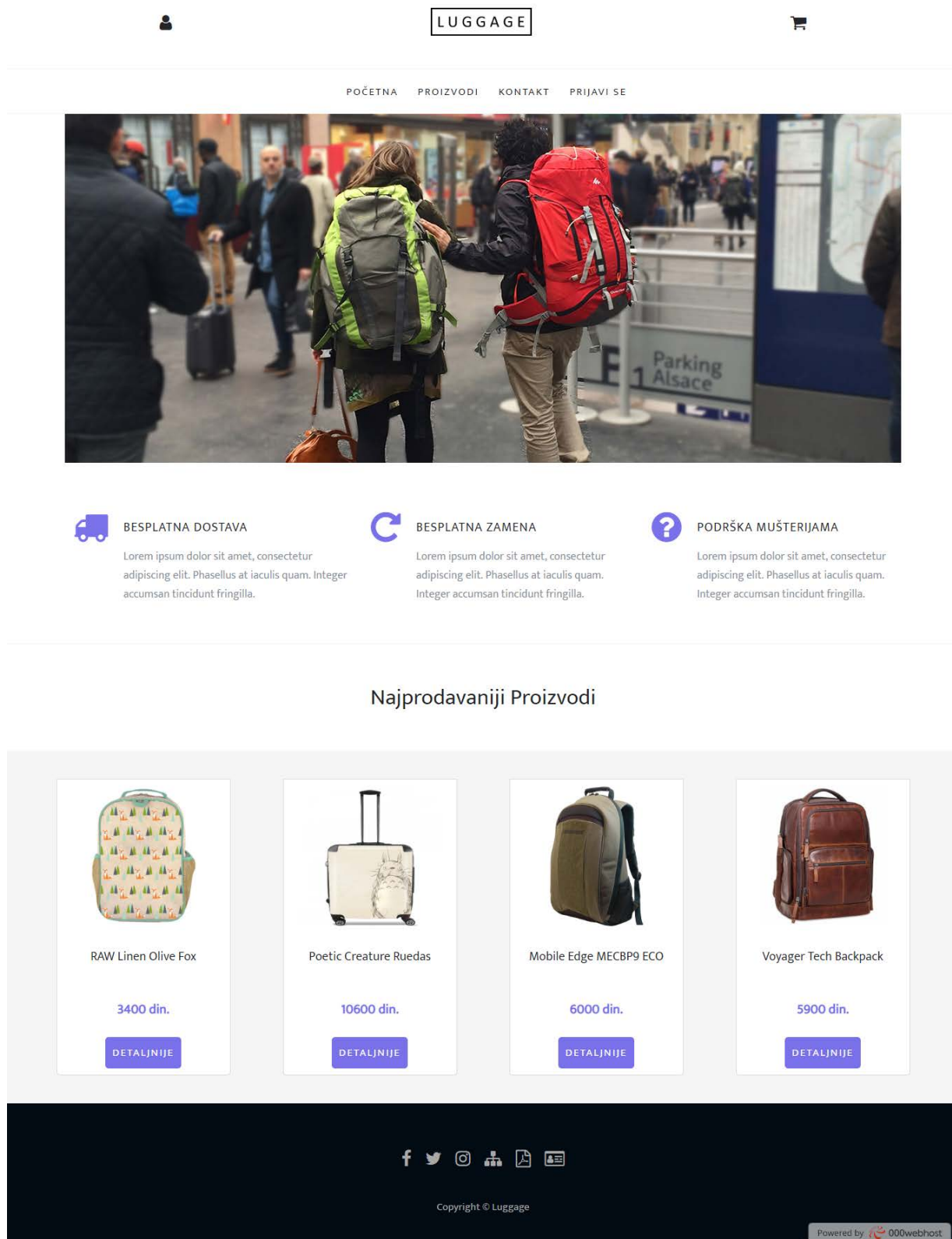
```
<changefreq>yearly</changefreq>
<priority>0.9</priority>
</url>
<url>
  <loc>
    https://luggage-online.000webhostapp.com/index.php?page=login
  </loc>
  <lastmod>2019-02-15</lastmod>
  <changefreq>yearly</changefreq>
  <priority>1</priority>
</url>
<url>
  <loc>
    https://luggage-online.000webhostapp.com/index.php?page=register
  </loc>
  <lastmod>2019-02-15</lastmod>
  <changefreq>yearly</changefreq>
  <priority>1</priority>
</url>
<url>
  <loc>
    https://luggage-online.000webhostapp.com/index.php?page=shop
  </loc>
  <lastmod>2019-02-15</lastmod>
  <changefreq>yearly</changefreq>
  <priority>1</priority>
</url>
<url>
  <loc>
    https://luggage-online.000webhostapp.com/index.php?page=author
  </loc>
  <lastmod>2019-02-15</lastmod>
  <changefreq>yearly</changefreq>
  <priority>0.5</priority>
</url>
<url>
  <loc>
    https://luggage-online.000webhostapp.com/index.php?page=userprofile
  </loc>
  <lastmod>2019-02-15</lastmod>
  <changefreq>yearly</changefreq>
  <priority>0.9</priority>
</url>
</urlset>
```



## 2.3) Slike stranica i opis funkcionalnosti

### 2.3.1) home


Ova stranica je namenjena da prikaže 4 najprodavanija proizvoda i služi kao smernica ka ostalim stranicama.




## 2.3.2) products

Ova stranica je namenjena da prikaže sve proizvode.

Funkcionalnosti: stranicenje, filtriranje i sortiranje proizvoda



LUGGAGE



POČETNA   PROIZVODI   KONTAKT   PRIJAVI SE

### Svi Proizvodi

1   2   3

Filtriranje i sortiranje

Sortiraj :  

Izaberite ▼

Kategorija :  
☐ Ranac  
☐ Kofer

Brend :  
☐ Venosis  
☐ Periveo  
☐ Panore Mobile  
☐ Avanyx

Pol :  
☐ Muški  
☐ Ženski


Cena :  

Cena od

Cena do

Poništavanje svega :  


Poništi sve



Professional Weekender

4900 din.


DETALJNIJE



Hatchet Backpack Sun

3000 din.


DETALJNIJE



Mobile Edge MECBP9 ECO

6000 din.


DETALJNIJE



Poetic Creature Ruedas

10600 din.


DETALJNIJE



Backpack Vintage Canvas

7200 din.


DETALJNIJE



RAW Linen Olive Fox

3400 din.


DETALJNIJE



Minnie Mouse

7300 din.

DETALJNIJE









Evening Standard


6200 din.

DETALJNIJE

1   2   3



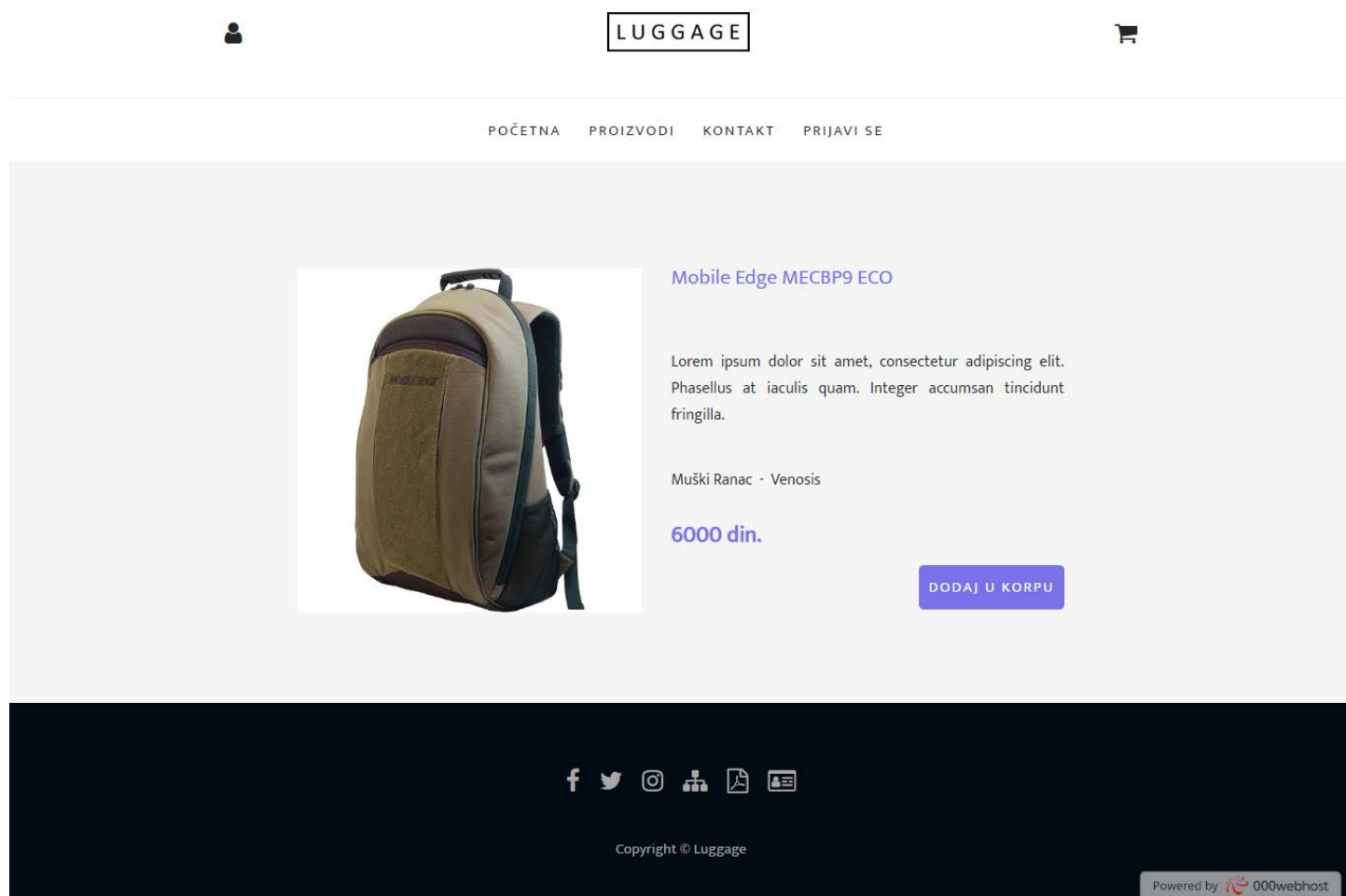
Copyright © Luggage

Powered by  000webhost

### 2.3.3) product&id='x'

Ova stranica je namenjena da prikaže detaljnije informacije o određenom proizvodu. Administrator umesto opcije za dodavanje u korpu ima opciju za menjanje i brisanje proizvoda.


Funkcionalnosti: dodavanje u korpu i menjanje i brisanje proizvoda




### 2.3.4) contact

Ova stranica je namenjena za kontaktiranje vlasnika sajta i informisanje o firmi.

Funkcionalnosti: anketa i kontaktiranje vlasnika sajta



LUGGAGE



POČETNA   PROIZVODI   KONTAKT   PRIJAVI SE

Obratite nam se putem formulara

Adresa 📍 Kruševac, Vidovdanska 10

Telefon 📞 037/581-321







Kratka anketa

Izaberite pitanje ▼


Email

Poruka

POŠALJI




Copyright © Luggage

Powered by  000webhost


### 2.3.5) login

Ova stanica je namenjena za prijavljivanje korisnika.

Funkcionalnosti: prijavljivanje korisnika, resetovanje lozinke



LUGGAGE



POČETNA   PROIZVODI   KONTAKT   PRIJAVI SE

Prijavite se

Korisničko ime

Lozinka

PRIJAVI SE

[Niste registrovani?](#)

[Zaboravili ste lozinku?](#)




Copyright © Luggage

Powered by  000webhost


### 2.3.6) register

Ova stanica je namenjena za registraciju korisnika.

Funkcionalnosti: registracija korisnika



LUGGAGE



POČETNA   PROIZVODI   KONTAKT   PRIJAVI SE

## Registrujte se

Ime

Prezime

Email

Korisničko ime

Lozinka

Ponovite lozinku

REGISTRUJ SE



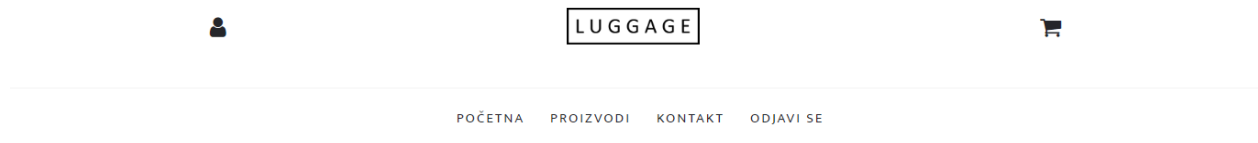
Copyright © Luggage

Powered by  000webhost

### 2.3.7) shop

Ova stranica je namenjena da prikaže proizvode u korpi i kupljene proizvode. Kada nema proizvoda u korpi ili kada nema kupljenih proizvoda izgled stranice se menja.




Funkcionalnosti: izbacivanje proizvoda iz korpe i kupovina proizvoda, administrator nema opciju za izbacivanje proizvoda iz korpe



#### U korpi :

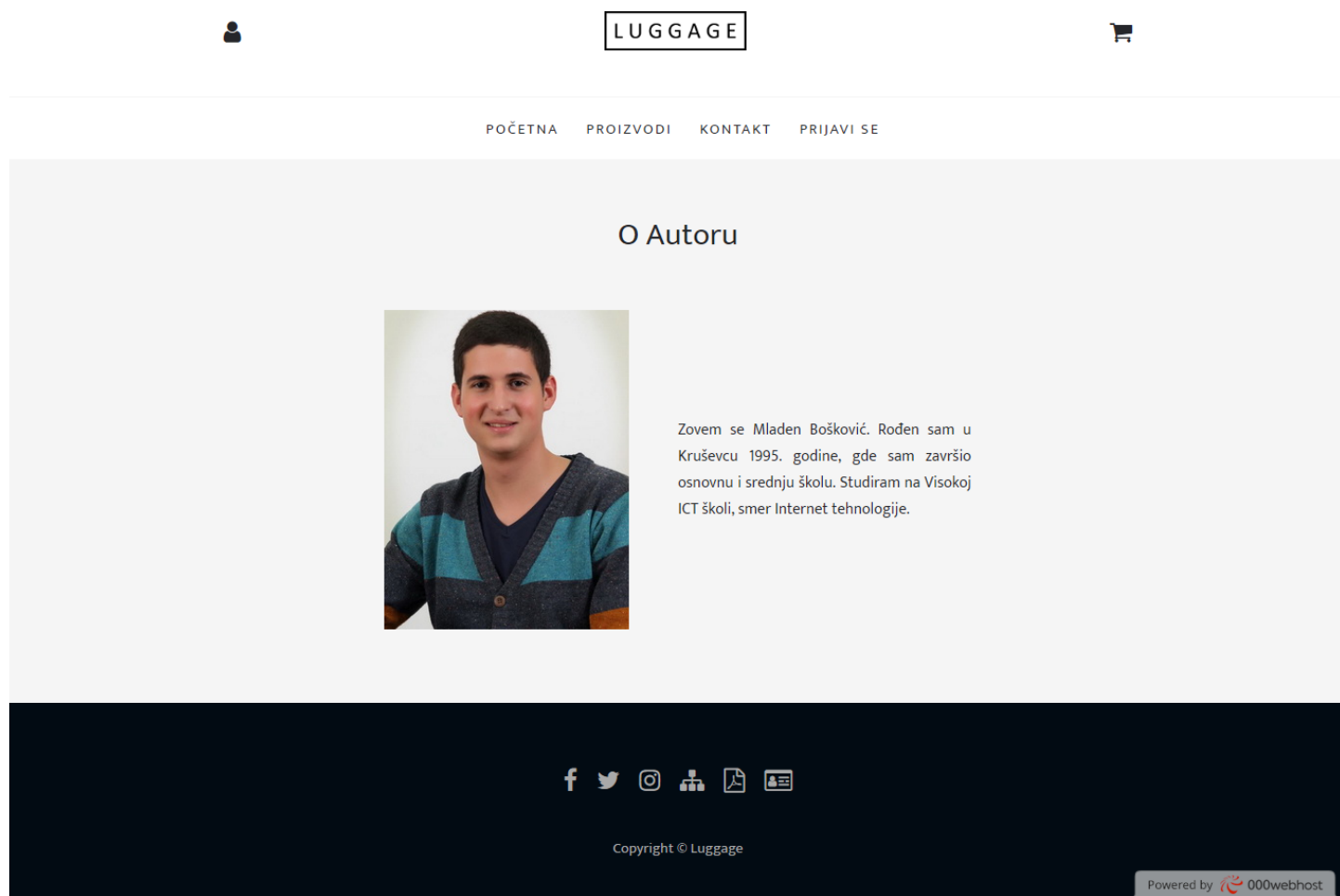
SLIKA	NAZIV	CENA	IZBACI IZ KORPE
	Hatchet Backpack Sun	3000 din.	<a href="#">Izbaci iz korpe</a>
	Poetic Creature Ruedas	10600 din.	<a href="#">Izbaci iz korpe</a>
Ukupno: 13600 din.			
<a href="#">KUPI</a>			

#### Kupljeno :

SLIKA	NAZIV	CENA
	RAW Linen Olive Fox	3400 din.
	Professional Weekender	4900 din.
	American Heritage Leather	12300 din.

### 2.3.8) author

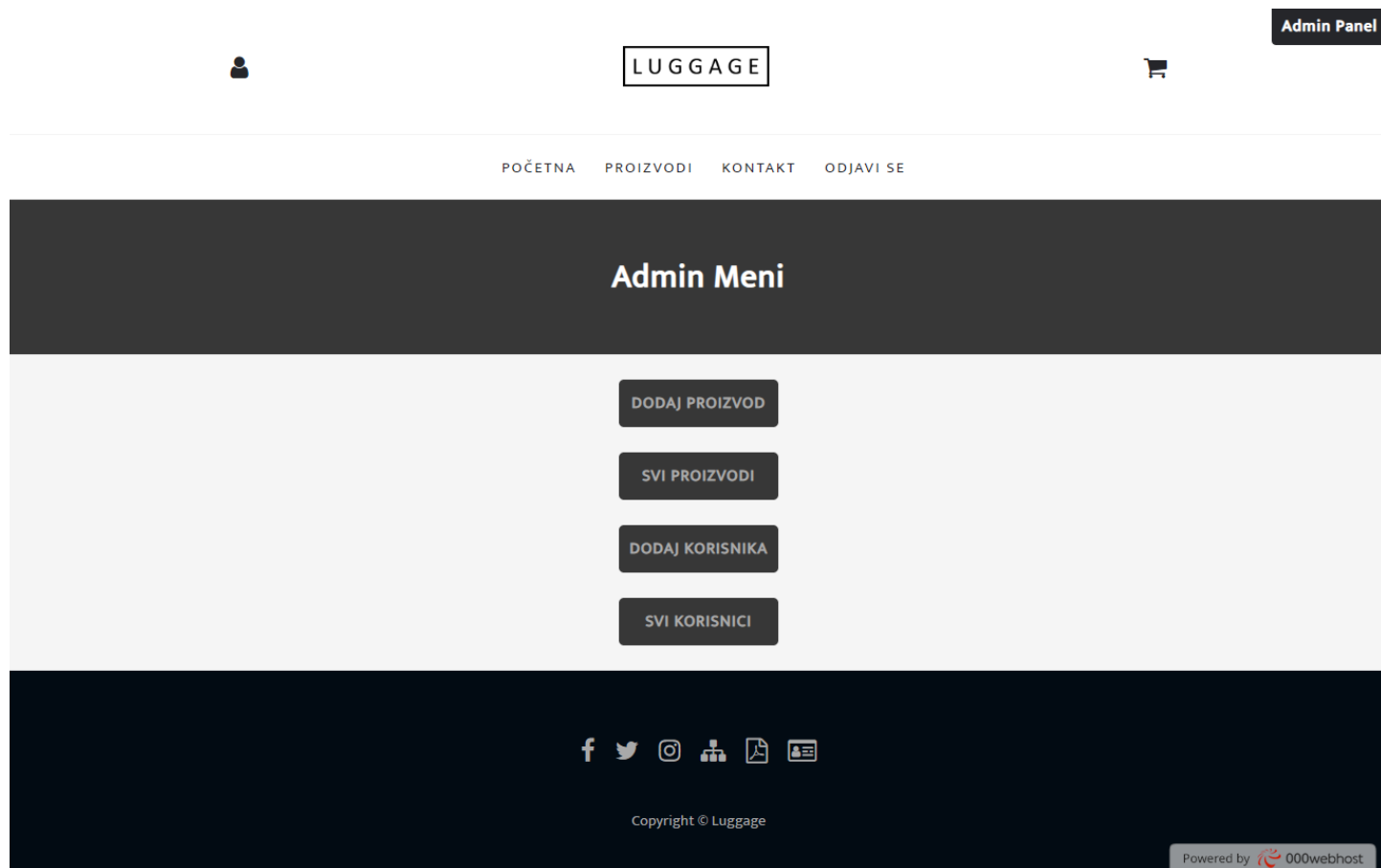
Ova stranica je namenjena da prikaže informacije o autoru sajta.





### 2.3.9.1) adminhome


Ova stranica je navigaciona stranica za administratora.




### 2.3.9.2) admin&adminaction=addproduct

Ova stranica je namenjena za dodavanje proizvoda.

Funkcionalnosti: dodavanje proizvoda



LUGGAGE



Admin Panel

POČETNA   PROIZVODI   KONTAKT   ODJAVI SE

## Admin Meni

DODAJ PROIZVOD

SVI PROIZVODI

DODAJ KORISNIKA

SVI KORISNICI

### Dodajte Proizvod

Naziv

Opis

Slika

Choose File | No file chosen

Cena

Brend

Izaberite ▼







Pol

Izaberite ▼


Kategorija

Izaberite ▼

DODAJ PROIZVOD



Copyright © Luggage

Powered by  000webhost

2.3.9.3) admin&adminaction=allproducts

Ova stranica je namenjena da prikaže sve proizvode. Kada nema proizvoda stranica se menja.

Funkcionalnosti: menjanje i brisanje proizvoda

LUGGAGE

Admin Panel

POČETNA

PROIZVODI

KONTAKT

ODJAVI SE

Admin Meni

DODAJ PROIZVOD

SVI PROIZVODI

DODAJ KORISNIKA

SVI KORISNICI

Svi Proizvodi

SLIKA	NAZIV	OPIS	CENA	POL	BREND	KATEGORIJA	IZMENI	OBRIŠI
	Professional Weekender	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	4900 din.	Muški	Verosis	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	Hatchet Backpack Sun	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	3000 din.	Ženski	Verosis	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	Mobile Edge MECBP9 ECO	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	6000 din.	Muški	Verosis	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	Poetic Creature Ruedas	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	10600 din.	Ženski	Verosis	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>
	Backpack Vintage Canvas	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	7200 din.	Muški	Perineo	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	BMW Linen Olive Fox	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	3400 din.	Ženski	Perineo	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	Minnie Mouse	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	7500 din.	Ženski	Perineo	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>
	Evening Standard	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	6200 din.	Muški	Perineo	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>
	Stay True Wheeled Cabin Travel Suitcase	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	6500 din.	Ženski	Perineo	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>
	Dinosaur Backpack	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	3700 din.	Ženski	Panore Mobile	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	Olympia USA Hopkins	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	5100 din.	Ženski	Panore Mobile	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	Monopoly Leather Backpack	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	5600 din.	Ženski	Panore Mobile	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	The Nostalgic Chatelet	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	8500 din.	Ženski	Panore Mobile	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>
	Voyager Tech Backpack	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	5900 din.	Muški	Avanyx	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	Chiltern Backpack Oak Natural Grain Leather	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	6100 din.	Ženski	Avanyx	Ranac	<div>IZMENI</div>	<div>OBRIŠI</div>
	New Horizon Wheeled Cabin	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	5100 din.	Muški	Avanyx	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>
	American Heritage Leather	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	12100 din.	Muški	Avanyx	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>
	Base Spinner	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.	8600 din.	Ženski	Avanyx	Kofer	<div>IZMENI</div>	<div>OBRIŠI</div>

f

t

@

u

id


Copyright © Luggage

Powered by LUGGAGE


### 2.3.9.4) admin&adminaction=updateproduct&idupdate='x'

Ova stranica je namenjena za menjanje proizvoda.

Funkcionalnosti: menjanje i brisanje proizvoda



LUGGAGE



Admin Panel


POČETNA   PROIZVODI   KONTAKT   ODJAVI SE

Admin Meni

DODAJ PROIZVOD   SVI PROIZVODI   DODAJ KORISNIKA   SVI KORISNICI

Izmenite Proizvod

Obriši



Naziv

Professional Weekender

Opis

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at iaculis quam. Integer accumsan tincidunt fringilla.

Slika

Choose File

No file chosen

Cena

4900

Brend

Venosis







Pol

Muški


Kategorija

Ranac

IZMENI PROIZVOD




Copyright © Luggage

Powered by  000webhost


### 2.3.9.5) admin&adminaction=adduser

Ova stranica je namenjena za dodavanje korisnika.

Funkcionalnosti: dodavanje korisnika



LUGGAGE



Admin Panel

POČETNA   PROIZVODI   KONTAKT   ODJAVI SE

## Admin Meni

DODAJ PROIZVOD

SVI PROIZVODI







DODAJ KORISNIKA

SVI KORISNICI


### Dodajte Korisnika

Ime	<input type="text"/>	Prezime	<input type="text"/>
Email	<input type="text"/>	Korisničko ime	<input type="text"/>
Lozinka	<input type="password"/>	Ponovite lozinku	<input type="password"/>
Uloga	<div>Izaberite ▼</div>	Aktivan	<div>Izaberite ▼</div>

DODAJ KORISNIKA




Copyright © Luggage

Powered by  000webhost


### 2.3.9.6) admin&adminaction=allusers

Ova stranica je namenjena da prikaže sve korisnike. Kada nema korisnika izgled stranice se menja.

Funkcionalnosti: menjanje i brisanje korisnika



LUGGAGE



Admin Panel

POČETNA   PROIZVODI   KONTAKT   ODJAVI SE

## Admin Meni

DODAJ PROIZVOD







SVI PROIZVODI

DODAJ KORISNIKA


SVI KORISNICI

## Svi Korisnici

IME	PREZIME	EMAIL	KORISNIČKO IME	ULOGA	AKTIVAN	IZMENI	OBRISI
Seed	Database	seed_database@luggage.com	seed_database	Korisnik	Ne	<div>Izmeni</div>	<div>Obriši</div>
Korisnik	Korisnik	korisnik@luggage.com	korisnik	Korisnik	Da	<div>Izmeni</div>	<div>Obriši</div>
Admin	Luggage	admin@luggage.com	admin	Admin	Da	<div>Izmeni</div>	<div>Obriši</div>
Korisnik	Korisnik	korisnik2@luggage.com	korisnik2	Korisnik	Da	<div>Izmeni</div>	<div>Obriši</div>




Copyright © Luggage

Powered by  000webhost


### 2.3.9.7) admin&adminaction=updateuser&idupdate='x'

Ova stranica je namenjena za menjanje korisnika.

