Dinamička web aplikacija za organizaciju rada fitnes centra

Negovanović, Strahinja

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:924434

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-16**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Stručni studij

Dinamička web aplikacija za organizaciju rada fitnes centra

Završni rad

Strahinja Negovanović

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Osijek, 29.06.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu završnog rada na preddiplomskom stručnom studiju

Ime i prezime studenta:	Strahinja Negovanović
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI4300, 29.08.2013.
OIB studenta:	99706763617
Mentor:	Doc.dr.sc. Alfonzo Baumgartner
Sumentor:	Marina Peko
Sumentor iz tvrtke:	Tomislav Jakopec
Predsjednik Povjerenstva:	Doc.dr.sc. Tomislav Keser
Član Povjerenstva:	Dr. sc. Tomislav Galba
Naslov završnog rada:	Dinamička web aplikacija za organizaciju rada fitnes centra
Znanstvena grana rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak završnog rada	Izraditi dinamičku web aplikaciju koja će služiti kao pomoć za rad fitnes centra. Kreirati najmanje dva korisnička profila koja se razlikuju po ovlastima. Administratorski profil treba imati mogućnost dodavanja, izmjene i brisanja korisničkih profila. Korisničkom profilu omogućiti bilježenje i pregled trenutno aktivnih korisnika te evidenciju plaćanja članarina te grafički pregled rasporeda programa vježbanja. Tehnologije za izradu zadatka: HTML, CSS, JavaScript, PHP, MySQL.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	29.06.2017.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 21.07.2017.

Ime i prezime studenta:	Strahinja Negovanović
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	AI4300, 29.08.2013.
Ephorus podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Dinamička web aplikacija za organizaciju rada fitnes centra**

izrađen pod vodstvom mentora Doc.dr.sc. Alfonzo Baumgartner

i sumentora Marina Peko

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA

Ja, Strahinja Negovanović, OIB: 99706763617, student/ica na studiju: Preddiplomski stručni studij Elektrotehnika, smjer Informatika, dajem suglasnost Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek da pohrani i javno objavi moj **završni rad**:

Dinamička web aplikacija za organizaciju rada fitnes centra

u javno dostupnom fakultetskom, sveučilišnom i nacionalnom repozitoriju.

Osijek, 01.08.2017.

_____
potpis

Sadržaj:

1. Uvod 1
1.1. Zadatak završnog rada 1
2. Općenito o *web* aplikaciji 2
2.1. *HTML* i *CSS* 6
2.2. *PHP* 9
2.3. Ključne datoteke 9
3. Članarina i raspored 11
4. Kontakt forma 12
4.1. JavaScript 13
4.2. Era 13
5. Prijava i registracija korisnika 15
5.1. Postavljanje *web* aplikacije lokalno i općenito o bazama 15
5.2. Postavljanje *hosting-a* i stvaranje nove baze 20
5.3. Spajanje na bazu 20
5.4. Prijava i registracija – forme i datoteke 21
6. Zaključak 32
7. Literatura i popis upotrijebljenih oznaka 33
8. Sažetak i ključne riječi 34
9. Naslov, sažetak i ključne riječi na engleskom jeziku 35
10. Životopis 36
11. Prilozi 37

1. Uvod

Cilj ove *web* aplikacije je omogućiti voditeljima fitnes centra pregled i uređivanje aktivnih članova i članarina, pregled i pisanje rasporeda, te pregled i pisanje vijesti. Članovi fitnes centra, kad se prijave, mogu pregledati sve aktivne članove i uređivati slike na početnoj stranici *web* aplikacije. Voditelji također mogu uređivati slike na početnoj stranici.

Za završni rad sam odlučio izraditi dinamičku *web* aplikaciju koja će služiti kao pomoć za rad fitnes centra zato što volim odlaziti i vježbati u teretani, član sam XXL fitnes centra i jer sam oduvijek htio znati kako se prave *web* aplikacije za fitnes centre i kako bih ja to sam napravio.

Prije nego što sam krenuo pisati *web* aplikaciju, najveći izazov mi je bio osmisliti njen izgled i odabir *front-end framework-a* koji bi mi pomogao u tome. Za dizajn i izgled stranice sam odlučio koristiti *Bootstrap 3 framework*. Prijava i registracija korisnika je napravljena pomoću *PHP* i *MySQL* tehnologija, *phpMyAdmin* sučelja i *XAMPP-a*. Alati (programi) korišteni za izradu ove *web* aplikacije su *Aptana Studio 3* i *Notepad++*, a operacijski sustav *Windows 10*.

