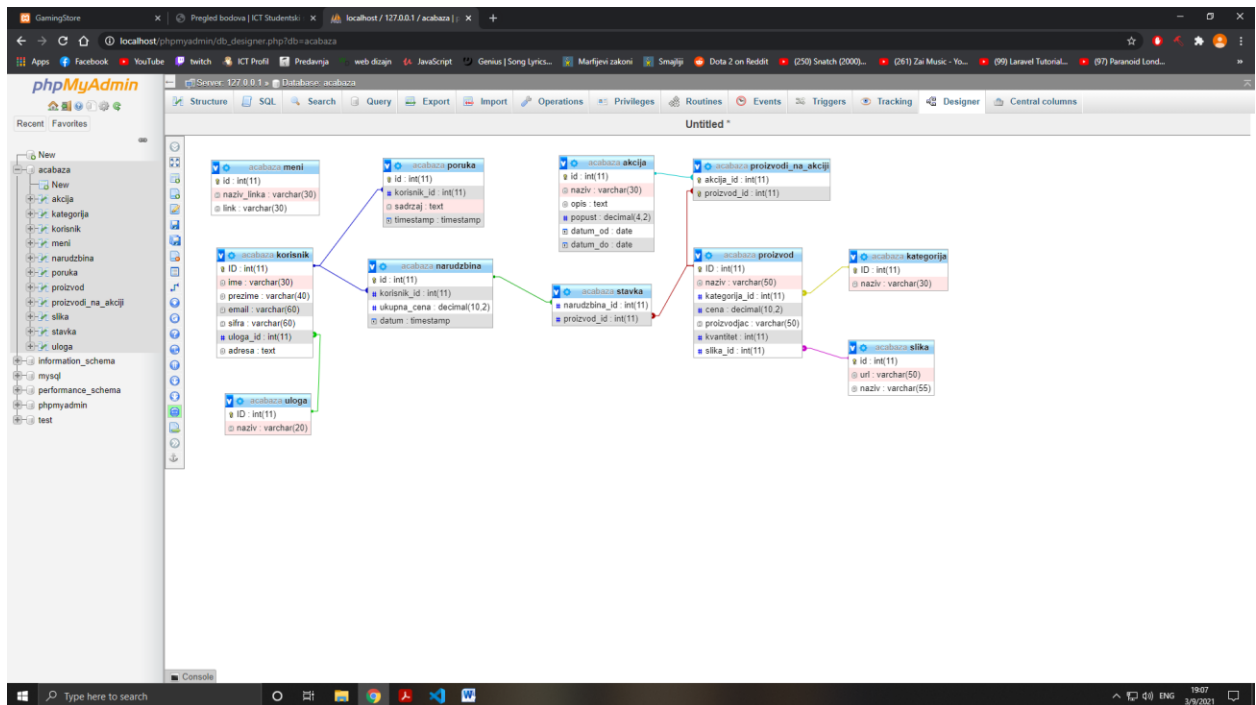


## SADRZAJ

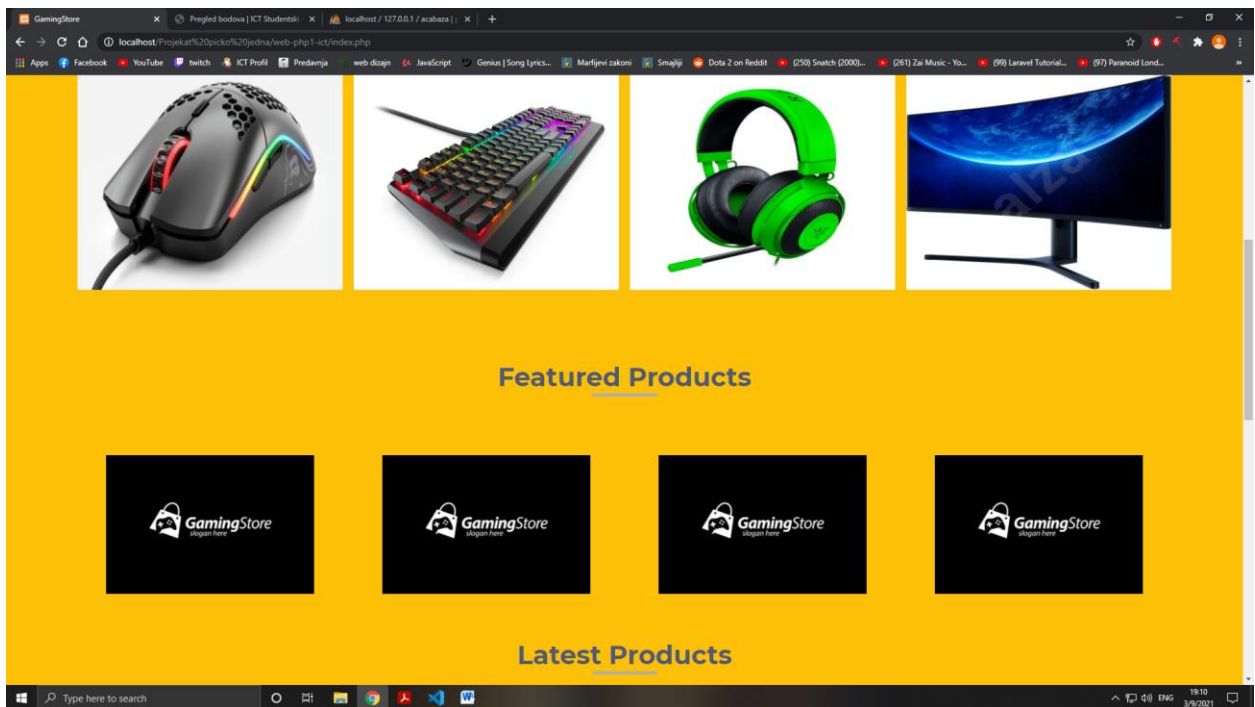
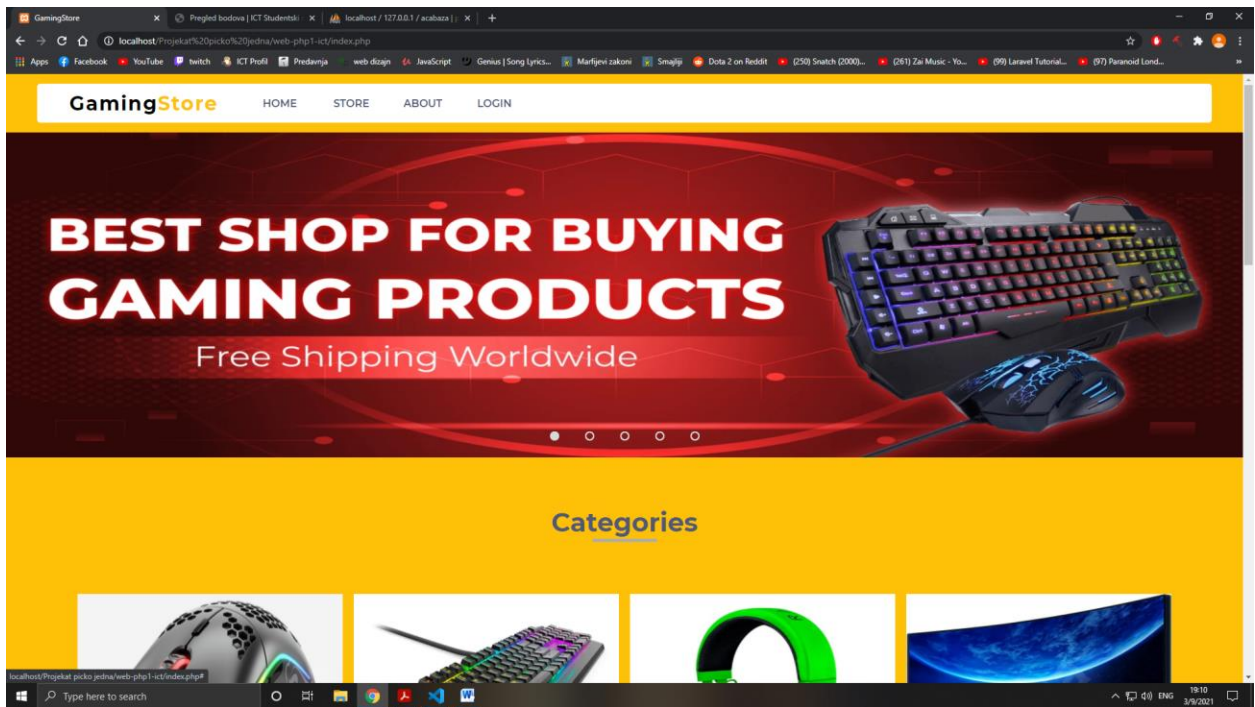
1. Skica baze podataka
2. Skice stranica
3. Javascript kod
4. PHP kod