Funkcionalnosti: menjanje i brisanje korisnika



LUGGAGE



Admin Panel

POČETNA   PROIZVODI   KONTAKT   ODJAVI SE

## Admin Meni

DODAJ PROIZVOD

SVI PROIZVODI

DODAJ KORISNIKA

SVI KORISNICI

### Izmenite Korisnika

Obriši

Ime

Seed

Prezime

Database

Email

seed\_database@luggage.com

Korisničko ime

seed\_database







Uloga

Korisnik


Aktivan

Ne

IZMENI KORISNIKA

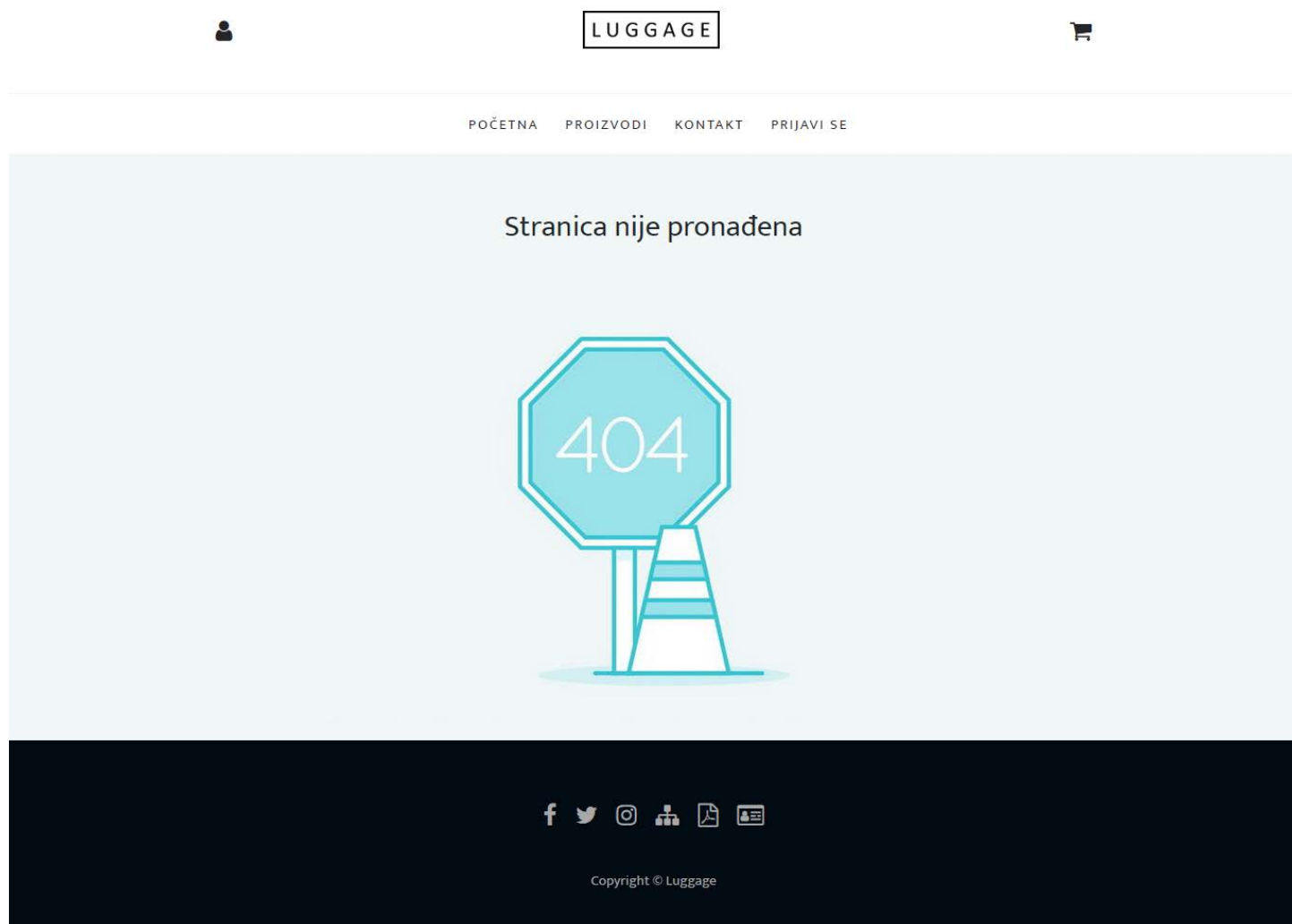


Copyright © Luggage

Powered by  000webhost

### 2.3.10) 404

Ova stranica je namenjena da se na nju redirektuje pri grešci 404 ili 403.







### 2.3.11) userprofile

Ova stranica je namenjena za pregled informacija o prijavljenom korisniku.

Funkcionalnost: promena lozinke



LUGGAGE



Admin Panel

POČETNA   PROIZVODI   KONTAKT   ODJAVI SE

Vaš Profil







Podaci	
Ime	Admin
Prezime	Luggage
Korisničko ime	admin
Uloga	Admin
Datum registracije	14.02.2019

Trenutna lozinka


Nova lozinka

Ponovite novu lozinku

PROMENI LOZINKU




Copyright © Luggage

Powered by  000webhost


### 2.3.12) resetpassword

Ova stranica je namenjena za resetovanje lozinke.

Funkcionalnost: resetovanje lozinke



LUGGAGE



POČETNA   PROIZVODI   KONTAKT   PRIJAVI SE

## Resetujte Lozinku

Nova lozinka

Ponovite novu lozinku

PROMENI LOZINKU



Copyright © Luggage

Powered by  000webhost

## 3. KODOVI

### 3.1) HTML (+ njemu odgovarajući PHP)

**3.1.0)** Ovdje počinju kodovi stranica. Prvo slede kodovi za **index.php**, **head** i **nav** bez kojih naredni kodovi ne bi bili funkcionalni.

- **index.php** (osnovna stranica)

```
<?php
session_start();

function __autoload($nazivKlase)
{
    $putanja = "classes/" . $nazivKlase . ".php";
    require_once ($putanja);
}

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

$korisnikLogovan = "";
if(isset($_SESSION['korisnik'])){
    $korisnikLogovan = $_SESSION['korisnik']->korisnikID;
}

include "views/head.php";
include "views/nav.php";
switch ($page)
{
    case "home" : include "views/home.php"; break;
    case "register" : if(isset($_SESSION['korisnik'])){
        include "views/home.php";
    } else {
        include "views/register.php";
    } break;
    case "login" : if(isset($_SESSION['korisnik'])){
        include "views/home.php";
    } else {
        include "views/login.php";
    } break;
    case "shop" : include "views/shop.php"; break;
    case "contact" : include "views/contact.php"; break;
    case "products" : include "views/products.php"; break;
    case "product" : include "views/product.php"; break;
    case "author" : include "views/author.php"; break;
    case "404" : include "views/404.php"; break;
    case "admin" : if(isset($_SESSION['korisnik'])){
        if($_SESSION['korisnik']->uloga == "Admin"){
            include "views/admin/admin.php"; break;
        } else{
            include "views/home.php"; break;
        }
    } else{
        include "views/home.php"; break;
    } break;
    case "userprofile" : include "views/userprofile.php"; break;
    case "resetpassword" : include "views/resetpassword.php"; break;
    default : include "views/home.php"; break;
}
```

```
}  
include "views/footer.php";
```

## - head

```
<!DOCTYPE html>  
<html lang="sr">  
<head>  
    <title>  
        <?php  
            switch ($page){  
                case "home" : echo "Početna"; break;  
                case "products" : echo "Proizvodi"; break;  
                case "product" : echo "Proizvod"; break;  
                case "contact" : echo "Kontakt"; break;  
                case "login" : echo "Prijava"; break;  
                case "register" : echo "Registracija"; break;  
                case "admin" : echo "Admin Panel"; break;  
                case "author" : echo "Autor"; break;  
                case "shop" : echo "Korpa"; break;  
                case "404" : echo "Stranica ne postoji"; break;  
                case "userprofile" : echo "Profil"; break;  
                case "resetpassword" : echo "Resetovanje lozinke"; break;  
                default : echo "Početna"; break;  
            }  
        ?>  
    </title>  
    <link rel="shortcut icon" href="images/favicon.ico"/>  
    <link rel="stylesheet" type="text/css" href="css/style.css"/>  
    <meta charset="UTF-8">  
    <meta name="author" content="Mladen Bošković"/>  
    <meta name="description" content="Luggage"/>  
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome/4.7.0/css/font-awesome.min.css"/>  
    <link rel="stylesheet" type="text/css" href="css/simpleLightbox.min.css"/>  
  
</head>  
<body>
```

## - nav

```
<?php  
include "functions/nav.php";  
?>  
  
<div id="beforenav">  
    <div id="beforenav_drzac">  
        <div>  
            <?php if(isset($_SESSION['korisnik'])): ?>  
                <a href="index.php?page=userprofile" title="Profil">  
                    <span class="fa fa-user" aria-hidden="true"></span>  
                </a>  
            <?php else: ?>  
                <a href="#" title="Profil" id="profilLink">  
                    <span class="fa fa-user" aria-hidden="true"></span>  
                </a>  
            <?php endif; ?>  
        </div>  
        <div><a href="index.php"></a></div>  
        <div>  
            <?php if(isset($_SESSION['korisnik'])): ?>  
                <a href="index.php?page=shop" title="Korpa">  
                    <span class="fa fa-shopping-cart" aria-hidden="true"></span>  
                </a>  
            </div>
```

```
        <?php else: ?>
            <a href="#" title="Korpa" id="korpaLink">
                <span class="fa fa-shopping-cart" aria-hidden="true"></span>
            </a>
        <?php endif; ?>
    </div>
</div>
</div>
</div>

<div id="nav2">
    <div id="nav_drzac">
        <div id="nav">
            <?php nav(); ?>

        </div>
    </div>
</div>

<?php if(isset($_SESSION['korisnik'])): ?>
    <?php if($_SESSION['korisnik']->uloga == "Admin"): ?>
        <a href="index.php?page=admin" id="adminLink">Admin Panel</a>
    <?php endif; ?>
<?php endif; ?>
```

### 3.1.1) home

```
<?php
$slajderObj = new Slajder(Baza::instanca());
$slajder = $slajderObj->dohvatiSve();
$br = 1;
?>

<div id="slajder">
    
    <?php foreach ($slajder as $s): ?>
        "/>
    <?php endforeach; ?>
</div>

<div id="pogodnosti">
    <div id="pogodnosti_drzac">
        <table>
            <tr>
                <td>
                    <a>
                        <span class="fa fa-truck" aria-hidden="true"></span>
                    </a>
                </td>
                <td>
                    <p class="pogodnosti_naslov">BESPLATNA DOSTAVA</p>
                </td>
                <td>
                    <a>
                        <span class="fa fa-rotate-right" aria-hidden="true"></span>
                    </a>
                </td>
                <td>
                    <p class="pogodnosti_naslov">BESPLATNA ZAMENA</p>
                </td>
                <td>
                    <a>
                        <span class="fa fa-question-circle" aria-hidden="true"></span>
                    </a>
                </td>
                <td>
                    <p class="pogodnosti_naslov">PODRŠKA MUŠTERIJAMA</p>
                </td>
            </tr>
            <tr>
                <td>
                </td>
                <td>
                </td>
                <td>
                    <p class="pogodnosti_tekst">
                        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at
                        iaculis quam. Integer accumsan tincidunt fringilla.
                    </p>
                </td>
                <td>
                </td>
                <td>
                </td>
                <td>
                    <p class="pogodnosti_tekst">
                        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at
                        iaculis quam. Integer accumsan tincidunt fringilla.
                    </p>
                </td>
                <td>
                </td>
                <td>
                </td>
                <td>
                    <p class="pogodnosti_tekst">
                        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus at
                        iaculis quam. Integer accumsan tincidunt fringilla.
                    </p>
                </td>
            </tr>
        </table>
    </div>
</div>
```

```

        </p>
    </td>
</tr>
</table>
</div>
</div>
<div class="naslov_svaki">
    <p>Najprodavaniji Proizvodi</p>
</div>

<?php
$proizvodObj = new Proizvod(Baza::instanca());
$rezultatProizvodi = $proizvodObj->dohvatiCetiriNajprodavanijaProizvoda();
?>
<div id="proizvodi_home">
    <div id="proizvodi_drzac_home">
        <?php foreach ($rezultatProizvodi as $proizvod): ?>
            <div class="proizvod">
                alt) ?>"/>
                <div class="detaljnije2">
                    <p class="ps_naziv"><?= stripslashes($proizvod->proizvod) ?></p>
                    <p class="ps_cena"><?= $proizvod->cena ?> din.</p>
                    <a href="index.php?page=product&id=<?= $proizvod->proizvodID ?>"
class="detaljnije">DETALJNIJE</a>
                </div>
            </div>
        <?php endforeach; ?>
    </div>
</div>

```

### 3.1.2) products

```
<?php
$katgorija = "";
if(isset($_GET['category'])){
    $katgorija = $_GET['category'];
}

$katgorijaObj = new Kategorija(Baza::instanca());
$resultatKategorija = $katgorijaObj->dohvatiSve();

$brendObj = new Brend(Baza::instanca());
$resultatBrend = $brendObj->dohvatiSve();

$proizvodObj = new Proizvod(Baza::instanca());
$resultatProizvodi = $proizvodObj->dohvatiSve();

$str = "1";
if(isset($_GET['str'])){
    $str = $_GET['str'];
}

?>

<input type="hidden" value="<?= $str ?>" id="kojaStranica" name="kojaStranica"/>

<div class="naslov_svaki">
    <p>Svi Proizvodi</p>
</div>

<div class="stranice">
</div>

<div id="proizvodi_products">
    <div id="proizvodi_drzac">
        <div id="proizvodi_filter">
            <h3>Filtriranje i sortiranje</h3>
            <div id="sortirajDiv">
                <p>Sortiraj :</p>
                <select id="sortiraj">
                    <option value="0">Izaberite</option>
                    <option value="1">Cena rastuća</option>
                    <option value="2">Cena opadajuća</option>
                    <option value="3">Naziv A-Z</option>
                    <option value="4">Naziv Z-A</option>
                </select>
            </div>
            <div>
                <p>Kategorija :</p>
                <?php foreach ($rezultatKategorija as $kat): ?>
                    <?php if(($katgorija) && ($katgorija == strtolower($kat->katgorija))): ?>
                        <input type="checkbox" name="katgorija_filter" class="katgorija_filter"
value="<?= $kat->katgorijaID ?>" checked/> <?= $kat->katgorija ?> <br/>
                    <?php else: ?>
                        <input type="checkbox" name="katgorija_filter" class="katgorija_filter"
value="<?= $kat->katgorijaID ?>"/> <?= $kat->katgorija ?> <br/>
                    <?php endif; ?>
                <?php endforeach; ?>
            </div>

            <div>
                <p>Brend :</p>
                <?php foreach ($rezultatBrend as $brend): ?>
                    <input type="checkbox" name="brend_filter" class="brend_filter" value="<?=
$brend->brendID ?>"/> <?= $brend->brend ?> <br/>
                <?php endforeach; ?>
            </div>
        </div>
    </div>
</div>
```



```
        <div>
            <p>Pol :</p>
            <input type="checkbox" name="pol_filter" value="m" class="pol_filter"/> Muški <br/>
            <input type="checkbox" name="pol_filter" value="z" class="pol_filter"/> Ženski
        <br/>
        </div>

        <div>
            <p>Cena :</p>
            <input type="number" id="minCena" name="minCena" placeholder="Cena od"
class="inputCenaFilter"/> <br/>
            <input type="number" id="maxCena" name="maxCena" placeholder="Cena do"
class="inputCenaFilter"/> <br/>
        </div>

        <div>
            <p>Poništavanje svega :</p>
            <input type="button" id="ponistiSve" name="ponistiSve" value="Poništi sve"
class="" />
        </div>

    </div>

    <div id="proizvodi_sadrzaj">
    </div>

</div>

<div class="stranice">
</div>

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

### 3.1.3) product&id='x'

```
<?php
$proizvod = null;
if(isset($_GET['id'])){
    $id = $_GET['id'];
    $proizvodObj = new Proizvod(Baza::instanca());
    $proizvodObj->setProizvodID($id);
    $proizvod = $proizvodObj->dohvatiJedan();
    if(!$proizvod){
        header("Location: https://luggage-online.000webhostapp.com/index.php");
    }
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
?>

<div id="product_sadrzaj">
    <div id="product_drzac">
        <a href="images/upload/<?= stripslashes($proizvod->src) ?>" class="vecaSlika">alt)
?>"/></a>
        <div id="o_proizvodu">
            <p id="p_naziv"><?= stripslashes($proizvod->proizvod) ?></p>
            <p id="p_opis"><?= stripslashes($proizvod->opis) ?></p>
            <p id="p_dodatno"><?php echo $proizvod->pol=="m" ? "Muški" : "Ženski"; ?> <?=
$proizvod->kategorija ?> &nbsp;   <?= $proizvod->brend ?></p>
            <p id="p_cena"><?= $proizvod->cena ?> din.</p>
            <?php if(isset($_SESSION['korisnik'])):
                if($_SESSION['korisnik']->uloga == "Admin"): ?>
                    <a href="index.php?page=admin&adminaction=updateproduct&idupdate=<?= $proizvod-
>proizvodID ?>" class="p_opcije">IZMENI PROIZVOD</a>
                    <?php else: ?>
                        <a href="#" data-id="<?= $proizvod->proizvodID ?>" class="p_opcije dodaj">DODAJ
U KORPU</a>
                    <?php endif; ?>
                <?php else: ?>
                    <a href="#" data-id="<?= $proizvod->proizvodID ?>" class="p_opcije dodaj">DODAJ U
KORPU</a>
                <?php endif; ?>
                <input type="hidden" id="sessionIdKor" name="sessionIdKor" value="<?= $korisnikLogovan
?>"/>
            </div>
        </div>
    </div>
</div>
```

### 3.1.4) contact

```
<?php
$pitiranjeObj = new Pitiranje(Baza::instanca());
$rezPitiranje = $pitiranjeObj->dohvatiAktivnaPitiranja();

?>
<div class="naslov_svaki">
    <p>Obratite nam se putem formulara</p>
</div>

<div id="contact_sadrzaj">
    <div id="contact_drzac">

        <div>
            <ul>
                <li>Adresa <span class="fa fa-home"></span> Kruševac, Vidovdanska 10</li>
                <li>Telefon <span class="fa fa-phone"></span> 037/581-321</li>
                <li><span class="contactSpan"></span></li>
            </ul>

            <div id="kontakt_anketa">
                <?php if(count($rezPitiranje)): ?>
                    <div id="anketa">
                        <h3>Kratka anketa</h3><br/>
                        <p>
                            <select id="selectAnketa">
                                <option value="0">Izaberite pitanje</option>
                                <?php foreach ($rezPitiranje as $pitiranje): ?>
                                    <option value="<?= $pitiranje->pitanjeID ?>"><?= $pitiranje-
>pitanje ?></option>
                                <?php endforeach; ?>
                            </select>
                        </p>
                    </div>
                <?php endif; ?>

                <div id="odgovori">
                </div>

                <div id="glasajDugmeDiv">
                    <p><input type="button" value="Glasaj" name="glasaj" id="glasaj"/></p>
                    <p><input type="button" name="rezultatiDugme" id="rezultatiDugme"
value="Rezultati"/></p>
                    <input type="hidden" id="contactKorisnikLogovan" name="contactKorisnikLogovan"
value="<?= $korisnikLogovan ?>" />
                </div>

                <div id="izaberiteOdgovor">
                </div>

            </div>
        </div>

        <div>
            <form id="forma_contact" name="forma_contact">
                <table id="tabela_contact">
                    <tr>
                        <td>Email</td>
                    </tr>
                    <tr>
                        <td><input type="text" id="contactEmail" name="contactEmail"></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="contactEmailGreska" class="ispisGreskeContact"></div>
                        </td>
                    </tr>
                </table>
            </form>
        </div>
    </div>
</div>
```

```
        </tr>
        <tr>
            <td>Poruka</td>
        </tr>
        <tr>
            <td><textarea id="poruka" name="poruka"></textarea></td>
        </tr>
        <tr>
            <td>
                <div id="contactPorukaGreska" class="ispisGreskeContact"></div>
            </td>
        </tr>
        <tr>
            <td><input type="button" id="contactDugme" name="contactDugme"
value="POŠALJI"/></td>
        </tr>
    </table>
</form>

    <div id="contact_greske">
    </div>

</div>
</div>

<div class="greske">

</div>
```

### 3.1.5) login

```
<div class="naslov_svaki">
    <p>Prijavite se</p>
</div>

<div id="login_sadrzaj">
    <div id="login_drzac">
        <form id="forma_login" name="forma_login">
            <table id="tabela_login">
                <tr>
                    <td>Korisničko ime</td>
                </tr>
                <tr>
                    <td><input type="text" id="loginKorIme" name="loginKorIme" placeholder=" "
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="loginKorImeGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Lozinka</td>
                </tr>
                <tr>
                    <td><input type="password" id="loginLozinka" name="loginLozinka" placeholder=" "
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="loginLozinkaGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td><input type="button" id="loginDugme" name="loginDugme" value="PRIJAVI SE"
class="inputi"/></td>
                </tr>
            </table>
            <br/>
            <a href="index.php?page=register">Niste registrovani?</a><br/>
            <p id="zab_loz_p">Zaboravili ste lozinku?</p>
        </form>
    </div>
</div>

<div class="greske">
</div>

<div id="zab_loz_div">
    <table id="resetPassEmailTabela">
        <tr>
            <td><h3>- Resetovanje lozinke -</h3></td>
        </tr>
        <tr>
            <td>Unesite email</td>
        </tr>
        <tr>
            <td>
                <input type="text" id="resetPassEmail" name="resetPassEmail" placeholder=" "
autocomplete="on" class="inputi"/>
            </td>
        </tr>
        <tr>
            <td>
                <div id="resetPassEmailGreska" class="ispisGreske"></div>
            </td>
        </tr>
        <tr>
            <td>
```

```
        <input type="button" id="resetPassEmailDugme" name="resetPassEmailDugme"
value="POTVRDI EMAIL" class="inputi"/>
    </td>
</tr>

</table>
</div>

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

### 3.1.6) register

```
<div class="naslov_svaki">
    <p>Registrujte se</p>
</div>

<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="forma_register" name="forma_register">
            <table id="tabela_register">
                <tr>
                    <td>Ime</td>
                    <td>Prezime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regIme" name="regIme" placeholder=""
autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regPrezime" name="regPrezime" placeholder=""
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="regImeGreska" class="ispisGreske"></div>
                    </td>
                    <td>
                        <div id="regPrezimeGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Email</td>
                    <td>Korisničko ime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regEmail" name="regEmail" placeholder=""
autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regKorIme" name="regKorIme" placeholder=""
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="regEmailGreska" class="ispisGreske"></div>
                    </td>
                    <td>
                        <div id="regKorImeGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Lozinka</td>
                    <td>Ponovite lozinku</td>
                </tr>
                <tr>
                    <td><input type="password" id="regLozinka" name="regLozinka" placeholder=""
autocomplete="on" class="inputi"/></td>
                    <td><input type="password" id="regLozinka2" name="regLozinka2" placeholder=""
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="regLozinkaGreska" class="ispisGreske"></div>
                    </td>
                    <td>
                        <div id="regLozinka2Greska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td colspan="2"><input type="button" id="regDugme" name="regDugme"
value="REGISTRUIJ SE" class="inputi"/></td>
                </tr>
            </table>
        </form>
```

```
    </div>
</div>

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



### 3.1.7) shop

```
<?php
if(!isset($_SESSION['korisnik'])){
    header("Location: https://luggage-online.000webhostapp.com/index.php");
} else{
    $korisnikID = $_SESSION['korisnik']->korisnikID;
    $uloga = $_SESSION['korisnik']->uloga;

    $korpaObj = new Korpa(Baza::instanca());
    $korpaObj->setKorisnikID($korisnikID);
    $korpa = "";

    switch ($uloga)
    {
        case "Korisnik" : $korpa = $korpaObj->dohvatiNekupljenoKorisnik(); break;
        case "Admin" : $korpa = $korpaObj->dohvatiNekupljenoAdmin(); break;
    }
}

if(!isset($_SESSION['korisnik'])){
    header("Location: https://luggage-online.000webhostapp.com/index.php");
} else{
    $korisnikID2 = $_SESSION['korisnik']->korisnikID;
    $uloga2 = $_SESSION['korisnik']->uloga;

    $korpaObj2 = new Korpa(Baza::instanca());
    $korpaObj2->setKorisnikID($korisnikID2);
    $korpa2 = "";

    switch ($uloga2)
    {
        case "Korisnik" : $korpa2 = $korpaObj2->dohvatiKupljenoKorisnik(); break;
        case "Admin" : $korpa2 = $korpaObj2->dohvatiKupljenoAdmin(); break;
    }
}

$korisnikID3 = $_SESSION['korisnik']->korisnikID;
$korpaObj3 = new Korpa(Baza::instanca());
$korpaObj3->setKorisnikID($korisnikID2);
$suma = $korpaObj3->dohvatiSumuNekupljenihProizvoda();

?>

<div class="naslov_svaki">
    <p>U korpi :</p>
</div>

<?php if(count($korpa)): ?>
    <div id="shop_sadrzaj">
        <table class="tabela_shop">
            <tr>
                <th>SLIKA</th>
                <th>NAZIV</th>
                <th>CENA</th>
                <?php if($uloga == "Korisnik"): ?>
                    <th>IZBACI IZ KORPE</th>
                <?php endif;?>
            </tr>

            <?php foreach ($korpa as $k): ?>
                <tr>
                    <td><a href="index.php?page=product&id=<?= $k->proizvodID ?>">alt ?>" class="malaSlika"/><a/></td>
```

```

        <td><?= $k->proizvod ?></td>
        <td><?= $k->cena ?> din.</td>
        <?php if($uloga == "Korisnik"): ?>
            <td><a href="#" data-id="<?= $k->proizvodID ?>" class="izbaciIzKorpe
obrisi">Izbaci iz korpe</a></td>
        <?php endif; ?>
        <input type="hidden" name="idKorIzbaci" id="idKorIzbaci" value="<?= $k-
>korisnikID ?>"/>
    </tr>
    <?php endforeach; ?>
    <?php if($uloga == "Korisnik"): ?>
        <tr>
            <td colspan="4">
                <h3>Ukupno: <?= $suma->suma ?> din.</h3>
            </td>
        </tr>
        <tr>
            <td colspan="4">
                <a href="#" data-id="<?= $k->korisnikID ?>" class="kupiDugme">KUPI</a>
            </td>
        </tr>
    <?php endif; ?>
</table>
</div>
<?php else: ?>
    <div class="naslov_svaki">
        <?php if($uloga == "Korisnik"): ?>
            <p>Vaša korpa je prazna</p><br/>
        <?php else: ?>
            <p>Korpa je prazna</p><br/>
        <?php endif; ?>
        
        
        <br/><br/>
    </div>
<?php endif; ?>

<div class="naslov_svaki">
    <p>Kupljeno :</p>
</div>

<?php if(count($korpa2)): ?>
    <div id="shop_sadrzaj2">
        <table class="tabela_shop">
            <tr>
                <th>SLIKA</th>
                <th>NAZIV</th>
                <th>CENA</th>
            </tr>

            <?php foreach ($korpa2 as $k2): ?>
                <tr>
                    <td><a href="index.php?page=product&id=<?= $k2->proizvodID ?>">alt ?>" class="malaSlika"/></a></td>
                    <td><?= $k2->proizvod ?></td>
                    <td><?= $k2->cena ?> din.</td>
                </tr>
            <?php endforeach; ?>
        </table>
    </div>
<?php else: ?>
    <div class="naslov_svaki">
        <?php if($uloga == "Korisnik"): ?>
            <p>Ništa niste kupili za sada</p><br/>
        <?php else: ?>
            <p>Ništa nije kupljeno za sada</p><br/>
        <?php endif; ?>
    </div>
<?php endif; ?>

```

### 3.1.8) author

```
<div id="author_naslov">
  <p>0 Autoru</p>
</div>

<div id="author_sadrzaj">
  <div id="author_drzac">
    
    <div id="author_tekst">
      Zovem se Mladen Bošković. Rođen sam u Kruševcu 1995. godine, gde sam završio osnovnu i
srednju školu. Studiram na Visokoj ICT školi, smer Internet tehnologije.
    </div>
  </div>
</div>
```

**3.1.9.0)** Ovde počinju kodovi admin stranica. Prvo slede kodovi za **admin** i **adminmenu** bez kojih naredni kodovi ne bi bili funkcionalni.

#### - admin

```
<?php
if(!isset($_SESSION['korisnik'])){
    header("Location: https://luggage-online.000webhostapp.com/index.php");
} else{
    if($_SESSION['korisnik']->uloga != "Admin"){
        header("Location: https://luggage-online.000webhostapp.com/index.php");
    }
}

$adminPage = null;
if(isset($_GET['adminaction'])) {
    $adminPage = $_GET['adminaction'];
}

switch ($adminPage){
    case "adduser" : include "adminmenu.php"; include "adduser.php"; break;
    case "allusers" : include "adminmenu.php"; include "allusers.php"; break;
    case "updateuser" : include "adminmenu.php"; include "updateuser.php"; break;
    case "allproducts" : include "adminmenu.php"; include "allproducts.php"; break;
    case "addproduct" : include "adminmenu.php"; include "addproduct.php"; break;
    case "updateproduct" : include "adminmenu.php"; include "updateproduct.php"; break;
    default : include "adminhome.php"; break;
}
```

#### - adminmenu

```
<div id="admin_meni">
    <h1>Admin Meni</h1>
</div>

<div id="admin_sadrzaj2">
    <div id="admin_drzac2">
        <ul>
            <li>
                <a href="index.php?page=admin&adminaction=addproduct">DODAJ PROIZVOD</a>
            </li>
            <li>
                <a href="index.php?page=admin&adminaction=allproducts">SVI PROIZVODI</a>
            </li>
            <li>
                <a href="index.php?page=admin&adminaction=adduser">DODAJ KORISNIKA</a>
            </li>
            <li>
                <a href="index.php?page=admin&adminaction=allusers">SVI KORISNICI</a>
            </li>
        </ul>
    </div>
</div>
```

### 3.1.9.1) adminhome

```
<div id="admin_meni">
  <h1>Admin Meni</h1>
</div>

<div id="admin_sadrzaj">
  <div id="admin_drzac">
    <ul>
      <li>
        <a href="index.php?page=admin&adminaction=addproduct">DODAJ PROIZVOD</a>
      </li>
      <li>
        <a href="index.php?page=admin&adminaction=allproducts">SVI PROIZVODI</a>
      </li>
      <li>
        <a href="index.php?page=admin&adminaction=adduser">DODAJ KORISNIKA</a>
      </li>
      <li>
        <a href="index.php?page=admin&adminaction=allusers">SVI KORISNICI</a>
      </li>
    </ul>
  </div>
</div>
```

### 3.1.9.2) admin&adminaction=addproduct