1.1. Zadatak završnog rada

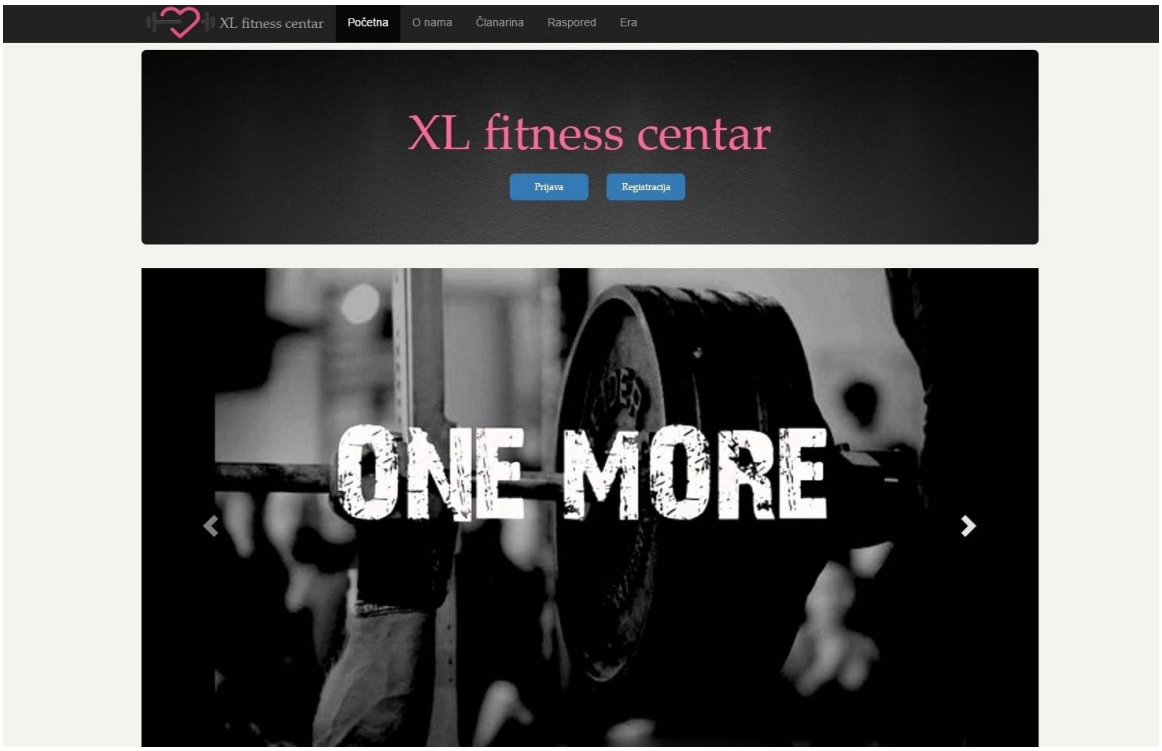
Zadatak završnog rada je: „*Izraditi dinamičku web aplikaciju koja će služiti kao pomoć za rad fitnes centra. Kreirati najmanje dva korisnička profila koja se razlikuju po ovlastima. Administratorski profil treba imati mogućnost dodavanja, izmjene i brisanja korisničkih profila. Korisničkom profilu omogućiti bilježenje i pregled trenutno aktivnih korisnika te evidenciju plaćanja članarina te grafički pregled rasporeda programa vježbanja. Tehnologije za izradu zadatka: HTML, CSS, JavaScript, PHP, MySQL.*“

2. Općenito o *web* aplikaciji

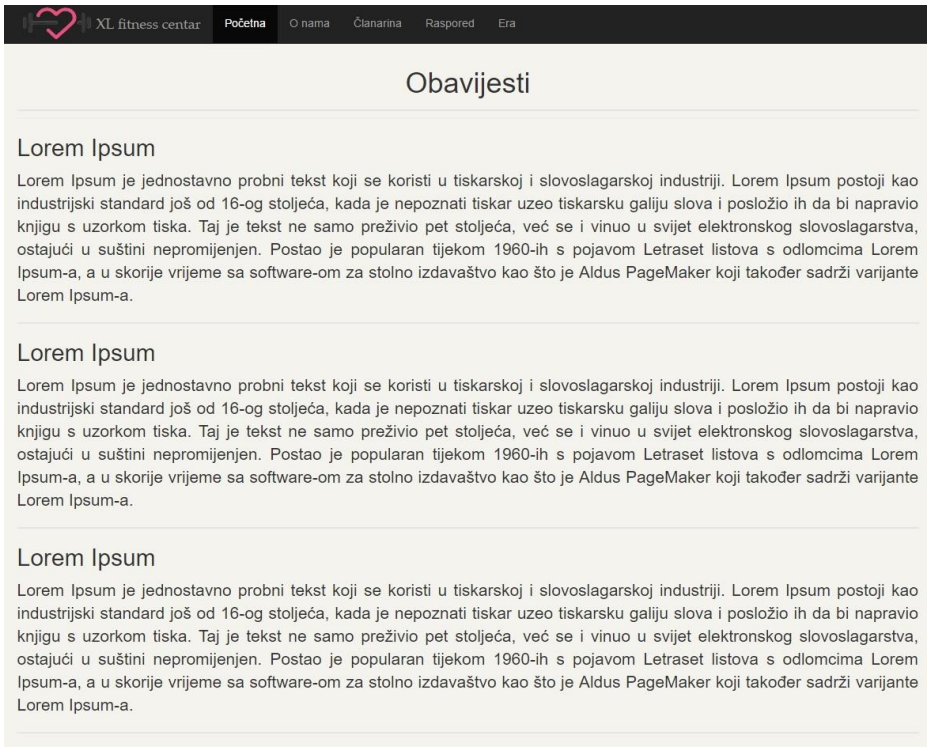
Čitav izgled i dizajn *web* aplikacije je napravljen pomoću *Bootstrap 3 framework-a.* Odlučio sam koristiti *Bootstrap 3 framework* zbog toga da *web* aplikacija ima lijep izgled, dizajn i da bude dinamička tj. eng. *responsive*. *Bootstrap* predstavlja *open-source JavaScript framework* tj. kombinaciju *HTML, CSS i JavaScript* tehnologija, koji služi za lakši razvoj *web* stranica (naročito izgleda) i naprednih *web* komponenti zbog čega je jedan od najpopularnijih *front-end framework-a.* [1] Drugim riječima, *Bootstrap* u sebi ima ugrađenu *JQuery* knjižnicu (eng. *library)* pisanih *CSS* i *JavaScript* alata, skripti i funkcija koje omogućavaju stvaranje brze, dinamične i lijepe *web* aplikacije koja se prikazuje jednako na svim pretraživačima, neovisno o rezoluciji monitora i na mobitelima. [1] *Bootstrap* je besplatan i kada se preuzme, dobije se osnovni izgled *web* stranice koju programer nadograđuje prema svojim potrebama.