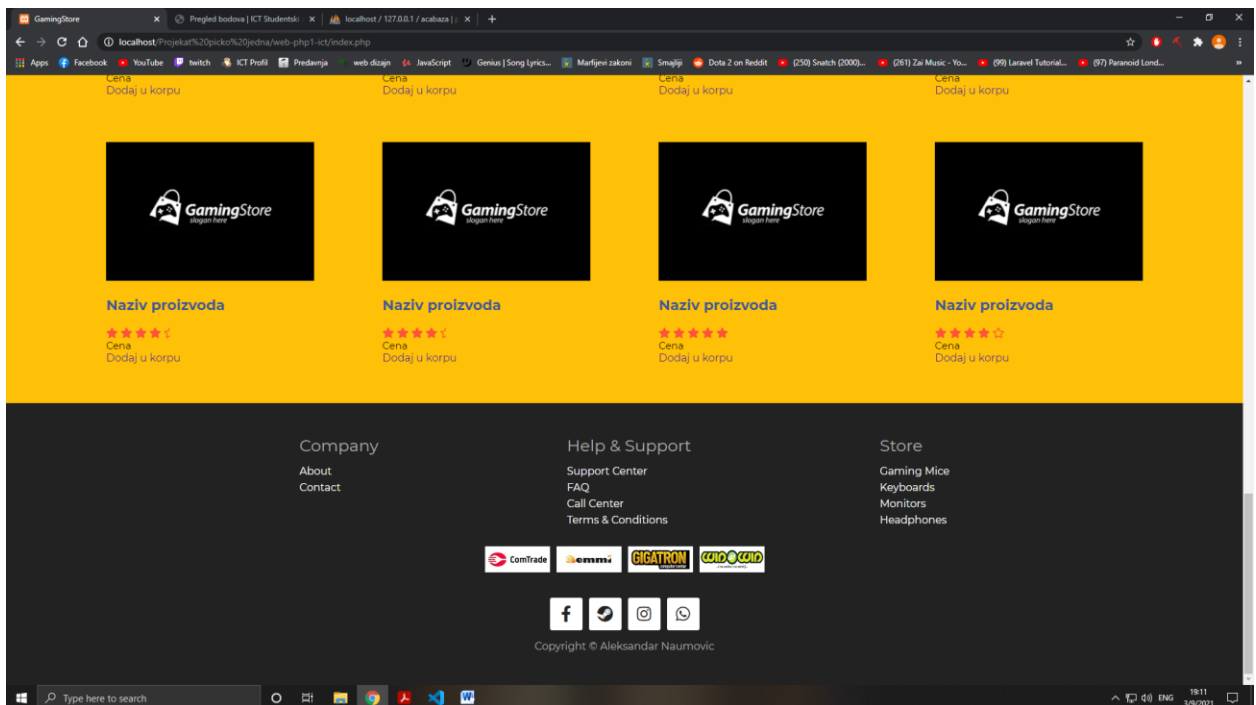
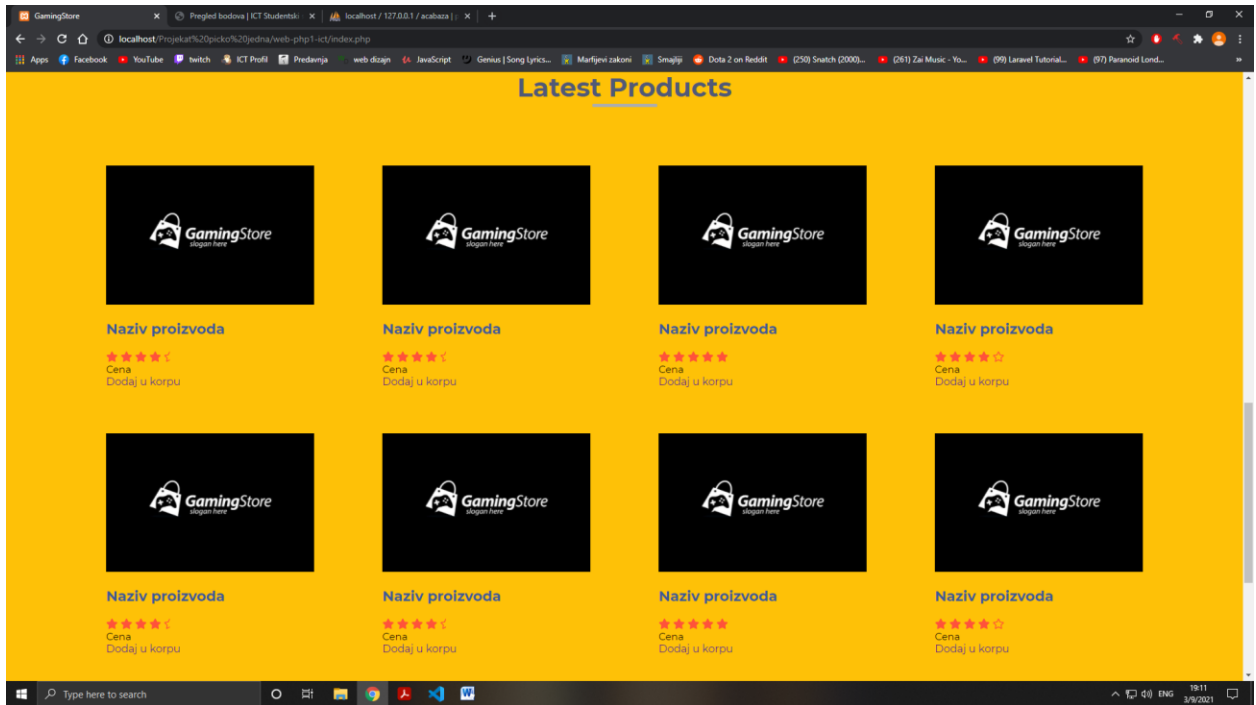
# BAZA