```
<?php
$brendObj = new Brend(Baza::instanca());
$rezultatBrend = $brendObj->dohvatiSve();

$kategoriyaObj = new Kategorija(Baza::instanca());
$rezultatKategorija = $kategoriyaObj->dohvatiSve();
?>

<div class="naslov_svaki_admin">
    <p>Dodajte Proizvod</p>
</div>

<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="formaProizvod" name="formaProizvod" enctype="multipart/form-data" method="post"
action="modules/admin.php">
            <table>
                <tr>
                    <td colspan="3">
                        <p>Naziv</p>
                        <textarea id="insertProizvodNaziv" name="insertProizvodNaziv"
maxlength="50"></textarea>
                    </td>
                </tr>
                <tr>
                    <td colspan="3">
                        <p>Opis</p>
                        <textarea id="insertProizvodOpis" name="insertProizvodOpis"
maxlength="200"></textarea>
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <p>Slika</p>
                        <input type="file" id="insertProizvodSlika" name="insertProizvodSlika"/>
                    </td>
                    <td>
                        <p id="formaProizvodCena_t">Cena</p><br/>
                        <input type="number" id="insertProizvodCena" name="insertProizvodCena"/>
                    </td>
                </tr>
                <tr>
                    <td>
                        <p>Brend</p>
                        <select id="insertProizvodBrend" name="insertProizvodBrend">
                            <option value="0">Izaberite</option>
                            <?php foreach ($rezultatBrend as $brend): ?>
                                <option value="<?= $brend->brendID ?>"><?= $brend->brend ?></option>
                            <?php endforeach; ?>
                        </select>
                    </td>
                    <td>
                        <p id="formaProizvodPol_t">Pol</p><br/>
                        <select id="insertProizvodPol" name="insertProizvodPol">
                            <option value="0">Izaberite</option>
                            <option value="m">Muški</option>
                            <option value="z">Ženski</option>
                        </select>
                    </td>
                    <td>
                        <p id="formaProizvodKategorija_t">Kategorija</p><br/>
                        <select id="insertProizvodKategorija" name="insertProizvodKategorija">
                            <option value="0">Izaberite</option>
                            <?php foreach ($rezultatKategorija as $kategoriya): ?>
                                <option value="<?= $kategoriya->kategorijaID ?>"><?= $kategoriya->
```

```
>kategorija ?></option>
        <?php endforeach; ?>
    </select>
</td>
</tr>
<tr>
    <td colspan="3"><input type="submit" value="DODAJ PROIZVOD"
id="addProductDugme" name="addProductDugme" /></td>
</tr>
</table>
</form>
</div>
</div>

<div class="admin_greske">

</div>
```

### 3.1.9.3) admin&adminaction=allproducts

```
<?php
$proizvodObj = new Proizvod(Baza::instanca());
$rezultatAllProducts = $proizvodObj->dohvatiSve();

?>

<div class="naslov_svaki_admin">
    <p>Svi Proizvodi</p>
</div>

<?php if(count($rezultatAllProducts)): ?>
<div id="allproducts_sadrzaj">
    <table class="tabela">
        <tr>
            <th>SLIKA</th>
            <th>NAZIV</th>
            <th>OPIS</th>
            <th>CENA</th>
            <th>POL</th>
            <th>BREND</th>
            <th>KATEGORIJA</th>
            <th>IZMENI</th>
            <th>OBRIŠI</th>
        </tr>
        <?php foreach ($rezultatAllProducts as $allproducts): ?>
            <tr>
                <td><a href="index.php?page=product&id=<?= $allproducts->proizvodID ?>">alt)
?>" class="malaSlika"/></a></td>
                <td><?= stripslashes($allproducts->proizvod) ?></td>
                <td><?= stripslashes($allproducts->opis) ?></td>
                <td><?= $allproducts->cena ?> din.</td>
                <td><?= $allproducts->pol == 'm' ? "Muški" : 'Ženski'; ?></td>
                <td><?= $allproducts->brend ?></td>
                <td><?= $allproducts->kategorija ?></td>
                <td><a href="index.php?page=admin&adminaction=updateproduct&idupdate=<?=
$allproducts->proizvodID ?>" class="detaljnijeD">Izmeni</a></td>
                <td><a href="#" data-id="<?= $allproducts->proizvodID ?>" class="obrisiProizvod
obrisi">Obriši</a></td>
            </tr>
        <?php endforeach; ?>
    </table>
</div>
<?php else: ?>
    <div class="naslov_svaki_admin">
        <p>Trenutno nema proizvoda</p>
    </div>
<?php endif; ?>
```



### 3.1.9.4) admin&adminaction=updateproduct&idupdate='x'

```
<?php
$brendObj = new Brend(Baza::instanca());
$rezultatBrend = $brendObj->dohvatiSve();

$kategoriyaObj = new Kategorija(Baza::instanca());
$rezultatKategorija = $kategoriyaObj->dohvatiSve();

$idUpdateP = $_GET['idupdate'];
$proizvodObj = new Proizvod(Baza::instanca());
$proizvodObj->setProizvodID($idUpdateP);
$rezultatProizvod = $proizvodObj->dohvatiJedan();
?>

<div class="naslov_svaki_admin">
    <p>Izmenite Proizvod</p>
</div>

<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="formaProizvod" name="formaProizvod" enctype="multipart/form-data" method="post"
action="modules/admin.php">
            <table>
                <tr>
                    <td colspan="3">
                        <a href="#" data-id="<?= $rezultatProizvod->proizvodID ?>"
class="obrisiProizvod obrisi">Obriši</a>
                    </td>
                </tr>
                <tr>
                    <td colspan="2" rowspan="2">alt) ?>"/></td>
                    <td>
                        <p>Naziv</p>
                        <textarea id="updateProizvodNaziv" name="updateProizvodNaziv"
maxlength="50"><?= stripslashes($rezultatProizvod->proizvod) ?></textarea>
                    </td>
                </tr>
                <tr>
                    <td>
                        <p>Opis</p>
                        <textarea id="updateProizvodOpis" name="updateProizvodOpis"
maxlength="200"><?= stripslashes($rezultatProizvod->opis) ?></textarea>
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <p>Slika</p>
                        <input type="file" id="insertProizvodSlika" name="insertProizvodSlika"/>
                    </td>
                    <td>
                        <p id="formaProizvodCena_t">Cena</p><br/>
                        <input type="number" id="insertProizvodCena" name="insertProizvodCena"
value="<?= $rezultatProizvod->cena ?>"/>
                    </td>
                </tr>
                <tr>
                    <td>
                        <p>Brend</p>
                        <select id="insertProizvodBrend" name="insertProizvodBrend">
                            <option value="0">Izaberite</option>
                            <?php foreach ($rezultatBrend as $brend): ?>
                                <?php if($rezultatProizvod->brendID == $brend->brendID): ?>
                                    <option value="<?= $brend->brendID ?>" selected><?= $brend-
>brend ?></option>
                                <?php else: ?>
                                    <option value="<?= $brend->brendID ?>"><?= $brend->brend
?></option>
                            </select>
                    </td>
                </tr>
            </table>
        </form>
    </div>
</div>
```

```

        <?php endif; ?>
    <?php endforeach; ?>
</select>
</td>
<td>
    <p id="formaProizvodPol_t">Pol</p><br/>
    <select id="insertProizvodPol" name="insertProizvodPol">
        <?php
            $nizOptionP = ["Izaberite"=>"0", "Muški"=>"m", "Ženski"=>"z"];
            foreach ($nizOptionP as $in => $vr): ?>
                <?php if($rezultatProizvod->pol == $vr): ?>
                    <option value="<?=$vr ?>" selected><?=$in ?></option>
                <?php else: ?>
                    <option value="<?=$vr ?>"><?=$in ?></option>
                <?php endif; ?>
            <?php endforeach; ?>
        </select>
    </td>
<td>
    <p id="formaProizvodKategorija_t">Kategorija</p><br/>
    <select id="insertProizvodKategorija" name="insertProizvodKategorija">
        <option value="0">Izaberite</option>
        <?php foreach ($rezultatKategorija as $kategorija): ?>
            <?php if($rezultatProizvod->kategorijaID == $kategorija-
>kategorijaID): ?>
                <option value="<?=$kategorija->kategorijaID ?>" selected><?=$
$kategorija->kategorija ?></option>
            <?php else: ?>
                <option value="<?=$kategorija->kategorijaID ?>"><?=$
$kategorija->kategorija ?></option>
            <?php endif; ?>
        <?php endforeach; ?>
    </select>
</td>
</tr>
<tr>
    <td colspan="3"><input type="submit" value="IZMENI PROIZVOD"
id="updateProductDugme" name="updateProductDugme"/></td>
</tr>
</table>
    <input type="hidden" value="<?=$idUpdateP ?>" name="updatePrID" id="updatePrID"/>
</form>
</div>
</div>

<div class="admin_greske">

</div>

```

### 3.1.9.5) admin&adminaction=adduser

```
<?php
$ulogaObj = new Uloga(Baza::instanca());
$rezultatUloga = $ulogaObj->dohvatiSve();
?>

<div class="naslov_svaki_admin">
    <p>Dodajte Korisnika</p>
</div>

<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="forma_register" name="forma_register">
            <table id="tabela_register">
                <tr>
                    <td>Ime</td>
                    <td>Prezime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regIme" name="regIme" placeholder=""
autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regPrezime" name="regPrezime" placeholder=""
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>Email</td>
                    <td>Korisničko ime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regEmail" name="regEmail" placeholder=""
autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regKorIme" name="regKorIme" placeholder=""
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>Lozinka</td>
                    <td>Ponovite lozinku</td>
                </tr>
                <tr>
                    <td><input type="password" id="regLozinka" name="regLozinka" placeholder=""
autocomplete="on" class="inputi"/></td>
                    <td><input type="password" id="regLozinka2" name="regLozinka2" placeholder=""
autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>Uloga</td>
                    <td id="add_aktivan_t">Aktivan</td>
                </tr>
                <tr>
                    <td>
                        <select id="add_uloga" name="add_uloga">
                            <option value="0">Izaberite</option>
                            <?php foreach($rezultatUloga as $uloga): ?>
                                <option value="<?= $uloga->ulogaID ?>"><?= $uloga->uloga ?></option>
                            <?php endforeach; ?>
                        </select>
                    </td>
                    <td>
                        <select id="add_aktivan" name="add_aktivan">
                            <option value="2">Izaberite</option>
                            <option value="1">Da</option>
                            <option value="0">Ne</option>
                        </select>
                    </td>
                </tr>
                <tr>
                    <td colspan="2"><input type="button" id="addUserDugme" name="addUserDugme"

```

```
value="DODAJ KORISNIKA" class="inputi"/></td>
      </tr>

    </table>
  </form>
</div>
</div>

<div class="admin_greske">

</div>
```

### 3.1.9.6) admin&adminaction=allusers

```
<?php
$korisnikObj = new Korisnik(Baza::instanca());
$rezultatAllUsers = $korisnikObj->dohvatiSve();

?>

<div class="naslov_svaki_admin">
    <p>Svi Korisnici</p>
</div>

<?php if(count($rezultatAllUsers): ?>
<div id="allusers_sadrzaj">
    <table class="tabela">
        <tr>
            <th>IME</th>
            <th>PREZIME</th>
            <th>EMAIL</th>
            <th>KORISNIČKO IME</th>
            <th>ULOGA</th>
            <th>AKTIVAN</th>
            <th>IZMENI</th>
            <th>OBRIŠI</th>
        </tr>
        <?php foreach ($rezultatAllUsers as $allusers): ?>
            <tr>
                <td><?= stripslashes($allusers->ime) ?></td>
                <td><?= stripslashes($allusers->prezime) ?></td>
                <td><?= stripslashes($allusers->email) ?></td>
                <td><?= stripslashes($allusers->korisnicko_ime) ?></td>
                <td><?= $allusers->uloga ?></td>
                <td><?= $allusers->aktivan == 0 ? "Ne" : "Da"; ?></td>
                <td><a href="index.php?page=admin&adminaction=updateuser&idupdate=<?= $allusers->korisnikID ?>" class="detaljnijeD">Izmeni</a></td>
                <td><a href="#" data-id="<?= $allusers->korisnikID ?>" class="obrisiKorisnika
obrisi">Obriši</a></td>
            </tr>
        <?php endforeach; ?>
    </table>
</div>
<?php else: ?>
    <div class="naslov_svaki_admin">
        <p>Trenutno nema korisnika</p>
    </div>
<?php endif; ?>
```

### 3.1.9.7) admin&adminaction=updateuser&idupdate='x'

```
<?php
$ulogaObj = new Uloga(Baza::instanca());
$rezultatUloga = $ulogaObj->dohvatiSve();

$idUpdate = $_GET['idupdate'];
$korisnikObj = new Korisnik(Baza::instanca());
$korisnikObj->setKorisnikID($idUpdate);
$rezultatUpdateUser = $korisnikObj->dohvatiJednog();
?>

<div class="naslov_svaki_admin">
    <p>Izmenite Korisnika</p>
</div>

<div id="register_sadrzaj">
    <div id="register_drzac">
        <form id="forma_register" name="forma_register">
            <table id="tabela_register">
                <tr>
                    <td colspan="2">
                        <a href="#" data-id="<?=$rezultatUpdateUser->korisnikID ?>"
class="obrisiKorisnika obrisi">Obriši</a>
                    </td>
                </tr>
                <tr>
                    <td>Ime</td>
                    <td>Prezime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regIme" name="regIme" value="<?=$stripslashes($rezultatUpdateUser->ime); ?>" placeholder="" autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regPrezime" name="regPrezime" value="<?=$stripslashes($rezultatUpdateUser->prezime); ?>" placeholder="" autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>Email</td>
                    <td>Korisničko ime</td>
                </tr>
                <tr>
                    <td><input type="text" id="regEmail" name="regEmail" value="<?=$stripslashes($rezultatUpdateUser->email); ?>" placeholder="" autocomplete="on" class="inputi"/></td>
                    <td><input type="text" id="regKorIme" name="regKorIme" value="<?=$stripslashes($rezultatUpdateUser->korisnicko_ime); ?>" placeholder="" autocomplete="on" class="inputi"/></td>
                </tr>
                <tr>
                    <td>Uloga</td>
                    <td id="add_aktivan_t">Aktivan</td>
                </tr>
                <tr>
                    <td>
                        <select id="add_uloga" name="add_uloga">
                            <option value="0">Izaberite</option>
                            <?php foreach($rezultatUloga as $uloga): ?>
                                <?php if($rezultatUpdateUser->ulogaID == $uloga->ulogaID): ?>
                                    <option value="<?=$uloga->ulogaID ?>" selected><?=$uloga->uloga
                                ?></option>
                                <?php else: ?>
                                    <option value="<?=$uloga->ulogaID ?>"><?=$uloga->uloga
                                ?></option>
                                <?php endif; ?>
                            <?php endforeach; ?>
                        </select>
                    </td>
                </tr>
            </table>
        </form>
    </div>
</div>
```

```

        <td>
            <select id="add_aktivan" name="add_aktivan">
                <?php
                    $nizOption = ['Izaberite', 'Da', 'Ne'];
                    $br = 2;
                    foreach($nizOption as $option): ?>
                        <?php if($rezultatUpdateUser->aktivan == $br): ?>
                            <option value="<?= $br ?>" selected><?= $option ?></option>
                        <?php else: ?>
                            <option value="<?= $br ?>"><?= $option ?></option>
                        <?php endif; ?>
                        <?php $br--;
                    endforeach; ?>
            </select>
        </td>
    </tr>
    <tr>
        <td colspan="2"><input type="button" id="updateUserDugme"
name="updateUserDugme" value="IZMENI KORISNIKA" class="inputi"/></td>
    </tr>
</table>
<input type="hidden" value="<?= $idUpdate ?>" name="idUpdate" id="idUpdate"/>
</form>
</div>
</div>
<div class="admin_greske">
</div>

```

### 3.1.10) 404

```
<div id="error404">
  <p>Stranica nije pronađena</p>
  
</div>
```



### 3.1.11) userprofile

```
<?php
if(!isset($_SESSION['korisnik'])) {
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}

$id = $_SESSION['korisnik']->korisnikID;
$korisnikObj = new Korisnik(Baza::instanca());
$korisnikObj->setKorisnikID($id);
$korisnik = $korisnikObj->dohvatiJednog();
?>

<div class="naslov_svaki">
    <p>Vaš Profil</p>
</div>

<div id="profile_sadrzaj">
    <div id="profile_drzac">

        <div id="o_korisniku">
            <table>
                <tr>
                    <th colspan="2"> Podaci</th>
                </tr>
                <tr>
                    <td>Ime</td>
                    <td><?= stripslashes($korisnik->ime) ?></td>
                </tr>
                <tr>
                    <td>Prezime</td>
                    <td><?= stripslashes($korisnik->prezime) ?></td>
                </tr>
                <tr>
                    <td>Korisničko ime</td>
                    <td><?= stripslashes($korisnik->korisnicko_ime) ?></td>
                </tr>
                <tr>
                    <td>Uloga</td>
                    <td><?= $korisnik->uloga ?></td>
                </tr>
                <tr>
                    <td>Datum registracije</td>
                    <td><?= date("d.m.Y", $korisnik->datum_registracije) ?></td>
                </tr>
            </table>
        </div>

        <div id="promena_lozinke">
            <form id="formaPromenaLozinke" name="formaPromenaLozinke">
                <table id="tabela_promeni_lozinku">
                    <tr>
                        <td>Trenutna lozinka</td>
                    </tr>
                    <tr>
                        <td><input type="password" id="trenutnaLozinka" name="trenutnaLozinka"
placeholder="" class="inputi"/></td>
                    </tr>
                    <tr>
                        <td>
                            <div id="trenutnaLozinkaGreska" class="ispisGreske"></div>
                        </td>
                    </tr>
                    <tr>
                        <td>Nova lozinka</td>
                    </tr>
                    <tr>
                        <td><input type="password" id="novaLozinka" name="novaLozinka"
```

```

placeholder="" class="inputi"/></td>
    </tr>
    <tr>
        <td>
            <div id="novaLozinkaGreska" class="ispisGreske"></div>
        </td>
    </tr>
    <tr>
        <td>Ponovite novu lozinku</td>
    </tr>
    <tr>
        <td><input type="password" id="novaLozinka2" name="novaLozinka2"
placeholder="" class="inputi"/></td>
    </tr>
    <tr>
        <td>
            <div id="novaLozinkaGreska2" class="ispisGreske"></div>
        </td>
    </tr>
    <tr>
        <td><input type="button" id="promenaLozinkeDugme"
name="promenaLozinkeDugme" value="PROMENI LOZNKU" class="inputi"/></td>
    </tr>
</table>
<input type="hidden" id="promenaLozinkeKorisnik" value="<?= $korisnik->korisnikID
?>"/>

</form>

<div id="promenaLozinkeGreske">
</div>
</div>
</div>
</div>

```

### 3.1.12) resetpassword

```
<?php
if(!isset($_GET['reset'])){
    header("Location: https://luggage-online.000webhostapp.com/index.php");
} else{
    $token = $_GET['reset'];
    $korisnik = new Korisnik(Baza::instanca());
    $korisnik->setResetPassword($token);

    $korisnikSelect = $korisnik->resetLozinkeTokenProvera();
    if(!$korisnikSelect){
        header('Location: https://luggage-online.000webhostapp.com/index.php');
    }
}
?>

<div class="naslov_svaki">
    <p>Resetujte Lozinku</p>
</div>

<div id="reset_lozinke_sadrzaj">
    <div id="reset_lozinke_drzac">
        <form id="formaPromenaLozinke" name="formaPromenaLozinke">
            <table id="tabela_promeni_lozinku">
                <tr>
                    <td>Nova lozinka</td>
                </tr>
                <tr>
                    <td><input type="password" id="novaLozinka" name="novaLozinka" placeholder=" "
class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="novaLozinkaGreska" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td>Ponovite novu lozinku</td>
                </tr>
                <tr>
                    <td><input type="password" id="novaLozinka2" name="novaLozinka2" placeholder=" "
class="inputi"/></td>
                </tr>
                <tr>
                    <td>
                        <div id="novaLozinkaGreska2" class="ispisGreske"></div>
                    </td>
                </tr>
                <tr>
                    <td><input type="button" id="resetLozinkeDugme" name="resetLozinkeDugme"
value="PROMENI LOZINKU" class="inputi"/></td>
                </tr>
            </table>
            <input type="hidden" id="promenaLozinkeKorisnikToken" value="<?= $token ?>"/>
        </form>
    </div>

    <div id="resetLozinkeGreske">
    </div>

</div>
```

### 3.2) CSS – style.css

```
/*-----HEAD-----*/
@import url('https://fonts.googleapis.com/css?family=Mukta');
body{
  font-family: 'Mukta', sans-serif;
  color: #25262a;
}
*{
  margin: 0px;
  padding: 0px;
  box-sizing: border-box;
}
ul{
  list-style-type: none;
}
a{
  text-decoration: none;
}

/*-----HEAD-----*/

/*-----FOOTER-----*/
#footer{
  width: 100%;
  height: 200px;
  background-color: #020B13;
  text-align: center;
}

#footer ul li{
  display: inline-block;
  font-size: 25px;
  padding: 10px;
}

#footer ul li a:link{
  color: #A9A9A9;
}

#footer ul li a:visited{
  color: #A9A9A9;
}

#footer ul li a:hover{
  color: #F5F5F5;
}

#footer p{
  color: #A9A9A9;
  font-size: 15px;
}

#footer ul{
  padding-top: 50px;
  padding-bottom: 25px;
}

/*-----FOOTER-----*/

/*-----NAV-----*/
#beforenav{
  width: 100%;
  height: 125px;
  display: flex;
  align-items: center;
  justify-content: center;
```

```

}

#beforenav_drzac{
    margin: 0px auto;
    width: 1300px;
    display: flex;
    align-items: center;
    justify-content: space-around;
}

#beforenav a span{
    font-size: 25px;
    color: #25262a;
}

#beforenav a span:hover{
    color: #7971ea;
}

/*-----NAV-----*/

/*-----NAV 222-----*/
#nav_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    align-items: center;
    padding-bottom: 5px;
    padding-top: 5px;
}

#nav2{
    width: 100%;
    height: 65px;

    border-top: 1px solid #F5F5F5;
    border-bottom: 1px solid #F5F5F5;
}

#nav{
    margin-left: 10px;
}

#nav ul{
    font-size: 0;
    margin: 0;
    padding: 0;
}

#nav ul li{
    display: inline-block;
    position: relative;
}

#nav ul li a{
    color: #25262a;
    display: block;
    font-size: 15px;
    padding: 15px 14px;
    transition: 0.3s linear;
    letter-spacing: 2px;
}

#nav ul li a:hover{
    background: #F5F5F5;
}

#nav ul li ul{
    border-top: 3px solid #7971ea;
}

```

```

display: none;
position: absolute;
width: 175px;
}

#nav ul li ul li:last-child{
border-bottom: 3px solid #7971ea;
}

#nav ul li ul li{
border-top: 1px solid #F5F5F5;
display: block;
}

#nav ul li ul li:first-child{
border-top: none;
}

#nav ul li ul li a{
background: white;
display: block;
padding: 10px 14px;
color: #7971ea;
}

#nav ul li ul li a:hover{
background: #F5F5F5;
color: black;
}

#adminLink{
position: fixed;
right: 0px;
top: 0px;
font-weight: bold;
font-size: 16px;
//color: #25262a;
color: white;
padding: 5px 10px;
//background-color: palegreen;
background-color: #25262a;
border-bottom-left-radius: 5px;
letter-spacing: 1px;
}

#adminLink:hover{
background-color: #7971ea;
}

/*-----NAV 222-----*/

/*-----REGISTER-----*/
#register_sadrzaj{
width: 100%;
background-color: #F5F5F5;
}

#register_drzac{
width: 1300px;
margin: 0px auto;
display: flex;
justify-content: center;
padding-top: 75px;
}

#tabela_register tr:last-child{
text-align: center;
}

```

```
#tabela_register tr td{
    padding-left: 25px;
    padding-right: 25px;
}

#tabela_register tr:last-child td{
    padding-top: 10px;
}

#regDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
}

#forma_register{
    padding-bottom: 40px;
}

.ispisGreske{
    width: 300px;
    height: 20px;
    margin-bottom: 15px;
    color: red;
    font-size: 15px;
}

/*-----REGISTER-----*/

/*-----LOGIN-----*/

#login_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#login_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    padding-top: 75px;
}

#tabela_login tr:last-child td{
    padding-top: 10px;
}

#loginDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
}

#login_greske{
    width: 100%;
    text-align: center;
    background-color: #F5F5F5;
}

#forma_login{
    padding-bottom: 40px;
}

#login_drzac a:link{
    color: blue;
}

#login_drzac a:visited{
    color: blue;
}
```

```

}

#login_drzac a:hover{
    color: red;
}

#zab_loz_p{
    color: blue;
}

#zab_loz_div{
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #F5F5F5;
}

#zab_loz_p:hover{
    color: red;
}

#resetPassEmailIspis{
    height: 50px;
    text-align: center;
    background-color: #F5F5F5;
}

#resetPassEmailDugme{
    font-weight: bold;
    letter-spacing: 1px;
}

#resetPassEmailTabela{
    margin-bottom: 30px;
    padding-top: 30px;
}

#resetPassEmailTabela tr:first-child td {
    text-align: center;
    padding-bottom: 20px;
}

/*-----LOGIN-----*/

/*-----CONTACT-----*/
#contact_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#contact_drzac{
    margin: 0px auto;
    width: 1300px;
    display: flex;
    justify-content: space-around;
    padding-top: 75px;
}

#contact_drzac ul li{
    padding: 3px;
}

#contact_drzac ul li span{
    color: #7971ea;
}

#contactEmail{
    font-family: 'Mukta', sans-serif;

```



```

border: 1px solid #A9A9A9;
text-indent: 5px;
width: 400px;
height: 35px;
font-size: 15px;
}

#poruka{
  font-family: 'Mukta', sans-serif;
  height: 150px;
  font-size: 15px;
  resize: vertical;
  min-height: 100px;
  max-height: 400px;
  border: 1px solid #A9A9A9;
  text-indent: 5px;
  width: 400px;
}

#contactDugme{
  font-family: 'Mukta', sans-serif;
  border: 1px solid #A9A9A9;
  width: 400px;
  height: 35px;
  font-size: 15px;
  font-weight: bold;
  margin-top: 10px;
  letter-spacing: 1px;
}

#forma_contact{
  padding-bottom: 10px;
}

.ispisGreskeContact{
  width: 400px;
  height: 20px;
  margin-bottom: 15px;
  color: red;
  font-size: 15px;
}

#contact_greske{
  text-align: center;
  background-color: #F5F5F5;
}

#contact_greske h4{
  padding-bottom: 5px;
}

#contact_greske h2{
  padding-bottom: 5px;
}

.contactSpan{
  width: 275px;
  height: 2px;
  background-color: #7971ea;
  display: block;
}

/*---anketa---*/
#kontakt_anketa{
  height: 550px;
  width: 400px;
}

#ankete h3{
  margin-top: 75px;
}

```

```

    margin-bottom: -10px;
}

#selectAnketa{
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    height: 35px;
    border: 1px solid #A9A9A9;
    margin-bottom: 10px;
}

#odgovori{
    background-color: #F5F5F5;
    margin-top: 10px;
}

#odgovori ul li{
    padding: 3px;
}

#glasaj{
    font-family: 'Mukta', sans-serif;
    border: 1px solid #A9A9A9;
    width: 100px;
    height: 35px;
    font-size: 15px;
    margin-top: 10px;
    margin-bottom: 30px;
    letter-spacing: 1px;
}

#glasajDugmeDiv{
    background-color: #F5F5F5;
}

#glasajDugmeDiv p{
    display: inline-block;
}

#rezultatiDugme{
    font-family: 'Mukta', sans-serif;
    border: 1px solid #A9A9A9;
    width: 100px;
    height: 35px;
    font-size: 15px;
    letter-spacing: 1px;
}

#odgovori h4{
    margin-bottom: 5px;
}

#odgovori ul li input{
    margin-right: 5px;
}

#izaberiteOdgovor{
    background-color: #F5F5F5;
}

#izaberiteOdgovor h4{
    margin-bottom: 5px;
}

#izaberiteOdgovor p{
    margin-bottom: 5px;
}

/*---anketa---*/
/*-----CONTACT-----*/

```

```
/*-----HOME-----*/
#slajder{
    width: 1200px;
    margin: 0px auto;
}

#slajder img{
    display: none;
}

#slajder .trenutna{
    display: block;
}

#proizvodi_home{
    width: 100%;
    background-color: #F5F5F5;
}

#proizvodi_drzac_home{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: space-around;
    flex-wrap: wrap;
    padding-top: 20px;
    padding-bottom: 20px;
}

