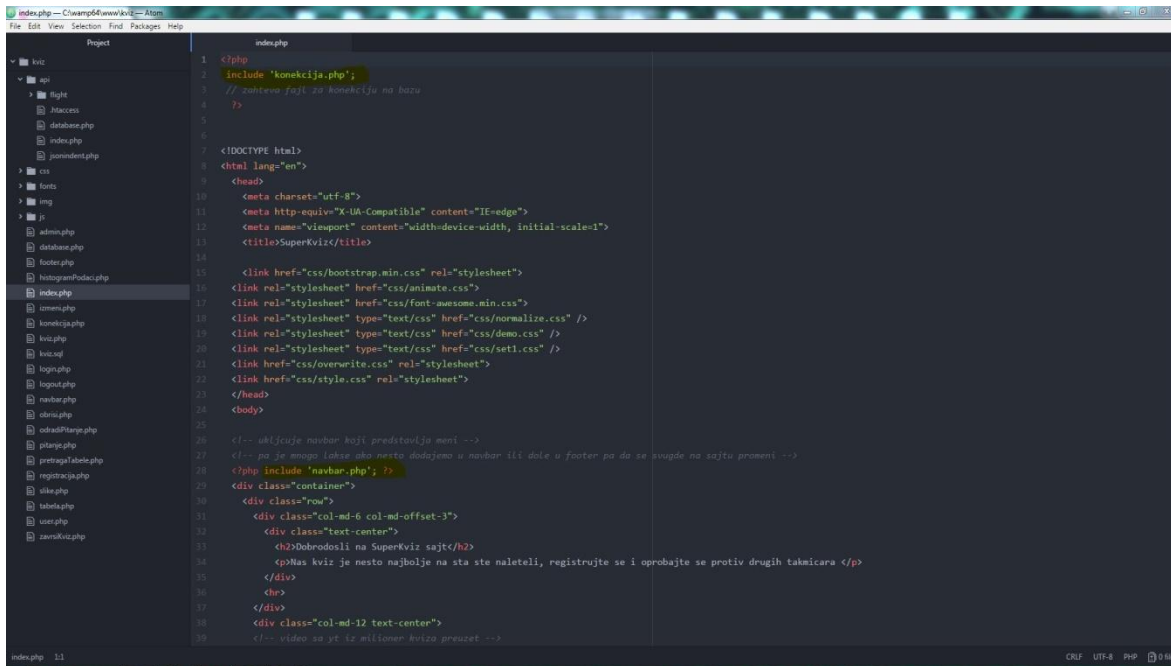


# Kviz projekat – II kolokvijum Veb programiranje

Obradjena veb aplikacija predstavlja igricu u I vidu kviza gde korisnici mogu da se registruju, zatim uloguju preko registrovanog naloga I igraju kviz. Zatim na kraju mogu da pregledaju svoje rezultate u tabeli rezultata. Korisnik koji je admin ima mogucnost i dodavanja novog pitanja kao pretrage korisnika i pitanja.

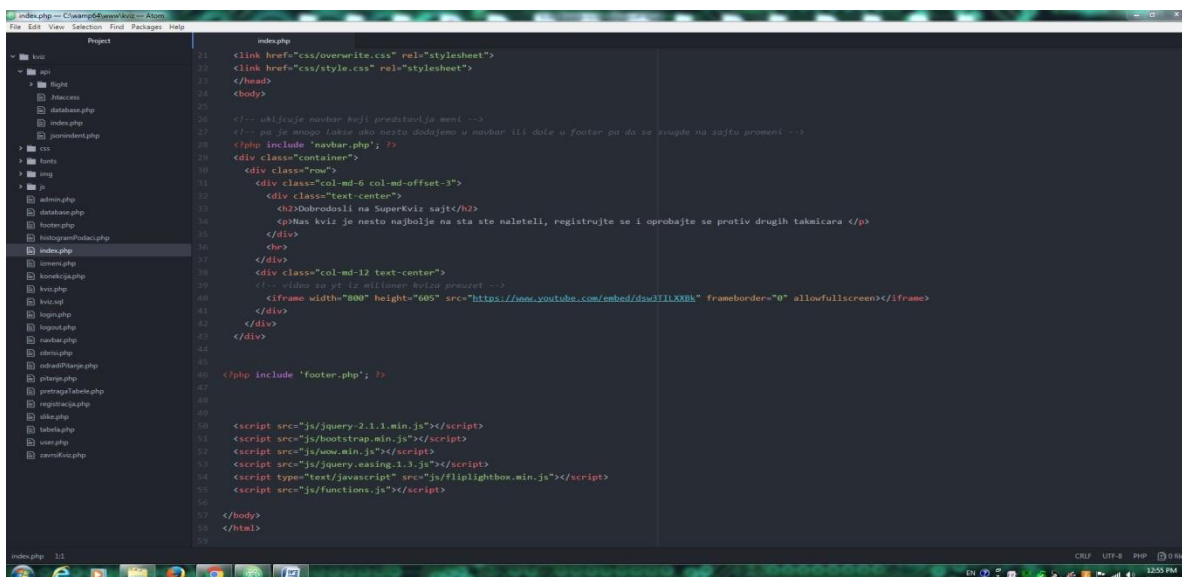
- 1) Stranica – index.php koja u sebi ukljucuje konekcija.php, navbar.php I footer.php . To su fajlovi koje skoro na svakoj stranici ukljucujemo pa ako se nesto menja dovoljno je jedno m u tom fajlu da se izmeni I svugde ce biti izmenjeno. Markirali smo ukljucenu konekciju I navbar. U zaglavlju imamo bootstrap biblioteku za css kao I ostale css fajlove.



```
1 <?php
2 include 'konekcija.php';
3 // zavisna fajl za konekciju na bazu
4 ?>
5
6 <!DOCTYPE html>
7 <html lang="en">
8 <head>
9
10 <meta charset="utf-8">
11 <meta http-equiv="X-UA-Compatible" content="IE=edge">
12 <meta name="viewport" content="width=device-width, initial-scale=1">
13 <title>SuperKviz</title>
14
15 <link href="css/bootstrap.min.css" rel="stylesheet">
16 <link rel="stylesheet" href="css/animate.css">
17 <link rel="stylesheet" href="css/font-awesome.min.css">
18 <link rel="stylesheet" type="text/css" href="css/normalize.css" />
19 <link rel="stylesheet" type="text/css" href="css/demo.css" />
20 <link rel="stylesheet" type="text/css" href="css/set1.css" />
21 <link href="css/overwrite.css" rel="stylesheet">
22 <link href="css/style.css" rel="stylesheet">
23 </head>
24
25
26 <!-- ukljucuje navbar koji predstavlja meni -->
27 <!-- pa je mnogo lakse ako nesto dodajemo u navbar ili dola u footer pa da se svugde na sajtu promeni -->
28 <?php include 'navbar.php'; ?>
29 <div class="container">
30 <div class="row">
31 <div class="col-md-6 col-md-offset-3">
32 <div class="text-center">
33 <h2>Dobrodosli na SuperKviz sajt</h2>
34 <p>Nas kviz je nesto najbolje na sta ste naleteli, registrujte se i probajte se protiv drugih takmicara </p>
35 </div>
36 </div>
37 </div>
38 <div class="col-md-12 text-center">
39 <!-- video sa yt iz milioner kviza prauzet -->
```

Na slici ispod je prikazana druga strana index.php fajla gde se koristi <iframe> tag za prikaz video sadržaja sa youtube servisa.

Takodje se vidi kako je uvezen **footer.php** fajl kao i javascript biblioteke I fajlovi. Koristio sam div kontejnere sa gotvovim nazivima klasa iz bootstrap.



```
21 <link href="css/overwrite.css" rel="stylesheet">
22 <link href="css/style.css" rel="stylesheet">
23 </head>
24 <body>
25
26 <!-- ukljucuje navbar koji predstavlja meni -->
27 <!-- pa je mnogo lakse ako nesto dodajemo u navbar ili dola u footer pa da se svugde na sajtu promeni -->
28 <?php include 'navbar.php'; ?>
29 <div class="container">
30 <div class="row">
31 <div class="col-md-6 col-md-offset-3">
32 <div class="text-center">
33 <h2>Dobrodosli na SuperKviz sajt</h2>
34 <p>Nas kviz je nesto najbolje na sta ste naleteli, registrujte se i probajte se protiv drugih takmicara </p>
35 </div>
36 </div>
37 </div>
38 <div class="col-md-12 text-center">
39 <!-- video sa yt iz milioner kviza prauzet -->
40 <iframe width="800" height="600" src="https://www.youtube.com/embed/daw71LXXBk" frameborder="0" allowfullscreen></iframe>
41 </div>
42 </div>
43
44 <?php include 'footer.php'; ?>
45
46
47 <script src="js/jquery-2.1.1.min.js"></script>
48 <script src="js/bootstrap.min.js"></script>
49 <script src="js/wow.min.js"></script>
50 <script src="js/jquery.easing.1.3.js"></script>
51 <script type="text/javascript" src="js/fliplightbox.min.js"></script>
52 <script src="js/functions.js"></script>
53
54 </body>
55 </html>
```