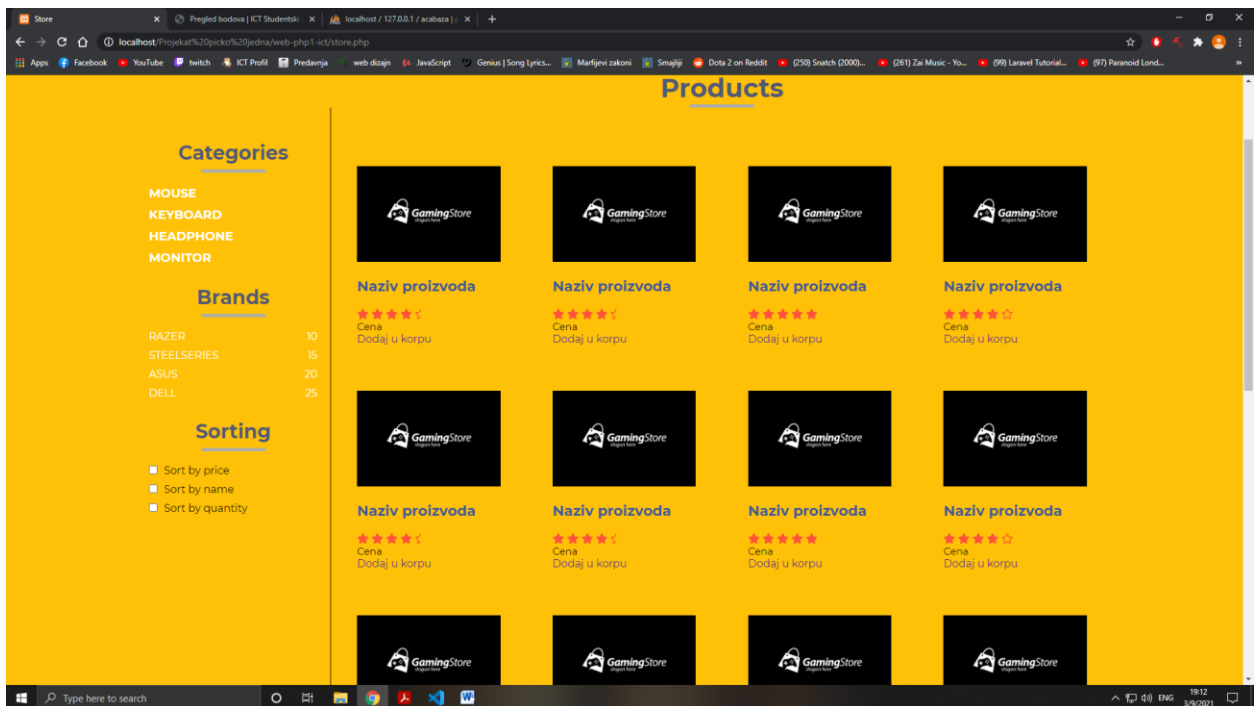
# STRANICE

Index.php

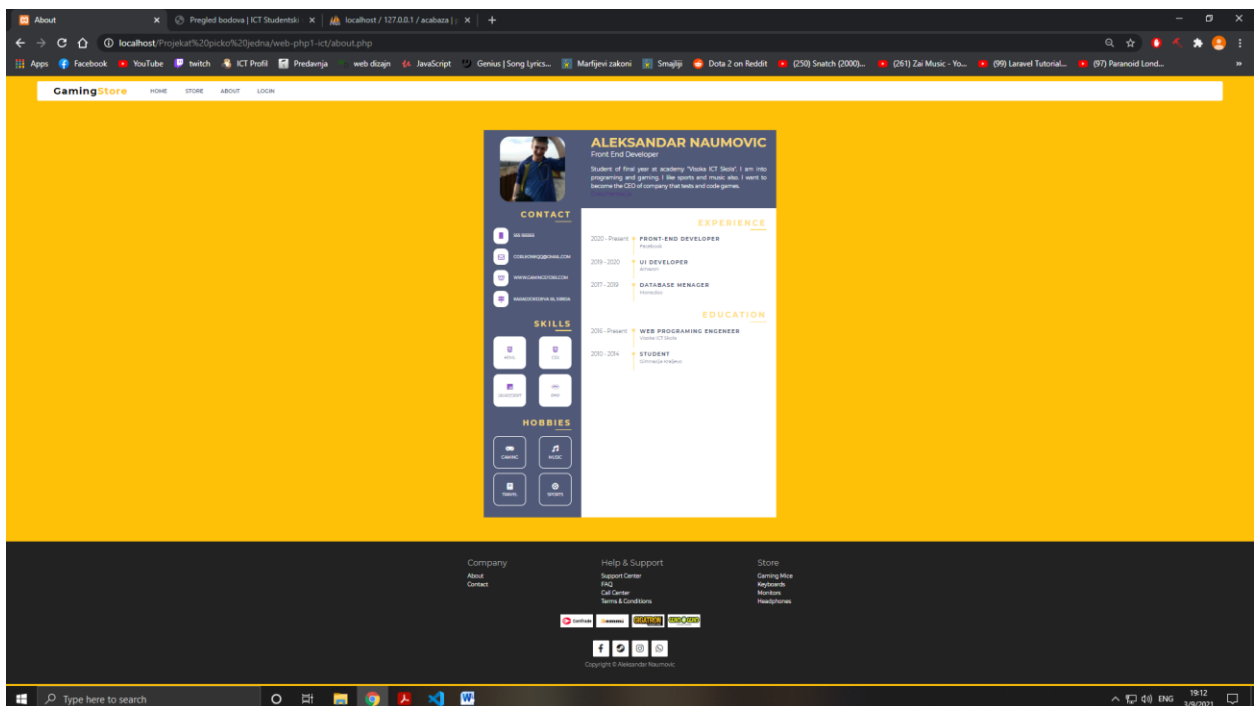




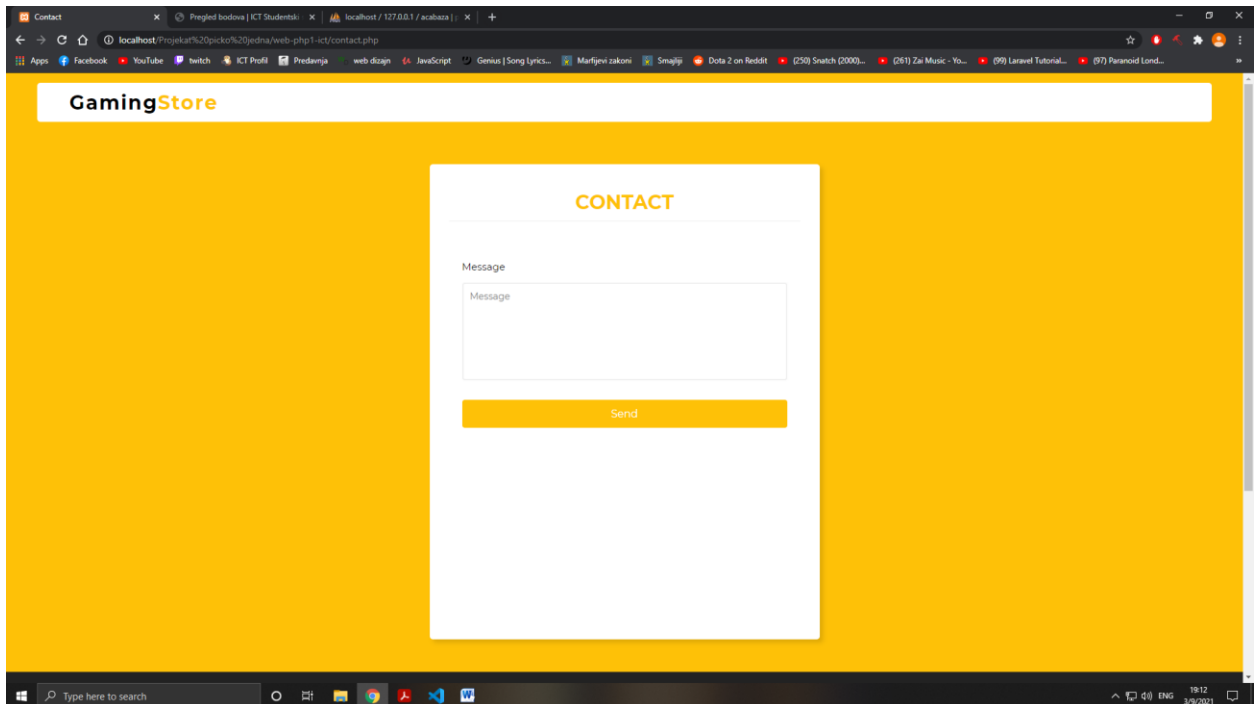
## Store.php



## About.php



## Contact.php



Contact

localhost / 127.0.0.1 / acalaza

localhost / Projekt%20picko%20jedna/web-php1-ict/contact.php

Apps Facebook YouTube twitch ICT Profil Predavanja web dizajn JavaScript Genius | Song Lyrics... Marijevi zakoni Smajli Dota 2 on Reddit (250) Snatch (2000... (261) Zai Music - Yo... (99) Laravel Tutorial... (97) Paranoid Lond...