#pogodnosti{
    width: 100%;
    border-bottom: 1px solid #F5F5F5;
}

#pogodnosti_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 250px;
}

#pogodnosti_drzac table{
    width: 1200px;
}

#pogodnosti_drzac table tr td{
    padding-left: 10px;
    padding-right: 10px;
}

#pogodnosti_drzac table tr td a span{
    font-size: 50px;
    color: #7971ea;
}

.pogodnosti_naslov{
    font-size: 18px;
    letter-spacing: 1px;
}

.pogodnosti_tekst{
    color: #8c92a0;          /* boja za svetliji tekst */
    font-size: 16px;
}

/*-----HOME-----*/
```

```

/*-----ZAJEDNICKO-----*/
.naslov_svaki{
    width: 100%;
    text-align: center;
    font-size: 32px;
}

.naslov_svaki p{
    padding-top: 50px;
    padding-bottom: 50px;
}

.inputi{
    height: 35px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 300px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
}

.greske{
    width: 100%;
    text-align: center;
    background-color: #F5F5F5;
}

.greske h4{
    padding-bottom: 5px;
}

.greske h2{
    padding-bottom: 5px;
}
/*-----ZAJEDNICKO-----*/

/*-----CSS ZA PROIZVOD-----*/
.proizvod{
    width: 250px;
    height: 425px;
    text-align: center;
    border: 1px solid #DCDCDC;
    background-color: white;
    margin: 20px 11px 20px 11px;
    border-bottom-left-radius: 5px;
    border-bottom-right-radius: 5px;
    float: left;
}

.ps_naziv{
    height: 70px;
    font-size: 18px;
    margin: 5px 10px 5px 10px;
}

.ps_cena{
    height: 55px;
    font-size: 19px;
    font-weight: bold;
    color: #7971ea;;
}

.detaljni2{
    text-align: center;
}

```

```

}

.proizvod img{
  margin-top: 10px;
  margin-bottom: 15px;
  width: 200px;
  height: 200px;
}

.detaljnije{
  font-size: 15px;
  padding: 10px;
  background-color: #7971ea;
  display: inline-block;
  margin-bottom: 10px;
  border-radius: 5px;
  letter-spacing: 2px;
}

.detaljnije:link{
  color: white;
}

.detaljnije:visited{
  color: white;
}

.detaljnije:hover{
  color: #25262a;
}
/*-----CSS ZA PROIZVOD-----*/

/*-----PRODUCTS-----*/
#proizvodi_products{
  width: 100%;
  background-color: #F5F5F5;
  border-top: 2px solid white;
  border-bottom: 2px solid white;
}

#proizvodi_sadrzaj{
  padding: 30px 0px 30px 35px;
  width: 1130px;
}

#proizvodi_drzac{
  width: 1300px;
  margin: 0px auto;
  display: flex;
}

#proizvodi_filter{
  width: 170px;
  margin-top: 25px;
  margin-bottom: 25px;
}

#proizvodi_filter div{
  margin-bottom: 20px;
}

#proizvodi_filter h3{
  margin-bottom: 20px;
  margin-top: 20px;
}

.inputCenaFilter{

```

```

height: 25px;
font-family: 'Mukta', sans-serif;
font-size: 15px;
width: 150px;
border: 1px solid #A9A9A9;
text-indent: 5px;
margin-bottom: 5px;
}

#ponistiSve{
height: 35px;
font-family: 'Mukta', sans-serif;
font-size: 15px;
width: 150px;
border: 1px solid #A9A9A9;
}

#sortiraj{
height: 35px;
font-family: 'Mukta', sans-serif;
font-size: 15px;
width: 150px;
border: 1px solid #A9A9A9;
text-indent: 5px;
}

#sortirajDiv{
padding-bottom: 30px;
}

#nemaProizvoda{
margin: 300px 0px 0px 375px;
}

.stranice{
width: 100%;
text-align: center;
padding-top: 20px;
padding-bottom: 20px;
background-color: #F5F5F5;
height: 70px;
}

.stranice a{
padding: 5px 10px 5px 10px;
margin-left: 5px;
margin-right: 5px;
}

.stranice a:link{
color: #25262a;
}

.stranice a:visited{
color: #25262a;
}

.stranice a:hover{
color: #25262a;
background-color: #7971ea;
border-radius: 5px;
}

/*-----PRODUCTS-----*/

/*-----1 PRODUCT 1-----*/
#product_sadrzaj{
width: 100%;
background-color: #F5F5F5;

```

```
}

#product_drzac{
  width: 1300px;
  margin: 0px auto;
  display: flex;
  justify-content: center;
  align-items: center;
  padding-top: 100px;
  padding-bottom: 75px;
}

#product_drzac img{
  width: 350px;
  height: 350px;
}

#o_proizvodu{
  width: 400px;
  height: 350px;
  margin-left: 30px;
  //margin-top: 5px;
  margin-bottom: 25px;
}

#p_naziv{
  font-size: 21px;
  height: 90px;
  color: #7971ea;
}

#p_opis{
  text-align: justify;
  height: 120px;
}

#p_dodatno{
  height: 50px;
}

#p_cena{
  font-size: 23px;
  height: 50px;
  font-weight: bold;
  color: #7971ea;
}

.p_opcije{
  font-size: 15px;
  padding: 10px;
  background-color: #7971ea;
  display: inline-block;
  border-radius: 5px;
  letter-spacing: 2px;
}

.p_opcije:link{
  color: white;
}

.p_opcije:visited{
  color: white;
}

.p_opcije:hover{
  color: #25262a;
}

.p_opcije{
  float: right;
}
```

```
}
/*-----1 PRODUCT 1-----*/

/*-----SHOP-----*/
#shop_sadrzaj{
  width: 100%;
  display: flex;
  justify-content: center;
}

#shop_sadrzaj table tr td:last-of-type{
  text-align: center;
}

#shop_sadrzaj2{
  width: 100%;
  display: flex;
  justify-content: center;
}

.obrisi{
  font-size: 15px;
  padding: 10px;
  font-weight: bold;
  background-color: darkred;
  display: inline-block;
  letter-spacing: 1px;
  border-radius: 5px;
}

.obrisi:link{
  color: #A9A9A9;
}

.obrisi:visited{
  color: #A9A9A9;
}

.obrisi:hover{
  color: white;
}

.kupiDugme{
  font-size: 15px;
  padding: 10px 50px 10px 50px;
  font-weight: bold;
  background-color: darkolivegreen;
  display: inline-block;
  margin-bottom: 5px;
  margin-top: 5px;
  letter-spacing: 2px;
  border-radius: 5px;
}

.kupiDugme:link{
  color: #A9A9A9;
}

.kupiDugme:visited{
  color: #A9A9A9;
}

.kupiDugme:hover{
  color: white;
}

.korpaS{
  margin-left: 50px;
}
```



```

        margin-right: 50px;
    }

/*-----SHOP-----*/

/*-----AUTHOR-----*/
#author_naslov{
    width: 100%;
    text-align: center;
    background-color: #F5F5F5;
    font-size: 32px;
}

#author_naslov p{
    padding-top: 50px;
    padding-bottom: 50px;
}

#author_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#author_drzac{
    margin: 0px auto;
    width: 600px;
    display: flex;
    align-items: center;
    justify-content: space-between;
    padding-bottom: 75px;
}

#author_tekst{
    text-align: justify;
    width: 300px;
}

/*-----AUTHOR-----*/

/*-----404-----*/
#error404{
    width: 100%;
    text-align: center;
    background-color: rgb(241, 247, 247);
    font-size: 32px;
}

#error404 p{
    padding-top: 50px;
}

/*-----404-----*/

/*-----ADMIN-----*/
.naslov_svaki_admin{
    width: 100%;
    text-align: center;
    font-size: 32px;
    background-color: #F5F5F5;
}

.naslov_svaki_admin p{
    padding-top: 50px;
    padding-bottom: 50px;
}

```

```
.admin_greske{
  width: 100%;
  text-align: center;
  background-color: #F5F5F5;
}

#admin_greske h4{
  padding-bottom: 5px;
}

#admin_sadrzaj{
  width: 100%;
  background-color: #F5F5F5;
}

#admin_drzac{
  width: 1300px;
  margin: 0px auto;
  display: flex;
  justify-content: center;
}

#admin_drzac ul li a{
  display: block;
  margin: 25px;
  font-weight: bold;
  padding: 10px;
  background-color: #373737;
  text-align: center;
  letter-spacing: 1px;
  border-radius: 5px;
}

#admin_drzac ul li a:link{
  color: #A9A9A9;
}

#admin_drzac ul li a:visited{
  color: #A9A9A9;
}

#admin_drzac ul li a:hover{
  color: #F5F5F5;
}

#admin_sadrzaj2{
  width: 100%;
  background-color: #F5F5F5;
}

#admin_drzac2{
  width: 1300px;
  margin: 0px auto;
  display: flex;
  justify-content: center;
}

#admin_drzac2 ul li{
  display: inline-block;
}

#admin_drzac2 ul li a{
  display: inline-block;
  margin: 25px;
  font-weight: bold;
  padding: 10px;
  background-color: #373737;
  text-align: center;
}
```

```
    letter-spacing: 1px;
    border-radius: 5px;
}

#admin_drzac2 ul li a:link{
    color: #A9A9A9;
}

#admin_drzac2 ul li a:visited{
    color: #A9A9A9;
}

#admin_drzac2 ul li a:hover{
    color: #F5F5F5;
}

#admin_meni{
    width: 100%;
    text-align: center;
    background-color: #373737;
    color: white;
    letter-spacing: 1px;
}

#admin_meni h1{
    padding-top: 50px;
    padding-bottom: 50px;
}

#add_uloga{
    height: 35px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 125px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
}

#add_aktivan{
    height: 35px;
    font-family: 'Mukta', sans-serif;
    font-size: 15px;
    width: 125px;
    border: 1px solid #A9A9A9;
    text-indent: 5px;
    float: right;
}

#add_aktivan_t{
    float: right;
}

#addUserDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
}

#updateUserDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
}

#formaProizvod tr td{
    padding: 3px;
}

#formaProizvod img{
```

```
width: 350px;
height: 350px;
}

#insertProizvodNaziv{
  font-family: 'Mukta', sans-serif;
  height: 100px;
  font-size: 15px;
  resize: none;
  border: 1px solid #A9A9A9;
  text-indent: 5px;
  width: 500px;
}

#insertProizvodOpis{
  font-family: 'Mukta', sans-serif;
  height: 100px;
  font-size: 15px;
  resize: none;
  border: 1px solid #A9A9A9;
  text-indent: 5px;
  width: 500px;
}

#updateProizvodNaziv{
  font-family: 'Mukta', sans-serif;
  height: 150px;
  font-size: 15px;
  resize: none;
  border: 1px solid #A9A9A9;
  text-indent: 5px;
  width: 300px;
}

#updateProizvodOpis{
  font-family: 'Mukta', sans-serif;
  height: 150px;
  font-size: 15px;
  resize: none;
  border: 1px solid #A9A9A9;
  text-indent: 5px;
  width: 300px;
}

#insertProizvodSlika{
  font-family: 'Mukta', sans-serif;
  font-size: 15px;
  float: left;
}

#insertProizvodCena{
  font-family: 'Mukta', sans-serif;
  font-size: 15px;
  text-indent: 5px;
  height: 35px;
  border: 1px solid #A9A9A9;
  float: right;
}

#formaProizvodCena_t{
  float: right;
}

#insertProizvodBrend{
  font-family: 'Mukta', sans-serif;
  font-size: 15px;
  text-indent: 5px;
  height: 35px;
  border: 1px solid #A9A9A9;
```

```
width: 110px;
}

#insertProizvodPol{
  font-family: 'Mukta', sans-serif;
  font-size: 15px;
  text-indent: 5px;
  height: 35px;
  border: 1px solid #A9A9A9;
  width: 110px;
  float: right;
}

#formaProizvodPol_t{
  float: right;
  margin-right: 50px;
}

#insertProizvodKategorija{
  font-family: 'Mukta', sans-serif;
  font-size: 15px;
  text-indent: 5px;
  height: 35px;
  border: 1px solid #A9A9A9;
  width: 110px;
  float: right;
}

#formaProizvodKategorija_t{
  float: right;
}

#addProductDugme{
  font-family: 'Mukta', sans-serif;
  font-size: 15px;
  text-indent: 5px;
  border: 1px solid #A9A9A9;
  font-weight: bold;
  height: 35px;
  width: 300px;
  margin-top: 10px;
  letter-spacing: 1px;
}

#updateProductDugme{
  font-family: 'Mukta', sans-serif;
  font-size: 15px;
  text-indent: 5px;
  border: 1px solid #A9A9A9;
  font-weight: bold;
  height: 35px;
  width: 300px;
  margin-top: 10px;
}

#formaProizvod tr:last-of-type{
  text-align: center;
}

#formaProizvod{
  padding-bottom: 40px;
}

.malaSlika{
  width: 100px;
}
```

```
#allusers_sadrzaj{
  width: 100%;
  display: flex;
  justify-content: center;
}

#allproducts_sadrzaj{
  width: 100%;
  display: flex;
  justify-content: center;
}

/*-----ADMIN-----*/

/*-----USERPROFILE-----*/
#profile_sadrzaj{
  width: 100%;
  background-color: #F5F5F5;
}

#profile_drzac{
  margin: 0px auto;
  width: 1300px;
  display: flex;
  justify-content: space-around;
  padding-top: 20px;
  padding-bottom: 80px;
}

#o_korisniku{
  width: 500px;
  height: 500px;
  display: flex;
  justify-content: center;
  align-items: center;
}

#o_korisniku table{
  font-size: 18px;
  width: 350px;
}

#o_korisniku table th{
  background-color: #373737;
  color: white;
  letter-spacing: 1px;
  border-radius: 5px;
}

#o_korisniku table tr:hover {
  background-color: #ddd;
}

#o_korisniku table tr td{
  padding-left: 10px;
  padding-right: 10px;
  border-bottom: 1px solid #A9A9A9;
  height: 40px;
}

#promena_lozinke{
  width: 500px;
  height: 500px;
}

#tabela_promeni_lozinku tr:last-child td{
  padding-top: 10px;
}
```

```
#tabela_promeni_lozinku{
    width: 400px;
}

#promenaLozinkeDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
}

#formaPromenaLozinke{
    margin: 60px 0px 0px 100px;
}

#promenaLozinkeGreske{
    width: 500px;
    text-align: center;
    margin-top: 20px;
}

#novaLozinkaGreska{
    width: 400px;
}

#novaLozinkaGreska2{
    width: 400px;
}

/*-----USERPROFILE-----*/

/*-----RESETPASSWORD-----*/

#reset_lozinke_sadrzaj{
    width: 100%;
    background-color: #F5F5F5;
}

#reset_lozinke_drzac{
    width: 1300px;
    margin: 0px auto;
    display: flex;
    justify-content: center;
    align-items: center;
}

#resetLozinkeDugme{
    font-weight: bold;
    text-indent: 0px;
    letter-spacing: 1px;
}

#resetLozinkeGreske{
    margin: 0px auto;
    text-align: center;
    height: 125px;
    padding-top: 30px;
}

/*-----RESETPASSWORD-----*/

/*-----*/

.malaSlika{
    width: 100px;
}

.detaljnijeD{
    font-size: 15px;
}
```

```
padding: 10px;
font-weight: bold;
background-color: darkolivegreen;
display: inline-block;
letter-spacing: 1px;
border-radius: 5px;
}

.detaljnijeD:link{
    color: #A9A9A9;
}

.detaljnijeD:visited{
    color: #A9A9A9;
}

.detaljnijeD:hover{
    color: white;
}

.tabela {
    border-collapse: collapse;
    width: 100%;
    margin-bottom: 100px;
}

.tabela td, .tabela th {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: center;
}

.tabela tr:nth-child(even){background-color: #f2f2f2;}

.tabela tr:hover {background-color: #ddd;}

.tabela th {
    padding-top: 12px;
    padding-bottom: 12px;
    text-align: center;
    background-color: #373737;
    color: white;
    letter-spacing: 1px;
}

.tabela_shop {
    border-collapse: collapse;
    width: 80%;
    margin-bottom: 100px;
}

.tabela_shop td, .tabela_shop th {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: center;
}

.tabela_shop tr:nth-child(even){background-color: #f2f2f2;}

.tabela_shop tr:hover {background-color: #ddd;}

.tabela_shop th {
    padding-top: 12px;
    padding-bottom: 12px;
    text-align: center;
    background-color: #373737;
    color: white;
    letter-spacing: 1px;
}
```



```
.tabela_shop h3{
    color: #7971ea;
}
/*-----*/

/*-----*/
.ucitavanje {
    display:    none;
    position:   fixed;
    z-index:    1000;
    top:        0;
    left:       0;
    height:     100%;
    width:      100%;
    background: rgba(0, 0, 0, 0.5)
    url('../images/ucitavanje.gif')
    50% 50%
    no-repeat;
}

body.loading .ucitavanje {
    overflow: hidden;
}

body.loading .ucitavanje {
    display: block;
}
/*-----*/
```

### 3.3) JavaScript – script.js

```
$(document).ready(function(){
    $('#regDugme').on('click', regProvera);
    $('#loginDugme').on('click', loginProvera);
    $('#contactDugme').on('click', contactProvera);

    $('#addUserDugme').on('click', addUserProvera);
    $('#updateUserDugme').on('click', updateUserProvera);
    $('#obrisiKorisnika').on('click', obrisiKorisnika);
    $('#obrisiProizvod').on('click', obrisiProizvod);

    $('#dodaj').on('click', dodajUKorpu);
    $('#izbaciIzKorpe').on('click', izbaciIzKorpe);
    $('#kupiDugme').on('click', kupi);

    $('#selectAnketa').on('change', dohvatiOdgovore);
    $('#glasajDugmeDiv').hide();
    $('#glasaj').on('click', glasaj);
    $('#rezultatiDugme').on('click', dohvatiRezultate);

    $('#korpaLink').on('click', korpaLogovanProvera);
    $('#profilLink').on('click', profilLogovanProvera);

    $('#sortiraj').on('change', filter);
    $('#kategorija_filter').on('click', filter);
    $('#brend_filter').on('click', filter);
    $('#pol_filter').on('click', filter);
    $('#minCena').on('keyup', filter);
    $('#maxCena').on('keyup', filter);
    $('#ponistiSve').on('click', ponistiSve);

    $('#promenaLozinkeDugme').on('click', promenaLozinke);
    $('#zab_loz_p').on('click', prikaziSakrijResetovanje);
    $('#zab_loz_p').css('cursor', 'pointer');
    $('#zab_loz_div').hide();
    $('#resetPassEmailDugme').on('click', resetPassEmailProvera);
    $('#resetLozinkeDugme').on('click', resetLozinke);

    $(window).on('keyup', function (e) {
        if(e.keyCode == 13)
        {
            $('#loginDugme').click();
            //$('#regDugme').click();
            //$('#addUserDugme').click();
            //$('#updateUserDugme').click();
            //$('#addProductDugme').click();
            //$('#updateProductDugme').click();
        }
    });

    $('#nav li').hover(
        function() {
            $('ul', this).stop().slideDown(200);
        },
        function() {
            $('ul', this).stop().slideUp(200);
        }
    );

    slajder();
});
```

```

$('.vecaSlika').simpleLightbox();

$('#loginKorIme').focus();

$('#loginDugme, #regDugme, #addProductDugme, #updateProductDugme, #addUserDugme,
#updateUserDugme, #glasaj, #rezultatiDugme, #contactDugme, #ponistiSve, #promenaLozinkeDugme,
#resetPassEmailDugme, #resetLozinkeDugme').hover(
    function(){
        $(this).css({backgroundColor : 'darkolivegreen'});
    },
    function(){
        $(this).css({backgroundColor : '#DCDCDC'});
    }
);

filter();

$(document).on({
    ajaxStart: function() { $('body').addClass("loading"); },
    ajaxStop: function() { $('body').removeClass("loading"); }
});

});

function resetLozinke() {
    var token = $('#promenaLozinkeKorisnikToken').val();
    var novaLozinka = $('#novaLozinka').val();
    var novaLozinka2 = $('#novaLozinka2').val();

    var promenaLozinkeGreske = [];

    if(novaLozinka === ""){
        promenaLozinkeGreske.push("Polje za novu lozinku mora biti popunjeno");
        $('#novaLozinkaGreska').html("Polje za novu lozinku mora biti popunjeno");
    } else if(novaLozinka.length < 6){
        promenaLozinkeGreske.push("Nova lozinka mora imati bar 6 karaktera");
        $('#novaLozinkaGreska').html("Nova lozinka mora imati bar 6 karaktera");
    } else if((novaLozinka !== novaLozinka2) && (novaLozinka2 !== "") && (novaLozinka2.length >=
6)){
        promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne poklapaju");
        $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
        $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
    } else{
        $('#novaLozinkaGreska').html("");
    }

    if(novaLozinka2 === ""){
        promenaLozinkeGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
        $('#novaLozinkaGreska2').html("Polje za ponovljenu lozinku mora biti popunjeno");
    } else if(novaLozinka2.length < 6){
        promenaLozinkeGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
        $('#novaLozinkaGreska2').html("Ponovljena lozinka mora imati bar 6 karaktera");
    } else if((novaLozinka !== novaLozinka2) && (novaLozinka !== "") && (novaLozinka.length >= 6)){
        promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne poklapaju");
        $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
        $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
    } else{
        $('#novaLozinkaGreska2').html("");
    }

    if(promenaLozinkeGreske.length == 0){
        var data =
        {
            token : token,
            novaLozinka : novaLozinka,
            novaLozinka2 : novaLozinka2,

```

```

        resetLozinkeDugme : "ok"
    };

$.ajax({
    url : "modules/resetpassword.php",
    method : "POST",
    dataType : "json",
    data : data,
    success : function (data) {
        $('#resetLozinkeGreske').html("<h2>Uspešno ste promenili lozinku!</h2>");
        setTimeout(function() {
            window.location.href = "https://luggage-
online.000webhostapp.com/index.php?page=login";
        }, 5000);
    },
    error : function (xhr, status, error) {
        let greska = "";
        if(xhr.responseText != null)
        {
            try {
                greska = JSON.parse(xhr.responseText).message;
            } catch (e) {
                greska = "";
            }
        }
        let poruka = "";
        switch (xhr.status){
            case 404 :
                poruka = "Trenutno nije moguće promeniti lozinku, pokušajte kasnije";
                break;
            case 422:
                poruka = "<ul>";
                for(let i=0; i<greska.length; i++){
                    poruka += "<li>" + greska[i] + "</li>";
                }
                poruka += "</ul>";
                break;
            case 500:
                poruka = "Greška pri promeni lozinke, pokušajte kasnije";
                break;
        }
        if(poruka !== "")
            $('#resetLozinkeGreske').html("<h4>" + poruka + "</h4>");
    }
});
}

function resetPassEmailProvera() {
    var email = $.trim($('#resetPassEmail').val());
    var proveraEmail = /^[\\w]+[\\.\\w\\d]*[\\w\\d]+\\@[\\w]+([\\.][\\w]+)+$/;
    var greska = "";

    if(email === ""){
        greska = "Polje za email mora biti popunjeno";
        $('#resetPassEmailGreska').html("Polje za email mora biti popunjeno");
    } else if(!proveraEmail.test(email)){
        greska = "Email nije u dobrom formatu";
        $('#resetPassEmailGreska').html("Email nije u dobrom formatu");
    } else{
        $('#resetPassEmailGreska').html("");
    }

    if(greska === ""){
        var data = {
            email : email,
            resetPassEmailDugme : "ok"
        };
    }
}

```

```

$.ajax({
    url : "modules/resetpassword.php",
    method : "POST",
    dataType : "json",
    data : data,
    success : function (data) {
        $('#resetPassEmailIspis').html("<h3>Proverite email za nastavak procesa . . . .
.</h3>");

        setTimeout(function() {
            window.location.href = "https://luggage-online.000webhostapp.com/index.php";
        }, 5000);
    },
    error : function (xhr, status, error) {
        let greska = "";
        if(xhr.responseText != null)
        {
            try {
                greska = JSON.parse(xhr.responseText).message;
            } catch (e) {
                greska = "";
            }
        }
        let poruka = "";
        switch (xhr.status){
            case 404 :
                poruka = "Trenutno nije moguće resetovati lozinku, pokušajte kasnije";
                break;
            case 409:
                poruka = "Ne postoji nalog sa unetim email-om";
                break;
            case 422:
                poruka = greska;
                break;
            case 500:
                poruka = "Greška pri resetovanju lozinke, pokušajte kasnije";
                break;
        }
        if(poruka !== "")
            $('#resetPassEmailIspis').html("<h4>" + poruka + "</h4>");
    }
});
}
}
}

```

```

function prikaziSakrijResetovanje() {
    $('#zab_loz_div').toggle();
}

```

```

function promenaLozinke() {
    var korisnikID = $('#promenaLozinkeKorisnik').val();
    var trenutnaLozinka = $('#trenutnaLozinka').val();
    var novaLozinka = $('#novaLozinka').val();
    var novaLozinka2 = $('#novaLozinka2').val();

    var promenaLozinkeGreske = [];

    if(trenutnaLozinka === ""){
        promenaLozinkeGreske.push("Polje za trenutnu lozinku mora biti popunjeno");
        $('#trenutnaLozinkaGreska').html("Polje za trenutnu lozinku mora biti popunjeno");
    } else if(trenutnaLozinka.length < 6){
        promenaLozinkeGreske.push("Trenutna lozinka mora imati bar 6 karaktera");
        $('#trenutnaLozinkaGreska').html("Trenutna lozinka mora imati bar 6 karaktera");
    } else{
        $('#trenutnaLozinkaGreska').html("");
    }
}

```

```

if(novaLozinka === ""){
    promenaLozinkeGreske.push("Polje za novu lozinku mora biti popunjeno");
    $('#novaLozinkaGreska').html("Polje za novu lozinku mora biti popunjeno");
} else if(novaLozinka.length < 6){
    promenaLozinkeGreske.push("Nova lozinka mora imati bar 6 karaktera");
    $('#novaLozinkaGreska').html("Nova lozinka mora imati bar 6 karaktera");
} else if((novaLozinka !== novaLozinka2) && (novaLozinka2 !== "") && (novaLozinka2.length >=
6)){
    promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne poklapaju");
    $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
    $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
} else{
    $('#novaLozinkaGreska').html("");
}

if(novaLozinka2 === ""){
    promenaLozinkeGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
    $('#novaLozinkaGreska2').html("Polje za ponovljenu lozinku mora biti popunjeno");
} else if(novaLozinka2.length < 6){
    promenaLozinkeGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
    $('#novaLozinkaGreska2').html("Ponovljena lozinka mora imati bar 6 karaktera");
} else if((novaLozinka !== novaLozinka2) && (novaLozinka !== "") && (novaLozinka.length >= 6)){
    promenaLozinkeGreske.push("Nova lozinka i ponovljena lozinka se ne poklapaju");
    $('#novaLozinkaGreska').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
    $('#novaLozinkaGreska2').html("Nova lozinka i ponovljena lozinka se ne poklapaju");
} else{
    $('#novaLozinkaGreska2').html("");
}

if(promenaLozinkeGreske.length == 0){
    var data = {
        korisnikID : korisnikID,
        trenutnaLozinka : trenutnaLozinka,
        novaLozinka : novaLozinka,
        novaLozinka2 : novaLozinka2,
        promenaLozinkeDugme : "ok"
    };
    $.ajax({
        url : "modules/changepassword.php",
        method : "POST",
        dataType : "json",
        data : data,
        success : function (data) {
            $('#promenaLozinkeGreske').html("<h2>Uspešno ste promenili lozinku!</h2>");
            setTimeout(function() {
                window.location.href = "https://luggage-online.000webhostapp.com/index.php";
            }, 5000);
        },
        error : function (xhr, status, error) {
            let greska = "";
            if(xhr.responseText != null)
            {
                try {
                    greska = JSON.parse(xhr.responseText).message;
                } catch (e) {
                    greska = "";
                }
            }
            let poruka = "";
            switch (xhr.status){
                case 404 :
                    poruka = "Trenutno nije moguće promeniti lozinku, pokušajte kasnije";
                    break;
                case 409:
                    $('#trenutnaLozinkaGreska').html("Trenutna lozinka nije ispravna");
                    break;
                case 422:
                    poruka = "<ul>";
                    for(let i=0; i<greska.length; i++){

```

```

        poruka += "<li>" + greska[i] + "</li>";
    }
    poruka += "</ul>";
    break;
case 500:
    poruka = "Greška pri promeni lozinke, pokušajte kasnije";
    break;
}
if(poruka !== "")
    $('#promenaLozinkeGreske').html("<h4>" + poruka + "</h4>");
});
}
}
}

```