# Kviz projekat – II kolokvijum Veb programiranje

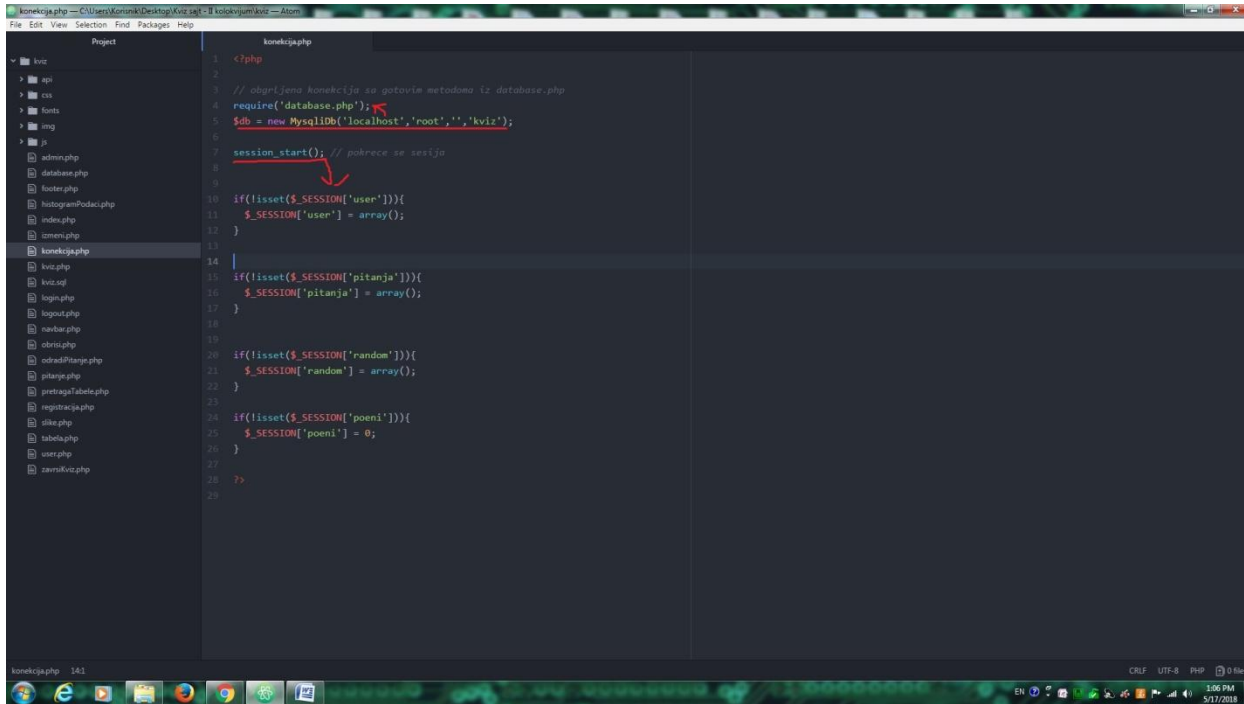
- 2) Konekcija.php je fajl koji je svugde uključen I tu se kreira konekcija na bazu preko localhost veze, određenim imenom, lozinkom I na određenu bazu. Zahtevali smo database.php fajl koji sam našao na internetu. To je klasa za instanciranje objekata za konekciju sa bazom u konstruktoru I određenim gotovim funkcijama za SQL klauzule I naredbe. Nakon toga naredbom **session\_start()** pokrenemo sesiju I ako nisu setovane vrednosti za određena polja koja će se posle objasniti setuj ih za kasnije provere u logovanju I kvizu.

**\$\_SESSION['pitanja']** će biti niz koji se puni sa pitanjima koja dolaze u kvizu.

**\$\_SESSION['user']** će biti niz koji se puni sa podacima korisnika kada se uloguje uspešno za kviz.

**\$\_SESSION['random']** će biti brojevi za ID pitanja gde će se proveravati da se ne izvuče pitanje koje je već bilo.

**\$\_SESSION['poeni']** će biti broj poena koji je korisnik osvojio.



- 3) **Tabela.php** je fajl u kome su prikazani rezultati korisnika koji su odigrali kviz. Mogu se sortirati po poenima, nazivima korisnika ili redosledom izvucenih pitanja. Kodom ispod smo prvo inicijalizovali zahtev za tabelom rezultata od api. Zatim smo setovali opciju za zahtev da vrati neke podatke kao transfer. Izvršavanjem zahteva dobijamo odgovor u json formatu. Dekodovanjem tog json formata sa listom objekata smestamo ih u niz koji zovemo tabela I zatvaramo zahtev za servisom. Preko foreach petlje za svaki element niza (element niza je jedna odigrana igra) ispisujemo za svako polje tabele Ime korisnika, broj poena I niz odradjenih pitanja.

**Na drugoj slici od tabela.php** je kreiranje histograma koji prikazuje pojavljivanje određenih rezultata u kvizu. Uvezli smo javascript google biblioteku za rad sa podacima u tabeli I dodatnim funkcionalnostima za sortiranja po definisanim kolonama.

Zatim sam pozvao **google funkcije** da se pri učitavanju stranice učitava **corechart histogram** I da se on iscrta pri učitavanju stranice.

Onda se preko ajax vade podaci sa histogramPodaci.php I parsiraju iz json I objekte koji se posle u funkciji ispod u **drawChart()** pripremaju za podatke za vizuelni prikaz I bira se naslov I pozicija za legend. Zatim se uzima div blok unutar koga se se iscrta histogram.

**#chart\_div** je blok koji se napuni tim grafikonom histograma kada se učitava stranica u funkciji **setOnLoadCallBack(drawChart)**

I na kraju se funkcijom **draw()** nad chart promeľljivom (sto je html div zapravo) iscrta histogram sa prosledjenim podacima I opcijama.

## Kviz projekat – II kolokvijum Veb programiranje

```
43 <thead>
44 <tr>
45 <th>Takmicar</th>
46 <th>Broj poena</th>
47 <th>Pitanja</th>
48 </tr>
49 </thead>
50 <tbody>
51 <?php
52     $zahtev = curl_init("http://localhost/kviz/api/tabela");
53     curl_setopt($zahtev, CURLOPT_RETURNTRANSFER, 1);
54     $odgovor = curl_exec($zahtev);
55     $tabela=json_decode($odgovor, true);
56     curl_close($zahtev);
57     foreach ($tabela as $t) {
58         <tr>
59             <td><?php echo $t['imePrezime'] ?></td>
60             <td><?php echo $t['brojPoena'] ?></td>
61             <td>
62                 <?php
63                 echo $t['pitanja']
64             </td>
65         </tr>
66     </tbody>
67 </table>
68 <?php
69 }
70 ?>
71 </tbody>
72 </table>
73 </tbody>
74 </table>
75 </tbody>
76 </table>
77 </tbody>
78 </table>
79 </tbody>
80 </table>
81 </tbody>
82 </table>
```

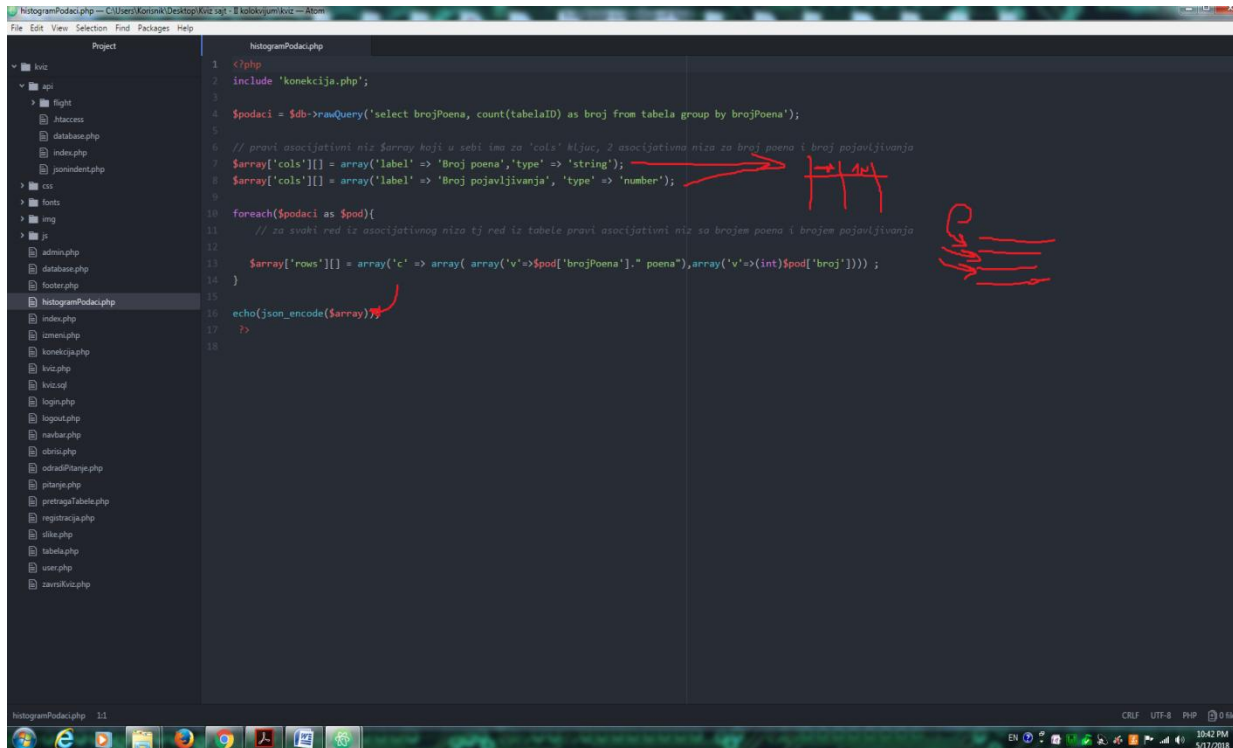
```
104 <!-- jquery biblioteka za bolji prikaz tabele i gomilom funkcija za tabele -->
105 <script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
106 <script>
107     $(document).ready(function(){
108         $('#tabela').DataTable();
109     });
110 </script>
111 <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
112 <script type="text/javascript">
113     google.charts.load("current", {packages:["corechart"]});
114     google.charts.setOnLoadCallback(drawChart);
115     var a;
116     $.ajax({
117         type: "POST",
118         url: 'histogramPodaci.php',
119         success: function (data) {
120             a = JSON.parse(data);
121         }
122     });
123     // funkcija za prikaz histograma
124     function drawChart() {
125         var data = new google.visualization.DataTable(a);
126         var options = {
127             title: 'Raspodela poena u kvizu',
128             legend: { position: 'none' },
129         };
130         var chart = new google.visualization.Histogram(document.getElementById('chart_div'));
131         chart.draw(data, options);
132     }
133 </script>
134 </body>
```

4) **histogramPodaci.php** je skripta koja iz baze izvlaci podatke za histogram I salje ih u json format preko ajax.

Prvo se skrpta poveze sa bazom pa se preko upita funkcije `rawQuery()` vade podaci o brojem rezultata sa istim brojem poena.