GamingStore

### CONTACT

Message

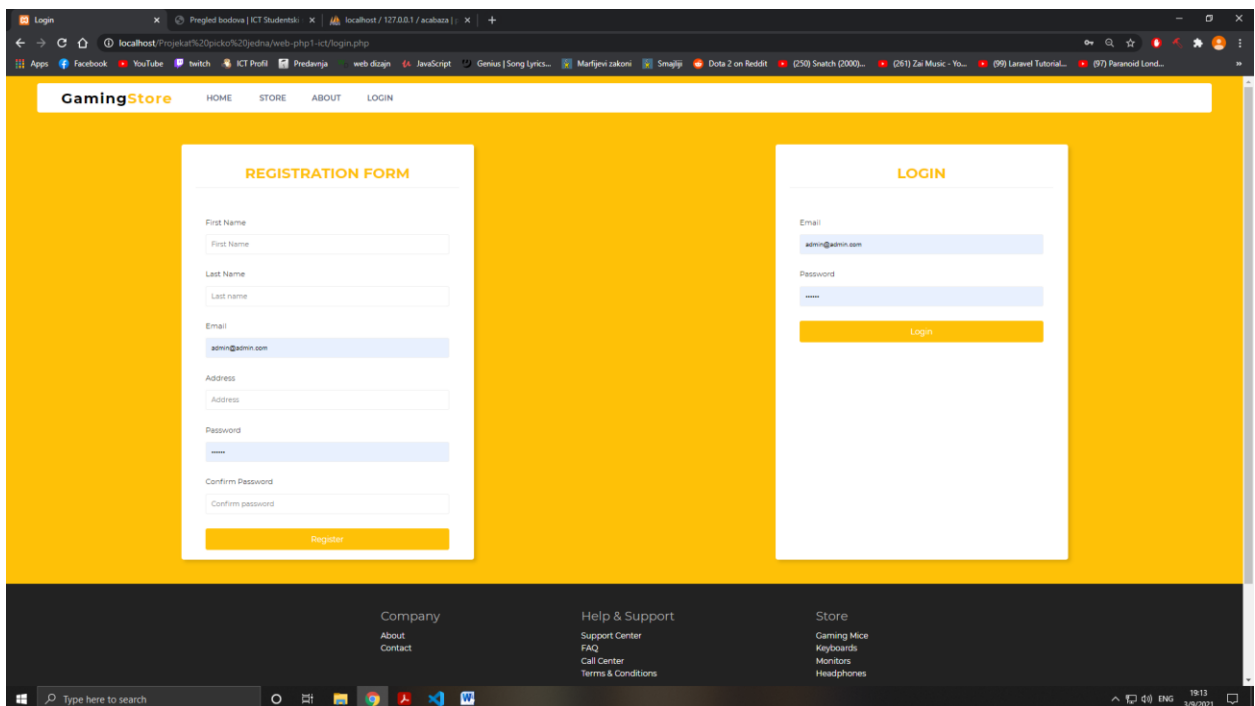
Message

Send

Type here to search

19:12 3/9/2021

## Login.php



Login

localhost / 127.0.0.1 / acalaza

localhost / Projekt%20picko%20jedna/web-php1-ict/login.php

Apps Facebook YouTube twitch ICT Profil Predavanja web dizajn JavaScript Genius | Song Lyrics... Marijevi zakoni Smajli Dota 2 on Reddit (250) Snatch (2000... (261) Zai Music - Yo... (99) Laravel Tutorial... (97) Paranoid Lond...

GamingStore HOME STORE ABOUT LOGIN

### REGISTRATION FORM

First Name

First Name

Last Name

Last name

Email

admin@admin.com

Address

Address

Password

.....

Confirm Password

Confirm password

Register

### LOGIN

Email

admin@admin.com

Password

.....

Login

Company

About

Contact

Help & Support

Support Center

FAQ

Call Center

Terms & Conditions

Store

Gaming Mice

Keyboards

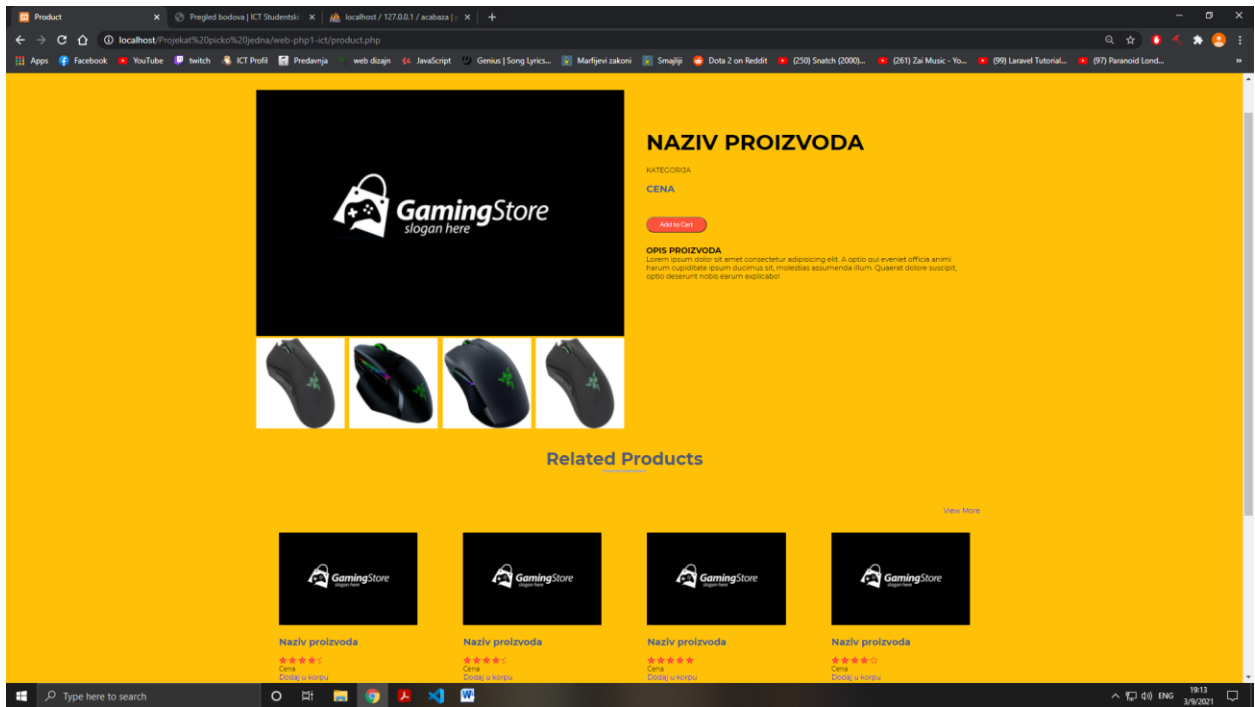
Monitors

Headphones

Type here to search

19:13 3/9/2021

# Product.php



# JAVASCRIPT

Ajax call back – prikaz slika kao kategorije

Slajder – ispis i animacija

```
js > JS main.js > ...
1  const BASEURL = "data/";
2
3  // ajax callback
4  function ajaxCallBack(url, method, result){
5      $.ajax({
6          url: url,
7          method: method,
8          dataType: "json",
9          success: result,
10         error: function(xhr){console.log(xhr);}
11     });
12 }
13
14
15 // kategorije f-ja proba - USPELO
16 function printCat(data){
17     let html = '';
18     for(let elementCat of data){
19         html += `<div class="col-3">
20             <a href="${elementCat.href}"></a>
21             </div>`;
22     }
23     $("#kategorije").html(html);
24 }
25
26
27 window.onload = function(){
28
29
30     ajaxCallBack(BASEURL + "categoriestest.json", "get", function(result){
31         printCat(result);
32     });
33
34     // SLAJDER ISPIS I ANIMACIJA
35
36     var slajder = [
37         {
38             "id" : 1,
39             src : "images/banner1.jpg",
40             alt : "banner1",
41         },
42         {
43             "id" : 2,
44             src : "images/banner2.jpg",
45             alt : "banner2",
46         },
47         {
48             "id" : 3,
49             src : "images/banner3.jpg",
```