```

function filter() {
    var sortiraj = $('#sortiraj').val();
    var kategorija = document.getElementsByName("kategorija_filter");
    var brend = document.getElementsByName("brend_filter");
    var pol = document.getElementsByName("pol_filter");
    var minCena = $('#minCena').val();
    var maxCena = $('#maxCena').val();

    var kategorijaNiz = [];
    var brendNiz = [];
    var polNiz = [];

    for(let i=0; i<kategorija.length; i++)
    {
        if(kategorija[i].checked)
            kategorijaNiz.push("'" + kategorija[i].value + "'");
    }

    for(let i=0; i<brend.length; i++)
    {
        if(brend[i].checked)
            brendNiz.push("'" + brend[i].value + "'");
    }

    for(let i=0; i<pol.length; i++)
    {
        if(pol[i].checked)
            polNiz.push("'" + pol[i].value + "'");
    }

    var kategorijeStr = "";
    var brendoviStr = "";
    var poloviStr = "";

    if(kategorijaNiz.length){
        kategorijeStr = kategorijaNiz.join(", ");
    }

    if(brendNiz.length){
        brendoviStr = brendNiz.join(", ");
    }

    if(polNiz.length){
        poloviStr = polNiz.join(", ");
    }

    var data = {
        sortiraj : sortiraj,
        kategorijeStr : kategorijeStr,
        brendoviStr : brendoviStr,
        poloviStr : poloviStr,
        minCena : minCena,
        maxCena : maxCena,
        filterDugme : "ok"
    }
}

```





```

function paginacija() {
    var kojaStranica = $(this).data('id');
    var poStranici = 8;
    var from = poStranici * (kojaStranica - 1);

    var sortiraj = $('#sortiraj').val();
    var kategorija = document.getElementsByName("kategorija_filter");
    var brend = document.getElementsByName("brend_filter");
    var pol = document.getElementsByName("pol_filter");
    var minCena = $('#minCena').val();
    var maxCena = $('#maxCena').val();

    var kategorijaNiz = [];
    var brendNiz = [];
    var polNiz = [];

    for(let i=0; i<kategorija.length; i++)
    {
        if(kategorija[i].checked)
            kategorijaNiz.push("'" + kategorija[i].value + "'");
    }

    for(let i=0; i<brend.length; i++)
    {
        if(brend[i].checked)
            brendNiz.push("'" + brend[i].value + "'");
    }

    for(let i=0; i<pol.length; i++)
    {
        if(pol[i].checked)
            polNiz.push("'" + pol[i].value + "'");
    }

    var kategorijeStr = "";
    var brendoviStr = "";
    var poloviStr = "";

    if(kategorijaNiz.length){
        kategorijeStr = kategorijaNiz.join(", ");
    }

    if(brendNiz.length){
        brendoviStr = brendNiz.join(", ");
    }

    if(polNiz.length){
        poloviStr = polNiz.join(", ");
    }

    var data = {
        sortiraj : sortiraj,
        kategorijeStr : kategorijeStr,
        brendoviStr : brendoviStr,
        poloviStr : poloviStr,
        minCena : minCena,
        maxCena : maxCena,
        from : from,
        poStranici : poStranici,
        paginacijaDugme : "ok"
    };

    $.ajax({
        url : "modules/pagination.php",
        method : "POST",
        dataType : "json",
        data : data,
        success : function (data) {

```

```

        var proizvodi = data.message;

        if(proizvodi != null)
        {
            var ispis = "";
            for(let i=0; i<proizvodi.length; i++){
                ispis += "<div class='proizvod'>";
                ispis += "<img src='images/upload/' + proizvodi[i].src + " alt='"+
proizvodi[i].alt + "' />";
                ispis += "<div class='detaljnije2'>";
                ispis += "<p class='ps_naziv'>" + proizvodi[i].proizvod + "</p>";
                ispis += "<p class='ps_cena'>" + proizvodi[i].cena + " din.</p>";
                ispis += "<a href='index.php?page=product&id=" + proizvodi[i].proizvodID + "'
class='detaljnije'>DETALJNIJE</a>";
                ispis += "</div>";
                ispis += "</div>";
            }
            $('#proizvodi_sadrzaj').html(ispis);

            var sveStraniceLinkovi = document.getElementsByClassName('straniceLink');
            for(let i=0; i<sveStraniceLinkovi.length; i++)
            {
                if(sveStraniceLinkovi[i].innerHTML == kojaStranica){
                    sveStraniceLinkovi[i].style.border = "1px solid #7971ea";
                    sveStraniceLinkovi[i].style.borderRadius = "5px";
                } else{
                    sveStraniceLinkovi[i].style.border = "none";
                }
            }

            } else{
                var nemaProizvoda = "<div id='nemaProizvoda'><h3>Nema proizvoda za izabrane
filtere</h3></div>";
                $('#proizvodi_sadrzaj').html(nemaProizvoda);
                $('.stranice').html("");
            }
        },
        error : function (xhr, status, error) {
        }
    });
}

function ponistiSve() {
    $('#sortiraj').val("0");
    var kategorija = document.getElementsByName("kategorija_filter");
    var brend = document.getElementsByName("brend_filter");
    var pol = document.getElementsByName("pol_filter");
    $('#minCena').val("");
    $('#maxCena').val("");

    for(let i=0; i<kategorija.length; i++)
    {
        if(kategorija[i].checked)
            kategorija[i].checked = false;
    }

    for(let i=0; i<brend.length; i++)
    {
        if(brend[i].checked)
            brend[i].checked = false;;
    }

    for(let i=0; i<pol.length; i++)
    {
        if(pol[i].checked)
            pol[i].checked = false;
    }
}

```



```

        $('#stranice').html("");
    }

    },
    error : function (xhr, status, error) {
    }
});
}

function regProvera() {
    var regIme = $.trim($('#regIme').val());
    var regPrezime = $.trim($('#regPrezime').val());
    var regEmail = $.trim($('#regEmail').val());
    var regKorIme = $.trim($('#regKorIme').val());
    var regLozinka = $('#regLozinka').val();
    var regLozinka2 = $('#regLozinka2').val();

    var proveraImePrezime = /^[A-ZŠĐČĆŽ][a-zšđććž]{2,12}(\s[A-ZŠĐČĆŽ][a-zšđććž]{2,12}){0,1}$/;
    var proveraEmail = /^[\\w]+[\\.|\\w\\d]*[\\w\\d]+\\@[\\w]+([\\.|\\w]+)$/;
    var proveraKorIme = /^[\\w\\d\\.\\_]{5,15}$/;
    var regGreske = [];

    if(regIme === ""){
        regGreske.push("Polje za ime mora biti popunjeno");
        $('#regImeGreska').html("Polje za ime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regIme)){
        regGreske.push("Ime nije u dobrom formatu");
        $('#regImeGreska').html("Ime nije u dobrom formatu");
    } else{
        $('#regImeGreska').html("");
    }

    if(regPrezime === ""){
        regGreske.push("Polje za prezime mora biti popunjeno");
        $('#regPrezimeGreska').html("Polje za prezime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regPrezime)){
        regGreske.push("Prezime nije u dobrom formatu");
        $('#regPrezimeGreska').html("Prezime nije u dobrom formatu");
    } else{
        $('#regPrezimeGreska').html("");
    }

    if(regEmail === ""){
        regGreske.push("Polje za email mora biti popunjeno");
        $('#regEmailGreska').html("Polje za email mora biti popunjeno");
    } else if(!proveraEmail.test(regEmail)){
        regGreske.push("Email nije u dobrom formatu");
        $('#regEmailGreska').html("Email nije u dobrom formatu");
    } else{
        $('#regEmailGreska').html("");
    }

    if(regKorIme === ""){
        regGreske.push("Polje za korisničko ime mora biti popunjeno");
        $('#regKorImeGreska').html("Polje za korisničko ime mora biti popunjeno");
    } else if(!proveraKorIme.test(regKorIme)){
        regGreske.push("Korisničko ime mora imati 5-15 karaktera");
        $('#regKorImeGreska').html("Korisničko ime mora imati 5-15 karaktera");
    } else{
        $('#regKorImeGreska').html("");
    }

    if(regLozinka === ""){
        regGreske.push("Polje za lozinku mora biti popunjeno");
        $('#regLozinkaGreska').html("Polje za lozinku mora biti popunjeno");
    } else if(regLozinka.length < 6){
        regGreske.push("Lozinka mora imati bar 6 karaktera");
        $('#regLozinkaGreska').html("Lozinka mora imati bar 6 karaktera");
    }
}

```

```

    } else if((regLozinka !== regLozinka2) && (regLozinka2 !== "") && (regLozinka2.length >= 6)){
        regGreske.push("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinkaGreska').html("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinka2Greska').html("Lozinka i ponovljena lozinka se ne poklapaju");
    } else{
        $('#regLozinkaGreska').html("");
    }

    if(regLozinka2 === ""){
        regGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
        $('#regLozinka2Greska').html("Polje za ponovljenu lozinku mora biti popunjeno");
    } else if(regLozinka2.length < 6){
        regGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
        $('#regLozinka2Greska').html("Ponovljena lozinka mora imati bar 6 karaktera");
    } else if((regLozinka !== regLozinka2) && (regLozinka !== "") && (regLozinka.length >= 6)){
        regGreske.push("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinkaGreska').html("Lozinka i ponovljena lozinka se ne poklapaju");
        $('#regLozinka2Greska').html("Lozinka i ponovljena lozinka se ne poklapaju");
    } else{
        $('#regLozinka2Greska').html("");
    }

    if(regGreske.length == 0){
        var data = {
            regIme : regIme,
            regPrezime : regPrezime,
            regEmail : regEmail,
            regKorIme : regKorIme,
            regLozinka : regLozinka,
            regLozinka2 : regLozinka2,
            regDugme : "ok"
        };
        $.ajax({
            url : "modules/register.php",
            method : "POST",
            dataType : "json",
            data : data,
            success : function (data) {
                $('.greske').html("<h2>Uspešno ste se registrovali!</h2><h4>Proverite email i aktivirajte nalog</h4>");
                setTimeout(function() {
                    window.location.href = "https://luggage-online.000webhostapp.com/index.php";
                }, 10000);
            },
            error : function (xhr, status, error) {
                let greska = "";
                if(xhr.responseText != null)
                {
                    try {
                        greska = JSON.parse(xhr.responseText).message;
                    } catch (e) {
                        greska = "";
                    }
                }
                let poruka = "";
                switch (xhr.status){
                    case 404 :
                        poruka = "Trenutno nije moguća registracija novih korisnika, pokušajte kasnije";
                        break;
                    case 409:
                        poruka = "Korisničko ime ili email već postoje";
                        break;
                    case 422:
                        poruka = "<ul>";
                        for(let i=0; i<greska.length; i++){
                            poruka += "<li>"+ greska[i] +"</li>";
                        }
                        poruka += "</ul>";
                }
            }
        });
    }

```

```

        break;
    case 500:
        poruka = "Greška pri registraciji, pokušajte kasnije";
        break;
    }
    if(poruka != "")
        $('#greske').html("<h4>" + poruka + "</h4>");
    });
}
}

function loginProvera() {
    var loginKorIme = $.trim($('#loginKorIme').val());
    var loginLozinka = $('#loginLozinka').val();
    var reLoginKorIme = /^[\\w\\d\\.\\_]{5,15}$/;
    var loginGreske = [];

    if(loginKorIme === ""){
        loginGreske.push("Polje za korisničko ime mora biti popunjeno");
        $('#loginKorImeGreska').html("Polje za korisničko ime mora biti popunjeno");
    } else if(!reLoginKorIme.test(loginKorIme)){
        loginGreske.push("Korisničko ime mora imati 5-15 karaktera");
        $('#loginKorImeGreska').html("Korisničko ime mora imati 5-15 karaktera");
    } else{
        $('#loginKorImeGreska').html("");
    }

    if(loginLozinka === ""){
        loginGreske.push("Polje za lozinku mora biti popunjeno");
        $('#loginLozinkaGreska').html("Polje za lozinku mora biti popunjeno");
    } else if(loginLozinka.length < 6){
        loginGreske.push("Lozinka mora imati bar 6 karaktera");
        $('#loginLozinkaGreska').html("Lozinka mora imati bar 6 karaktera");
    } else{
        $('#loginLozinkaGreska').html("");
    }

    if(loginGreske.length == 0){
        var data = {
            loginKorIme : loginKorIme,
            loginLozinka : loginLozinka,
            loginDugme : "ok"
        };

        $.ajax({
            url : "modules/login.php",
            method : "POST",
            dataType : "json",
            data : data,
            success : function (data) {
                var uloga = data.message;
                if(uloga === "Admin"){
                    window.location.href = "https://luggage-online.000webhostapp.com/index.php?page=admin";
                } else{
                    window.location.href = "https://luggage-online.000webhostapp.com/index.php";
                }
            },
            error : function (xhr, status, error) {
                let greska = "";
                if(xhr.responseText != null)
                {
                    try {
                        greska = JSON.parse(xhr.responseText).message;
                    } catch (e) {
                        greska = "";
                    }
                }
            }
        });
    }
}

```

```

        let poruka = "";
        switch (xhr.status){
            case 404 :
                poruka = "Trenutno nije moguće prijavljivanje, pokušajte kasnije";
                break;
            case 409:
                $('#loginKorImeGreska').html("Pogrešno korisničko ime ili lozinka");
                $('#loginLozinkaGreska').html("Pogrešno korisničko ime ili lozinka");
                break;
            case 422:
                poruka = "<ul>";
                for(let i=0; i<greska.length; i++){
                    poruka += "<li>"+ greska[i] +"</li>";
                }
                poruka += "</ul>";
                break;
        }
        if(poruka !== "")
            $('#greske').html("<h4>"+ poruka +"</h4>");
    });
}

function contactProvera() {
    var contactEmail = $.trim($('#contactEmail').val());
    var poruka= $('#poruka').val();
    var reContactEmail = /^[\\w]+[\\.\\w\\d]*[\\w\\d]+\\@[\\w]+([\\\\.][\\w]+)+$/;
    var contactGreske = [];

    if(contactEmail === ""){
        contactGreske.push("Polje za email mora biti popunjeno");
        $('#contactEmailGreska').html("Polje za email mora biti popunjeno");
    } else if(!reContactEmail.test(contactEmail)){
        contactGreske.push("Email nije u dobrom formatu");
        $('#contactEmailGreska').html("Email nije u dobrom formatu");
    } else{
        $('#contactEmailGreska').html("");
    }

    if(poruka === ""){
        contactGreske.push("Polje za poruku mora biti popunjeno");
        $('#contactPorukaGreska').html("Polje za poruku mora biti popunjeno");
    } else if(poruka.length < 15 || poruka.length > 200){
        contactGreske.push("Poruka mora imati 15-200 karaktera");
        $('#contactPorukaGreska').html("Poruka mora imati 15-200 karaktera");
    } else{
        $('#contactPorukaGreska').html("");
    }

    if(contactGreske.length == 0){
        var data = {
            contactEmail : contactEmail,
            poruka : poruka,
            contactDugme : "ok"
        };

        $.ajax({
            url : "modules/mailadmin.php",
            method : "POST",
            dataType : "json",
            data : data,
            success : function (data) {
                $('#contact_greske').html("<h2>Poruka uspešno poslata!</h2>");
                setTimeout(function() {
                    window.location.href = "https://luggage-online.000webhostapp.com/index.php";
                }, 5000);
            },
            error : function (xhr, status, error) {

```

```

        let greska = "";
        if(xhr.responseText != null)
        {
            try {
                greska = JSON.parse(xhr.responseText).message;
            } catch (e) {
                greska = "";
            }
        }
        let poruka = "";
        switch (xhr.status){
            case 404 :
                poruka = "Trenutno nije moguće poslati poruku, pokušajte kasnije";
                break;
            case 422:
                poruka = "<ul>";
                for(let i=0; i<greska.length; i++){
                    poruka += "<li>"+ greska[i] +"</li>";
                }
                poruka += "</ul>";
                break;
            case 500:
                poruka = "Greška pri slanju poruke, pokušajte kasnije";
                break;
        }
        if(poruka != "")
            $('#contact_greske').html("<h4>"+ poruka +"</h4>");
    });
}
}

```

```

function addUserProvera(){
    var regIme = $.trim($('#regIme').val());
    var regPrezime = $.trim($('#regPrezime').val());
    var regEmail = $.trim($('#regEmail').val());
    var regKorIme = $.trim($('#regKorIme').val());
    var regLozinka = $('#regLozinka').val();
    var regLozinka2 = $('#regLozinka2').val();
    var uloga = $('#add_uloga').val();
    var aktivan = $('#add_aktivan').val();

    var proveraImePrezime = /^[A-ZŠĐČĆŽ][a-zšđććž]{2,12}(\s[A-ZŠĐČĆŽ][a-zšđććž]{2,12}){0,1}$/;
    var proveraEmail = /^[\\w]+[\\.\w\\d]*[\\w\\d]+\\@[\\w]+([\\.\w]+)+$/;
    var proveraKorIme = /^[\\w\\d\\.\\_]{5,15}$/;
    var regGreske = [];

    if(regIme === ""){
        regGreske.push("Polje za ime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regIme)){
        regGreske.push("Ime nije u dobrom formatu");
    }
    if(regPrezime === ""){
        regGreske.push("Polje za prezime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regPrezime)){
        regGreske.push("Prezime nije u dobrom formatu");
    }
    if(regEmail === ""){
        regGreske.push("Polje za email mora biti popunjeno");
    } else if(!proveraEmail.test(regEmail)){
        regGreske.push("Email nije u dobrom formatu");
    }
    if(regKorIme === ""){
        regGreske.push("Polje za korisničko ime mora biti popunjeno");
    } else if(!proveraKorIme.test(regKorIme)){
        regGreske.push("Korisničko ime mora imati 5-15 karaktera");
    }
    if(regLozinka === ""){

```



```

    regGreske.push("Polje za lozinku mora biti popunjeno");
} else if(regLozinka.length < 6){
    regGreske.push("Lozinka mora imati bar 6 karaktera");
}
if(regLozinka2 === ""){
    regGreske.push("Polje za ponovljenu lozinku mora biti popunjeno");
} else if(regLozinka2.length < 6){
    regGreske.push("Ponovljena lozinka mora imati bar 6 karaktera");
}
if(regLozinka !== regLozinka2){
    regGreske.push("Lozinka i ponovljena lozinka se ne poklapaju");
}
if(uloga === "0"){
    regGreske.push("Morate odabrati ulogu");
}
if(aktivan === "2"){
    regGreske.push("Morate odabrati aktivnost korisnika");
}

if(regGreske.length){
    var regGreskeIspis = "<ul>";
    for(let i=0; i<regGreske.length; i++){
        regGreskeIspis += "<li><h4>"+ regGreske[i] + "</h4></li>";
    }
    regGreskeIspis += "</ul>";
    $(''.admin_greske').html(regGreskeIspis);
} else{
    var data = {
        regIme : regIme,
        regPrezime : regPrezime,
        regEmail : regEmail,
        regKorIme : regKorIme,
        regLozinka : regLozinka,
        regLozinka2 : regLozinka2,
        add_uloga : uloga,
        add_aktivan : aktivan,
        addUserDugme : "ok"
    };
    $.ajax({
        url : "modules/admin.php",
        method : "POST",
        dataType : "json",
        data : data,
        success : function (data) {
            $(''.admin_greske').html("<h2>Uspešno ste dodali korisnika!</h2><h4>Uskoro ćete biti prebačeni na admin stranicu sa svim korisnicima</h4>");
            setTimeout(function() {
                window.location.href = "https://luggage-online.000webhostapp.com/index.php?page=admin&adminaction=allusers";
            }, 5000);
        },
        error : function (xhr, status, error) {
            let greska = "";
            if(xhr.responseText != null)
            {
                try {
                    greska = JSON.parse(xhr.responseText).message;
                } catch (e) {
                    greska = "";
                }
            }
            let poruka = "";
            switch (xhr.status){
                case 404 :
                    poruka = "Trenutno nije moguće dodavanje novog korisnika, pokušajte kasnije";
                    break;
                case 409:
                    poruka = "Korisničko ime ili email već postoje";
                    break;
            }
        }
    });
}

```

```

        case 422:
            poruka = "<ul>";
            for(let i=0; i<greska.length; i++){
                poruka += "<li>" + greska[i] + "</li>";
            }
            poruka += "</ul>";
            break;
        }
        if(poruka != "")
            $(' .admin_greske').html("<h4>" + poruka + "</h4>");
    }
    });
}

function obrisiKorisnika() {
    if(confirm("Da li želite da obrišete korisnika?")){
        var id = $(this).data('id');
        var data =
            {
                korisnikID : id,
                deleteUserDugme : "ok"
            };
        $.ajax({
            url : "modules/admin.php",
            method : "POST",
            data : data,
            success : function (data) {
                alert('Korisnik uspešno obrisao!');
                window.location.href = "https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allusers";
            },
            error : function (xhr,status, error) {
                let poruka = "";
                switch (xhr.status){
                    case 404 :
                        poruka = "Trenutno nije moguće obrisati korisnika, pokušajte kasnije";
                        break;
                    case 500 :
                        poruka = "Greška pri brisanju korisnika, pokušajte kasnije";
                        break;
                }
                alert(poruka);
            }
        });
    }
}

function updateUserProvera(){
    var regIme = $.trim($('#regIme').val());
    var regPrezime = $.trim($('#regPrezime').val());
    var regEmail = $.trim($('#regEmail').val());
    var regKorIme = $.trim($('#regKorIme').val());
    var uloga = $('#add_uloga').val();
    var aktivan = $('#add_aktivan').val();
    var idUpdate = $('#idUpdate').val();

    var proveraImePrezime = /^[A-ZŠĐČĆŽ][a-zšđčćž]{2,12}(\s[A-ZŠĐČĆŽ][a-zšđčćž]{2,12}){0,1}$/;
    var proveraEmail = /^[a-zA-Z0-9_!@#$%^&*~\.\|\/\?]{1,64}@([a-zA-Z0-9_]{1,4}\.){1,4}[a-zA-Z0-9]{1,4}$/;
    var proveraKorIme = /^[a-zA-Z0-9_]{5,15}$/;
    var regGreske = [];

    if(regIme === ""){
        regGreske.push("Polje za ime mora biti popunjeno");
    } else if(!proveraImePrezime.test(regIme)){
        regGreske.push("Ime nije u dobrom formatu");
    }
}

```

```

if(regPrezime === ""){
    regGreske.push("Polje za prezime mora biti popunjeno");
} else if(!proveraImePrezime.test(regPrezime)){
    regGreske.push("Prezime nije u dobrom formatu");
}
if(regEmail === ""){
    regGreske.push("Polje za email mora biti popunjeno");
} else if(!proveraEmail.test(regEmail)){
    regGreske.push("Email nije u dobrom formatu");
}
if(regKorIme === ""){
    regGreske.push("Polje za korisničko ime mora biti popunjeno");
} else if(!proveraKorIme.test(regKorIme)){
    regGreske.push("Korisničko ime mora imati 5-15 karaktera");
}
if(uloga === "0"){
    regGreske.push("Morate odabrati ulogu");
}
if(aktivan === "2"){
    regGreske.push("Morate odabrati aktivnost korisnika");
}

if(regGreske.length){
    var regGreskeIspis = "<ul>";
    for(let i=0; i<regGreske.length; i++){
        regGreskeIspis += "<li><h4>" + regGreske[i] + "</h4></li>";
    }
    regGreskeIspis += "</ul>";
    $('#admin_greske').html(regGreskeIspis);
} else{
    var data = {
        regIme : regIme,
        regPrezime : regPrezime,
        regEmail : regEmail,
        regKorIme : regKorIme,
        add_uloga : uloga,
        add_aktivan : aktivan,
        idUpdate : idUpdate,
        updateUserDugme : "ok"
    };
    $.ajax({
        url : "modules/admin.php",
        method : "POST",
        dataType : "json",
        data : data,
        success : function (data) {
            $('#admin_greske').html("<h2>Uspešno ste izmenili korisnika!</h2><h4>Uskoro ćete biti prebačeni na admin stranicu sa svim korisnicima</h4>");
            setTimeout(function() {
                window.location.href = "https://luggage-online.000webhostapp.com/index.php?page=admin&adminaction=allusers";
            }, 5000);
        },
        error : function (xhr, status, error) {
            let greska = "";
            if(xhr.responseText != null)
            {
                try {
                    greska = JSON.parse(xhr.responseText).message;
                } catch (e) {
                    greska = "";
                }
            }
            let poruka = "";
            switch (xhr.status){
                case 404 :
                    poruka = "Trenutno nije moguće izmeniti korisnika, pokušajte kasnije";
                    break;
                case 409:
                    poruka = "Korisničko ime ili email već postoje";
            }
        }
    });
}

```

```

        break;
    case 422:
        poruka = "<ul>";
        for(let i=0; i<greska.length; i++){
            poruka += "<li>"+ greska[i] +"</li>";
        }
        poruka += "</ul>";
        break;
    }
    if(poruka != "")
        $(' .admin_greske').html("<h4>"+ poruka +"</h4>");
    }
    });
}

function obrisiProizvod() {
    if(confirm("Da li želite da obrišete proizvod?")){
        var proizvodID = $(this).data('id');
        var data =
            {
                proizvodID : proizvodID,
                deleteProductDugme : "ok"
            };
        $.ajax({
            url : "modules/admin.php",
            method : "POST",
            data : data,
            success : function (data) {
                alert('Proizvod uspešno obrisano!');
                window.location.href = "https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allproducts";
            },
            error : function (xhr, status, error) {
                let poruka = "";
                switch (xhr.status){
                    case 404 :
                        poruka = "Trenutno nije moguće obrisati korisnika, pokušajte kasnije";
                        break;
                    case 500 :
                        poruka = "Greška pri brisanju korisnika, pokušajte kasnije";
                        break;
                }
                alert(poruka);
            }
        });
    }
}

function dodajUKorpu() {
    var proizvodID = $(this).data('id');
    var korisnikID = $('#sessionIdKor').val();

    if(korisnikID == ""){
        alert("Morate se prijaviti da bi dodali proizvod u korpu");
    } else{
        var data =
            {
                proizvodID : proizvodID,
                korisnikID : korisnikID,
                dodajUKorpuDugme : "ok"
            };
        $.ajax({
            url : "modules/shop.php",
            method : "POST",
            data : data,
            success : function (data) {

```

```

        alert("Proizvod je dodat u korpu!");
    },
    error : function (xhr, status, error) {
        let poruka = "";
        switch (xhr.status){
            case 404 :
                poruka = "Trenutno nije moguće dodavati u korpu, pokušajte kasnije";
                break;
            case 500:
                poruka = "Greška pri dodavanju u korpu, pokušajte kasnije";
                break;
        }
        alert(poruka);
    }
});
}

function izbaciiZKorpe() {
    if(confirm("Da li želite da izbacite proizvod iz korpe?")){
        var proizvodID = $(this).data('id');
        var korisnikID = $('#idKorIzbaci').val();
        var data =
        {
            proizvodID : proizvodID,
            korisnikID : korisnikID,
            izbaciiZKorpeDugme : "ok"
        };
        $.ajax({
            url : "modules/shop.php",
            method : "POST",
            data : data,
            success : function (data) {
                window.location.href = "https://luggage-
online.000webhostapp.com/index.php?page=shop";
            },
            error : function (xhr, status, error) {
                let poruka = "";
                switch (xhr.status){
                    case 404 :
                        poruka = "Trenutno nije moguće izbaciti proizvod iz korpe, pokušajte
kasnije";
                        break;
                    case 500 :
                        poruka = "Greška pri izbacivanju proizvoda iz korpe, pokušajte kasnije";
                        break;
                }
                alert(poruka);
            }
        });
    }
}

function kupi() {
    if(confirm("Da li želite da izvršite kupovinu?")){
        var korisnikID = $(this).data('id');
        var data =
        {
            korisnikID : korisnikID,
            kupiDugme : "ok"
        };
        $.ajax({
            url : "modules/shop.php",
            method : "POST",
            data : data,
            success : function (data) {
                alert("Kupovina uspešno izvršena!");
            }
        });
    }
}

```

```

        window.location.href = "https://luggage-
online.000webhostapp.com/index.php?page=shop";
    },
    error : function (xhr, status, error) {
        let poruka = "";
        switch (xhr.status){
            case 404 :
                poruka = "Trenutno nije moguće izvršiti kupovinu, pokušajte kasnije";
                break;
            case 500 :
                poruka = "Greška pri izvršavanju kupovine, pokušajte kasnije";
                break;
        }
        alert(poruka);
    }
});
}
}
}

```

```

function dohvatiOdgovore() {
    var pitanjeID = $('#selectAnketa').val();
    $('#izaberiteOdgovor').html("");
    if(pitanjeID != "0"){
        var data =
        {
            pitanjeID : pitanjeID,
            dohvatiOdgovoreDugme : "ok"
        };
        $.ajax({
            url : "modules/survey.php",
            method : "POST",
            dataType : "json",
            data : data,
            success : function (data) {
                let podaci = data.message;

                let ispisPodaci = "<ul>";
                for(let i=0; i<podaci.length; i++){
                    ispisPodaci += "<li><input type='radio' value='"+ podaci[i].odgovorID +"'
id='glasajOdgovori' name='glasajOdgovori' />"+ podaci[i].odgovor + "</li>";
                }

                ispisPodaci += "</ul>";