Svoju *web* aplikaciju sam nazvao „XL *fitness* centar“. Ona se sastoji od 5 izbornika, a svaki dio ima zaglavlje i podnožje:

1. Početna stranica (Sl.2.1.) - sadrži naziv fitnes centra sa *logom*, dio sa prijavom i registracijom korisnika (gumbi za prijavu i registraciju korisnika), *JavaScript slider* sa slikama i dio sa vijestima (gdje imamo ubačeni probni tekst sa interneta, slika 2.2., koji predstavlja nekakve vijesti)

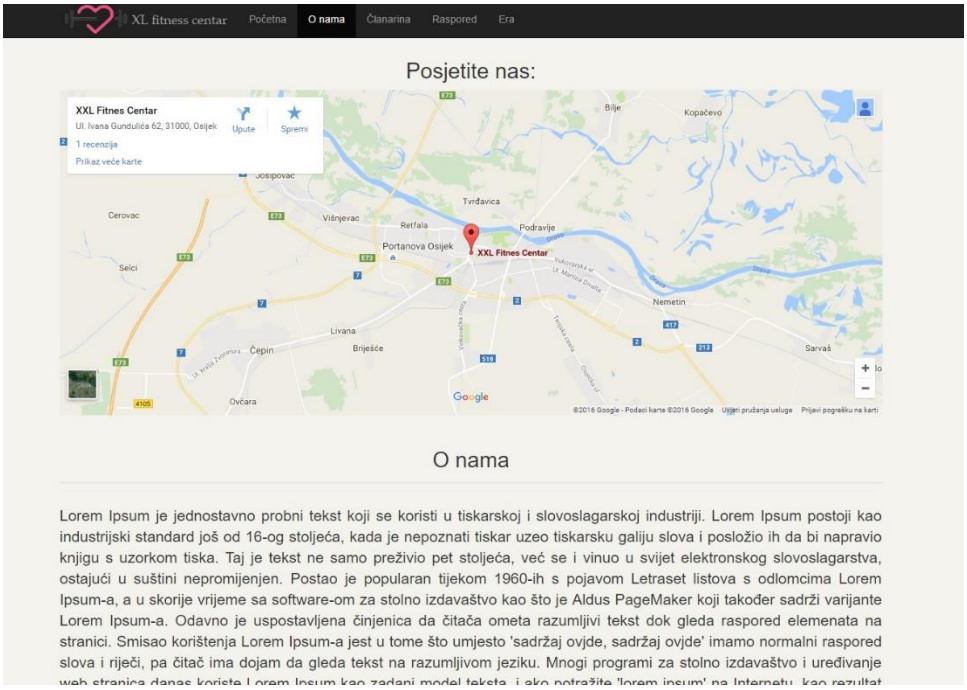


Slika 2.1. Početna

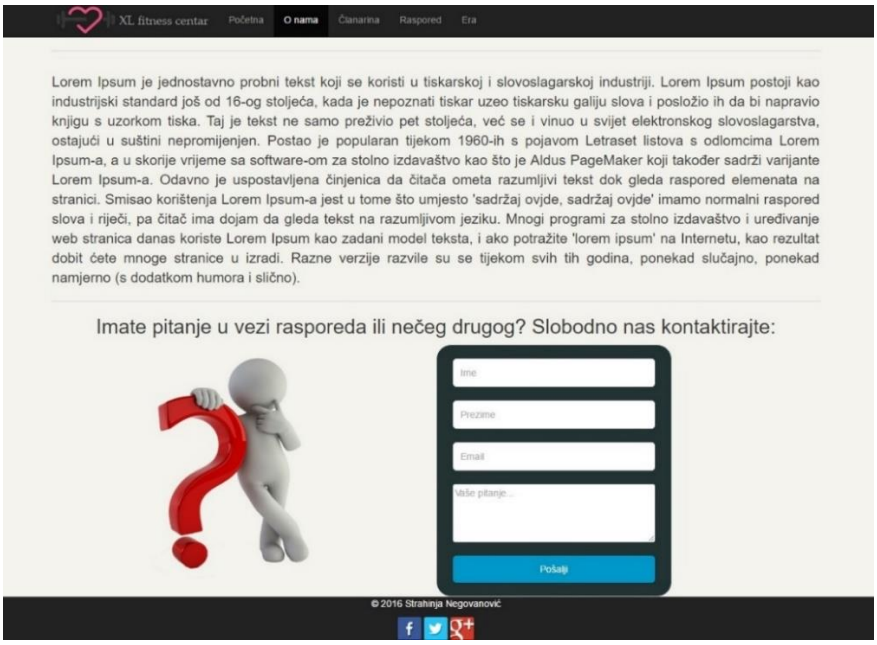


Slika 2.2. Nastavak početne

2. O nama (Sl.2.3.) – prvi dio sadrži prikaz lokacije XXL fitnes centra u Osijeku koji bi imaginarno predstavljao lokaciju mog „XL *fitness* centra“. Isto tako, tu se nalazi dio „o nama“ gdje je također ubačen probni tekst, a koji predstavlja informacije o „XL fitnes centru.“