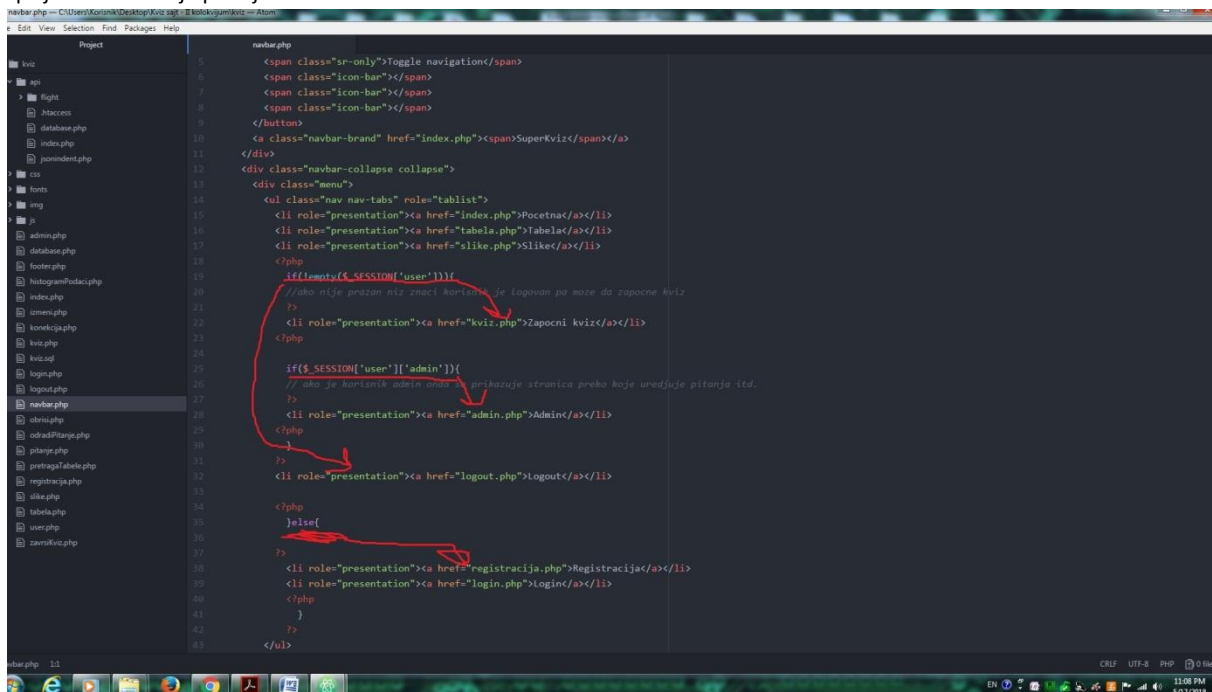
U \$rray asocijativni niz pravimo 2 zaglavlja za kolone Broj poena I Broj pojavljivanja, a ispod toga foreach petljom za svaki zapis iz upita unosimo u svaki red niza kao trodimenzionalni niz I na kraju pakujemo u JSON format koji ce se proslediti preko ajax nazad u **tabela.php** fajl nakon cega ce google funkcija vizualizacije da kreira histogram grafikon na osnovu ovih podataka.

## Kviz projekt – II kolokvijum Veb programiranje



```
1 <?php
2 include 'konekcija.php';
3
4 $podaci = $db->rawQuery('select brojPoema, count(tabelaID) as broj from tabela group by brojPoema');
5
6 // pravi asocijativni niz $array koji u sebi ima za 'cols' ključ, 2 asocijativna niza za broj poema i broj pojavljivanja
7 $array['cols'] = array('label' => 'Broj poema', 'type' => 'string');
8 $array['cols'][] = array('label' => 'Broj pojavljivanja', 'type' => 'number');
9
10 foreach($podaci as $pod){
11     // za svaki red iz asocijativnog niza tj red iz tabele pravi asocijativni niz sa brojem poema i brojem pojavljivanja
12     $array['rows'][] = array('c' => array( array('v'=>$pod['brojPoema'], " poema"), array('v'=>(int)$pod['broj']));
13 }
14
15 echo(json_encode($array));
16
17 ?>
```

- 5) Treba napomeniti samo da u **navbar.php** imamo php deo koda tj u meniju na sajtu koji prikazuje stranice za registrovanje i logovanje ako nismo ulogovani. Zatim ako smo ulogovani kao obican korisnik prikazuje stranicu za kviz, a ako smo ulogovani kao admin prikazuje dodatnu opciju za editovanje pitanja na admin stranici.



```
5 <span class="sr-only">Toggle navigation</span>
6 <span class="icon-bar"></span>
7 <span class="icon-bar"></span>
8 <span class="icon-bar"></span>
9 </button>
10 <a class="navbar-brand" href="index.php"><span>SuperKviz</span></a>
11 </div>
12 <div class="navbar-collapse collapse">
13 <div class="menu">
14 <ul class="nav navbar-nav" role="tablist">
15 <li role="presentation"><a href="index.php">Pocetna</a></li>
16 <li role="presentation"><a href="tabela.php">Tabela</a></li>
17 <li role="presentation"><a href="slike.php">Slike</a></li>
18 </ul>
19
20 <?php
21 // ako nije prazan niz znaci korisnik je logovan pa moze da zapocne kviz
22 <li role="presentation"><a href="kviz.php">Zapocni kviz</a></li>
23 </ul>
24
25 <?php
26 if($SESSION['user'] == 'admin'){
27     // ako je korisnik admin onda prikazuje stranica preba koje unodjuje pitanja itd.
28     <li role="presentation"><a href="admin.php">Admin</a></li>
29 }
30 </ul>
31 <li role="presentation"><a href="logout.php">Logout</a></li>
32 </ul>
33
34 <?php
35 }else{
36     <li role="presentation"><a href="registracija.php">Registracija</a></li>
37     <li role="presentation"><a href="login.php">Login</a></li>
38 }
39 </ul>
```

- 6) **Slike.php** je stranica koja kupi slike sa javnog Flickr apija ciji se url po određenim opcijama prosledjuje preko AJAX I pakuju u JSON format. Zatim se ti podaci preko each petlje za svaki item u nizu data.items pravi tag za sliku sa adresom koja se nalazi u svakom itemu pod media pojmem pa u njemu pod m poljem i nadodajemo u blok sa slikama. To se moze I proveriti kada se kao adresa unese [http://api.flickr.com/services/feeds/photos\\_public.gne?tags=quiz&tagmode=any&format=json](http://api.flickr.com/services/feeds/photos_public.gne?tags=quiz&tagmode=any&format=json) pa se nadje items pa unutar njega media pa m.

- 7) **Login.php** sadrži html formu (*desna slika*) po Bootstrap strukturi i prihvata podatke za username i password (autentifikacija) koji se prosledjuju php kodu (*leva slika*). Uvezena je skripta sa konekcijom na bazu kako bi smo proverili da li je korisnik u bazi. Za to nam takodje treba **user.php** skripta koja u sebi sadrži definiciju klase za korisnika i funkcije za validaciju registracije i logovanja. Funkcija **login()** iz klase user vraća Boolean vrednost nakon koje se kreira adekvatna poruka u zavisnosti od vrednosti promenljive **\$sacuvano**. I ta poruka se ispisuje na pocetku html forme. **Takodje ako je vracena true vrednost kreira se fajl 'sesija.txt' u modu za upis i kreira novi** ako je prethodno vec postojao sa istim imenom. Upisuje se poruka da je korisnik ulogovan i ispod iz POST polja username i password korisnika i na kraju se zatvara fajl. **Naredbe** `$r` znače prelazak u novi red radi bolje preglednosti sadržaja fajla.

```

5 <form id="forma" method="post" action="" role="form">
6 <?php if($poruka != ''){ ?>
7 <!-- Ako poruka od vrha nije prazna prikazi da li je uspesno ili neuspesno logovanje -->
8 <div class="well"><?php echo $poruka ?></div>
9 <?php ?>
10 <div class="row">
11 <div class="col-md-12">
12 <div class="form-group">
13 <label for="opis">Username *</label>
14 <input id="username" type="text" name="username" class="form-control" placeholder="Username *" >
15 </div>
16 </div>
17 </div>
18 <div class="row">
19 <div class="col-md-12">
20 <div class="form-group">
21 <label for="password">Password *</label>
22 <input id="password" type="password" name="password" class="form-control" placeholder="Password *" >
23 </div>
24 </div>
25 </div>
26
27 <div class="col-md-12">
28 <!-- preko POST metoda se salju username i password gore u php skriptu za proveru logovanja-->
29 <input id="login" name="login" type="submit" class="btn btn-success" value="Login">
30 </div>

```

- 8) **User.php** je klasa koja se koristi na stranicama za registraciju i logovanje. Konstruktor uzima polje kojim se instancira konekcija na bazu. Prva funkcija je za logovanje i u njoj se prvo username i password čiste na krajevima stringa i funkcijom **escape(definisana u database.php omotac klasi)** uklanjaju potencijalni štetni karakteri **povodom SQL Injection** napada. Zatim se ti sredjeni podaci prosledjuju promenljivama koje se čuvaju u niz **\$params**. Pa se preko **rawQuery(definisana u database.php omotac klasi)** u kome su bind-ovani podaci vadi taj korisnik. Pa se proverom dužine niza da je >0 inicijalizuje 'user' vrednost u sesiji i vraća true. U suprotnom se vraća false. U drugoj funkciji se radi isto sa podacima u početku samo što se oni nakon ciscenja pakuju u asocijativni niz i koduju u JSON format.



## Kviz projekat – II kolokvijum Veb programiranje

Zatim se preko zahtev koji se nalazi u api koji vraća **Flight::route** za noviKorisnik (**definisano u api/index.php**) metod inicijalizuje zahtev za dodavanje korisnika. Setuju se opcije da se iz POST nizu salju podaci i da se vrati transfer kad se izvrši zahtev. Nakon toga se odgovor dekoduje iz json u odgovor - **\$json\_objekat** (ima vrednost **OK** ili **GRESKA** - definisano u **api/index.php**) na osnovu kojeg se vraća boolean kao rezultat funkcije.

```
<?php

// klasa user za proveru logovanja i registracije
class User {

    private $db;

    // konstruktor preko koga se korisnik povezuje sa bazom
    public function __construct($db) {
        $this->db = $db;
    }

    // funkcija za login vraća true ako se korisnik nalazi u bazi jer znaci da je vec registrovan pa se moze logovati
    public function login() {

        $username = $this->db->escape(trim($_POST['username']));
        $password = $this->db->escape(trim($_POST['password']));

        $params = Array($username, $password);

        // rezultat upita se smesta u users promenljivu tj niz
        $users = $this->db->rawQuery("SELECT * FROM korisnik WHERE username = ? AND password = ? LIMIT 1", $params);

        if(count($users) > 0) {

            // inicijalizuj korisnika iz $users za sesiju
            $_SESSION['user'] = $users[0];

            return true;

        } else { // ako nema korisnika sa tom sifrom i lozinkom vrati false
            return false;
        }

    }

}

39 public function registracija(){
40
41     $ime = $this->db->escape(trim($_POST['ime']));
42     $username = $this->db->escape(trim($_POST['username']));
43     $password = $this->db->escape(trim($_POST['password']));
44
45
46     $data = Array (
47         "imePrezime" => $ime,
48         "username" => $username,
49         "password" => $password
50     );
51
52     $podaci = json_encode($data);
53
54     $zahtev = curl_init("http://localhost/kviz/api/noviKorisnik");
55
56     curl_setopt($zahtev, CURLOPT_POST, TRUE);
57     curl_setopt($zahtev, CURLOPT_POSTFIELDS, $podaci);
58     curl_setopt($zahtev, CURLOPT_RETURNTRANSFER, 1);
59
60     $odgovor = curl_exec($zahtev);
61
62     // dekoduje json i vraća u obliku statusnog odgovora "OK" ili "GRESKA"
63     // definisano za obradu json u ovom slucaju je u api/index.php fajlu
64     $json_objekat=json_decode($odgovor, true);
65     curl_close($zahtev);
66
67     if($json_objekat == "OK!") {
68         return true;
69     }
70     else {
71         return false;
72     }
73
74
75 }
```

- 9) **Registracija.php** je slicno kao login samo je dodato da se unese ime i prezime i password je tekstualnog tipa karaktera kako bi korisnik video sta je uneo pre nego sto se registruje. Podaci kad se unesu i kad se submituje forma preko POST metoda se podaci salju gore da se obrade u php kod. Zatim u php kod opet uključujemo konekciju pa proveravamo da li je setovano polje za **register** tj submit odradjen. Ako jeste onda se uvozi klasa User i kreira objekat kome se prosledjuje instance konekcije nad bazom i pozivima metod **registracija()** koji smo gore objasnili i koji vraća true ili false pa na osnovu toga kreiramo poruku koja ce se prikazati na pocetku forme u html.