                $('#odgovori').html(ispisPodaci);
                $('#glasajDugmeDiv').show();
            },
            error : function (xhr, status, error) {
                let poruka = "";
                switch (xhr.status){
                    case 404 :
                        poruka = "Trenutno nema odgovora za izabrano pitanje";
                        break;
                    case 409:
                        poruka = "Trenutno nema odgovora za izabrano pitanje";
                        break;
                }
                if(poruka != "")
                    $('#odgovori').html("<h4>"+ poruka + "</h4>");
            }
        });
    } else{
        $('#odgovori').html("");
        $('#glasajDugmeDiv').hide();
    }
}
}

```

```

function glasaj() {

```

```

var korisikLogovan = $('#contactKorisnikLogovan').val();
var odgovori = document.getElementsByName("glasajOdgovori");
var odgovorID = "";
var pitanjeID = $('#selectAnketa').val();
for(let i=0; i<odgovori.length; i++){
    if(odgovori[i].checked){
        odgovorID = odgovori[i].value;
    }
}

if(korisikLogovan === ""){
    $('#izaberiteOdgovor').html("<h4>Morate se prijaviti da bi glasali</h4>");
} else if(odgovorID === ""){
    $('#izaberiteOdgovor').html("<h4>Izaberite odgovor</h4>");
} else{
    var data =
    {
        pitanjeID : pitanjeID,
        odgovorID : odgovorID,
        glasajDugme : "ok"
    };
    $.ajax({
        url : "modules/survey.php",
        dataType : "json",
        method : "POST",
        data : data,
        success : function (data) {
            $('#izaberiteOdgovor').html("<h4>Uspešno ste glasali!</h4>");
        },
        error : function (xhr, status, error) {
            let poruka = "";
            switch (xhr.status){
                case 404 :
                    poruka = "Trenutno nije moguće glasati na anketu";
                    break;
                case 422:
                    poruka = "Na ovu anketu ste već glasali";
                    break;
                case 500:
                    poruka = "Greska, pokušajte kasnije";
                    break;
            }
            if(poruka != "")
                $('#izaberiteOdgovor').html("<h4>" + poruka + "</h4>");
        }
    });
}
}

function dohvatiRezultate() {
    var korisikLogovan = $('#contactKorisnikLogovan').val();
    var pitanjeID = $('#selectAnketa').val();

    if(korisikLogovan === ""){
        $('#izaberiteOdgovor').html("<h4>Morate se prijaviti da bi videli rezultate glasanja</h4>");
    } else{
        var data =
        {
            pitanjeID : pitanjeID,
            dohvatiRezultateDugme : "ok"
        };
        $.ajax({
            url : "modules/survey.php",
            method : "POST",
            dataType : "json",
            data : data,
            success : function (data) {
                let podaci = data.message;

```

```

    let ispisPodaci = "<ul>";
    for(let i=0; i<podaci.length; i++){

        ispisPodaci += "<li><p>" + podaci[i].odgovor + " - " + podaci[i].broj_glasova;
        if(podaci[i].broj_glasova == 1){
            ispisPodaci += " glas";
        } else if((podaci[i].broj_glasova > 1) && (podaci[i].broj_glasova < 5)){
            ispisPodaci += " glasa";
        } else{
            ispisPodaci += " glasova";
        }
        ispisPodaci += "</p></li>";
    }

    ispisPodaci += "</ul>";

    $('#izaberiteOdgovor').html(ispisPodaci);
},
error : function (xhr, status, error) {
    let poruka = "";
    switch (xhr.status) {
        case 404 :
            poruka = "Trenutno nije moguće videti rezultate glasanja";
            break;
        case 422:
            poruka = "Trenutno nema rezultata za izabrano pitanje";
            break;
    }
    if(poruka != "")
        $('#izaberiteOdgovor').html("<h4>" + poruka + "</h4>");
    });
}

function korpaLogovanProvera() {
    alert("Morate se prijaviti da bi pristupili korpi");
}

function profilLogovanProvera() {
    alert("Morate se prijaviti da bi pristupili profilu");
}

function slajder(){
    var trenutna = $('.trenutna');
    var sledeca = trenutna.next().length ? trenutna.next() : trenutna.parent().children(':first');
    trenutna.hide().removeClass('trenutna');
    sledeca.fadeIn('slow').addClass('trenutna');
    setTimeout(slajder, 5000);
}

```



## 3.4) PHP

### 3.4.1) functions

#### 3.4.1.1) nav

```
<?php
$navigaviijaObj = new Navigacija(Baza::instanca());
function nav()
{
    global $navigaviijaObj;
    function navigacija($roditelj){
        global $navigaviijaObj;
        $navigaviijaObj->setRoditelj($roditelj);
        $rezultatNav = $navigaviijaObj->dohvatiPoRoditelju();

        if ($rezultatNav){
            echo "<ul>";
        }
        foreach($rezultatNav as $nav){
            echo "<li><a href='". $nav->href. "'>". $nav->navigacija. "</a>";
            navigacija($nav->navigacijaID);
            echo "</li>";
        }
        if ($rezultatNav){
            echo "</ul>";
        }
    }

    $rezultatLinkovi = $navigaviijaObj->dohvatiPrviRed();

    echo "<ul id='navigacija'>";
    foreach($rezultatLinkovi as $link){
        if(isset($_SESSION['korisnik']) and ($link->navigacija == "PRIJAVI SE" or $link->navigacija
== "REGISTRACIJA")){
            continue;
        }
        if(!isset($_SESSION['korisnik']) and $link->navigacija == "ODJAVI SE"){
            continue;
        }

        echo "<li><a href='". $link->href. "'>". $link->navigacija. "</a>";
        navigacija($link->navigacijaID);
        echo "</li>";
    }
    echo "</ul>";
}
```

### 3.4.2) modules

#### 3.4.2.1) activate

```
<?php
require_once "../classes/SuperKlasa.php";
require_once "../classes/Korisnik.php";
require_once "../classes/Baza.php";

if(isset($_GET['activate'])){
    $korisnik = new Korisnik(Baza::instanca());
    $token = $_GET['activate'];
    $korisnik->setToken($token);

    $korisnikSelect = $korisnik->activateProvera();
    if($korisnikSelect){
        if($korisnikSelect->aktivan){
            echo "<h3>Nalog je već aktiviran</h3><h4><a href='https://luggage-
online.000webhostapp.com/index.php'>Početna stranica</a></h4>";
        } else{
            $korisnikActivate = $korisnik->activate();
            if($korisnikActivate){
                echo "<h3>Uspešna aktivacija naloga!</h3><h4><a href='https://luggage-
online.000webhostapp.com/index.php?page=login'>Prijavite se</a></h4>";
            } else {
                echo "<h3>Došlo je do greške</h3><h4><a href='https://luggage-
online.000webhostapp.com/index.php'>Početna stranica</a></h4>";
            }
        }
    } else{
        echo "<h3>Niste registrovani</h3><h4><a href='https://luggage-
online.000webhostapp.com/index.php&page=register'>Registrujte se</a></h4>";
    }
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
```

### 3.4.2.2) admin

```
<?php
session_start();

require_once "../classes/SuperKlasa.php";
require_once "../classes/Korisnik.php";
require_once "../classes/Proizvod.php";
require_once "../classes/Baza.php";

if(!isset($_SESSION['korisnik']))
{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
else
{
    if($_SESSION['korisnik']->uloga != "Admin")
    {
        header("Location: https://luggage-online.000webhostapp.com/index.php");
    }
    else
    {
        /*-----INSERT USER-----*/
        if(isset($_POST['addUserDugme'])) {
            header("Content-Type: application/json");
            $code = 404;
            $data = null;

            $regIme = trim($_POST['regIme']);
            $regPrezime = trim($_POST['regPrezime']);
            $regEmail = trim($_POST['regEmail']);
            $regKorIme = trim($_POST['regKorIme']);
            $regLozinka = md5($_POST['regLozinka']);
            $regLozinka2 = md5($_POST['regLozinka2']);
            $add_uloga = $_POST['add_uloga'];
            $add_aktivan = $_POST['add_aktivan'];

            $proveraImePrezime = "/^[A-ZŠĐČĆŽ][a-zšđćčž]{2,12}(\s[A-ZŠĐČĆŽ][a-zšđćčž]{2,12}){0,1}$/";
            $proveraEmail = "/^[\\w]+[\\.\\w\\d]*[\\w\\d]+@[\\w]+([\\.][\\w]+)+$/";
            $proveraKorIme = "/^[\\w\\d\\.\\_]{5,15}$/";
            $insPrGreske = [];

            if(!$regIme){
                $insPrGreske[] = "Polje za ime mora biti popunjeno";
            } else if(!preg_match($proveraImePrezime, $regIme)){
                $insPrGreske[] = "Ime nije u dobrom formatu";
            }
            if(!$regPrezime){
                $insPrGreske[] = "Polje za prezime mora biti popunjeno";
            } else if(!preg_match($proveraImePrezime, $regPrezime)){
                $insPrGreske[] = "Prezime nije u dobrom formatu";
            }
            if(!$regEmail){
                $insPrGreske[] = "Polje za email mora biti popunjeno";
            } else if(!preg_match($proveraEmail, $regEmail)){
                $insPrGreske[] = "Email nije u dobrom formatu";
            }
            if(!$regKorIme){
                $insPrGreske[] = "Polje za korisničko ime mora biti popunjeno";
            } else if(!preg_match($proveraKorIme, $regKorIme)){
                $insPrGreske[] = "Korisničko ime mora imati 5-15 karaktera";
            }
            if($_POST['regLozinka']){
                $insPrGreske[] = "Polje za lozinku mora biti popunjeno";
            } else if(strlen($_POST['regLozinka']) < 6){
                $insPrGreske[] = ("Lozinka mora imati bar 6 karaktera");
            }
            if($_POST['regLozinka2']){

```

```

        $insPrGreske[] = "Polje za ponovljenu lozinku mora biti popunjeno";
    } else if(strlen($_POST['regLozinka2']) < 6){
        $insPrGreske[] = "Ponovljena lozinka mora imati bar 6 karaktera";
    }
    if($_POST['regLozinka'] != $_POST['regLozinka2']){
        $insPrGreske[] = "Lozinka i ponovljena lozinka se ne poklapaju";
    }
    if($add_uloga == "0"){
        $insPrGreske[] = "Morate odabrati ulogu";
    }
    if($add_aktivan == "2"){
        $insPrGreske[] = "Morate odabrati aktivnost naloga";
    }
}

$time = addslashes($regIme);
$prezime = addslashes($regPrezime);
$email = addslashes($regEmail);
$korisnicko_ime = addslashes($regKorIme);
$lozinka = $regLozinka;
$aktivan = $add_aktivan;
$ulogaID = $add_uloga;

if(count($insPrGreske)){
    $code = 422;
    $data = $insPrGreske;
} else {
    $token = md5(sha1($email . $korisnicko_ime . time()));
    $reset_password = md5(sha1($ime . $email . time() . rand()));
    $datum_registracije = time();

    $korisnik = new Korisnik(Baza::instanca());
    $korisnik->setIme($ime);
    $korisnik->setPrezime($prezime);
    $korisnik->setEmail($email);
    $korisnik->setKorisnickoIme($korisnicko_ime);
    $korisnik->setLozinka($lozinka);
    $korisnik->setToken($token);
    $korisnik->setAktivan($aktivan);
    $korisnik->setDatumRegistracije($datum_registracije);
    $korisnik->setUlogaID($ulogaID);
    $korisnik->setResetPassword($reset_password);

    $code = $korisnik->insertKorisnik() ? 201 : 409;
}
http_response_code($code);
echo json_encode(['message' => $data]);
}

/*-----DELETE USER-----*/
if(isset($_POST['deleteUserDugme'])){
    $code = 404;
    $data = null;

    $korisnikID = $_POST['korisnikID'];

    $korisnik = new Korisnik(Baza::instanca());
    $korisnik->setKorisnikID($korisnikID);

    $code = $korisnik->deleteKorisnik() ? 204 : 500;

    http_response_code($code);
}

/*-----UPDATE USER-----*/
if(isset($_POST['updateUserDugme'])){
    $code = 404;

```

```

$data = null;

$regIme = trim($_POST['regIme']);
$regPrezime = trim($_POST['regPrezime']);
$regEmail = trim($_POST['regEmail']);
$regKorIme = trim($_POST['regKorIme']);
$add_uloga = $_POST['add_uloga'];
$add_aktivan = $_POST['add_aktivan'];
$idUpdate = $_POST['idUpdate'];

$proveraImePrezime = "/^[A-ZŠĐČĆŽ][a-zšđććž]{2,12}(\s[A-ZŠĐČĆŽ][a-
zšđććž]{2,12}){0,1}$/";
$proveraEmail = "/^[\\w]+[\\.\\w\\d]*[\\w\\d]+\\@[\\w]+(\\.\\w+)+$/";
$proveraKorIme = "/^[\\w\\d\\.\\_]{5,15}$/";
$insPrGreske = [];

if(!$regIme){
    $insPrGreske[] = "Polje za ime mora biti popunjeno";
} else if(!preg_match($proveraImePrezime, $regIme)){
    $insPrGreske[] = "Ime nije u dobrom formatu";
}
if(!$regPrezime){
    $insPrGreske[] = "Polje za prezime mora biti popunjeno";
} else if(!preg_match($proveraImePrezime, $regPrezime)){
    $insPrGreske[] = "Prezime nije u dobrom formatu";
}
if(!$regEmail){
    $insPrGreske[] = "Polje za email mora biti popunjeno";
} else if(!preg_match($proveraEmail, $regEmail)){
    $insPrGreske[] = "Email nije u dobrom formatu";
}
if(!$regKorIme){
    $insPrGreske[] = "Polje za korisničko ime mora biti popunjeno";
} else if(!preg_match($proveraKorIme, $regKorIme)){
    $insPrGreske[] = "Korisničko ime mora imati 5-15 karaktera";
}
if($add_uloga == "0"){
    $insPrGreske[] = "Morate odabrati ulogu";
}
if($add_aktivan == "2"){
    $insPrGreske[] = "Morate odabrati aktivnost naloga";
}

$ime = addslashes($regIme);
$prezime = addslashes($regPrezime);
$email = addslashes($regEmail);
$korisnicko_ime = addslashes($regKorIme);
$aktivan = $add_aktivan;
$ulogaID = $add_uloga;
$korisnikID = $idUpdate;

if(count($insPrGreske)){
    $code = 422;
    $data = $insPrGreske;
} else {
    $korisnik = new Korisnik(Baza::instanca());
    $korisnik->setIme($ime);
    $korisnik->setPrezime($prezime);
    $korisnik->setEmail($email);
    $korisnik->setKorisnickoIme($korisnicko_ime);
    $korisnik->setAktivan($aktivan);
    $korisnik->setUlogaID($ulogaID);
    $korisnik->setKorisnikID($korisnikID);

    $code = $korisnik->updateKorisnik() ? 204 : 409;
}
http_response_code($code);
echo json_encode(['message' => $data]);
}

```

```

/*-----INSERT PRODUCT-----*/
if(isset($_POST['addProductDugme'])){

    $insertProizvodNaziv = trim($_POST['insertProizvodNaziv']);
    $insertProizvodOpis = trim($_POST['insertProizvodOpis']);
    $insertProizvodSlika = $_FILES['insertProizvodSlika'];
    $insertProizvodCena = $_POST['insertProizvodCena'];
    $insertProizvodBrend = $_POST['insertProizvodBrend'];
    $insertProizvodPol = $_POST['insertProizvodPol'];
    $insertProizvodKategorija = $_POST['insertProizvodKategorija'];

    $insPrGreske = [];

    if(!$insertProizvodNaziv){
        $insPrGreske[] = "Polje za naziv mora biti popunjeno";
    } else if(strlen($insertProizvodNaziv) > 50){
        $insPrGreske[] = "Naziv ne sme biti duži od 50 karaktera";
    }
    if(!$insertProizvodOpis){
        $insPrGreske[] = "Polje za opis mora biti popunjeno";
    } else if(strlen($insertProizvodOpis) > 200){
        $insPrGreske[] = "Opis ne sme biti duži od 200 karaktera";
    }
    if(!$insertProizvodCena){
        $insPrGreske[] = "Polje za cenu mora biti popunjeno";
    }
    if($insertProizvodBrend == "0"){
        $insPrGreske[] = "Morate odabrati brend";
    }
    if($insertProizvodPol == "0"){
        $insPrGreske[] = "Morate odabrati pol";
    }
    if($insertProizvodKategorija == "0"){
        $insPrGreske[] = "Morate odabrati kategoriju";
    }
    if(!$insertProizvodSlika){
        $insPrGreske[] = "Morate odabrati sliku";
    }

    $uploadFolder = "../images/upload/";
    $tmpName = $insertProizvodSlika['tmp_name'];
    $velicinaSlike = $insertProizvodSlika['size'];
    $maxVelicina = 2048000;
    $nazivSlike = time() . $insertProizvodSlika['name'];
    $tip = $insertProizvodSlika['type'];
    $putanja = $uploadFolder . $nazivSlike;

    if($velicinaSlike > $maxVelicina){
        $insPrGreske[] = "Slika ne sme biti veća od 2MB";
    }

    $reSlika = "/image\jpg|image/jpeg|image/png/i";

    if(!preg_match($reSlika, $tip)){
        $insPrGreske[] = "Slika mora biti jpg, jpeg ili png formata";
    }

    $naziv = addslashes($insertProizvodNaziv);
    $opis = addslashes($insertProizvodOpis);
    $cena = $insertProizvodCena;
    $pol = $insertProizvodPol;
    $src = addslashes($nazivSlike);
    $salt = addslashes($insertProizvodNaziv);
    $kategorijaID = $insertProizvodKategorija;
    $brendID = $insertProizvodBrend;

    $prebacivanjeSlike = move_uploaded_file($tmpName, $putanja);

```

```

        if(!$prebacivanjeSlike){
            $insPrGreske[] = "Greška pri dodavanju slike";
        }

        if(count($insPrGreske)){
            foreach ($insPrGreske as $greska){
                echo "<h3>$greska</h3>";
            }
        } else {
            $proizvod = new Proizvod(Baza::instanca());
            $proizvod->setProizvod($naziv);
            $proizvod->setOpis($opis);
            $proizvod->setCena($cena);
            $proizvod->setPol($pol);
            $proizvod->setSrc($src);
            $proizvod->setAlt($alt);
            $proizvod->setKategorijaID($kategorijaID);
            $proizvod->setBrendID($brendID);

            $insertKorisnik = $proizvod->insertProizvod();
            if($insertKorisnik)
            {
                header('Refresh: 5; URL=https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allproducts');
                echo "<h3>Uspešno ste dodali proizvod!</h3><h4>Uskoro ćete biti prebačeni na
admin stranicu sa svim proizvodima</h4>";
            }
            else
            {
                header('Refresh: 5; URL=https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allproducts');
                echo "<h3>Greška pri dodavanju proizvoda, pokušajte kasnije.</h3><h4>Uskoro
ćete biti prebačeni na admin stranicu sa svim proizvodima</h4>";
            }
        }
    }

    /*-----DELETE PRODUCT-----*/
    if(isset($_POST['deleteProductDugme'])){
        $code = 404;
        $data = null;

        $proizvodID = $_POST['proizvodID'];

        $proizvod = new Proizvod(Baza::instanca());
        $proizvod->setProizvodID($proizvodID);

        $code = $proizvod->deleteProizvod() ? 204 : 500;

        http_response_code($code);
    }

    /*-----UPDATE PRODUCT-----*/
    if(isset($_POST['updateProductDugme'])){

        $updateProizvodNaziv = trim($_POST['updateProizvodNaziv']);
        $updateProizvodOpis = trim($_POST['updateProizvodOpis']);
        $updateProizvodSlika = $_FILES['insertProizvodSlika'];
        $updateProizvodCena = $_POST['insertProizvodCena'];
        $updateProizvodBrend = $_POST['insertProizvodBrend'];
        $updateProizvodPol = $_POST['insertProizvodPol'];
        $updateProizvodKategorija = $_POST['insertProizvodKategorija'];
        $updatePrID = $_POST['updatePrID'];

        $updPrGreske = [];
    }

```

```

if(!$updateProizvodNaziv){
    $updPrGreske[] = "Polje za naziv mora biti popunjeno";
} else if(strlen($updateProizvodNaziv) > 50){
    $updPrGreske[] = "Naziv ne sme biti duži od 50 karaktera";
}
if(!$updateProizvodOpis){
    $updPrGreske[] = "Polje za opis mora biti popunjeno";
} else if(strlen($updateProizvodOpis) > 200){
    $updPrGreske[] = "Opis ne sme biti duži od 200 karaktera";
}
if(!$updateProizvodCena){
    $updPrGreske[] = "Polje za cenu mora biti popunjeno";
}
if($updateProizvodBrend == "0"){
    $updPrGreske[] = "Morate odabrati brend";
}
if($updateProizvodPol == "0"){
    $updPrGreske[] = "Morate odabrati pol";
}
if($updateProizvodKategorija == "0"){
    $updPrGreske[] = "Morate odabrati kategoriju";
}

//AKO NIJE UPLOAD-ovao SLIKU
if($updateProizvodSlika['name'] == ""){
    $naziv = addslashes($updateProizvodNaziv);
    $opis = addslashes($updateProizvodOpis);
    $cena = $updateProizvodCena;
    $pol = $updateProizvodPol;
    $alt = addslashes($updateProizvodNaziv);
    $kategorijaID = $updateProizvodKategorija;
    $brendID = $updateProizvodBrend;
    $proizvodID = $updatePrID;

    if(count($updPrGreske)){
        foreach ($updPrGreske as $greska){
            echo "<h3>$greska</h3>";
        }
    } else {
        $proizvod = new Proizvod(Baza::instanca());
        $proizvod->setProizvod($naziv);
        $proizvod->setOpis($opis);
        $proizvod->setCena($cena);
        $proizvod->setPol($pol);
        $proizvod->setAlt($alt);
        $proizvod->setKategorijaID($kategorijaID);
        $proizvod->setBrendID($brendID);
        $proizvod->setProizvodID($proizvodID);

        $updateBezSlikeKorisnik = $proizvod->updateProizvodBezSlike();
        if($updateBezSlikeKorisnik)
        {
            header('Refresh: 5; URL=https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allproducts');
            echo "<h3>Uspešno ste izmenili proizvod!</h3><h4>Uskoro ćete biti prebačeni
na admin stranicu sa svim proizvodima</h4>";
        }
        else
        {
            header('Refresh: 5; URL=https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allproducts');
            echo "<h3>Greška pri izmeni proizvoda.</h3><h4>Uskoro ćete biti prebačeni
na admin stranicu sa svim proizvodima</h4>";
        }
    }
}

// AKO JESTE UPLOAD-ovao SLIKU

```



```

    } else {
        $uploadFolder = "../images/upload/";
        $tmpName = $updateProizvodSlika['tmp_name'];
        $velicinaSlike = $updateProizvodSlika['size'];
        $maxVelicina = 2048000;
        $nazivSlike = time() . $updateProizvodSlika['name'];
        $tip = $updateProizvodSlika['type'];
        $putanja = $uploadFolder . $nazivSlike;

        if($velicinaSlike > $maxVelicina){
            $updPrGreske[] = "Slika ne sme biti veća od 2MB";
        }

        $reSlika = "/image\jpg|image\jpeg|image/png/i";

        if(!preg_match($reSlika, $tip)){
            $updPrGreske[] = "Slika mora biti jpg, jpeg ili png formata";
        }

        $naziv = addslashes($updateProizvodNaziv);
        $opis = addslashes($updateProizvodOpis);
        $cena = $updateProizvodCena;
        $pol = $updateProizvodPol;
        $src = addslashes($nazivSlike);
        $alt = addslashes($updateProizvodNaziv);
        $kategorijaID = $updateProizvodKategorija;
        $brendID = $updateProizvodBrend;
        $proizvodID = $updatePrID;

        $prebacivanjeSlike = move_uploaded_file($tmpName, $putanja);

        if(!$prebacivanjeSlike){
            $updPrGreske[] = "Greška pri dodavanju slike";
        }

        if(count($updPrGreske)){
            foreach ($updPrGreske as $greska){
                echo "<h3>$greska</h3>";
            }
        } else {
            $proizvod = new Proizvod(Baza::instanca());
            $proizvod->setProizvod($naziv);
            $proizvod->setOpis($opis);
            $proizvod->setCena($cena);
            $proizvod->setPol($pol);
            $proizvod->setSrc($src);
            $proizvod->setAlt($alt);
            $proizvod->setKategorijaID($kategorijaID);
            $proizvod->setBrendID($brendID);
            $proizvod->setProizvodID($proizvodID);

            $updateSaSlikomKorisnik = $proizvod->updateProizvodSaSlikom();
            if ($updateSaSlikomKorisnik) {
                header('Refresh: 5; URL=https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allproducts');
                echo "<h3>Uspešno ste izmenili proizvod!</h3><h4>Uskoro ćete biti prebačeni
na admin stranicu sa svim proizvodima</h4>";
            } else {
                header('Refresh: 5; URL=https://luggage-
online.000webhostapp.com/index.php?page=admin&adminaction=allproducts');
                echo "<h3>Greška pri izmeni proizvoda.</h3><h4>Uskoro ćete biti prebačeni
na admin stranicu sa svim proizvodima</h4>";
            }
        }
    }
}

if(empty($_POST['addUserDugme']) and empty($_POST['deleteUserDugme']) and
empty($_POST['updateUserDugme']) and empty($_POST['addProductDugme']) and

```

```
empty($_POST['deleteProductDugme']) and empty($_POST['updateProductDugme'])) {  
    header("Location: https://luggage-online.000webhostapp.com/index.php");  
}  
  
}
```

### 3.4.2.3) changepassword

```
<?php
session_start();

require_once "../classes/SuperKlasa.php";
require_once "../classes/Korisnik.php";
require_once "../classes/Baza.php";

if(!isset($_SESSION['korisnik']))
{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
else
{
    if(isset($_POST['promenaLozinkeDugme']))
    {
        header("Content-Type: application/json");
        $code = 404;
        $data = null;

        $korisnikID = $_POST['korisnikID'];
        $trenutnaLozinka = md5($_POST['trenutnaLozinka']);
        $novaLozinka = md5($_POST['novaLozinka']);
        $novaLozinka2 = md5($_POST['novaLozinka2']);

        $promenaLozinkeGreske = [];

        if(!$_POST['trenutnaLozinka'])
        {
            $promenaLozinkeGreske[] = "Polje za trenutnu lozinku mora biti popunjeno";
        }
        else if(strlen($_POST['trenutnaLozinka']) < 6)
        {
            $promenaLozinkeGreske[] = ("Trenutna lozinka mora imati bar 6 karaktera");
        }

        if(!$_POST['novaLozinka'])
        {
            $promenaLozinkeGreske[] = "Polje za novu lozinku mora biti popunjeno";
        }
        else if(strlen($_POST['novaLozinka']) < 6)
        {
            $promenaLozinkeGreske[] = ("Nova lozinka mora imati bar 6 karaktera");
        }

        if(!$_POST['novaLozinka2'])
        {
            $promenaLozinkeGreske[] = "Polje za ponovljenu lozinku mora biti popunjeno";
        }
        else if(strlen($_POST['novaLozinka2']) < 6)
        {
            $promenaLozinkeGreske[] = ("Ponovljena lozinka mora imati bar 6 karaktera");
        }

        if($_POST['novaLozinka'] !== $_POST['novaLozinka2'])
        {
            $promenaLozinkeGreske[] = "Nova lozinka i ponovljena lozinka se ne poklapaju";
        }

        if(count($promenaLozinkeGreske))
        {
            $code = 422;
            $data = $promenaLozinkeGreske;
        }
        else
        {
            $korisnik = new Korisnik(Baza::instanca());
            $korisnik->setKorisnikID($korisnikID);
            $korisnik->setLozinka($trenutnaLozinka);

            $korisnikSelect = $korisnik->promenaLozinkeProvera();

            if($korisnikSelect)
            {
                $korisnik->setKorisnikID($korisnikID);
                $korisnik->setLozinka($novaLozinka);

                $code = $korisnik->promeniLozinku() ? 204 : 500;
            }
            else
            {
                $code = 409;
            }
        }
    }
}
```

```
    }  
  }  
  http_response_code($code);  
  echo json_encode(['message' => $data]);  
} else{  
  header("Location: https://luggage-online.000webhostapp.com/index.php");  
}  
}
```

### 3.4.2.4) filter

```
<?php
require_once "../classes/SuperKlasa.php";
require_once "../classes/Proizvod.php";
require_once "../classes/Baza.php";

if(isset($_POST['filterDugme']))
{
    header("Content-Type: application/json");
    $data = null;

    $sortiraj = $_POST['sortiraj'];
    $kategorijeStr = $_POST['kategorijeStr'];
    $brendoviStr = $_POST['brendoviStr'];
    $poloviStr = $_POST['poloviStr'];
    $minCena = $_POST['minCena'];
    $maxCena = $_POST['maxCena'];