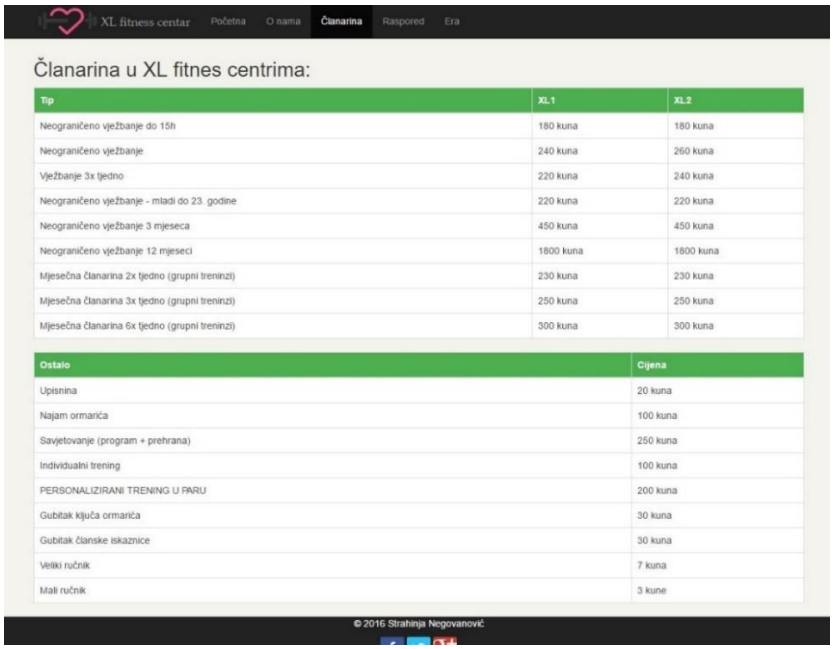
Slika 2.3. O nama prvi dio



Slika 2.4. O nama drugi dio

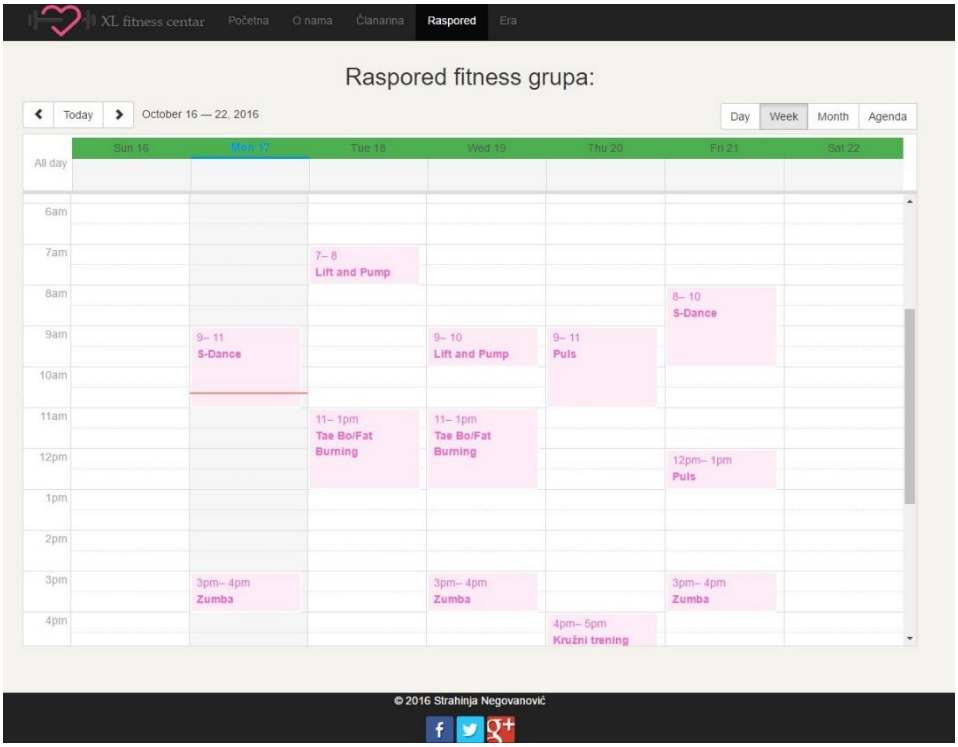
Drugi dio (Sl.2.4.) sadrži dio informacija o „XL fitnes centru“ i formu za upit gdje se može poslati email voditeljima u vezi savjeta, pitanja, prijedloga itd.

3. Članarina (Sl.2.5.) – sadrži tablicu članarina tj. izmišljeni cjenik za „XL fitnes centar“ napisan u *HTML-u* i *CSS*-u sa informacijama o tipu, nazivu i cijeni članarine za „XL fitnes centar.“



Slika 2.5. Članarina

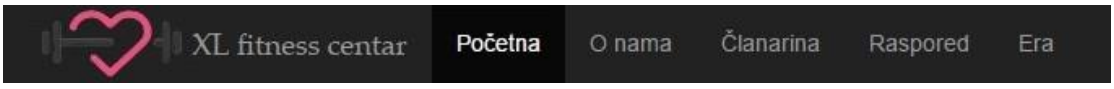
4. Raspored (Sl.2.6.) – sadrži raspored izmišljenih fitnes grupa, napisan pomoću *JavaScript*-e i *JQuery*-a, a preuzet s stranice *ALOYUI.*



Slika 2.6. Raspored

5. Era – sadrži era dijagram baze fitnes, odnosno baze *web* aplikacije „XL *fitness* centar.“

Zaglavlje (Sl.2.7.) sadrži logo i naziv stranice, kao i poveznicu na svaki od navedenih 5 dijelova. Podnožje (Sl.2.8.) se nalazi na dnu svakog izbornika, odnosno na dnu *web* aplikacije, a sadrži informacije o programeru i poveznice na njegove društvene račune (eng. *social account*).