## Kviz projekat – II kolokvijum Veb programiranje

```
1 <?php
2
3 include 'konekcija.php';
4 $poruka = '';
5 // ako je uspesno prosledjena forma za registraciju u
6 if(isset($_POST['register'])) {
7
8     include('user.php');
9
10    $user = new User($db);
11    // pokretanje registracije tog korisnika i njegov up
12    $sacuvano = $user->registracija();
13
14    if($sacuvano){
15        $poruka= 'Uspesno registrovan korisnik';
16    }else{
17        $poruka= 'Neuspesno registrovan korisnik';
18    }
19 }
20 ?>
21
22 <!DOCTYPE html>
```

```
33 <form id="forma" method="post" action="" role="form">
34     <?php if($poruka != ''){ ?>
35         <div class="well"><?php echo $poruka ?></div>
36     <?php ?>
37     <div class="row">
38         <div class="col-md-12">
39             <div class="form-group">
40                 <label for="ime">Ime i prezime *</label>
41                 <input id="ime" type="text" name="ime" class="form-control" placeholder="Ime i prezime *" >
42             </div>
43         </div>
44     </div>
45     <div class="row">
46         <div class="col-md-12">
47             <div class="form-group">
48                 <label for="username">Username *</label>
49                 <input id="username" type="text" name="username" class="form-control" placeholder="Username *" >
50             </div>
51         </div>
52     </div>
53     <div class="row">
54         <div class="col-md-12">
55             <div class="form-group">
56                 <label for="password">Password *</label>
57                 <input id="password" type="text" name="password" class="form-control" placeholder="Password *" >
58             </div>
59         </div>
60     </div>
61
62     <div class="col-md-12">
63         <input id="login" name="register" type="submit" class="btn btn-success btn-lg" value="Register">
64     </div>
```

- 10) Prelazimo u api folder gde imamo index.php u kome se nalaze sve rute – Flight::route za api zahteve koji ce se obradivati zajedno sa fajlom **database.php** u **tom api folderu** (to nije database.php u korenu projekta). Zahtevamo klasu **Flight.php** i **jsonindent.php** koji je skroz preuzet sa internet i sluzi za lepsi prikaz u JSON format. Registrujemo bazu i pravimo rute za GET i POST metode transfera (kod ovih funkcija je samo GET metoda). Definiseimo zaglavlje da je json i utf8 i uzimamo registrovanu bazu koja je registrovana na pocetku fajla. Pozivamo u sve 3 funkcije odredjene metode iz **database.php** klase koji vracaju sve podatke iz neke tabele. Zatim se u svakoj od tih funkcija while petljom red po red koji se kupi iz rezultata upita ubacuje u element niza i na kraju koduje u json da se salje preko curl funkcija.

```
<?php
require 'flight/Flight.php';
require 'jsonindent.php';

// registrujemo bazu preko Flight klase - FLIGHT je zapravo dodatni
Flight::register('db', 'Database', array('niz'));

// definisemo koren u bazi
Flight::route('/', function(){

    echo('GET i POST metode');

});

//pribavljanje svih pitanja - preko GET metoda uzimamo tabelu pitanja
Flight::route('GET /pitanja', function()
{
    header("Content-Type: application/json; charset=utf-8");

    $db = Flight::db();
    // pribavlja sva pitanja iz objekta $db koji pretstavlja objekat b
    $db->svePitanja();

    $niz = array();
    $iterator = 0;

    // u getResult se smestaju rezultati upita pa se kroz petlju uzima
    while ($red = $db->getResult()->fetch_object())
    {
        $niz[$iterator] = $red;
        $iterator += 1;
    }

    echo indent(json_encode($niz));
});
```

```
Flight::route('GET /tabela', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $db->sveRezultati();

    $niz = array();
    $iterator = 0;
    while ($red = $db->getResult()->fetch_object())
    {
        $niz[$iterator] = $red;
        $iterator += 1;
    }

    echo indent(json_encode($niz));
});

//pribavljanje svih korisnika
Flight::route('GET /korisnici', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();
    $db->sveKorisnici();

    $niz = array();
    $iterator = 0;
    while ($red = $db->getResult()->fetch_object())
    {
        $niz[$iterator] = $red;
        $iterator += 1;
    }

    echo indent(json_encode($niz));
});
```

# Kviz projekat – II kolokvijum Veb programiranje

Jedino je ruta za funkciju dodavanja novog korisnika preko POST metode. Sve je isto sto se tice pocetka fukcije samo se preko funkcije file\_get\_contents() kupe podaci iz input polja u json format koji se dekoduju u niz \$json\_data. Prosledjujemo funkciji za dodavanje novog korisnika te json podatke. Ako se upit uspesno izvrso vrati OK, ako nije vrati Greska! . Na kraju ispisemo u json format odgovor preko indent() funkcije za lepsi prikaz (jedni ispod drugih objekti)

```
// slanje podataka i dodavanje novog korisnika
Flight::route('POST /noviKorisnik', function()
{
    header("Content-Type: application/json; charset=utf-8");
    $db = Flight::db();

    // pribalvjam preko funkcije file_get_contents sve podatke koji su uneti
    $post_data = file_get_contents('php://input');

    $json_data = json_decode($post_data,true);

    $db->noviKorisnik($json_data);

    if($db->getResult())
    {
        $response = "OK!";
    }
    else
    {
        $response = "Greska!";
    }

    echo indent(json_encode($response));

});

Flight::start();
?>
```

- 11) U database.php fajlu smo definisali klasu koja kreira svoju konekciju preko prosledjenih parametara (hostname, username, password, dbname). Definisali smo polja za konekciju na pocetku klase a ispod smo u metodu za konekciju instancirali konekciju pa proverili ako je doslo do greske u povezivanju sa bazom da izbaci poruku preko printf (za formatiranje ispis) i pozove i prikaze error zatim izadje. Ako nije doslo do greske setuj utf8 set karaktera. Zatim smo ispod implementirali funkcije koje kreiraju i izrsavaju raw upite za izvlacenje svih pitanja, korisnika i rezultata (u upitu za rezultate kviza se koristi prirosno spajanje 2 tabele i vrsi sortiranje u opadajucem poretku). Jedino je drugačija funkcija za dodavanje novog korisnika jer se prvo ciste podaci za bazu preko escape i onda se preko INSERT naredbe i izrsavanjem upita se rezultat setuje na true ili false u zavisnosti od toga da li se upit dobro izvrso, dok kod drugih funkcija se rezultat setuje da bude ceo niz vracen iz rezultata upita koji bi posle koristili. Ispod na dnu se nalazi core funkcija koju sam preuzeo sa neta za izrsavanje upita

```
<?php
class Database {
    private $hostname = "localhost";
    private $username = "root";
    private $password = "root";
    private $dbname = "kviz";
    private $dblink;
    private $result = true; // moze biti ili boolean ili rezultat upita
    private $records;
    private $affectedRows;

    // konstruktor i instancira ime baze i konekciju
    function __construct($dbname)
    {
        $this->$dbname = $dbname;
        $this->Connect();
    }

    // pribavlja rezultate ili upita ili boolean vrednost
    public function getResult()
    {
        return $this->result;
    }

    // destruktor i zatvaranje konekcije baze
    function __destruct()
    {
        $this->$dblink->close();
    }

    // funkcija za konekciju na bazu
    function Connect()
    {
        // kreiranje konekcije
        $this->$dblink = new mysqli($this->hostname, $this->username, $this->password, $this->dbname);
        if($this->$dblink->connect_errno) // ako je doslo do greske u konekciji
        {
            printf("Konekcija neuspesna: %s\n", $mysqli->connect_error);
            exit();
        }
        $this->$dblink->set_charset("utf8"); // ako je uspesna konekcija setuj set karaktera na utf8
    }
}
```

```
function noviKorisnik($data) {
    $mysqli = new mysqli("localhost", "root", "root", "kviz");

    // filtrira i sredjuje string iz json formata
    $ime = mysqli_real_escape_string($mysqli,$data['imePrezime']);
    $username = mysqli_real_escape_string($mysqli,$data['username']);
    $password = mysqli_real_escape_string($mysqli,$data['password']);

    $values = "(".$ime.", ".$username.", ".$password.", 0)";

    // upit koji dodaje podatke u novom korisniku
    $query = "INSERT into korisnik (imePrezime, username, password, admin) VALUES ".$values;

    if($mysqli->query($query))
    {
        $this->result = true;
    }
    else
    {
        $this->result = false;
    }
    $mysqli->close();
}

// pribavlja upit o svim korisnicima
function sviKorisnici() {
    $mysqli = new mysqli("localhost", "root", "", "kviz");
    $q = "SELECT * FROM korisnik ";
    $this->result = $mysqli->query($q);
    $mysqli->close();
}

// pribavlja sve rezultate korisnika u opadajucem redosledu
function sviRezultati() {
    $mysqli = new mysqli("localhost", "root", "", "kviz");
    $q = "SELECT * FROM tabela t join korisnik k on t.korisnikID = k.korisnikID order by t.brojPoeni desc";
    $this->result = $mysqli->query($q);
    $mysqli->close();
}

// izvlaci sve pitanja koja postoje u bazi
function svaPitanja() {
    $mysqli = new mysqli("localhost", "root", "", "kviz");
    $q = "SELECT * FROM pitanje ";
    $this->result = $mysqli->query($q);
    $mysqli->close();
}
```