    $nizPodaci = [];
    if($kategorijeStr)
        $nizPodaci['kategorijaID'] = $kategorijeStr;
    if($brendoviStr)
        $nizPodaci['brendID'] = $brendoviStr;
    if($poloviStr)
        $nizPodaci['pol'] = $poloviStr;
    if($minCena)
        $nizPodaci['minCena'] = $minCena;
    if($maxCena)
        $nizPodaci['maxCena'] = $maxCena;

    $upit = "SELECT * FROM proizvod p INNER JOIN kategorija k ON p.kategorijaID = k.kategorijaID
            INNER JOIN brend b ON p.brendID = b.brendID";

    $nizDeloviUpita = [];
    if(count($nizPodaci))
    {
        foreach($nizPodaci as $in => $vr)
        {
            if($in == "minCena")
            {
                $nizDeloviUpita[] = "p.cena >= " . $vr;
            }
            else if($in == "maxCena")
            {
                $nizDeloviUpita[] = "p.cena <= " . $vr;
            }
            else
            {
                $nizDeloviUpita[] = "p." . $in . " IN " . "(" . $vr . ")";
            }
        }
    }

    $podaciStr = "";

    if(count($nizDeloviUpita))
        $podaciStr = implode(" AND ", $nizDeloviUpita);

    if($podaciStr)
        $upit = $upit . " WHERE ";

    $upit = $upit . $podaciStr;

    $dodatakSortiranje = "";
    if($sortiraj != 0)
    {
        if($sortiraj == 1)
        {

```

```
        $dodatakSortiranje = " ORDER BY p.cena ASC";
    }
    else if($sortiraj == 2)
    {
        $dodatakSortiranje = " ORDER BY p.cena DESC";
    }
    else if($sortiraj == 3)
    {
        $dodatakSortiranje = " ORDER BY p.proizvod ASC";
    }
    else
    {
        $dodatakSortiranje = " ORDER BY p.proizvod DESC";
    }
}

$upit = $upit . $dodatakSortiranje;

$proizvod = new Proizvod(Baza::instanca());
$data = $proizvod->filter($upit);
if(!$data){
    $data = null;
}

echo json_encode(['message' => $data]);
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
```

### 3.4.2.5) login

```
<?php
session_start();

require_once "../classes/SuperKlasa.php";
require_once "../classes/Korisnik.php";
require_once "../classes/Baza.php";

if(isset($_POST['loginDugme'])){
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $loginKorIme = trim($_POST['loginKorIme']);
    $loginLozinka = md5($_POST['loginLozinka']);
    $reKorIme = "/^[\\w\\d\\.\\_]{5,15}$/";
    $loginGreske = [];

    if(!$loginKorIme){
        $loginGreske[] = "Polje za korisničko ime mora biti popunjeno";
    } else if(!preg_match($reKorIme, $loginKorIme)){
        $loginGreske[] = "Korisničko ime mora imati 5-15 karaktera";
    }
    if(!$_POST['loginLozinka']){
        $loginGreske[] = "Polje za lozinku mora biti popunjeno";
    } else if(strlen($_POST['loginLozinka']) < 6){
        $loginGreske[] = ("Lozinka mora imati bar 6 karaktera");
    }

    $korisnicko_ime = addslashes($loginKorIme);
    $lozinka = $loginLozinka;

    if(count($loginGreske)){
        $code = 422;
        $data = $loginGreske;
    } else{
        $korisnik = new Korisnik(Baza::instanca());
        $korisnik->setKorisnickoIme($korisnicko_ime);
        $korisnik->setLozinka($lozinka);

        $korisnikSelect = $korisnik->loginProvera();
        if($korisnikSelect){
            $_SESSION['korisnik'] = $korisnikSelect;
            $data = $korisnikSelect->uloga;
            $code = 200;
        } else{
            $code = 409;
        }
    }
    http_response_code($code);
    echo json_encode(['message' => $data]);
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
```

### 3.4.2.6) logout

```
<?php
session_start();
if(isset($_SESSION['korisnik'])){
    unset($_SESSION['korisnik']);
    session_destroy();
    header("Location: https://luggage-online.000webhostapp.com/index.php");
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
```



### 3.4.2.7) mailadmin

```
<?php
session_start();
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'phpmailer/src/Exception.php';
require 'phpmailer/src/PHPMailer.php';
require 'phpmailer/src/SMTP.php';

if(isset($_POST['contactDugme'])) {
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $contactEmail = trim($_POST['contactEmail']);
    $contactPoruka = $_POST['poruka'];
    $reEmail = "/^[\\w]+[\\.\\w\\d]*[\\w\\d]+@[\\w]+([\\.]?[\\w]+)+$/";
    $contactGreske = [];

    if(!$contactEmail) {
        $contactGreske[] = "Polje za email mora biti popunjeno";
    } else if(!preg_match($reEmail, $contactEmail)) {
        $contactGreske[] = "Email nije u dobrom formatu";
    }

    if(!$contactPoruka) {
        $contactGreske[] = "Polje za poruku mora biti popunjeno";
    } else if(strlen($contactPoruka) < 15 or strlen($contactPoruka) > 200) {
        $contactGreske[] = "Poruka mora imati 15-200 karaktera";
    }

    if(count($contactGreske)) {
        $code = 422;
        $data = $contactGreske;
    } else {
        $mail = new PHPMailer(true);
        try {
            //Server settings
            // $mail->SMTPDebug = 2; // Enable verbose
            debug output
            /* $mail->SMTPOptions = array(
                'ssl' => array(
                    'verify_peer' => false,
                    'verify_peer_name' => false,
                    'allow_self_signed' => true
                )
            ); */
            $mail->isSMTP(); // Set mailer to
            use SMTP
            $mail->Host = 'smtp.gmail.com'; // Specify main and
            backup SMTP servers
            $mail->SMTPAuth = true; // Enable SMTP
            authentication
            $mail->Username = ''; // SMTP username
            $mail->Password = ''; // SMTP password
            $mail->SMTPSecure = 'tls'; // Enable TLS
            encryption, `ssl` also accepted
            $mail->Port = 587; // TCP port to
            connect to

            //Recipients
            $mail->setFrom('', $contactEmail);
            $mail->addAddress(""); // Add a recipient
            // $mail->addAddress('ellen@example.com'); // Name is optional
            // $mail->addReplyTo('info@example.com', 'Information');
            // $mail->addCC('cc@example.com');
```

```

        // $mail->addBCC('bcc@example.com');

        // Attachments
        // $mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments
        // $mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name

        // Content
        $mail->isHTML(true); // Set email format
to HTML

        $mail->Subject = 'Poruka za admina';
        $mail->Body = "$contactPoruka<br/><br/>Email: $contactEmail";
        // $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

        $mail->send();
        //echo 'Message has been sent';
        $code = 200;
    } catch (Exception $e) {
        //echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
        $code = 500;
    }
}
http_response_code($code);
echo json_encode(['message' => $data]);
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}

```

### 3.4.2.8) register

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'phpmailer/src/Exception.php';
require 'phpmailer/src/PHPMailer.php';
require 'phpmailer/src/SMTP.php';

require_once '../classes/SuperKlasa.php';
require_once '../classes/Korisnik.php';
require_once '../classes/Baza.php';

if(isset($_POST['regDugme'])) {
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $regIme = trim($_POST['regIme']);
    $regPrezime = trim($_POST['regPrezime']);
    $regEmail = trim($_POST['regEmail']);
    $regKorIme = trim($_POST['regKorIme']);
    $regLozinka = md5($_POST['regLozinka']);
    $regLozinka2 = md5($_POST['regLozinka2']);

    $proveraImePrezime = "/^[A-ZŠĐČĆŽ][a-zšđćčž]{2,12}(\s[A-ZŠĐČĆŽ][a-zšđćčž]{2,12}){0,1}$/";
    $proveraEmail = "/^[\\w]+[\\.\\w\\d]*[\\w\\d]+@[\\w]+([\\.][\\w]+)+$/";
    $proveraKorIme = "/^[\\w\\d\\.\\_]{5,15}$/";
    $regGreske = [];

    if(!$regIme) {
        $regGreske[] = "Polje za ime mora biti popunjeno";
    } else if(!preg_match($proveraImePrezime, $regIme)) {
        $regGreske[] = "Ime nije u dobrom formatu";
    }
    if(!$regPrezime) {
        $regGreske[] = "Polje za prezime mora biti popunjeno";
    } else if(!preg_match($proveraImePrezime, $regPrezime)) {
        $regGreske[] = "Prezime nije u dobrom formatu";
    }
    if(!$regEmail) {
        $regGreske[] = "Polje za email mora biti popunjeno";
    } else if(!preg_match($proveraEmail, $regEmail)) {
        $regGreske[] = "Email nije u dobrom formatu";
    }
    if(!$regKorIme) {
        $regGreske[] = "Polje za korisničko ime mora biti popunjeno";
    } else if(!preg_match($proveraKorIme, $regKorIme)) {
        $regGreske[] = "Korisničko ime mora imati 5-15 karaktera";
    }
    if(!$_POST['regLozinka']) {
        $regGreske[] = "Polje za lozinku mora biti popunjeno";
    } else if(strlen($_POST['regLozinka']) < 6) {
        $regGreske[] = "Lozinka mora imati bar 6 karaktera";
    }
    if(!$_POST['regLozinka2']) {
        $regGreske[] = "Polje za ponovljenu lozinku mora biti popunjeno";
    } else if(strlen($_POST['regLozinka2']) < 6) {
        $regGreske[] = "Ponovljena lozinka mora imati bar 6 karaktera";
    }
    if($_POST['regLozinka'] !== $_POST['regLozinka2']) {
        $regGreske[] = "Lozinka i ponovljena lozinka se ne poklapaju";
    }

    $ime = addslashes($regIme);
    $prezime = addslashes($regPrezime);
    $email = addslashes($regEmail);
    $korisnicko_ime = addslashes($regKorIme);
    $lozinka = $regLozinka;
```

```

if(count($regGreske)){
    $code = 422;
    $data = $regGreske;
} else{
    $token = md5(shal($email . $korisnicko_ime . time()));
    $reset_password = md5(shal($ime . $email . time() . rand()));
    $datum_registracije = time();

    $korisnik = new Korisnik(Baza::instanca());
    $korisnik->setIme($ime);
    $korisnik->setPrezime($prezime);
    $korisnik->setEmail($email);
    $korisnik->setKorisnickoIme($korisnicko_ime);
    $korisnik->setLozinka($lozinka);
    $korisnik->setToken($token);
    $korisnik->setAktivan(0);
    $korisnik->setDatumRegistracije($datum_registracije);
    $korisnik->setResetPassword($reset_password);

    $code = $korisnik->registerKorisnik() ? 201 : 409;
    if($code == 201){
        $mail = new PHPMailer(true);
        try {
            //Server settings
            //$mail->SMTPDebug = 2; // Enable verbose debug

            /*$mail->SMTPOptions = array(
                'ssl' => array(
                    'verify_peer' => false,
                    'verify_peer_name' => false,
                    'allow_self_signed' => true
                )
            );*/
            $mail->isSMTP(); // Set mailer to use

            $mail->Host = 'smtp.gmail.com'; // Specify main and
            backup SMTP servers
            $mail->SMTPAuth = true; // Enable SMTP
            authentication
            $mail->Username = ''; // SMTP username
            $mail->Password = ''; // SMTP password
            $mail->SMTPSecure = 'tls'; // Enable TLS
            encryption, `ssl` also accepted
            $mail->Port = 587; // TCP port to connect
            to

            //Recipients
            $mail->setFrom('', 'Luggage');
            $mail->addAddress($email, $ime . " " . $prezime); // Add a recipient
            //$mail->addAddress('ellen@example.com'); // Name is optional
            //$mail->addReplyTo('info@example.com', 'Information');
            //$mail->addCC('cc@example.com');
            //$mail->addBCC('bcc@example.com');

            //Attachments
            //$mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments
            //$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name

            //Content
            $mail->isHTML(true); // Set email format to

            $mail->Subject = 'Aktivacija naloga';
            $mail->Body = 'Verifikacija email-a i aktivacija naloga: <a
href="https://luggage-online.000webhostapp.com/modules/activate.php?activate='.$token.'"
target="_blank">KLIKNITE OVDE</a>';
            //$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

            $mail->send();
            //echo 'Message has been sent';

```

```
        $code = 200;
    } catch (Exception $e) {
        //echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
        $code = 500;
    }
}
http_response_code($code);
echo json_encode(['message' => $data]);
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
```

### 3.4.2.9) shop

```
<?php
session_start();
require_once "../classes/SuperKlasa.php";
require_once "../classes/Korpa.php";
require_once "../classes/Baza.php";

if(!isset($_SESSION['korisnik']))
{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
else
{
    if(isset($_POST['dodajUKorpuDugme'])) {
        $code = 404;

        $korisnikID = $_POST['korisnikID'];
        $proizvodID = $_POST['proizvodID'];

        $korpa = new Korpa(Baza::instanca());
        $korpa->setKorisnikID($korisnikID);
        $korpa->setProizvodID($proizvodID);

        $code = $korpa->dodajUKorpu() ? 201 : 500;

        http_response_code($code);
    }

    if(isset($_POST['izbaciIzKorpeDugme'])) {
        $code = 404;

        $proizvodID = $_POST['proizvodID'];
        $korisnikID = $_POST['korisnikID'];

        $korpa = new Korpa(Baza::instanca());
        $korpa->setProizvodID($proizvodID);
        $korpa->setKorisnikID($korisnikID);

        $code = $korpa->izbaciIzKorpe() ? 204 : 500;

        http_response_code($code);
    }

    if(isset($_POST['kupiDugme'])) {
        $code = 404;

        $korisnikID = $_POST['korisnikID'];

        $korpa = new Korpa(Baza::instanca());
        $korpa->setKorisnikID($korisnikID);

        $code = $korpa->kupi() ? 204 : 500;

        http_response_code($code);
    }

    if(empty($_POST['dodajUKorpuDugme']) and empty($_POST['izbaciIzKorpeDugme']) and
empty($_POST['kupiDugme'])) {
        header("Location: https://luggage-online.000webhostapp.com/index.php");
    }
}
```

### 3.4.2.10) survey

```
<?php
session_start();
require_once "../classes/SuperKlasa.php";
require_once "../classes/Odgovor.php";
require_once "../classes/Anketa.php";
require_once "../classes/Baza.php";

if(isset($_POST['dohvatiOdgovoreDugme'])) {
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $pitanjeID = $_POST['pitanjeID'];

    $odgovor = new Odgovor(Baza::instanca());
    $odgovor->setPitanjeID($pitanjeID);

    $odgovori = $odgovor->dohvatiOdgovore();

    $code = $odgovori ? 200 : 409;
    $data = $odgovori;

    http_response_code($code);
    echo json_encode(['message' => $data]);
}

if(isset($_POST['glasajDugme'])) {
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $odgovorID = $_POST['odgovorID'];
    $pitanjeID = $_POST['pitanjeID'];
    $korisnikID = $_SESSION['korisnik']->korisnikID;

    $anketa = new Anketa(Baza::instanca());
    $anketa->setOdgovorID($odgovorID);
    $anketa->setPitanjeID($pitanjeID);
    $anketa->setKorisnikID($korisnikID);

    $odgovor = $anketa->glasanjeProvera();
    if($odgovor) {
        $code = 422;
    } else {
        $code = $anketa->glasaj() ? 201 : 500;
    }

    http_response_code($code);
    echo json_encode(['message' => $data]);
}

if(isset($_POST['dohvatiRezultateDugme'])) {
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $pitanjeID = $_POST['pitanjeID'];

    $anketa = new Anketa(Baza::instanca());
    $anketa->setPitanjeID($pitanjeID);

    $rezultati = $anketa->dohvatiRezultateGlasanja();
    if($rezultati) {
        $data = $rezultati;
    }
}
```

```
        $code = 200;
    } else{
        $code = 422;
    }

    http_response_code($code);
    echo json_encode(['message' => $data]);
}

if(empty($_POST['dohvatiOdgovoreDugme']) and empty($_POST['glasajDugme']) and
empty($_POST['dohvatiRezultateDugme'])) {
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
```



### 3.4.2.11) pagination

```
<?php
require_once "../classes/SuperKlasa.php";
require_once "../classes/Proizvod.php";
require_once "../classes/Baza.php";

if(isset($_POST['paginacijaDugme']))
{
    header("Content-Type: application/json");
    $data = null;

    $sortiraj = $_POST['sortiraj'];
    $kategorijeStr = $_POST['kategorijeStr'];
    $brendoviStr = $_POST['brendoviStr'];
    $poloviStr = $_POST['poloviStr'];
    $minCena = $_POST['minCena'];
    $maxCena = $_POST['maxCena'];
    $from = $_POST['from'];
    $poStranici = $_POST['poStranici'];

    $nizPodaci = [];
    if($kategorijeStr)
        $nizPodaci['kategorijaID'] = $kategorijeStr;
    if($brendoviStr)
        $nizPodaci['brendID'] = $brendoviStr;
    if($poloviStr)
        $nizPodaci['pol'] = $poloviStr;
    if($minCena)
        $nizPodaci['minCena'] = $minCena;
    if($maxCena)
        $nizPodaci['maxCena'] = $maxCena;

    $upit = "SELECT * FROM proizvod p INNER JOIN kategorija k ON p.kategorijaID = k.kategorijaID
            INNER JOIN brend b ON p.brendID = b.brendID";

    $nizDeloviUpita = [];
    if(count($nizPodaci))
    {
        foreach($nizPodaci as $in => $vr)
        {
            if($in == "minCena")
            {
                $nizDeloviUpita[] = "p.cena >= " . $vr;
            }
            else if($in == "maxCena")
            {
                $nizDeloviUpita[] = "p.cena <= " . $vr;
            }
            else
            {
                $nizDeloviUpita[] = "p." . $in . " IN " . "(" . $vr . ")";
            }
        }
    }

    $podaciStr = "";

    if(count($nizDeloviUpita))
        $podaciStr = implode(" AND ", $nizDeloviUpita);

    if($podaciStr)
        $upit = $upit . " WHERE ";

    $upit = $upit . $podaciStr;

    $dodatakSortiranje = "";
    if($sortiraj != 0)
    {

```

```
if($sortiraj == 1)
{
    $dodatakSortiranje = " ORDER BY p.cena ASC";
}
else if($sortiraj == 2)
{
    $dodatakSortiranje = " ORDER BY p.cena DESC";
}
else if($sortiraj == 3)
{
    $dodatakSortiranje = " ORDER BY p.proizvod ASC";
}
else
{
    $dodatakSortiranje = " ORDER BY p.proizvod DESC";
}

$upit = $upit . $dodatakSortiranje;

$dodatakLimit = " LIMIT " . $from . "," . $poStranici;
$upit = $upit . $dodatakLimit;

$proizvod = new Proizvod(Baza::instanca());
$data = $proizvod->filter($upit);
if(!$data){
    $data = null;
}

echo json_encode(['message' => $data]);
} else{
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}
```

### 3.4.2.12) resetpassword

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'phpmailer/src/Exception.php';
require 'phpmailer/src/PHPMailer.php';
require 'phpmailer/src/SMTP.php';

require_once '../classes/SuperKlasa.php';
require_once '../classes/Korisnik.php';
require_once '../classes/Baza.php';

if(isset($_POST['resetPassEmailDugme'])) {
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $email = trim($_POST['email']);
    $proveraEmail = "/^[\\w]+[\\.\\w\\d]*[\\w\\d]+\\@[\\w]+([\\.][\\w]+)+$/";
    $greska = "";

    if(!$email){
        $greska = "Polje za email mora biti popunjeno";
    } else if(!preg_match($proveraEmail, $email)){
        $greska = "Email nije u dobrom formatu";
    }

    if($greska !== ""){
        $code = 422;
        $data = $greska;
    } else {
        $korisnik = new Korisnik(Baza::instanca());
        $korisnik->setEmail($email);
        $korisnikSelect = $korisnik->resetLozinkeEmailProvera();

        if(!$korisnikSelect){
            $code = 409;
        } else {
            $token = md5(sha1($email . rand() . time() . rand()));
            $korisnik->setResetPassword($token);
            $korisnik->setEmail($email);
            $korisnik->resetLozinkeEmailUspesnaProvera();

            $mail = new PHPMailer(true);
            try {
                //Server settings
                // $mail->SMTPDebug = 2; // Enable verbose debug

                /*$mail->SMTPOptions = array(
                    'ssl' => array(
                        'verify_peer' => false,
                        'verify_peer_name' => false,
                        'allow_self_signed' => true
                    )
                );*/
                $mail->isSMTP(); // Set mailer to use

                $mail->Host = 'smtp.gmail.com'; // Specify main and
                backup SMTP servers
                $mail->SMTPAuth = true; // Enable SMTP
                authentication
                $mail->Username = 'auditorne.php@gmail.com'; // SMTP username
                $mail->Password = 'Sifra123'; // SMTP password
                $mail->SMTPSecure = 'tls'; // Enable TLS
                encryption, `ssl` also accepted
                $mail->Port = 587; // TCP port to connect
                to
```

```

        //Recipients
$mail->setFrom('auditorne.php@gmail.com', 'Luggage');
$mail->addAddress($email); // Add a recipient
$mail->addAddress('ellen@example.com'); // Name is optional
$mail->addReplyTo('info@example.com', 'Information');
$mail->addCC('cc@example.com');
$mail->addBCC('bcc@example.com');

//Attachments
$mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments
$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name

//Content
$mail->isHTML(true); // Set email format to HTML

$mail->Subject = 'Resetovanje lozinke';
$mail->Body = 'Resetovanje lozinke: <a href="https://luggage-
online.000webhostapp.com/modules/resetpassword.php?reset='. $token ." target="_blank">KLIKNITE
OVDE</a>';

$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';

$mail->send();
//echo 'Message has been sent';
$code = 200;
} catch (Exception $e) {
    //echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
    $code = 500;
}
}

http_response_code($code);
echo json_encode(['message' => $data]);
}

if(isset($_GET['reset'])){
    $token = $_GET['reset'];
    $korisnik = new Korisnik(Baza::instanca());
    $korisnik->setResetPassword($token);

    $korisnikSelect = $korisnik->resetLozinkeTokenProvera();
    if(!$korisnikSelect){
        header('Refresh: 5; URL=https://luggage-online.000webhostapp.com/index.php');
        echo "<h3>Nevažeći zahtev za resetovanje lozinke</h3>";
    } else{
        header('Location: https://luggage-
online.000webhostapp.com/index.php?page=resetpassword&reset=' . $token);
    }
}

if(isset($_POST['resetLozinkeDugme'])){
    header("Content-Type: application/json");
    $code = 404;
    $data = null;

    $token = $_POST['token'];
    $novaLozinka = md5($_POST['novaLozinka']);
    $novaLozinka2 = md5($_POST['novaLozinka2']);
    $promenaLozinkeGreske = [];

    if(!$_POST['novaLozinka']){
        $promenaLozinkeGreske[] = "Polje za novu lozinku mora biti popunjeno";
    } else if(strlen($_POST['novaLozinka']) < 6){
        $promenaLozinkeGreske[] = ("Nova lozinka mora imati bar 6 karaktera");
    }
}

```

```

if(!$_POST['novaLozinka2']){
    $promenaLozinkeGreske[] = "Polje za ponovljenu lozinku mora biti popunjeno";
} else if(strlen($_POST['novaLozinka2']) < 6){
    $promenaLozinkeGreske[] = ("Ponovljena lozinka mora imati bar 6 karaktera");
}

if($_POST['novaLozinka'] !== $_POST['novaLozinka2']){
    $promenaLozinkeGreske[] = "Nova lozinka i ponovljena lozinka se ne poklapaju";
}

if(count($promenaLozinkeGreske)){
    $code = 422;
    $data = $promenaLozinkeGreske;
} else{
    $noviToken = md5(sha1(time() . rand() . time() . rand() . time() . rand()));
    $korisnik = new Korisnik(Baza::instanca());
    $korisnik->setToken($token);
    $korisnik->setResetPassword($noviToken);
    $korisnik->setLozinka($novaLozinka);

    $code = $korisnik->resetLozinke() ? 204 : 500;
}

http_response_code($code);
echo json_encode(['message' => $data]);
}

if(empty($_POST['resetPassEmailDugme']) && empty($_GET['reset']) &&
empty($_POST['resetLozinkeDugme'])){
    header("Location: https://luggage-online.000webhostapp.com/index.php");
}

```

### 3.5) XML – sitemap.xml (već priloženo u poglavlju 2.2)

## 4. KLASA (classes)

### 4.1) Baza

```
<?php

class Baza
{
    private $host;
    private $baza;
    private $username;
    private $password;

    private $konekcija;
    private static $instanca;

    private function __construct()
    {
        $this->host = "localhost";
        $this->baza = "php2sajt1baza";
        $this->username = "root";
        $this->password = "";

        $this->connect();
    }

    public static function instanca()
    {
        if(self::$$instanca == null){
            self::$$instanca = new Baza();
        }
        return self::$$instanca;
    }

    private function connect()
    {
        try
        {
            $this->konekcija = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->baza .
";charset=utf8", $this->username, $this->password);
            $this->konekcija -> setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $this->konekcija -> setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
        }
        catch (PDOException $e)
        {
            echo "Konekcija sa bazom nije uspela: " . $e->getMessage();
        }
    }

    public function izvrsiSelect($upit, $podaci = null)
    {
        $priprema = $this->konekcija->prepare($upit);
        try
        {
            if($podaci == null)
            {
                $priprema->execute();
            }
            else
            {
                $priprema->execute($podaci);
            }
            return $priprema;
        }
        catch (PDOException $e)
        {
            echo "Došlo je do greške: " . $e->getMessage();
        }
    }
}
```

```
}  
}  
  
public function izvrsiInsertUpdateDelete($upit, $podaci)  
{  
    $priprema = $this->konekcija->prepare($upit);  
    try  
    {  
        $priprema->execute($podaci);  
        return $priprema;  
    }  
    catch (PDOException $e)  
    {  
        return;  
        //echo "Došlo je do greške: " . $e->getMessage();  
    }  
}
```

## 4.2) Anketa

```
<?php

class Anketa extends SuperKlasa
{
    private $anketaID;
    private $korisnikID;
    private $odgovorID;
    private $pitanjeID;

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    //region Get i Set
    public function getAnketaID()
    {
        return $this->anketaID;
    }

    public function setAnketaID($anketaID)
    {
        $this->anketaID = $anketaID;
    }

    public function getKorisnikID()
    {
        return $this->korisnikID;
    }

    public function setKorisnikID($korisnikID)
    {
        $this->korisnikID = $korisnikID;
    }

    public function getOdgovorID()
    {
        return $this->odgovorID;
    }

    public function setOdgovorID($odgovorID)
    {
        $this->odgovorID = $odgovorID;
    }

    public function getPitanjeID()
    {
        return $this->pitanjeID;
    }

    public function setPitanjeID($pitanjeID)
    {
        $this->pitanjeID = $pitanjeID;
    }
    //endregion

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM anketa";
        return parent::dohvatiSveSuper($upit);
    }

    public function glasanjeProvera()
    {
        $upit = "SELECT * FROM anketa WHERE pitanjeID = :pitanjeID AND korisnikID = :korisnikID";
        $podaci =
            [
                "pitanjeID" => $this->pitanjeID,
```



```

        "korisnikID" => $this->korisnikID
    ];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}

public function glasaj()
{
    $upit = "INSERT INTO anketa (pitanjeID, odgovorID, korisnikID) VALUES (:pitanjeID,
:odgovorID, :korisnikID)";
    $podaci =
    [
        "pitanjeID" => $this->pitanjeID,
        "odgovorID" => $this->odgovorID,
        "korisnikID" => $this->korisnikID
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function dohvatiRezultateGlasanja()
{
    $upit = "SELECT COUNT(a.korisnikID) AS broj_glasova, o.odgovor FROM anketa a INNER JOIN
odgovor o ON a.odgovorID = o.odgovorID
        WHERE a.pitanjeID = :pitanjeID GROUP BY o.odgovor";
    $podaci = ["pitanjeID" => $this->pitanjeID];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetchAll();
}
}

```

### 4.3) Brend

```
<?php

class Brend extends SuperKlasa
{
    private $brendID;
    private $brend;

    //region Get i Set
    public function getBrendID()
    {
        return $this->brendID;
    }

    public function setBrendID($brendID)
    {
        $this->brendID = $brendID;
    }

    public function getBrend()
    {
        return $this->brend;
    }

    public function setBrend($brend)
    {
        $this->brend = $brend;
    }
    //endregion

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM brend";
        return parent::dohvatiSveSuper($upit);
    }
}
```

#### 4.4) Footer

```
<?php

class Footer extends SuperKlasa
{
    private $footerID;
    private $href;
    private $class;

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    //region Get i Set
    public function getFooterID()
    {
        return $this->footerID;
    }

    public function setFooterID($footerID)
    {
        $this->footerID = $footerID;
    }

    public function getHref()
    {
        return $this->href;
    }

    public function setHref($href)
    {
        $this->href = $href;
    }

    public function getClass()
    {
        return $this->class;
    }

    public function setClass($class)
    {
        $this->class = $class;
    }
    //endregion