index.php

# styles.css

about.php

JS main.js

X

JS meni.js

Database.php

js &gt; JS main.js &gt; onload

```
50         alt : "banner3",
51     },
52     {
53         "id" : 4,
54         src : "images/banner4.jpg",
55         alt : "banner4",
56     },
57     {
58         "id" : 5,
59         src : "images/banner5.jpg",
60         alt : "banner5",
61     }
62 ]
63
64 var ispis1='';
65 for (let i = 0; i < slajder.length; i++) {
66     if(i==0){
67         ispis1+=`<input type="radio" name="radio-btn" id="radio1"/>
68         <input type="radio" name="radio-btn" id="radio2"/>
69         <input type="radio" name="radio-btn" id="radio3"/>
70         <input type="radio" name="radio-btn" id="radio4"/>
71         <input type="radio" name="radio-btn" id="radio5"/>
72
73         <div class="slide first">
74             <a href="#"></a>
75         </div>
76
77         <div class="navigation-auto">
78             <div class="auto-btn1"></div>
79             <div class="auto-btn2"></div>
80             <div class="auto-btn3"></div>
81             <div class="auto-btn4"></div>
82             <div class="auto-btn5"></div>
83         </div>
84
85         <div class="navigation-manual">
86             <label for="radio1" class="manual-btn"></label>
87             <label for="radio2" class="manual-btn"></label>
88             <label for="radio3" class="manual-btn"></label>
89             <label for="radio4" class="manual-btn"></label>
90             <label for="radio5" class="manual-btn"></label>
91         </div>`;
92     }else{
93         ispis1+=`<div class="slide">
94             <a href="#"></a>
95         </div>`;
96     }
97 }
98
```

## validacija.js

validacija podataka na klijentskoj strani

js > JS validacija.js > ...

```
1
2  const formReg = document.getElementById("formReg");
3  const formLog = document.getElementById("formLog");
4  const emailLog = document.getElementById("emailLog");
5  const passwordLog = document.getElementById("passwordLog");
6  const firstname = document.getElementById("firstname");
7  const lastname = document.getElementById("lastname");
8  const email = document.getElementById("email");
9  const address = document.getElementById("address");
10 const password = document.getElementById("password");
11 const password2 = document.getElementById("confpassword");
12
13 formLog.addEventListener("submit", (e) => {
14     // e.preventDefault();
15     proveraLog();
16 });
17
18 formReg.addEventListener("submit", (e) => {
19     e.preventDefault();
20     proveraReg();
21 });
22
23 function proveraReg(){
24     const firstnameValue = firstname.value.trim();
25     const lastnameValue = lastname.value.trim();
26     const emailValue = email.value.trim();
27     const addressValue = address.value.trim();
28     const passwordValue = password.value.trim();
29     const password2Value = password2.value.trim();
30
31     if(firstnameValue === ''){
32         setErrorFor(firstname, 'First name cannot be blank');
33     } else if(!isUsername(firstnameValue)){
34         setErrorFor(firstname, 'First name is not valid');
35     } else{
```

js > JS validacija.js > ...

```
30
31     if(firstnameValue === ''){
32         setErrorFor(firstname, 'First name cannot be blank');
33     }else if(!isUsername(firstnameValue)){
34         setErrorFor(firstname, 'First name is not valid');
35     }else{
36         setSuccessFor(firstname);
37     }
38
39     if(lastnameValue === ''){
40         setErrorFor(lastname, 'Last name cannot be blank');
41     }else if(!isUsername(lastnameValue)){
42         setErrorFor(lastname, 'Last name is not valid');
43     }else{
44         setSuccessFor(lastname);
45     }
46
47     if(emailValue === ''){
48         setErrorFor(email, 'Email cannot be blank');
49     }else if(!isEmail(emailValue)){
50         setErrorFor(email, 'Email is not valid');
51     }else{
52         setSuccessFor(email);
53     }
54
55     if(addressValue === ''){
56         setErrorFor(address, 'Address cannot be blank');
57     }else if(!isAddress(addressValue)){
58         setErrorFor(address, 'Address is not valid');
59     }else{
60         setSuccessFor(address);
61     }
62
63     if(passwordValue === ''){
64         setErrorFor(password, 'Password cannot be blank');
65     }else if(!isPassword(passwordValue)){
66         setErrorFor(password, 'Password is not valid');
67     }else{
68         setSuccessFor(password);
69     }
70
71     if(password2Value === ''){
72         setErrorFor(password2, 'Password cannot be blank');
73     }else if(!isPassword(password2Value)){
74         setErrorFor(password2, 'Password is not valid');
75     }else if(passwordValue !== password2Value){
76         setErrorFor(password2, 'Passwords does not match');
77     }else{
78         setSuccessFor(password2);
```

```

68         setPasswordFor(password);
69     }
70
71     if(password2Value === ''){
72         setErrorFor(password2, 'Password cannot be blank');
73     }else if(!isPassword(password2Value)){
74         setErrorFor(password2, 'Password is not valid');
75     }else if(passwordValue !== password2Value){
76         setErrorFor(password2, 'Passwords does not match');
77     }else{
78         setPasswordFor(password2);
79     }
80 }
81
82 function proveraLog(){
83     const emailLogValue = emailLog.value.trim();
84     const passwordValueLog = passwordLog.value.trim();
85
86     if(emailLogValue === ''){
87         setErrorFor(emailLog, 'Username cannot be blank');
88     }else if(!isEmail(emailLogValue)){
89         setErrorFor(emailLog, 'Username is not valid');
90     }else{
91         setPasswordFor(emailLog);
92     }
93
94     if(passwordValueLog === ''){
95         setErrorFor(passwordLog, 'Password cannot be blank');
96     }else if(!isPasswordLog(passwordValueLog)){
97         setErrorFor(passwordLog, 'Password is not valid');
98     }else{
99         setPasswordFor(passwordLog);
100     }
101 }
102
103 function setErrorFor(input,message){
104     const formControl = input.parentElement;
105     const small = formControl.querySelector(".small");
106
107     small.innerText = message;
108
109     formControl.className = "form-control error";
110 }
111
112 function setPasswordFor(input){
113     const formControl = input.parentElement;
114     formControl.className = "form-control success";
115 }
116