## Kviz projekat – II kolokvijum Veb programiranje

- 12) **Kviz.php** fajl sadrži blok u kome će se ispisati izvučeno pitanje, **#loadbar** koji se nakon učitavanja novog pitanja pravi animaciju prikazivanja. Prvo se izvlaci pitanje u php kodu. Proveravamo da li je slučajno broj pitanja dosao do 5. Ako jeste idi na kraj kviza preko **header funkcije** **završiKviz.php**. Inace se izvlace pitanja preko **get** funkcije u **database.php** koja izvlaci 1 smesta u niz sva pitanja. Zatim se preko **rand** funkcije izvlaci nasumican broj od 0 do brojPitanja-1. Pa se preko uslova petlje gde se proverava da li je taj izvučen broj u nizu random izvucenih brojeva (ID pitanja koja su vec bila). Ako jeste ponovo se izvlaci broj za ID pitanja. Na kraju se izvuče pitanje iz niza sa indeksom tog broja i taj broj se smesti u niz random izvucenih brojeva.

```
1 <?php
2
3 include 'konekcija.php';
4
5 // proveravamo da li je niz napunjen do 5 pitanja pa ako jeste
6 if(count($_SESSION['pitanja']) == 5){
7     header("Location: završiKviz.php");
8 }
9
10 // pribavi sva pitanja preko upita a metod get je definisan u
11 $pitanja = $db->get('pitanje');
12
13 $broj = rand(0, count($pitanja)-1);
14
15 // proveravamo da li se nasumicni broj nalazi kao ID pitanja
16 while(in_array($broj, $_SESSION['random'])){
17     $broj = rand(0, count($pitanja)-1);
18 }
19
20 $pitanje = $pitanja[$broj];
21 array_push($_SESSION['random'], $broj); // i na kraju se smesti
22
23 ?>
24
25 <!DOCTYPE html>
26 <html lang="en">
27 <head>
```

Prikaze se u **h3** tagu izvučeno pitanje i dole su kao labele dati ponudjeni odgovori koji sadrže u sebi input polja u vidu radio button sa **value {A, B, C, D}** kao ponudjeni odgovori i kada se klikne na njih taj izbor se salje u **functions.js** skriptu koja izmedju ostalog obradjuje izbor pa kad obradi vraca se nazad na **kviz.php** za novo pitanje. Takodje tu se nalazi i jQuery funkcija za obradu **loadbar** kojim se pri učitavanju pravi animacija za blok pitanja.

```
1 <div class="container-fluid bg-info">
2   <div class="modal-dialog">
3     <div class="modal-content">
4       <div class="modal-header">
5         <!-- ispisuje se pitanje koje je nasumicno izvučeno iz baze preko ID kao redni broj i tekst pitanja se ispisuje -->
6         <h3><span class="label label-warning" value="2" id="qid"><?php echo $pitanje['pitanjeID']; ></span> <?php echo $pitanje['pitanje']; ></h3>
7       </div>
8       <div class="modal-body">
9         <div class="col-xs-3 col-xs-offset-5">
10           <div id="loadbar" style="display: none;">
11             <div class="block6" id="rotate6_01"></div>
12             <div class="block6" id="rotate6_02"></div>
13             <div class="block6" id="rotate6_03"></div>
14             <div class="block6" id="rotate6_04"></div>
15             <div class="block6" id="rotate6_05"></div>
16             <div class="block6" id="rotate6_06"></div>
17             <div class="block6" id="rotate6_07"></div>
18             <div class="block6" id="rotate6_08"></div>
19           </div>
20         </div>
21       </div>
22     </div>
23   </div>
24 </div>
25
26 <!-- Labele za ponudjenje odgovore u kvizu -->
27 <div class="quiz" id="quiz" data-toggle="buttons">
28   <!-- kada se klikne na neki od radio button kao odgovor pokrece se js/functions.js skripta gde je uglavnom jQuery -->
29   <label class="element-animation1 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="glyphicon glyphicon-chevron-right"></i></span> <input type="radio" name="q_answer" value="a"><?php echo $pitanje['A']; ></label>
30   <label class="element-animation2 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="glyphicon glyphicon-chevron-right"></i></span> <input type="radio" name="q_answer" value="b"><?php echo $pitanje['B']; ></label>
31   <label class="element-animation3 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="glyphicon glyphicon-chevron-right"></i></span> <input type="radio" name="q_answer" value="c"><?php echo $pitanje['C']; ></label>
32   <label class="element-animation4 btn btn-lg btn-primary btn-block"><span class="btn-label"><i class="glyphicon glyphicon-chevron-right"></i></span> <input type="radio" name="q_answer" value="d"><?php echo $pitanje['D']; ></label>
33 </div>
34
35 <div class="modal-footer text-muted">
36   <span id="answer"></span>
37   <div>
38     <div>
39     <div>
40     <div>
41   </div>
42   <div>
43     <div>
44     <div>
45     <div>
46   </div>
47   <div>
48     <div>
49     <div>
50     <div>
51   </div>
52   <div>
53     <div>
54     <div>
55     <div>
56   </div>
57   <div>
58     <div>
59     <div>
60     <div>
61   </div>
62   <div>
63     <div>
64     <div>
65     <div>
66   </div>
67   <div>
68     <div>
69     <div>
70     <div>
71   </div>
72   <div>
73     <div>
74     <div>
75     <div>
76   </div>
77   <div>
78     <div>
79     <div>
80     <div>
81   </div>
82   <div>
83     <div>
84     <div>
85     <div>
86   </div>
87   <div>
88     <div>
89     <div>
90     <div>
91   </div>
92   <div>
93     <div>
94     <div>
95     <div>
96   </div>
97   <div>
98     <div>
99     <div>
100    <div>
101  </div>
102  <div>
103    <div>
104    <div>
105    <div>
106  </div>
107  <div>
108    <div>
109    <div>
110    <div>
111  </div>
112  <div>
113    <div>
114    <div>
115    <div>
116  </div>
117  <div>
118    <div>
119    <div>
120    <div>
121  </div>
122  <div>
123    <div>
124    <div>
125    <div>
126  </div>
127  <div>
128    <div>
129    <div>
130    <div>
131  </div>
132  <div>
133    <div>
134    <div>
135    <div>
136  </div>
137  <div>
138    <div>
139    <div>
140    <div>
141  </div>
142  <div>
143    <div>
144    <div>
145    <div>
146  </div>
147  <div>
148    <div>
149    <div>
150    <div>
151  </div>
152  <div>
153    <div>
154    <div>
155    <div>
156  </div>
157  <div>
158    <div>
159    <div>
160    <div>
161  </div>
162  <div>
163    <div>
164    <div>
165    <div>
166  </div>
167  <div>
168    <div>
169    <div>
170    <div>
171  </div>
172  <div>
173    <div>
174    <div>
175    <div>
176  </div>
177  <div>
178    <div>
179    <div>
180    <div>
181  </div>
182  <div>
183    <div>
184    <div>
185    <div>
186  </div>
187  <div>
188    <div>
189    <div>
190    <div>
191  </div>
192  <div>
193    <div>
194    <div>
195    <div>
196  </div>
197  <div>
198    <div>
199    <div>
200    <div>
201  </div>
202  <div>
203    <div>
204    <div>
205    <div>
206  </div>
207  <div>
208    <div>
209    <div>
210    <div>
211  </div>
212  <div>
213    <div>
214    <div>
215    <div>
216  </div>
217  <div>
218    <div>
219    <div>
220    <div>
221  </div>
222  <div>
223    <div>
224    <div>
225    <div>
226  </div>
227  <div>
228    <div>
229    <div>
230    <div>
231  </div>
232  <div>
233    <div>
234    <div>
235    <div>
236  </div>
237  <div>
238    <div>
239    <div>
240    <div>
241  </div>
242  <div>
243    <div>
244    <div>
245    <div>
246  </div>
247  <div>
248    <div>
249    <div>
250    <div>
251  </div>
252  <div>
253    <div>
254    <div>
255    <div>
256  </div>
257  <div>
258    <div>
259    <div>
260    <div>
261  </div>
262  <div>
263    <div>
264    <div>
265    <div>
266  </div>
267  <div>
268    <div>
269    <div>
270    <div>
271  </div>
272  <div>
273    <div>
274    <div>
275    <div>
276  </div>
277  <div>
278    <div>
279    <div>
280    <div>
281  </div>
282  <div>
283    <div>
284    <div>
285    <div>
286  </div>
287  <div>
288    <div>
289    <div>
290    <div>
291  </div>
292  <div>
293    <div>
294    <div>
295    <div>
296  </div>
297  <div>
298    <div>
299    <div>
300    <div>
301  </div>
302  <div>
303    <div>
304    <div>
305    <div>
306  </div>
307  <div>
308    <div>
309    <div>
310    <div>
311  </div>
312  <div>
313    <div>
314    <div>
315    <div>
316  </div>
317  <div>
318    <div>
319    <div>
320    <div>
321  </div>
322  <div>
323    <div>
324    <div>
325    <div>
326  </div>
327  <div>
328    <div>
329    <div>
330    <div>
331  </div>
332  <div>
333    <div>
334    <div>
335    <div>
336  </div>
337  <div>
338    <div>
339    <div>
340    <div>
341  </div>
342  <div>
343    <div>
344    <div>
345    <div>
346  </div>
347  <div>
348    <div>
349    <div>
350    <div>
351  </div>
352  <div>
353    <div>
354    <div>
355    <div>
356  </div>
357  <div>
358    <div>
359    <div>
360    <div>
361  </div>
362  <div>
363    <div>
364    <div>
365    <div>
366  </div>
367  <div>
368    <div>
369    <div>
370    <div>
371  </div>
372  <div>
373    <div>
374    <div>
375    <div>
376  </div>
377  <div>
378    <div>
379    <div>
380    <div>
381  </div>
382  <div>
383    <div>
384    <div>
385    <div>
386  </div>
387  <div>
388    <div>
389    <div>
390    <div>
391  </div>
392  <div>
393    <div>
394    <div>
395    <div>
396  </div>
397  <div>
398    <div>
399    <div>
400    <div>
401  </div>
402  <div>
403    <div>
404    <div>
405    <div>
406  </div>
407  <div>
408    <div>
409    <div>
410    <div>
411  </div>
412  <div>
413    <div>
414    <div>
415    <div>
416  </div>
417  <div>
418    <div>
419    <div>
420    <div>
421  </div>
422  <div>
423    <div>
424    <div>
425    <div>
426  </div>
427  <div>
428    <div>
429    <div>
430    <div>
431  </div>
432  <div>
433    <div>
434    <div>
435    <div>
436  </div>
437  <div>
438    <div>
439    <div>
440    <div>
441  </div>
442  <div>
443    <div>
444    <div>
445    <div>
446  </div>
447  <div>
448    <div>
449    <div>
450    <div>
451  </div>
452  <div>
453    <div>
454    <div>
455    <div>
456  </div>
457  <div>
458    <div>
459    <div>
460    <div>
461  </div>
462  <div>
463    <div>
464    <div>
465    <div>
466  </div>
467  <div>
468    <div>
469    <div>
470    <div>
471  </div>
472  <div>
473    <div>
474    <div>
475    <div>
476  </div>
477  <div>
478    <div>
479    <div>
480    <div>
481  </div>
482  <div>
483    <div>
484    <div>
485    <div>
486  </div>
487  <div>
488    <div>
489    <div>
490    <div>
491  </div>
492  <div>
493    <div>
494    <div>
495    <div>
496  </div>
497  <div>
498    <div>
499    <div>
500    <div>
501  </div>
502  <div>
503    <div>
504    <div>
505    <div>
506  </div>
507  <div>
508    <div>
509    <div>
510    <div>
511  </div>
512  <div>
513    <div>
514    <div>
515    <div>
516  </div>
517  <div>
518    <div>
519    <div>
520    <div>
521  </div>
522  <div>
523    <div>
524    <div>
525    <div>
526  </div>
527  <div>
528    <div>
529    <div>
530    <div>
531  </div>
532  <div>
533    <div>
534    <div>
535    <div>
536  </div>
537  <div>
538    <div>
539    <div>
540    <div>
541  </div>
542  <div>
543    <div>
544    <div>
545    <div>
546  </div>
547  <div>
548    <div>
549    <div>
550    <div>
551  </div>
552  <div>
553    <div>
554    <div>
555    <div>
556  </div>
557  <div>
558    <div>
559    <div>
560    <div>
561  </div>
562  <div>
563    <div>
564    <div>
565    <div>
566  </div>
567  <div>
568    <div>
569    <div>
570    <div>
571  </div>
572  <div>
573    <div>
574    <div>
575    <div>
576  </div>
577  <div>
578    <div>
579    <div>
580    <div>
581  </div>
582  <div>
583    <div>
584    <div>
585    <div>
586  </div>
587  <div>
588    <div>
589    <div>
590    <div>
591  </div>
592  <div>
593    <div>
594    <div>
595    <div>
596  </div>
597  <div>
598    <div>
599    <div>
600    <div>
601  </div>
602  <div>
603    <div>
604    <div>
605    <div>
606  </div>
607  <div>
608    <div>
609    <div>
610    <div>
611  </div>
612  <div>
613    <div>
614    <div>
615    <div>
616  </div>
617  <div>
618    <div>
619    <div>
620    <div>
621  </div>
622  <div>
623    <div>
624    <div>
625    <div>
626  </div>
627  <div>
628    <div>
629    <div>
630    <div>
631  </div>
632  <div>
633    <div>
634    <div>
635    <div>
636  </div>
637  <div>
638    <div>
639    <div>
640    <div>
641  </div>
642  <div>
643    <div>
644    <div>
645    <div>
646  </div>
647  <div>
648    <div>
649    <div>
650    <div>
651  </div>
652  <div>
653    <div>
654    <div>
655    <div>
656  </div>
657  <div>
658    <div>
659    <div>
660    <div>
661  </div>
662  <div>
663    <div>
664    <div>
665    <div>
666  </div>
667  <div>
668    <div>
669    <div>
670    <div>
671  </div>
672  <div>
673    <div>
674    <div>
675    <div>
676  </div>
677  <div>
678    <div>
679    <div>
680    <div>
681  </div>
682  <div>
683    <div>
684    <div>
685    <div>
686  </div>
687  <div>
688    <div>
689    <div>
690    <div>
691  </div>
692  <div>
693    <div>
694    <div>
695    <div>
696  </div>
697  <div>
698    <div>
699    <div>
700    <div>
701  </div>
702  <div>
703    <div>
704    <div>
705    <div>
706  </div>
707  <div>
708    <div>
709    <div>
710    <div>
711  </div>
712  <div>
713    <div>
714    <div>
715    <div>
716  </div>
717  <div>
718    <div>
719    <div>
720    <div>
721  </div>
722  <div>
723    <div>
724    <div>
725    <div>
726  </div>
727  <div>
728    <div>
729    <div>
730    <div>
731  </div>
732  <div>
733    <div>
734    <div>
735    <div>
736  </div>
737  <div>
738    <div>
739    <div>
740    <div>
741  </div>
742  <div>
743    <div>
744    <div>
745    <div>
746  </div>
747  <div>
748    <div>
749    <div>
750    <div>
751  </div>
752  <div>
753    <div>
754    <div>
755    <div>
756  </div>
757  <div>
758    <div>
759    <div>
760    <div>
761  </div>
762  <div>
763    <div>
764    <div>
765    <div>
766  </div>
767  <div>
768    <div>
769    <div>
770    <div>
771  </div>
772  <div>
773    <div>
774    <div>
775    <div>
776  </div>
777  <div>
778    <div>
779    <div>
780    <div>
781  </div>
782  <div>
783    <div>
784    <div>
785    <div>
786  </div>
787  <div>
788    <div>
789    <div>
790    <div>
791  </div>
792  <div>
793    <div>
794    <div>
795    <div>
796  </div>
797  <div>
798    <div>
799    <div>
800    <div>
801  </div>
802  <div>
803    <div>
804    <div>
805    <div>
806  </div>
807  <div>
808    <div>
809    <div>
810    <div>
811  </div>
812  <div>
813    <div>
814    <div>
815    <div>
816  </div>
817  <div>
818    <div>
819    <div>
820    <div>
821  </div>
822  <div>
823    <div>
824    <div>
825    <div>
826  </div>
827  <div>
828    <div>
829    <div>
830    <div>
831  </div>
832  <div>
833    <div>
834    <div>
835    <div>
836  </div>
837  <div>
838    <div>
839    <div>
840    <div>
841  </div>
842  <div>
843    <div>
844    <div>
845    <div>
846  </div>
847  <div>
848    <div>
849    <div>
850    <div>
851  </div>
852  <div>
853    <div>
854    <div>
855    <div>
856  </div>
857  <div>
858    <div>
859    <div>
860    <div>
861  </div>
862  <div>
863    <div>
864    <div>
865    <div>
866  </div>
867  <div>
868    <div>
869    <div>
870    <div>
871  </div>
872  <div>
873    <div>
874    <div>
875    <div>
876  </div>
877  <div>
878    <div>
879    <div>
880    <div>
881  </div>
882  <div>
883    <div>
884    <div>
885    <div>
886  </div>
887  <div>
888    <div>
889    <div>
890    <div>
891  </div>
892  <div>
893    <div>
894    <div>
895    <div>
896  </div>
897  <div>
898    <div>
899    <div>
900    <div>
901  </div>
902  <div>
903    <div>
904    <div>
905    <div>
906  </div>
907  <div>
908    <div>
909    <div>
910    <div>
911  </div>
912  <div>
913    <div>
914    <div>
915    <div>
916  </div>
917  <div>
918    <div>
919    <div>
920    <div>
921  </div>
922  <div>
923    <div>
924    <div>
925    <div>
926  </div>
927  <div>
928    <div>
929    <div>
930    <div>
931  </div>
932  <div>
933    <div>
934    <div>
935    <div>
936  </div>
937  <div>
938    <div>
939    <div>
940    <div>
941  </div>
942  <div>
943    <div>
944    <div>
945    <div>
946  </div>
947  <div>
948    <div>
949    <div>
950    <div>
951  </div>
952  <div>
953    <div>
954    <div>
955    <div>
956  </div>
957  <div>
958    <div>
959    <div>
960    <div>
961  </div>
962  <div>
963    <div>
964    <div>
965    <div>
966  </div>
967  <div>
968    <div>
969    <div>
970    <div>
971  </div>
972  <div>
973    <div>
974    <div>
975    <div>
976  </div>
977  <div>
978    <div>
979    <div>
980    <div>
981  </div>
982  <div>
983    <div>
984    <div>
985    <div>
986  </div>
987  <div>
988    <div>
989    <div>
990    <div>
991  </div>
992  <div&gt
```