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM footer";
        return parent::dohvatiSveSuper($upit);
    }
}
```

## 4.5) Kategorija

```
<?php

class Kategorija extends SuperKlasa
{
    private $kategorijaID;
    private $kategorija;

    //region Get i Set
    public function getKategorijaID()
    {
        return $this->kategorijaID;
    }

    public function setKategorijaID($kategorijaID)
    {
        $this->kategorijaID = $kategorijaID;
    }

    public function getKategorija()
    {
        return $this->kategorija;
    }

    public function setKategorija($kategorija)
    {
        $this->kategorija = $kategorija;
    }
    //endregion

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM kategorija";
        return parent::dohvatiSveSuper($upit);
    }
}
```

#### 4.6) Korisnik

```
<?php

class Korisnik extends SuperKlasa
{
    private $korisnikID;
    private $ime;
    private $prezime;
    private $email;
    private $korisnicko_ime;
    private $lozinka;
    private $token;
    private $aktivan;
    private $datum_registracije;
    private $ulogaID;
    private $reset_password;

    //region Get i Set
    public function getKorisnikID()
    {
        return $this->korisnikID;
    }

    public function setKorisnikID($korisnikID)
    {
        $this->korisnikID = $korisnikID;
    }

    public function getIme()
    {
        return $this->ime;
    }

    public function setIme($ime)
    {
        $this->ime = $ime;
    }

    public function getPrezime()
    {
        return $this->prezime;
    }

    public function setPrezime($prezime)
    {
        $this->prezime = $prezime;
    }

    public function getEmail()
    {
        return $this->email;
    }

    public function setEmail($email)
    {
        $this->email = $email;
    }

    public function getKorisnickoIme()
    {
        return $this->korisnicko_ime;
    }

    public function setKorisnickoIme($korisnicko_ime)
    {
        $this->korisnicko_ime = $korisnicko_ime;
    }

    public function getLozinka()
```

```
{
    return $this->lozinka;
}

public function setLozinka($lozinka)
{
    $this->lozinka = $lozinka;
}

public function getToken()
{
    return $this->token;
}

public function setToken($token)
{
    $this->token = $token;
}

public function getAktivan()
{
    return $this->aktivan;
}

public function setAktivan($aktivan)
{
    $this->aktivan = $aktivan;
}

public function getDatumRegistracije()
{
    return $this->datum_registracije;
}

public function setDatumRegistracije($datum_registracije)
{
    $this->datum_registracije = $datum_registracije;
}

public function getUlogaID()
{
    return $this->ulogaID;
}

public function setUlogaID($ulogaID)
{
    $this->ulogaID = $ulogaID;
}

public function getResetPassword()
{
    return $this->reset_password;
}

public function setResetPassword($reset_password)
{
    $this->reset_password = $reset_password;
}
//endregion

public function __construct($baza)
{
    $this->baza = $baza;
}

public function dohvatiSve()
{
    $upit = "SELECT * FROM korisnik k INNER JOIN uloga u ON k.ulogaID = u.ulogaID";
    return parent::dohvatiSveSuper($upit);
}
```

```

public function dohvatiJednog()
{
    $upit = "SELECT * FROM korisnik k INNER JOIN uloga u ON k.ulogaID = u.ulogaID WHERE
k.korisnikID = :korisnikID";
    $podaci = ["korisnikID" => $this->korisnikID];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}

public function registerKorisnik()
{
    $upit = "INSERT INTO korisnik (ime, prezime, email, korisnicko_ime, lozinka, token,
aktivan, datum_registracije, ulogaID, reset_password)
VALUES (:ime, :prezime, :email, :korisnicko_ime, :lozinka, :token, :aktivan,
:datum_registracije, (SELECT ulogaID FROM uloga WHERE uloga = 'Korisnik'), :reset_password)";
    $podaci =
    [
        "ime" => $this->ime,
        "prezime" => $this->prezime,
        "email" => $this->email,
        "korisnicko_ime" => $this->korisnicko_ime,
        "lozinka" => $this->lozinka,
        "token" => $this->token,
        "aktivan" => $this->aktivan,
        "datum_registracije" => $this->datum_registracije,
        "reset_password" => $this->reset_password
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function activateProvera()
{
    $upit = "SELECT * FROM korisnik WHERE token = :token";
    $podaci = ["token" => $this->token];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}

public function activate()
{
    $upit = "UPDATE korisnik SET aktivan = 1 WHERE token = :token";
    $podaci = ["token" => $this->token];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function loginProvera()
{
    $upit = "SELECT * FROM korisnik k INNER JOIN uloga u ON k.ulogaID=u.ulogaID WHERE
korisnicko_ime = :korisnicko_ime AND lozinka = :lozinka AND aktivan = 1";
    $podaci =
    [
        "korisnicko_ime" => $this->korisnicko_ime,
        "lozinka" => $this->lozinka
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci)->fetch();
}

public function insertKorisnik()
{
    $upit = "INSERT INTO korisnik (ime, prezime, email, korisnicko_ime, lozinka, token,
aktivan, datum_registracije, ulogaID, reset_password)
VALUES (:ime, :prezime, :email, :korisnicko_ime, :lozinka, :token, :aktivan,
:datum_registracije, :ulogaID, :reset_password)";
    $podaci =
    [
        "ime" => $this->ime,
        "prezime" => $this->prezime,
        "email" => $this->email,
        "korisnicko_ime" => $this->korisnicko_ime,
        "lozinka" => $this->lozinka,
        "token" => $this->token,

```

```

        "aktivan" => $this->aktivan,
        "datum_registracije" => $this->datum_registracije,
        "ulogaID" => $this->ulogaID,
        "reset_password" => $this->reset_password
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function deleteKorisnik()
{
    $upitFkKorpa = "DELETE FROM korpa WHERE korisnikID = :korisnikID";
    $upitFkAnketa = "DELETE FROM anketa WHERE korisnikID = :korisnikID";
    $upit = "DELETE FROM korisnik WHERE korisnikID = :korisnikID";
    $podaci = ["korisnikID" => $this->korisnikID];
    $this->baza->izvrsiInsertUpdateDelete($upitFkKorpa, $podaci);
    $this->baza->izvrsiInsertUpdateDelete($upitFkAnketa, $podaci);
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function updateKorisnik()
{
    $upit = "UPDATE korisnik SET ime = :ime, prezime = :prezime, email = :email, korisnicko_ime
= :korisnicko_ime, aktivan = :aktivan, ulogaID = :ulogaID
    WHERE korisnikID = :korisnikID";
    $podaci =
    [
        "ime" => $this->ime,
        "prezime" => $this->prezime,
        "email" => $this->email,
        "korisnicko_ime" => $this->korisnicko_ime,
        "aktivan" => $this->aktivan,
        "ulogaID" => $this->ulogaID,
        "korisnikID" => $this->korisnikID
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function promenaLozinkeProvera()
{
    $upit = "SELECT * FROM korisnik WHERE korisnikID = :korisnikID AND lozinka = :lozinka";
    $podaci =
    [
        "korisnikID" => $this->korisnikID,
        "lozinka" => $this->lozinka
    ];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}

public function promeniLozinku()
{
    $upit = "UPDATE korisnik SET lozinka = :lozinka WHERE korisnikID = :korisnikID";
    $podaci =
    [
        "korisnikID" => $this->korisnikID,
        "lozinka" => $this->lozinka
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function resetLozinkeEmailProvera()
{
    $upit = "SELECT * FROM korisnik WHERE email = :email";
    $podaci = ["email" => $this->email];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}

public function resetLozinkeEmailUspesnaProvera()
{
    $upit = "UPDATE korisnik SET reset_password = :reset_password WHERE email = :email";
    $podaci = [

```



```

        "reset_password" => $this->reset_password,
        "email" => $this->email
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function resetLozinkeTokenProvera()
{
    $upit = "SELECT * FROM korisnik WHERE reset_password = :reset_password";
    $podaci =
    [
        "reset_password" => $this->reset_password
    ];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}

public function resetLozinke(){
    $upit = "UPDATE korisnik SET lozinka = :lozinka, reset_password = :reset_password WHERE
reset_password = :token";
    $podaci =
    [
        "lozinka" => $this->lozinka,
        "reset_password" => $this->reset_password,
        "token" => $this->token
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}
}

```

## 4.7) Korpa

```
<?php

class Korpa extends SuperKlasa
{
    private $korpaID;
    private $korisnikID;
    private $proizvodID;
    private $kupljeno;

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    //region Get i Set
    public function getKorpaID()
    {
        return $this->korpaID;
    }

    public function setKorpaID($korpaID)
    {
        $this->korpaID = $korpaID;
    }

    public function getKorisnikID()
    {
        return $this->korisnikID;
    }

    public function setKorisnikID($korisnikID)
    {
        $this->korisnikID = $korisnikID;
    }

    public function getProizvodID()
    {
        return $this->proizvodID;
    }

    public function setProizvodID($proizvodID)
    {
        $this->proizvodID = $proizvodID;
    }

    public function getKupljeno()
    {
        return $this->kupljeno;
    }

    public function setKupljeno($kupljeno)
    {
        $this->kupljeno = $kupljeno;
    }
    //endregion

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM korpa";
        return parent::dohvatiSveSuper($upit);
    }

    public function dodajUKorpu()
    {
        $upit = "INSERT INTO korpa (korisnikID, proizvodID, kupljeno) VALUES (:korisnikID,
:proizvodID, 0)";
        $podaci =
        [
```

```

        "korisnikID" => $this->korisnikID,
        "proizvodID" => $this->proizvodID
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function izbaciIzKorpe()
{
    $upit = "DELETE FROM korpa WHERE korisnikID = :korisnikID AND proizvodID = :proizvodID AND
    kupljeno = 0 LIMIT 1";
    $podaci = [
        "korisnikID" => $this->korisnikID,
        "proizvodID" => $this->proizvodID
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function kupi()
{
    $upit = "UPDATE korpa SET kupljeno = 1 WHERE korisnikID = :korisnikID";
    $podaci = ["korisnikID" => $this->korisnikID];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function dohvatiNekupljenoKorisnik()
{
    $upit = "SELECT * FROM korisnik k INNER JOIN korpa krp ON k.korisnikID = krp.korisnikID
    INNER JOIN proizvod p ON krp.proizvodID = p.proizvodID WHERE k.korisnikID =
    :korisnikID AND kupljeno = 0";
    $podaci = ["korisnikID" => $this->korisnikID];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetchAll();
}

public function dohvatiNekupljenoAdmin()
{
    $upit = "SELECT * FROM korisnik k INNER JOIN korpa krp ON k.korisnikID = krp.korisnikID
    INNER JOIN proizvod p ON krp.proizvodID = p.proizvodID WHERE kupljeno = 0";
    return $this->baza->izvrsiSelect($upit)->fetchAll();
}

public function dohvatiKupljenoKorisnik()
{
    $upit = "SELECT * FROM korisnik k INNER JOIN korpa krp ON k.korisnikID = krp.korisnikID
    INNER JOIN proizvod p ON krp.proizvodID = p.proizvodID WHERE k.korisnikID =
    :korisnikID AND kupljeno = 1";
    $podaci = ["korisnikID" => $this->korisnikID];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetchAll();
}

public function dohvatiKupljenoAdmin()
{
    $upit = "SELECT * FROM korisnik k INNER JOIN korpa krp ON k.korisnikID = krp.korisnikID
    INNER JOIN proizvod p ON krp.proizvodID = p.proizvodID WHERE kupljeno = 1";
    return $this->baza->izvrsiSelect($upit)->fetchAll();
}

public function dohvatiSumuNekupljenihProizvoda()
{
    $upit = "SELECT SUM(cena) as suma FROM korisnik k INNER JOIN korpa krp ON k.korisnikID =
    krp.korisnikID
    INNER JOIN proizvod p ON krp.proizvodID = p.proizvodID WHERE k.korisnikID =
    :korisnikID AND kupljeno = 0";
    $podaci = ["korisnikID" => $this->korisnikID];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}
}

```

## 4.8) Navigacija

```
<?php

class Navigacija extends SuperKlasa
{
    private $navigacijaID;
    private $navigacija;
    private $href;
    private $roditelj;

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    //region Get i Set
    public function getNavigacijaID()
    {
        return $this->navigacijaID;
    }

    public function setNavigacijaID($navigacijaID)
    {
        $this->navigacijaID = $navigacijaID;
    }

    public function getNavigacija()
    {
        return $this->navigacija;
    }

    public function setNavigacija($navigacija)
    {
        $this->navigacija = $navigacija;
    }

    public function getHref()
    {
        return $this->href;
    }

    public function setHref($href)
    {
        $this->href = $href;
    }

    public function getRoditelj()
    {
        return $this->roditelj;
    }

    public function setRoditelj($roditelj)
    {
        $this->roditelj = $roditelj;
    }
    //endregion

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM navigacija";
        return parent::dohvatiSveSuper($upit);
    }

    public function dohvatiPoRoditelju()
    {
        $upit = "SELECT * FROM navigacija WHERE roditelj = :roditelj";
        $podaci = ["roditelj" => $this->roditelj];
        return $this->baza->izvrsiSelect($upit, $podaci)->fetchAll();
    }
}
```

```
public function dohvatiPrviRed()  
{  
    $upit = "SELECT * FROM navigacija WHERE roditelj = 0";  
    return $this->baza->izvrsiSelect($upit)->fetchAll();  
}
```

## 4.9) Odgovor

```
<?php

class Odgovor extends SuperKlasa
{
    private $odgovorID;
    private $odgovor;
    private $pitanjeID;

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    //region Get i Set
    public function getOdgovorID()
    {
        return $this->odgovorID;
    }

    public function setOdgovorID($odgovorID)
    {
        $this->odgovorID = $odgovorID;
    }

    public function getOdgovor()
    {
        return $this->odgovor;
    }

    public function setOdgovor($odgovor)
    {
        $this->odgovor = $odgovor;
    }

    public function getPitanjeID()
    {
        return $this->pitanjeID;
    }

    public function setPitanjeID($pitanjeID)
    {
        $this->pitanjeID = $pitanjeID;
    }
    //endregion

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM odgovor";
        return parent::dohvatiSveSuper($upit);
    }

    public function dohvatiOdgovore()
    {
        $upit = "SELECT * FROM odgovor WHERE pitanjeID = :pitanjeID";
        $podaci = ["pitanjeID" => $this->pitanjeID];
        return $this->baza->izvrsiSelect($upit, $podaci)->fetchAll();
    }
}
```

#### 4.10) Pitanje

```
<?php

class Pitanje extends SuperKlasa
{
    private $pitanjeID;
    private $pitanje;
    private $aktivno;

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    //region Get i Set
    public function getPitanjeID()
    {
        return $this->pitanjeID;
    }

    public function setPitanjeID($pitanjeID)
    {
        $this->pitanjeID = $pitanjeID;
    }

    public function getPitanje()
    {
        return $this->pitanje;
    }

    public function setPitanje($pitanje)
    {
        $this->pitanje = $pitanje;
    }

    public function getAktivno()
    {
        return $this->aktivno;
    }

    public function setAktivno($aktivno)
    {
        $this->aktivno = $aktivno;
    }
    //endregion

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM pitanje";
        return parent::dohvatiSveSuper($upit);
    }

    public function dohvatiAktivnaPitanja(){
        $upit = "SELECT * FROM pitanje WHERE aktivno = 1";
        return $this->baza->izvrsiSelect($upit)->fetchAll();
    }
}
```

#### 4.11) Proizvod

```
<?php

class Proizvod extends SuperKlasa
{
    private $proizvodID;
    private $proizvod;
    private $opis;
    private $cena;
    private $pol;
    private $src;
    private $alt;
    private $kategorijaID;
    private $brendID;

    //region Get i Set
    public function getProizvodID()
    {
        return $this->proizvodID;
    }

    public function setProizvodID($proizvodID)
    {
        $this->proizvodID = $proizvodID;
    }

    public function getProizvod()
    {
        return $this->proizvod;
    }

    public function setProizvod($proizvod)
    {
        $this->proizvod = $proizvod;
    }

    public function getOpis()
    {
        return $this->opis;
    }

    public function setOpis($opis)
    {
        $this->opis = $opis;
    }

    public function getCena()
    {
        return $this->cena;
    }

    public function setCena($cena)
    {
        $this->cena = $cena;
    }

    public function getPol()
    {
        return $this->pol;
    }

    public function setPol($pol)
    {
        $this->pol = $pol;
    }

    public function getSrc()
    {
        return $this->src;
    }
}
```



```

}

public function setSrc($src)
{
    $this->src = $src;
}

public function getAlt()
{
    return $this->alt;
}

public function setAlt($alt)
{
    $this->alt = $alt;
}

public function getKategorijaID()
{
    return $this->kategorijaID;
}

public function setKategorijaID($kategorijaID)
{
    $this->kategorijaID = $kategorijaID;
}

public function getBrendID()
{
    return $this->brendID;
}

public function setBrendID($brendID)
{
    $this->brendID = $brendID;
}
//endregion

public function __construct($baza)
{
    $this->baza = $baza;
}

public function dohvatiSve()
{
    $upit = "SELECT * FROM proizvod p INNER JOIN kategorija k ON p.kategorijaID =
k.kategorijaID
            INNER JOIN brend b ON p.brendID = b.brendID";
    return parent::dohvatiSveSuper($upit);
}

public function dohvatiJedan()
{
    $upit = "SELECT * FROM proizvod p INNER JOIN kategorija k ON p.kategorijaID =
k.kategorijaID
            INNER JOIN brend b ON p.brendID = b.brendID WHERE proizvodID = :proizvodID";
    $podaci = ["proizvodID" => $this->proizvodID];
    return $this->baza->izvrsiSelect($upit, $podaci)->fetch();
}

public function insertProizvod()
{
    $upit = "INSERT INTO proizvod (proizvod, opis, cena, pol, src, alt, kategorijaID, brendID)
VALUES (:proizvod, :opis, :cena, :pol, :src, :alt, :kategorijaID, :brendID)";
    $podaci =
    [
        "proizvod" => $this->proizvod,
        "opis" => $this->opis,
        "cena" => $this->cena,
        "pol" => $this->pol,
    ]
}

```

```

        "src" => $this->src,
        "alt" => $this->alt,
        "kategorijaID" => $this->kategorijaID,
        "brendID" => $this->brendID
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function updateProizvodBezSlike()
{
    $upit = "UPDATE proizvod SET proizvod = :proizvod, opis = :opis, cena = :cena, pol = :pol,
alt = :alt,
        kategorijaID = :kategorijaID, brendID = :brendID WHERE proizvodID =
:proizvodID";
    $podaci =
    [
        "proizvod" => $this->proizvod,
        "opis" => $this->opis,
        "cena" => $this->cena,
        "pol" => $this->pol,
        "alt" => $this->alt,
        "kategorijaID" => $this->kategorijaID,
        "brendID" => $this->brendID,
        "proizvodID" => $this->proizvodID
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function updateProizvodSaSlikom()
{
    $upit = "UPDATE proizvod SET proizvod = :proizvod, opis = :opis, cena = :cena, pol = :pol,
src = :src, alt = :alt,
        kategorijaID = :kategorijaID, brendID = :brendID WHERE proizvodID =
:proizvodID";
    $podaci =
    [
        "proizvod" => $this->proizvod,
        "opis" => $this->opis,
        "cena" => $this->cena,
        "pol" => $this->pol,
        "src" => $this->src,
        "alt" => $this->alt,
        "kategorijaID" => $this->kategorijaID,
        "brendID" => $this->brendID,
        "proizvodID" => $this->proizvodID
    ];
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function deleteProizvod()
{
    $upitFkKorpa = "DELETE FROM korpa WHERE proizvodID = :proizvodID";
    $upit = "DELETE FROM proizvod WHERE proizvodID = :proizvodID";
    $podaci = ["proizvodID" => $this->proizvodID];
    $this->baza->izvrsiInsertUpdateDelete($upitFkKorpa, $podaci);
    return $this->baza->izvrsiInsertUpdateDelete($upit, $podaci);
}

public function dohvatiCetiriNajprodavanijaProizvoda()
{
    $upit = "SELECT COUNT(k.proizvodID) as kupljen_broj, p.proizvod, p.cena, p.src, p.alt,
p.proizvodID FROM korpa k
        INNER JOIN proizvod p ON k.proizvodID = p.proizvodID WHERE k.kupljeno = 1 GROUP BY
k.proizvodID ORDER BY kupljen_broj DESC LIMIT 0,4";
    return $this->baza->izvrsiSelect($upit)->fetchAll();
}

public function filter($upit)
{
    return $this->baza->izvrsiSelect($upit)->fetchAll();
}

```

}  
}

## 4.12) Slajder

```
<?php

class Slajder extends SuperKlasa
{
    private $slajderID;
    private $src;

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    //region Get i Set
    public function getSlajderID()
    {
        return $this->slajderID;
    }

    public function setSlajderID($slajderID)
    {
        $this->slajderID = $slajderID;
    }

    public function getSrc()
    {
        return $this->src;
    }

    public function setSrc($src)
    {
        $this->src = $src;
    }
    //endregion

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM slajder";
        return parent::dohvatiSveSuper($upit);
    }
}
```

#### 4.13) SuperKlasa

```
<?php

class SuperKlasa
{
    protected $baza;

    //region Get i Set
    protected function getBaza()
    {
        return $this->baza;
    }

    protected function setBaza($baza)
    {
        $this->baza = $baza;
    }
    //endregion

    /*protected function dohvatiSveSuper($upit) // ne odgovara pri ispisu svih redova, ukoliko
    postoji samo 1
    {
        $rezultat = $this->baza->izvrsiSelect($upit);
        if($rezultat->rowCount() == 1)
        {
            return $rezultat->fetch();
        }
        else
        {
            return $rezultat->fetchAll();
        }
    }*/

    protected function dohvatiSveSuper($upit)
    {
        $rezultat = $this->baza->izvrsiSelect($upit);
        return $rezultat->fetchAll();
    }
}
```

#### 4.14) Uloga

```
<?php

class Uloga extends SuperKlasa
{
    private $ulogaID;
    private $uloga;

    //region Get i Set
    public function getUlogaID()
    {
        return $this->ulogaID;
    }

    public function setUlogaID($ulogaID)
    {
        $this->ulogaID = $ulogaID;
    }

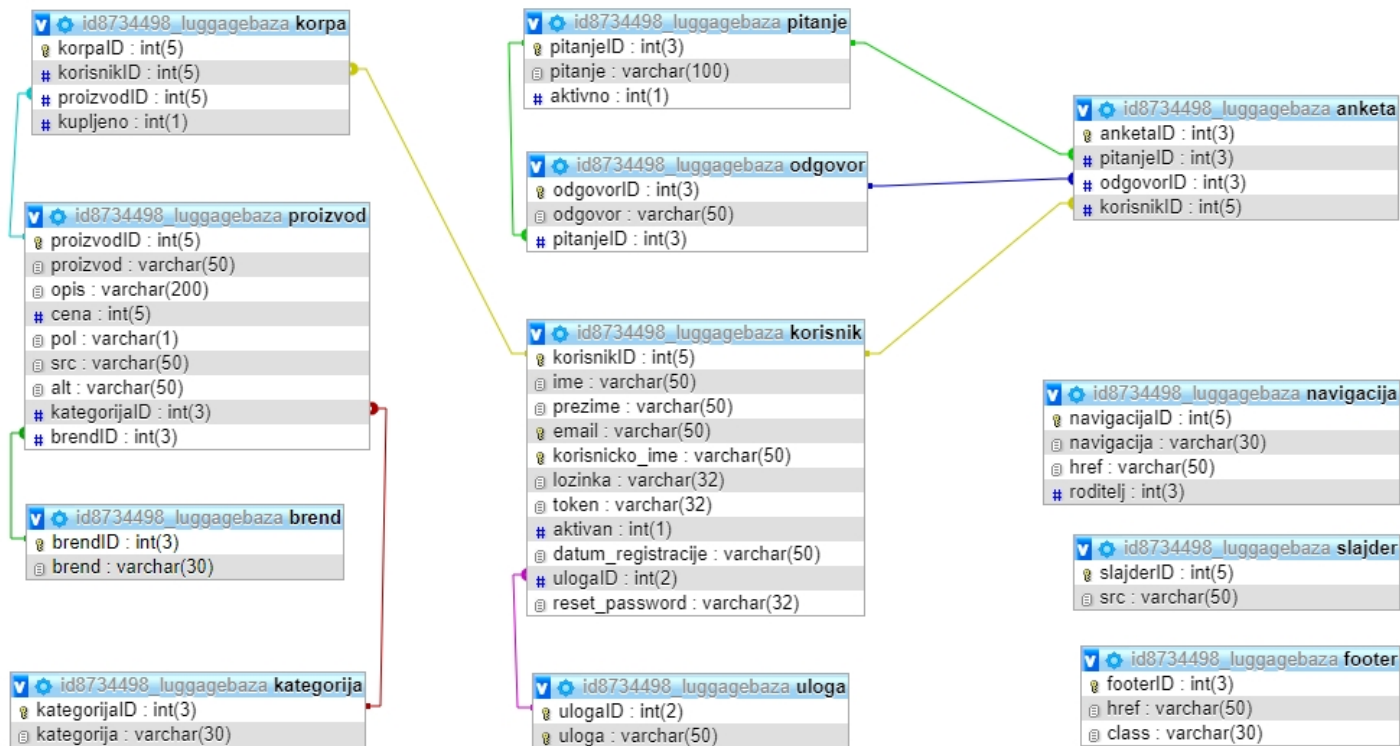
    public function getUloga()
    {
        return $this->uloga;
    }

    public function setUloga($uloga)
    {
        $this->uloga = $uloga;
    }
    //endregion

    public function __construct($baza)
    {
        $this->baza = $baza;
    }

    public function dohvatiSve()
    {
        $upit = "SELECT * FROM uloga";
        return parent::dohvatiSveSuper($upit);
    }
}
```

## 5. BAZA PODATAKA



## - anketa

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	anketaID	int(3)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  Spatial  More
<input type="checkbox"/>	2	pitanjeID	int(3)		No	None			Change  Drop  Primary  Unique  Index  Spatial  More
<input type="checkbox"/>	3	odgovorID	int(3)		No	None			Change  Drop  Primary  Unique  Index  Spatial  More
<input type="checkbox"/>	4	korisnikID	int(5)		No	None			Change  Drop  Primary  Unique  Index  Spatial  More

## - brend

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	brendID	int(3)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	brend	varchar(30)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  Index  More

## - footer

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	footerID	int(3)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	href	varchar(50)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	3	class	varchar(30)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  Index  More

## - korisnik

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	korisnikID	int(5)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  More
<input type="checkbox"/>	2	ime	varchar(50)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	3	prezime	varchar(50)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	4	email	varchar(50)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	5	korisnicko_ime	varchar(50)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	6	lozinka	varchar(32)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	7	token	varchar(32)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	8	aktivan	int(1)		No	0			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	9	datum_registracije	varchar(50)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	10	ulogalID	int(2)		No	None			Change  Drop  Primary  Unique  More
<input type="checkbox"/>	11	reset_password	varchar(32)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  More

## - korpa

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	korpaID	int(5)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  Spatial  More
<input type="checkbox"/>	2	korisnikID	int(5)		No	None			Change  Drop  Primary  Unique  Index  Spatial  More
<input type="checkbox"/>	3	proizvodID	int(5)		No	None			Change  Drop  Primary  Unique  Index  Spatial  More
<input type="checkbox"/>	4	kupljeno	int(1)		No	None			Change  Drop  Primary  Unique  Index  Spatial  More

## - kategorija

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	kategorijaID	int(3)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	kategorija	varchar(30)	utf8_unicode_ci	No	None			Change  Drop  Primary  Unique  Index  More



## - navigacija

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	navigacijaID	int(5)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	navigacija	varchar(30) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	3	href	varchar(50) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	4	roditelj	int(3)		No	None			Change  Drop  Primary  Unique  Index  More

## - odgovor

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	odgovorID	int(3)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	odgovor	varchar(50) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	3	pitanjeID	int(3)		No	None			Change  Drop  Primary  Unique  Index  More

## - pitanje

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	pitanjeID	int(3)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	pitanje	varchar(100) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	3	aktivno	int(1)		No	None			Change  Drop  Primary  Unique  Index  More

## - proizvod

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	proizvodID	int(5)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	proizvod	varchar(50) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	3	opis	varchar(200) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	4	cena	int(5)		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	5	pol	varchar(1) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	6	src	varchar(50) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	7	alt	varchar(50) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	8	kategorijaID	int(3)		No	None			Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	9	brendID	int(3)		No	None			Change  Drop  Primary  Unique  Index  More

## - slajder

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	slajderID	int(5)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	src	varchar(50) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More

## - uloga

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	ulogaID	int(2)		No	None		AUTO_INCREMENT	Change  Drop  Primary  Unique  Index  More
<input type="checkbox"/>	2	uloga	varchar(50) utf8_unicode_ci		No	None			Change  Drop  Primary  Unique  Index  More