```

## Regularni izrazi

```
js > JS validacija.js >  proveraReg
116
117 // REGULARNI IZRAZI
118
119 function isEmail(email){
120 |     return /^[^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*|(".+")@((\[[0-9]{1,3}\.
121 | }
122
123 function isFirstName(firstname){
124 |     return /^[A-ZČĆŽŠĐ][a-zčćžšđ]{2,14}(\s[A-ZČĆŽŠĐ][a-zčćžšđ]{2,19})+$/i.test(firstname);
125 | }
126
127 function isLastName(lastname){
128 |     return /^[A-ZČĆŽŠĐ][a-zčćžšđ]{2,14}(\s[A-ZČĆŽŠĐ][a-zčćžšđ]{2,19})+$/i.test(lastname);
129 | }
130
131 function isAdress(address){
132 |     return /^[a-z]+[0-9]$/i.test(address);
133 | }
134
135 function isPassword(password){
136 |     return /^[a-z]+[0-9]{1}$/i.test(password);
137 | }
138
139 function isPasswordLog(passwordLog){
140 |     return /^[a-z]+[0-9]{1}$/i.test(passwordLog);
141 | }
142
```

## validacijaCnt.js - validacija kontakt forme

```
js > JS validacijaCnt.js > ...
1  const formCnt = document.getElementById("formCnt");
2  const textareaCnt = document.getElementById("textareaCnt");
3
4  formCnt.addEventListener("submit", (e)=>{
5      e.preventDefault();
6      proveraCnt();
7  });
8
9  function proveraCnt(){
10     const textareaCntValue = textareaCnt.value.trim();
11
12     if(textareaCntValue === ''){
13         setErrorFor(textareaCnt, 'Message cannot be blank');
14     }else{
15         setSuccessFor(textareaCnt);
16     }
17 }
18
19 function setErrorFor(input, message){
20     const formControl = input.parentElement;
21     const small = formControl.querySelector(".small");
22
23     small.innerText = message;
24
25     formControl.className = "form-control error";
26 }
27
28 function setSuccessFor(input){
29     const formControl = input.parentElement;
30     formControl.className = "form-control success";
31 }
32
```

# PHP KOD

## db/Database.php

```
db > Database.php
1  <?php
2
3  class Database
4  {
5      private $hostname = "localhost";
6      private $username = "root";
7      private $password = "";
8      private $dbname;
9      private $dblink;
10     private $result;
11     private $records;
12     private $affected;
13
14     function __construct($par_dbname)
15     {
16         $this->dbname = $par_dbname;
17         $this->Connect();
18     }
19
20     function Connect(){
21         $this->dblink = new mysqli($this->hostname, $this->username, $this->password, $this->dbname);
22         if($this->dblink->connect_errno){
23             printf("Konekcija neuspješna: %s\n", $this->dblink->connect_error);
24             exit();
25         }
26         $this->dblink->set_charset("utf8");
27     }
28
29     function ExecuteQuery($query){
30         $this->result = $this->dblink->query($query);
31         if($this->result){
32             if(isset($this->result->num_rows)){
33                 $this->records = $this->result->num_rows;
34             }
35             if(isset($this->result->affected_rows)){
36                 $this->affected = $this->result->affected_rows;
37             }
38         }
39         return true;
40     }else{
41         return false;
42     }
43 }
44
45 function getResult(){
46     return $this->result;
47 }
```

## Select / insert / update / delete

// SELECT

```
function select($table="proizvod",$rows="",$join_table="kategorija",$join_key1="kategorija_id",$join_key2="id", $where =null, $order=null){
    $q = 'SELECT '.$rows.' FROM '.$table;
    //SELECT * FROM novosti
    if($join_table!=null){
        $q.=' JOIN '.$join_table.' ON '.$table.'.'.$join_key1.'='.$join_table.'.'.$join_key2;
        //SELECT * FROM novosti JOIN kategorije ON novosti.kategorija_id = kategorije.id
    }
    if($where!=null){
        $q.=' WHERE '.$where;
    }
    if($order!=null){
        $q.=' ORDER BY '.$order;
    }
    // echo $q;
    $this->ExecuteQuery($q);
}
```

// INSERT

```
function insert($table="proizvod",$rows="naziv,kategorija_id, cena, proizvođač, kvantitet, slika_id", $values){
    $query_values = implode(',',$values);
    $q = 'INSERT INTO '.$table;
    if($rows!=null){
        $q.='('.$rows.')';
    }
    $q.=" VALUES($query_values)";
    echo $q;
    if($this->ExecuteQuery($q)){
        return true;
    }else{
        return false;
    }
}
```

// UPDATE

```
function update($table, $id,$id_val, $keys,$values){
    $query_values = "";
    $set_query = array();
    for($i =0; $i<sizeof($keys); $i++){
        $set_query[] = "$keys[$i] = $values[$i]";
    }
    $query_values = implode(", ", $set_query);
    $q = "UPDATE $table SET $query_values WHERE $id=$id_val";
    echo $q;
    if($this->ExecuteQuery($q) && $this->affected>0){
        return true;
    }else{
        return false;
    }
}
```



```
//DELETE
```

```
function delete($table, $id, $id_value){  
    $q = "DELETE FROM $table WHERE $table.$id=$id_value";  
    // echo $q;  
    if($this->ExecuteQuery($q)){  
        return true;  
    }else{  
        return false;  
    }  
}
```

## control/Controler.php

```
<?php  
include ("db/Database.php");  
  
class Controller{  
    private $mydb;  
    private static $instance = null;  
  
    function __construct()  
    {  
        $this->mydb = new Database("acabaza");  
    }  
  
    public static function getInstance()  
    {  
        if (self::$instance == null)  
        {  
            self::$instance = new Controller();  
        }  
  
        return self::$instance;  
    }  
}
```

## Proizvodi

```
// proizvodi
public function getProizvod(){
    $this->mydb->select();
    return $this->mydb->getResult();
}
public function getProizvodFiler($filter){
    $this->mydb->select("proizvod","*",null,null,null,$filter,null);
    return $this->mydb->getResult();
}
public function insertProizvod($values){
    $this->mydb->insert("proizvod","naziv,kategorija_id, cena, proizvodjac, kvantitet, slika_id",$values);
    return $this->mydb->getResult();
}
public function deleteProizvod($id){
    $this->mydb->delete("proizvod", "ID",$id);
    return $this->mydb->getResult();
}
public function updateProizvod($id,$keys, $values)
{
    $this->mydb->update("proizvod","id",$id, $keys,$values);

    return $this->mydb->getResult();
}
```