## Kviz projekat – II kolokvijum Veb programiranje

- 13) **Functions.js** skripta koja obradjuje scrollup u footer.php koji nas vraća a vrh strane kad god odemo na dno preko provere za **scrollTop**, obradjuje animaciju za loader tako sto prikazuje i skriva **loadbar** kod ucitavanja svakog pitanja. Funkcija koja obradjuje izbor registruje klik na tag labela i izvuci izbor preko input polja tipa radio button i izvuci ID pitanja preko #qid sto je id html taga za ID pitanja pa se onda preko **ajax POST metodom salju podaci na obradu u obradiPitanje.php**. I na kraju se kad se obradi pitanje prikaze vratena poruka iz php skripte koja obradjuje pitanje i prikazuje se da li je odgovor tacan ili ne i broj poena. Pa se opet ucitava **kviz.php** i tako u krug dok se broj pitanja ne poveca do 5 sto je ogranicenje za kraj kviza.

```
1 (function ($) {
2 |
3   $(window).scroll(function(){
4     if ($(this).scrollTop() > 100) {
5       $('.scrollup').fadeIn();
6     } else {
7       $('.scrollup').fadeOut();
8     }
9   });
10  $('.scrollup').click(function(){
11    $("html, body").animate({ scrollTop: 0 }, 1000);
12    return false;
13  });
14
15  wow = new WOW({}).init();
16
17  $(function(){
18    var loading = $('#loadbar').hide();
19    $(document)
20      .ajaxStart(function () {
21        loading.show();
22      })
23      .ajaxStop(function () {
24        loading.hide();
25      });
26
27    $(".label.btn").on('click',function () {
28      var choice = $(this).find('input:radio').val();
29      console.log(choice);
30
31      var idPitanja = $('#qid').text();
32      console.log(idPitanja);
33
34      $.ajax({
35        method: "POST",
36        url: "odradiPitanje.php",
37        data: { izbor: choice, idPitanja: idPitanja }
38      })
39
40      .done(function( msg ) {
41        alert( msg );
42        window.location.href = "kviz.php"; // učitaj opet i predji kviz.php sledeću
43        // i tako u krug se vraća nazad na kviz.php dok se broj pitanja u nizu ne napuni do 5
44      });
45    });
46  });
47
48 })(jQuery);
```

- 14) **odradiPitanje.php** je skripta koja ukljucuje konekciju na bazu samim tim i omotac klasu sa predefinisanim funkcijama nad bazom. Podaci za izbor i ID pitanja koja smo dobili preko **ajax** cuvamo i zatim izvršimo upit sa where naredbom da nadje to jedno pitanje sa tim ID i sacuvamo ga kad nam vrati preko **getOne()** funkcije za tu tabelu 'pitanje'. Ubacimo u niz pitanja u sesiji taj ID pitanja. Zatim proveravamo da li je nas izbor jednak onom sto je u polju pod 'tacan' a to su vrednosti {a,b,c,d} pa ako jeste inkrementiraj broj poena i setuj poruku. Ako nije tacno samo setuj negativnu poruku i na kraju ispisi poruku koja ce se vratiti kao nazad u **functions.js** za **alert**.

```
1 <?php
2
3 include('konekcija.php');
4
5 // izvucem sto smo u ajax slali nas izbor i od tog pitanja id
6 $izbor = $_POST['izbor'];
7 $id = $_POST['idPitanja'];
8 $poruka = '';
9
10
11 $db->where('pitanjeID',$id);
12
13 // vraća nam se zapis (asocijativni niz) o tom pitanju iz baze preko funkcije
14 $pit = $db->getOne('pitanje');
15
16 array_push($_SESSION['pitanja'],$id); // ubaci u niz odradjenih pitanja id
17
18 if($pit['tacan'] == $izbor){
19   $_SESSION['poeni'] = $_SESSION['poeni'] + 1; // inkrementira se broj poena
20   $poruka = 'Tacno. Trenutni broj poena je : '.$_SESSION['poeni'];
21 }else{
22   $poruka = 'Netacno. Trenutni broj poena je : '.$_SESSION['poeni'];
23 }
24 echo($poruka);
25 ?>
26
```