Slika 2.7. Zaglavlje

Slika 2.8. Podnožje

Kod za podnožje (Sl.2.9.) sadrži oznaku *footer* koja je oznaka za podnožje u *Boostrap-u*, gdje imamo paragraf sa mojim imenom i prezimenom, kao i oznaku „a“ koja je oznaka za poveznicu u *HTML*-u, gdje sa *href* oznakom definiramo poveznicu tj. mjesto ili *web* adresu gdje će korisnika, klikom miša, prebaciti.

```
<footer class="footer">
        <p>@ 2016 Strahinja Negovanović</p>
        <a href="https://www.facebook.com/strahinja.negovanovic"><img title="Facebook" src="ikonice/facebook.png" width="35px" height="35px"/></a>
        <a href="https://twitter.com/Kistras"><img title="Twitter" src="ikonice/twitter.png" width="35px" height="35px"/></a>
        <a href="https://plus.google.com/u/0/103167825392726927406/posts"><img title="Google" src="ikonice/google.png" width="35px" height="35px"/></a>
</footer>
```

Slika 2.9. Kod za podnožje

2.1. *HTML* i *CSS*

HTML je oznaka za *HyperText Markup Language*. Svaka *web* stranica je napisana u *HTML-u*. [2] *HTML* predstavlja „kostur“ za svaku stranicu (osnovno što čini neku *web* stranicu) i ima osnovnu strukturu: [2]

```
<!DOCTYPE html>
<html>
  <head>
      <title> </title>
  </head>
  <body>
  </body>
</html>
```

Prvi dio „<!DOCTYPE html>, <html>“ govori nam da se radi o *HTML-u*. Svaki *HTML* dokument se sastoji od glave (*head)* i tijela (*body*), a svaki njegov element se otvara i zatvara „*tag-ovima“* (npr. element p:). Postoje i elementi koji se ne zatvaraju nego su „samozatvarajući“ (npr.
). U *head* dijelu se stavljaju metapodaci, tj. podaci o *HTML* dokumentu i obično ti podaci nisu prikazani (podaci o autoru, jeziku stranice i dr.), osim naslova naše stranice koji je prikazan na vrhu pretraživača, a definiran u *tag-u* „<title> </title>“. *Body* dio definira cijeli *HTML* dokument i sadrži sav sadržaj *HTML-a*: tekst, slike, poveznice, tablice i dr. *HTML* datoteka ima ekstenziju .html ili .php kada se *PHP* ugradi u *HTML* datoteku.

Web pretraživači nam ne prikazuju *HTML kod* i njegove *tag-ove*, već nam daju vizualni prikaz neke stranice, na osnovu onoga što je u njemu napisano i stavljeno. Međutim, na svakom *web* pretraživaču je veoma jednostavno moguće pogledati *HTML kod* neke stranice. Pored *HTML-a* koji predstavlja „kostur“ neke stranice, za izgled i opis neke stranice je zaslužan CSS bez kojeg *HTML* jezik nema smisla.

CSS predstavlja oznaku za *Cascading Style Sheets* i on je stilski jezik. [2] *CSS* datoteka ima ekstenziju „.css“. *CSS* se može ubaciti u *HTML* dokument na 3 načina:

1. Unutarnjim povezivanjem - umetanje CSS svojstva unutar *head* oznake pomoću oznake *style*: [2]

```
<html>
  <head>
          <style>
               <!-- CSS svojstvo -->
          </style>
  </head>
  <body>
  </body>
</html>
```

2. Vanjskim povezivanjem - *CSS* svojstva stavljamo u vanjsku datoteku npr. stil.css i povezujemo sa *HTML* datotekom [2]

```
<html>
  <head>
     <link rel = "stylesheet" href="stil.css" />
  </head>
  <body>
  </body>
</html>
```

3. Umetanjem *CSS* svojstva unutar samog *HTML* elementa koristeći *style* oznaku. [2]
   <h1 style= "color: red"> Naslov </h1>

Sintaksa *CSS-a* je jednostavna i koristi engleske riječi kako bi se pojedini elementi definirali. Primjer: [2]

selektor {

    svojstvo: vrijednost;

}

Selektor predstavlja *HTML* element kojem želimo dodijeliti neko svojstvo, a to svojstvo ima svoju vrijednost. Svaki selektor može imati više svojstava. Svojstvo i vrijednost su odvojeni dvotočkom i nalaze se unutar vitičastih zagrada. Iza vrijednosti dolazi točka-zarez koja odvaja svojstva. Ako neko svojstvo ima više vrijednosti, one su odvojene zarezom, a ako vrijednost svojstva ima više od jedne riječi, onda je stavljamo unutar dvostrukih navodnika. Primjer, kada

bismo htjeli da su nam svi naslovi u *HTML-u* npr. crvene boje i da nam je font „*Times New Roman“*, potrebno je u *CSS-u* napisati oznaku naslova (h1, h2, h3 i dr.), otvoriti i zatvoriti vitičaste zagrade i unutar napisati željenu boju i font: [2]

```
h1 {
        color: red;
        font-family: "Times New Roman", Times, serif;
}
```

Ovim *kodom,* svi naslovi oznake h1 tj. *tag-ovi* h1 će biti crvene boje i imat će font „*Times New Roman“*. Kada se ne koriste *HTML tag-ovi,* onda se koriste selektor identifikatora „id“ i klase. Selektor identifikatora se odnosi na *HTML* element kojemu je vrijednost *id* svojstva jednaka nazivu selektora. Primjer:

HTML:

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

CSS:

```
#izbornik {
        Color: red;
        Text-align: center;
}
```

Ovo *CSS* svojstvo se primjenjuje na *HTML* element koji ima postavljen *id*="izbornik", a pristupa se u *CSS-u* pomoću znaka „#“. Klase koristimo kada trebamo definirati svojstva više elemenata unutar *HTML-a,* a u *CSS-u* se točkom označava pristup toj klasi. Primjer:

HTML:

```
<h1 class="centar"> Naslov </h1>
<p class="centar">Tekst</p>
```

CSS:

```
.podvučeno {
        Text-align: center;
}
```

2.2. *PHP*

PHP je skriptni programski jezik i on se koristi u kombinaciji s *HTML kodom*. [3] Pomoću *PHP-a* možete kreirati HTML stranicu na serveru prije nego što je ona poslana klijentu. [3] Ovim načinom generiranja sadržaja klijent ne može vidjeti *kod* (koji je izgenerirao sadržaj koji klijent gleda) već ima pristup čistom HTML *kodu*. [3] Primarna uloga *PHP-a* je kreiranje dinamičkih *web* stranica. [3] *PHP kod* se sastoji od niza naredbi koje se izvršavaju jedna za drugom. Posljednja naredba označava kraj *PHP koda*. U *HTML-u* mi možemo uključiti više *PHP* skripti. Njegove skripte ne zahtijevaju prevođenje (*kompajliranje*) nego se one izvode pomoću *PHP interpretera* koji se nalazi na serveru i koji se poziva za izvršenje *PHP* skripti svaki put kada korisnička strana to od njega zahtijeva. [3] Kada se *PHP* skripta izvrši, server šalje rezultate klijentskoj strani podatke u obliku forme na određenoj *web* stranici. [3]

2.3. Ključne datoteke

Datoteka head.php (P.2.1.) sadrži metapodatke tj. podatke o *HTML* dokumentu, sa linkovima na *css* datoteke *Bootstrap-a* i moje *css* datoteke. $naslovAPP i $putanjaAPP su *PHP* varijable definirane u datoteci konfiguracija.php (P.2.12.) koje sadrže naslov *web* aplikacije i njenu putanju. Datoteka izbornik.php (P.2.2.) sadrži definirane izbornike za svaki dio (početna, o nama itd.).

Datoteka pocetna.php (P.2.3.) sadrži *div* naziva *jumbotron* gdje imamo ubačen stilizirani naslov i gumbe za prijavu i registraciju članova. *PHP* dio koda se odnosi na *SQL* upit gdje se odabire sve iz tablice galerija (koja se nalazi u bazi), a u pomoću *img* oznake smo izvukli putanju slika iz tablice galerija. Taj *PHP* kod nam omogućava prikazivanje slika u slideru na početnoj stranici. Datoteka skripte.php (P.2.4.) uključuje *JavaScript* skripte potrebne za rad *Bootstrap-a.*

Datoteka stil.css (P.2.5.) sadrži sva *CSS* svojstva potrebna za *web* aplikaciju "XL *fitness* centar". Ta datoteka pored *Bootstrap*-ovih *CSS* datoteka, definira izgled stranice tj. izgled i poziciju svih elemenata na *web* aplikaciji.

Mape images i ikonice (koje se nalaze unutar Fitnes mape) sadrže sve slike i ikonice koje se nalaze na *web* aplikaciji "XL *fitness* centar".

3. Članarina i raspored

Članarina i raspored sadrže izmišljene grupe i cjenik s ciljem da se samo prikaže njihova struktura, a kasnije je po zahtjevu korisnika moguće izmijeniti informacije. Članarina je napravljena pomoću tablica u *HTML-u*, a njen izgled u *CSS-u*. Za pravljenje tablice, u *HTML-u* se koristi oznaka *table* koja sadrži oznake *thead* i *tbody*. [2] *Thead* oznaka se odnosi na zaglavlje (vrh) tablice, a *tbody* na sadržaj tablice. „<tr>" oznaka se odnosi na red tablice, a „<td>" na ćeliju u tablici.

Dio koda u datoteci clanarina.php (P.3.6.) znači da će tablica biti dinamična, odnosno, kada se smanji rezolucija na rezoluciju tableta i mobitela, tablica se smanjuje i pojavljuje se *slider* pomoću kojeg se sadržaj tablice može vidjeti pomjeranjem u lijevo ili u desno. Tablica ima klasu "*table table-bordered*" zato što *Bootstrap* u sebi sadrži tu klasu, gdje je napravljena tablica koja ima okvir i oko ćelija.

Raspored za moju *web* aplikaciju sam prvo napravio sam pomoću "čistog" *HTML-a* i *CSS-a*, i po meni, izgled nije bio loš, ali za uređivanje tog rasporeda je bilo potrebno pisati jako puno koda u *HTML-u* pa i u *CSS-u*, zbog čega sam se odlučio za jedan od *JQuery scheduler-a* koji imaju potrebnu svu funkcionalnost i izgled za raspored. Odabrao sam *ALLOYUI scheduler* koji se jednostavno dodaje (dodaju se dvije linije koda u *head tag-u HTML-a*) i uređuje.

Datoteka raspored.php (P.3.7.) sadrži u oznaci *head* poveznice na *ALLOYUI scheduler* kojim se uključuje *scheduler* (hrv. raspored) u ovu *PHP* datoteku i bez kojih *scheduler* ne bi radio niti bi bio prikazan. *Scheduler* je ubačen pomoću *script HTML* oznake gdje imamo njegov *kod*. *Var events* stvara varijablu „events" u koju dodajemo datum i naziv grupa, te se na taj način prikazuju grupe po datumima (koje smo mi definirali).

Drugi dio datoteke raspored.php (P.3.8.) se odnosi na definiranje "pogleda" gdje možemo raspored pregledavati po danima, mjesecu, tjednu itd., a pomoću *new Y.Scheduler* oznake definiramo datum koji će se prikazati kada se otvori izbornik raspored.