## Akcije / proizvodi na akciji / kategorije

*//akcije*

```
public function getAkcija(){
    $this->mydb->select("akcija","*", null, null, null, null, null);
    return $this->mydb->getResult();
}

public function getAkcijaFilter($filter){
    $this->mydb->select("akcija","*", null, null, null, $filter, null);
    return $this->mydb->getResult();
}

public function insertAkcija($values){
    $this->mydb->insert("akcija","naziv,opis, popust, datum_od, datum_do",$values);
    return $this->mydb->getResult();
}

public function updateAkcija($id,$keys, $values)
{
    $this->mydb->update("akcija","id",$id, $keys,$values);

    return $this->mydb->getResult();
}

//proizvodi na akciji
public function getProizvodiNaAkciji($akcija){
    $this->mydb->select("proizvodi_na_akciji","*", null, null, null, " akcija_id=$akcija", null);
    return $this->mydb->getResult();
}

public function insertProizvodNaAkciji($values){
    $this->mydb->insert("proizvodi_na_akciji","akcija_id,proizvod_id",$values);
    return $this->mydb->getResult();
}

// kategorije

public function getKategorija(){
    $this->mydb->select("kategorija","*", null, null, null, null, null);
    return $this->mydb->getResult();
}
```

## Korisnici / meni / slika

// korisnici

```
public function getKorisnik($email, $sifra){
    $this->mydb->select("korisnik","*", null, null, null, " email='$email' AND sifra='$sifra'" , null);
    return $this->mydb->getResult();
}

public function insertKorisnik($values){
    $values[]=2;
    $this->mydb->insert("korisnik","ime,prezime, email, sifra, adresa, uloga_id",$values);
    return $this->mydb->getResult();
}
```

// meni

```
public function getMeniStandard(){
    $where = ' naziv_linka IN ("Home", "About","Login","Store")';
    $this->mydb->select("meni","*", null, null, null, $where, null);
    return $this->mydb->getResult();
}

public function getMeniUser(){
    $where = ' naziv_linka IN ("Home", "About","Contact","Store", "Logout","Korpa")';
    $this->mydb->select("meni","*", null, null, null, $where, null);
    return $this->mydb->getResult();
}

public function getMeniAdmin(){
    $where = ' naziv_linka IN ("Admin", "Logout")';
    $this->mydb->select("meni","*", null, null, null, $where , null);
    return $this->mydb->getResult();
}
```

// slika

```
public function getSlika(){
    $this->mydb->select("slika","*", null, null, null, null, null);
    return $this->mydb->getResult();
}

public function insertSlika($values){
    $values[]=2;
    $this->mydb->insert("korisnik","url,naziv",$values);
    return $this->mydb->getResult();
}
```

Uloga / poruke / narudzbine

```
// uloga
public function getUloga(){
    $this->mydb->select("uloga","*", null, null, null, null, null);
    return $this->mydb->getResult();
}

// poruke

public function getPoruka(){
    $this->mydb->select("poruka","*", null, null, null, null, null);
    return $this->mydb->getResult();
}
public function insertPoruka($values){
    $values[]="NOW()";
    $this->mydb->insert("poruka","korisnik_id, sadrzaj, timestamp",$values);
    return $this->mydb->getResult();
}

// narudzbine

public function insertStavka($values){
    $narudzbine = $this->getNaruzbina();
    $narudzbina_id = $narudzbine->fetch_object()->id;
    $values[]=$narudzbina_id;
    $this->mydb->insert("stavka","proizvod_id,narudzbina_id",$values);
    return $this->mydb->getResult();
}

public function getNaruzbina(){
    $this->mydb->select("narudzbina","*", null, null, null, null, " id DESC LIMIT 1");
    return $this->mydb->getResult();
}

public function insertNaruzbina($values,$stavke){
    $values[]="NOW()";
    $this->mydb->insert("narudzbina","korisnik_id, ukupna_cena, datum",$values);

    foreach ($stavke as $stavka) {
        $this->insertStavka($stavka);
    }
    return $this->mydb->getResult();
}
```

## /obradameni.php

obradameni.php

```
1  <?php
2  include("control/Controller.php");
3  $ctrl = Controller::getInstance();
4
5  $meni_array = array();
6
7  session_start();
8
9  if(!empty($_SESSION['loggeduser'])):
10     if($_SESSION['loggeduser'][1]=="1"):
11         $meni = $ctrl->getMeniAdmin();
12         while($row=$meni->fetch_object()):
13             $pom = array("href"=>$row->link,"textLink"=>$row->naziv_linka);
14             $meni_array[]=$pom;
15         endwhile;
16         echo json_encode($meni_array);
17     elseif($_SESSION['loggeduser'][1]=="2"):
18         $meni = $ctrl->getMeniUser();
19         while($row=$meni->fetch_object()):
20             $pom = array("href"=>$row->link,"textLink"=>$row->naziv_linka);
21             $meni_array[]=$pom;
22         endwhile;
23         echo json_encode($meni_array);
24     else:
25         $meni = $ctrl->getMeniStandard();
26         while($row=$meni->fetch_object()):
27             $pom = array("href"=>$row->link,"textLink"=>$row->naziv_linka);
28             $meni_array[]=$pom;
29         endwhile;
30         echo json_encode($meni_array);
31     endif;
32 else:
33     $meni = $ctrl->getMeniStandard();
34     while($row=$meni->fetch_object()):
35         $pom = array("href"=>$row->link,"textLink"=>$row->naziv_linka);
36         $meni_array[]=$pom;
37     endwhile;
38     echo json_encode($meni_array);
39 endif;
40 ?>
```

11

## /login.php

login.php

```
1  <?php
2
3  include("control/Controller.php");
4  $ctrl = Controller::getInstance();
5
6
7  session_start();
8
9  if (isset($_POST['emailLog']) && isset($_POST['passwordLog'])) {
10     echo "usao";
11     $email = $_POST['emailLog'];
12     $password = $_POST['passwordLog'];
13     $korisnik = $ctrl->getKorisnik($email, $password);
14     $row= $korisnik->fetch_object();
15     $_SESSION['loggeduser'] = [$row->ime,$row->uloga_id];
16     header('Location: index.php');
17     exit();
18 }
19
20
21 ?>
```

## /logout.php

logout.php

```
1  <?php
2  session_start();
3  session_destroy();
4  header('Location: index.php');
5
6  exit();
7
8  ?>
```