## Kviz projekat – II kolokvijum Veb programiranje

- 15) **zavrshiKviz.php** je skripta koja ucitava konekciju za rad sa bazom. Cuva u promenljive broj poena, ID korisnika i niz pitanja. Niz pitanja pretvara u string funkcijom **implode()** i umetanjem zarezima izmedju svakog ID pitanja. Posle sve pakuje u asocijativni niz i preko predefinisnog upita insertuje podatke nad tabelom 'tabela' koja vraca na kraju Boolean u zavisnosti od toga da li su uspesno uneti podaci. Pa ako je uspesno sacuvano **resetuju se vrednosti niza SESIJE** i ucitava se tabela.php stranica sa prikazom rezultata i histograma. Ako nije ispravno uneto izbacuje se poruka o gresci.

```
1 <?php
2 include("konekcija.php");
3
4 // pribavlja se broj poena, ID korisnika koji je logovan
5 $brPoena = $_SESSION['poeni'];
6 $korisnikID= $_SESSION['user']['korisnikID'];
7 $pitanjaString = implode(",",$SESSION['pitanja']); // pribavlja se string sa nizom ID pitanja
8
9 $data = Array (
10     "korisnikID" => $korisnikID,
11     "brojPoena" => $brPoena,
12     "pitanja" => $pitanjaString
13 );
14
15 $sacuvano = $db->insert('tabela', $data);
16
17 // ako je uspesno izvršen upit resetuj poene, ID-jeve pitanja i random izvučena pitanja
18 if($sacuvano){
19     $_SESSION['poeni'] = 0;
20     $_SESSION['pitanja'] = array();
21     $_SESSION['random'] = array();
22
23     header("location:tabela.php");
24 }
25 else{
26     echo 'GRESKA';
27 }
28 }>
```

- 16) **admin.php** je skripta na kojoj admin ima pregled svih pitanja kao i da dodaje, menja ili obriše pitanje. Gore pri vrhu na liniji 52 ce nam se ispisati statusna poruka da li je ispravno uneto pitanje. Zatim imamo input polja za unos pitanja, ponudjene i tacan odgovor kao select meni sa opcijama {a,b,c,d} i na kraju dugme za slanje preko POST metoda skripti na pocetku fajla za obradu.

```
48 <div class="col-md-12 text-center">
49     <!-- forma za unos novog pitanja kojom se salju podaci preko post metode-->
50     <form id="forma" method="post" action="" role="form">
51         <?php if($poruka != ''){ }>
52         <div class="well"><?php echo $poruka ?></div>
53         <?php }>
54         <div class="row">
55             <div class="col-md-12">
56                 <div class="form-group">
57                     <label for="pitanje">Pitanje *</label>
58                     <input id="pitanje" type="text" name="pitanje" class="form-control" placeholder="Pitanje *" >
59                 </div>
60             </div>
61             <div class="row">
62                 <div class="col-md-12">
63                     <div class="form-group">
64                         <label for="a">A</label>
65                         <input id="a" type="text" name="a" class="form-control" placeholder="A" >
66                     </div>
67                 </div>
68                 <div class="row">
69                     <div class="col-md-12">
70                         <div class="form-group">
71                             <label for="b">B</label>
72                             <input id="b" type="text" name="b" class="form-control" placeholder="B" >
73                         </div>
74                     </div>
75                     <div class="row">
76                         <div class="col-md-12">
77                             <div class="form-group">
78                                 <label for="c">C</label>
79                                 <input id="c" type="text" name="c" class="form-control" placeholder="C" >
80                             </div>
81                         </div>
82                         <div class="row">
83                             <div class="col-md-12">
84                                 <div class="form-group">
85                                     <label for="d">D</label>
86                                     <input id="d" type="text" name="d" class="form-control" placeholder="D" >
87                                 </div>
88                             </div>
89                             <div class="row">
90                                 <div class="col-md-12">
91                                     <div class="form-group">
92                                         <label for="tacan">Tacan odgovor *</label>
93                                         <select name="tacan" class="form-control">
94                                             <option value="a">A</option>
95                                             <option value="b">B</option>
96                                             <option value="c">C</option>
97                                             <option value="d">D</option>
98                                         </select>
99                                     </div>
100                                 </div>
101                             </div>
102                             <div class="col-md-12">
103                                 <input name="unosPitanja" type="submit" class="btn btn-success" value="Unesi pitanje" /> <!-- dugme za slanje podataka gde doduce -->
104                             </div>
105                         </div>
106                     </div>
107                 </div>
108             </div>
109         </div>
110     </form>
111 </div>
```

## Kviz projekat – II kolokvijum Veb programiranje

Kod koji obradjuje uneseno pitanje ukljucuje konekciju na bazu I proverava da li je setovana vrednost za submit u **POST** nizu. Ako jeste ukljucuje fajl **pitanje.php** koji u sebi sadrzi **klasu Pitanje** sa funkcijama za unos pitanja I izmenu pitanja koju cemo posle analizirati. Funkcija za unos vraca true ili false u zavisnosti od toga da li je uspesno uneto pitanje setuje vrednost poruke koja se prikazuje pri vrhu gore prikazane forme.

```
1 <?php
2 // pokrenuta konekcija na bazu
3 include 'konekcija.php';
4
5 $poruka = '';
6
7 // ako su preko forme uspeno preneseni podaci POST metodom
8 if(isset($_POST['unosPitanja'])) {
9     include('pitanje.php'); // uvezi klasu pitanje sa metodima
10    $pitanje = new Pitanje($db);
11    $sacuvano = $pitanje->unesiPitanje(); // unosi se pitanje i funkcija vraca bool
12    if($sacuvano){
13        $poruka= 'Uspesno ste uneli pitanje';
14    }else{
15        $poruka= 'Neuspesno uneto pitanje';
16    }
17 }
18 ?>
19
20
21 <!DOCTYPE html>
```

U ovom bloku ima pretragu po korisnicima u select tagu koja poziva js funkciju `ajaxPretraga()` za ostvarene rezultate tog korisnika I poziva na svaku promenu is pocetka. Prvo se u php kodu **inicijalizuje zahtev preko Flight ruta** I setuje se da vrati transfer podataka kada se izvrši zahtev I vraca ga u json format kao sve korisnike iz tabele koji se dekoduju u niz. Zatim se **foreach** petljom redom u svakom option tagu prikazuju korisnici iz niza (ID korisnika I ime I Prezime). Zatim se poziva funkcija na slici desno koja izvlaci iz **value** od **option** taga **ID korisnika** zatim se pokrece ajax koji salje kao podatak ID korisnika u **pretragaTabele.php** koja vraca rezultate za odredjenog korisnika. Zatim se nakon uspesno izvršene skripte vraca rezultat I konkatencijom se kreira sktruktura html tabele sa nazivima kolona u zaglavlju I kroz each petlju se izvlace podaci iz JSON u objekat I prikazuju u poljima vrseci konkatenciju na output-a I na kraju se taj output smesti u div blok **#tabela** ispod bloka sa korisnicima u html.

```
<div class="col-md-12 text-center">
<h1>Pretraga tabele po korisnicima</h1>
<div class="row">
<div class="col-md-12">
<div class="form-group">
<label for="korisnik">Korisnik</label>
<select id="korisnik" name="korisnik" class="form-control" onchange="ajaxPretraga()">
<?php

// inicijalizuje se zahtev preko veb servisa za korisnicima
$zahtev = curl_init("http://localhost/kviz/api/korisnici");

// zahtev vraca neke podatke kojim je to setovano
curl_setopt($zahtev, CURLOPT_RETURNTRANSFER, 1);
$dgovor = curl_exec($zahtev);
$korisnici=json_decode($dgovor, true); // dekodujemo json u niz
curl_close($zahtev);

foreach ($korisnici as $k) { // izlistamo svakog korisnika njegovo ime i prezime
}
<option value="<?php echo $k['korisnikID']; >"><?php echo $k['imePrezime']; ></option>
<?php
}
}

</select>
```

```
// funkcija koja koristi ajax za pretragu korisnika sa rezultatom njegovim
function ajaxPretraga(){

// izvućemo koji je izabran ID korisnika iz html preko value
var korisnikID = $("#korisnik").val();
$.ajax({
url: "pretragaTabele.php",
data: "id="+korisnikID, // salje se podatak ID korisnika

success: function(result){ // kada je uspesno obradjena php skripta
var textOutput = 'table table-hover';
textOutput+=''thead';
textOutput+=''thTakmicar</th>';
textOutput+=''thBroj poena</th>';
textOutput+=''thPitanja</th>';
textOutput+=''tr';
textOutput+=''thead';
textOutput+=''tbody';

// u telu tabele se kroz each petlju za svaki rezultat izvlace iz
$.each($.parseJSON(result), function(i, val) {
textOutput += 'tr';
textOutput += 'td'+val.imePrezime+'td';
textOutput += 'td'+val.brojPoena+'td';
textOutput += 'td'+val.pitanja+'td';
textOutput += 'tr';

});

textOutput+=''tbody></table>';
$("#tabela").html(textOutput); // prosledimo html-u citav output k

});
}

</script>
<script>
// preko jquery se pokrece funkcija za pretragu
$( document ).ready(function() {
ajaxPretraga();
});
```

## Kviz projekat – II kolokvijum Veb programiranje

- 17) **pretragaTabele.php** uključuje konekciju na bazu i uzima id prosledjen preko ajax zatim ako je id razlicit od nule **kreira where uslov** za mecovanje sa dobijenim ID i na kraju izvrši upit **prirodnim spajanjem** tabela 'pitanje' i 'korisnik' sa zadatim where uslovom koji ce vratit kao rezultat sve podatke o rezultatima korisnika u kvizovima koje je odigrao i te podatke **na kraju kodovati u json**.

```
1 <?php
2 include 'konekcija.php';
3
4 $id = $_GET['id']; // dobijemo id preko ajax
5
6 $whereUslov = '';
7
8 if($id != 0){
9     $whereUslov = ' where k.korisnikID = '.$id; // definisi uslov
10 }
11
12 // izvrši join upit za prikaz imena i prezimena korisnika i njegovog rezultata
13 $tabela = $db->rawQuery('select * from tabela t join korisnik k on t.korisnikID = k.korisnikID '.$whereUslov);
14 echo json_encode($tabela);
15
16 ?>
```