4. Kontakt forma

Ispod dijela “o nama” napravljena je kontakt forma (Sl.4.1. i Sl.4.2.) za dodatna pitanja, mišljenja, savjete itd. korisnika, gdje se nakon što se stisne pošalji, pošalje mail na: strahinja944@gmail.com. Međutim, korisnik neće moći poslati mail ako npr. nije dobro ispunio polje za email ili je ostavio negdje prazno polje zato što to datoteka „validacija.js” provjerava. Datoteka „validacija.js” (P.4.9.) sadrži u sebi *JavaScript* funkciju koja u sebi ima varijablu „x”. Varijabla prima unos korisnika za svako polje posebno, a sa if petljom u funkciji provjeravamo da li je taj unos prazan i da li smo dobili nulu. Ako smo dobili prazno polje, funkcija će ispisati grešku da moramo ispisati npr. ime ili prezime i izlazi iz petlje. Kada je korisnik unio sve ispravno, pošalje se mail.

Slika 4.1. Kontakt forma

```
<h2 align="center">Imate pitanje u vezi rasporeda ili nečeg drugog? Slobodno nas kontaktirajte:</h2>

<div class="row kontakt">
    <div class="col-md-6">
        <img src="images/pitanje.jpg"  height="370px" width="350px"/>
    </div>

    <script src="js/validacija.js"></script>

    <div class="col-md-6 forma">
    <form name="myForm" action="mailto:strahinja944@gmail.com" onsubmit="return validateForm()" method="post" enctype="text/plain">
        <input type="text" placeholder="Ime" name="ime"/>
        <input type="text" placeholder="Prezime" name="prezime"/>
        <input type="text" placeholder="Email" name="email"/>
        <textarea rows="4" cols="50" maxlength="500" name="tekst" placeholder="Vaše pitanje..."></textarea>
        <input type="submit">
    </form>
    </div>
</div>
</div>
```

Slika 4.2. HTML forma za kontakt formu, onama.php

4.1. *JavaScript*

JavaScript je skriptni jezik koji se koristi za kreiranje dinamičkih *web* stranica pisanih u *HTML-u* i *CSS-u,* za validaciju podataka, kreiranje kolačića (eng. *cookies*) itd. [3] Cilj kreiranja *JavaScript* jezika je bio dodavanje interaktivnosti *HTML* stranicama. [3] *Javascript* je interpreter što znači da izvršava naredbu po naredbu bez *kompajliranja* tj. prevođenja cijelog programa. *Javascript* nije isto što i *Java* jer je *Java* objektno-orjentirani programski jezik dok je *JavaScript* samo skriptni jezik. [3]

JavaScript mora biti u *HTML* dokumentu i mora započeti i završiti sa oznakom <*script*>. Primjer skripte koja ispisuje riječ pozdrav:

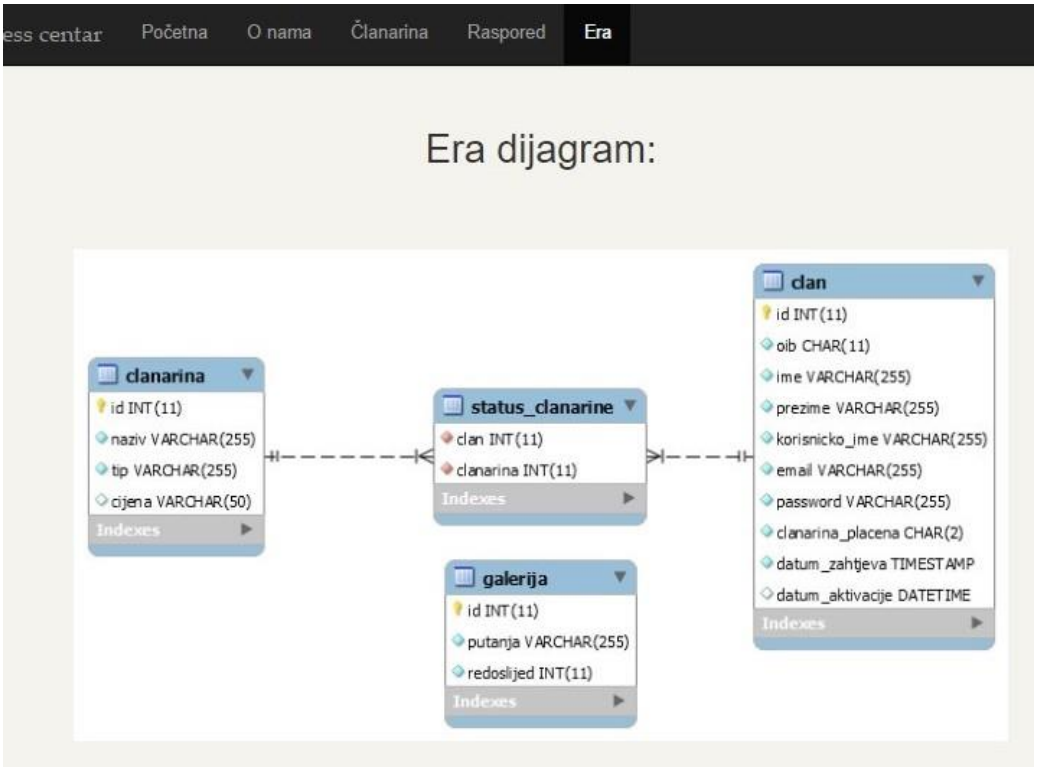
```
<html>
  <body>
    <script>
           document.write("Pozdrav. ");
    </script>
  </body>
</html>
```

JavaScript se može postaviti u *body* dio *HTML-a* (skripta se izvršava odmah kada se stranica učita), u *head* dio *HTML-a* (izvršit će se tek kad se pozove i obično se u ovom dijelu smještaju funkcije koje se kasnije pozivaju iz *body* dijela) i u vanjsku datoteku. *JavaScript* datoteke je korisno staviti u neku vanjsku datoteku kada više stranica treba koristiti jednu istu skriptu, gdje onda ta datoteka ima ekstenziju „js“, a poziva se u *HTML-u, npr.*:
<script src="js/validacija.js"></script>