Ovde smo napravili tabelu u html u kojoj ce se prikazati sva pitanja sa opcijama sa strane da se **izmene ili obrise**. Inicijalizujemo zahtev preko ruta za sva pitanja iz tabele i setujemo opciju za transfer da vrati te podatke kad se izvrši zahtev. Odgovor dobijen dekodiramo iz json u niz i onda foreach petljom prikazujemo sva polja za svako pitanje iz tabele u jednom redu i **dodatna polja** kao ikone za izmenu i brisanje koje pokrecu php skripte i salju preko GET metoda ID pitanja.

```
<div class="col-md-12">
<table class="table table-hover">
<thead>
<tr>
<th>Pitanje ID</th>
<th>Pitanje</th>
<th>Odgovor A</th>
<th>Odgovor B</th>
<th>Odgovor C</th>
<th>Odgovor D</th>
<th>Izmeni</th>
<th>Obrisi</th>
</tr>
</thead>
<tbody>
<?php
$zahtev = curl_init("http://localhost/kviz/api/pitanja");
curl_setopt($zahtev, CURLOPT_RETURNTRANSFER, 1);
$odgovor = curl_exec($zahtev);
$pitanja=json_decode($odgovor, true);
curl_close($zahtev);
foreach($pitanja as $pit): // petljom se za svako pitanje ispisuje podatak o pitanju u svaka polje tabele
<tr>
<td><?php echo ($pit['pitanjeID']);></td>
<td><?php echo ($pit['pitanje']);></td>
<td><?php echo ($pit['A']);></td>
<td><?php echo ($pit['B']);></td>
<td><?php echo ($pit['C']);></td>
<td><?php echo ($pit['D']);></td>
<td><?php echo ($pit['tacan']);></td>
<td><a href="izmeni.php?id=<?php echo ($pit['pitanjeID']);>"><i class="fa fa-refresh fa-1x"></i></a></td>
<td><a href="obrisi.php?id=<?php echo ($pit['pitanjeID']);>"><i class="fa fa-times fa-1x"></i></a></td>
</tr>
<?php endforeach;
</tbody>
</table>
```

- 18) **obrisi.php** uvozi konekciju i proverava da li je **setovan ID pitanja preko GET metoda**, pa ako nije vraća ga na admin. Inace cuva taj id i prosledjuje ga **where** funkciji objektu baze da nadje to pitanje koje ce smestiti u rezultat i onda samo izvrši delete upit sa id tog pitanja iz tabele i ucita opet admin stranicu.

```
1 <?php
2 include 'konekcija.php';
3
4
5 if(!isset($_GET['id'])){
6     header("Location: admin.php");
7 }
8
9 // sacuvamo id
10 $id=$_GET['id'];
11 $db->where('pitanjeID',$id);
12 $pitanjeJedno = $db->delete('pitanje'); //
13 header("Location: admin.php");
14 ?>
```



## Kviz projekat – II kolokvijum Veb programiranje

- 19) **izmeni.php** skripta ima deo forme koji ce prvo ucitati vrednosti u poljima koja su trenutno u bazi. Zatim admin unosi nove vrednosti za sva polja I na kraju klikne na submit gde se pokrece php kod na levoj slici. Tu je ukljucena konekcija **pa proverava da li je setovan ID pitanja preko GET metoda**, ako nije vrati ga na admin stranicu. **Inace sacuvaj ID pitanja I preko where** klauzule pronadji to pitanje I pribavi ga preko **getOne** funkcije. Ako je setovano polje za izmenu pitanja tj submitovana forma onda ukljuci **klasu Pitanje iz pitanje.php** gde ce se kreirati objekat pitanje I nad kojim ce se pokrenuti funkcija za izmenu pitanja koja vraća true ili false pa u zavisnosti od toga setuj adekvatnu poruku koja ce se prikazati na pocetku forme.

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

// ako nije setovan id pitanja vrati ga nazad na admin stranicu
if(!isset($_GET['id'])){
    header("Location: admin.php");
}

// sacuvamo id
$id=$_GET['id'];
$db->where('pitanjeID',$id);

$pitanjeJedno = $db->getOne('pitanje'); // izvucemo to nadjenje
$poruka = '';

if(isset($_POST['izmenaPitanja'])) {
    include('pitanje.php'); // uvezi pitanje klasu sa metodima

    $pitanje = new Pitanje($db);

    // ako je vrateno true za izmenu pitanja
    if($pitanje->izmenaPitanja($id)){
        $poruka= 'Uspesno ste izmenili pitanje';
    }else{
        $poruka= 'Neuspesno izmenjeno pitanje';
    }
}

?>
```

```

60 <form id="forma" method="post" action="" role="form">
61 <?php if($poruka != ''){ ?>
62 <div class="well"><?php echo $poruka ?></div>
63 <?php ?>
64 <div class="row">
65 <div class="col-md-12">
66 <div class="form-group">
67 <label for="pitanje">Pitanje </label>
68 <input id="pitanje" type="text" name="pitanje" class="form-control" value="<?php echo $pitanjeJedno['pitanje'] ?>" >
69 </div>
70 </div>
71 </div>
72 <div class="row">
73 <div class="col-md-12">
74 <div class="form-group">
75 <label for="a">A</label>
76 <input id="a" type="text" name="a" class="form-control" value="<?php echo $pitanjeJedno['A'] ?>" >
77 </div>
78 </div>
79 </div>
80 <div class="row">
81 <div class="col-md-12">
82 <div class="form-group">
83 <label for="b">B</label>
84 <input id="b" type="text" name="b" class="form-control" value="<?php echo $pitanjeJedno['B'] ?>" >
85 </div>
86 </div>
87 </div>
88 <div class="row">
89 <div class="col-md-12">
90 <div class="form-group">
91 <label for="c">C</label>
92 <input id="c" type="text" name="c" class="form-control" value="<?php echo $pitanjeJedno['C'] ?>" >
93 </div>
94 </div>
95 </div>
96 <div class="row">
97 <div class="col-md-12">
98 <div class="form-group">
99 <label for="d">D</label>
100 <input id="d" type="text" name="d" class="form-control" value="<?php echo $pitanjeJedno['D'] ?>" >
101 </div>
102 </div>
103 </div>
104 <div class="row">
105 <div class="col-md-12">
106 <div class="form-group">
107 <label for="tacan">Tacan odgovor </label>
108 <select name="tacan" class="form-control">
109 <!-- dodata je opcija da prikaze prvo tacan odgovor od izvucenog pitanja iz php-->
110 <option value="<?php echo $pitanjeJedno['tacan'] ?>"><?php echo strtoupper($pitanjeJedno['tacan']) ?></option>
111 <option value="a">A</option>
112 <option value="b">B</option>
113 <option value="c">C</option>
114 <option value="d">D</option>
115 </select>
116 </div>
117 </div>
118 </div>
119 <div class="col-md-12">
120 <input id="login" name="izmenaPitanja" type="submit" class="btn btn-success" value="Izmeni pitanje"> <!-- na klik se salju podaci ph
121
```

- 20) **pitanje.php** je skripta u kojoj je definisana klasa za pitanje sa funkcijama za unos novog pitanja I izmenu pitanja. Imamo definisano polje koje hvata konekciju na bazu I koje se setuje preko konstruktora. Zatim imamo funkciju **unesiPitanje** koje preko POST metoda pribavlja sva polja prosledjena preko inputa I cisti string od praznina I stetnih karaktera po bazu pa cuva u promenljive koje pakuje u asocijativni niz nakon cega **izvrsava insert upit nad tabelom 'pitanje' I bind-uje odredjene podatke** u upit koji vraća true ili false nakon cega I funkcija vraća true ili false sto ce sluziti posle za status poruke. Zatim ispod nje imamo funkciju za izmenu pitanja koja na isti nacin filtrira podatke koji su uneseni preko inputa I smesta ih u obican niz koji se **bind-uje rawQuery funkciji u update upitu** koji vraća Boolean u cijoj zavisnosti I cela funkcija za izmenu pitanja vraća true ili false.

## Kviz projekat – II kolokvijum Veb programiranje

```
1 <?php
2
3 class Pitanje {
4
5     private $db;
6
7     public function __construct($db) { // konstruktor
8         $this->db = $db;
9     }
10
11 // funkcija kojom se unosi pitanje
12 public function unesiPitanje(){
13
14     $pitanje = $this->db->escape(trim($_POST['pitanje']));
15     $a = $this->db->escape(trim($_POST['a']));
16     $b = $this->db->escape(trim($_POST['b']));
17     $c = $this->db->escape(trim($_POST['c']));
18     $d = $this->db->escape(trim($_POST['d']));
19     $tacan = $this->db->escape(trim($_POST['tacan']));
20
21     $data = Array (
22         "pitanje" => $pitanje,
23         "A" => $a,
24         "B" => $b,
25         "C" => $c,
26         "D" => $d,
27         "tacan" => $tacan
28     );
29
30 //tako upakovani se unose u bazu preko funkcije insert iz d
31 $sacuvano = $this->db->insert('pitanje', $data);
32
33 if($sacuvano) {
34     return true;
35 }
36 else {
37     return false;
38 }
39 }
40
41
42 // funkcija za izmenu pitanja
43 public function izmenaPitanja($id){
44
45     $pitanje = $this->db->escape(trim($_POST['pitanje']));
46     $a = $this->db->escape(trim($_POST['a']));
47     $b = $this->db->escape(trim($_POST['b']));
48     $c = $this->db->escape(trim($_POST['c']));
49     $d = $this->db->escape(trim($_POST['d']));
50     $tacan = $this->db->escape(trim($_POST['tacan']));
51
52     $data = Array ($pitanje, $a, $b, $c, $d, $tacan, $id);
53
54 // binduje u upit iz database.php i vraća true ako je uspesno izmenjeno
55 $sacuvano = $this->db->rawQuery('update pitanje set pitanje=?,A=?,B=?,C=?,D=?,tacan=? where pitanjeID=?', $data);
56
57 if($sacuvano) {
58     return true;
59 }
60 else {
61     return false;
62 }
63
64 }
65 }
```

- 21) **logout.php** je skriptu koja uključuje konekciju gde je pokrenuta sesija pa je zatim uništava. Zatim nadodaje na kraj fajla sesija.txt da je korisnik izlogovan, pa zatvara fajl i učitava index tj početnu stranicu sajta.

```
1 <?php
2 include('konekcija.php');
3 session_destroy();
4
5 $file=fopen("sesija.txt","a");
6 fwrite($file, "KORISNIK JE IZLOGOVAN !");
7 fclose($file);
8
9 header('Location:index.php'); // vrati se na index stranicu
10
11 >>
```