4.2. Era

Izbornik “Era” (Sl.4.3.) sadrži *era* dijagram baze fitnes i era.php datoteka (Sl.4.4.). Era.php je datoteka koja se odnosi na izbornik “Era”. Prvi dio gdje imamo *kod* php include_once “konfiguracija.php” se odnosi na *php* datoteku “konfiguracija.php” gdje sa *include_once* oznakom tražimo od datoteke era.php da uključi u sebe datoteku konfiguracija.php. Isto tako se uključuju datoteke head.php, izbornik.php, podnozje.php i skripte.php. Pomoću *img* oznake, mi smo ubacili sliku era.png gdje nam je prikazana kada se klikne na izbornik era.

Izbornik Era:



Slika 4.3. Era.png

```php
<?php include_once 'konfiguracija.php'; ?>

<!DOCTYPE html>
<html lang="hr">
  <head>
    <?php include_once 'predlozak/head.php'; ?>
  </head>

  <body>
    <?php include_once 'predlozak/izbornik.php'; ?>

    <div class="container era">
        <h2 align="center">Era dijagram:</h2>
        <img class="era2" src="images/era.png" width="720px"/>
    </div>

    <?php include_once 'predlozak/podnozje.php'; ?>
    <?php include_once 'predlozak/skripte.php'; ?>

  </body>
</html>
```

Slika 4.4. Era.php

5. Prijava i registracija korisnika

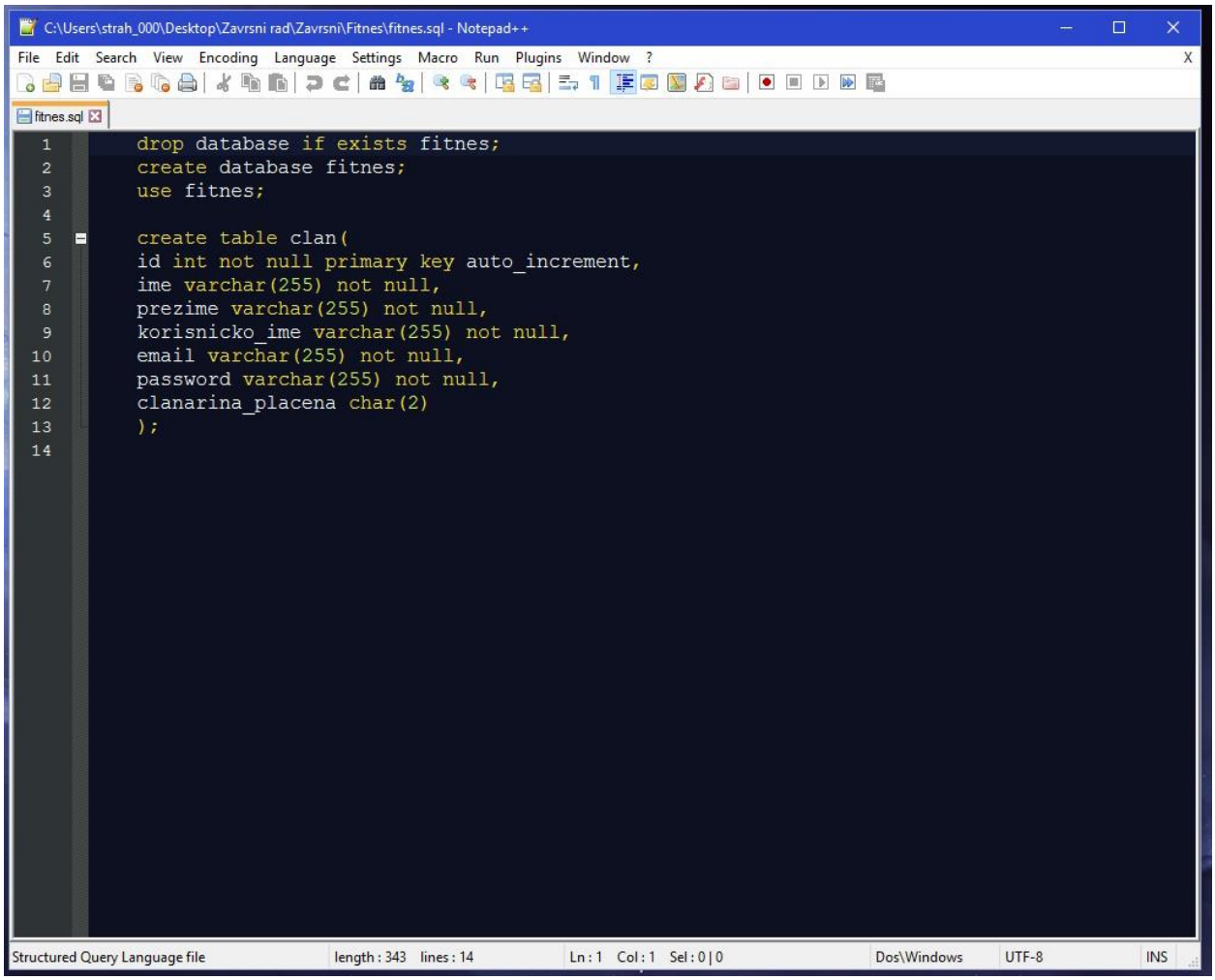
Prijava i registracija članova je napravljena pomoću *PHP-a 5.2.0* i *MySQL-a tj. MariaDB 10.1.10, phpMyAdmin* sučelja i *XAMPP-a v3.2.2.* Alati (programi) korišteni za izradu ove aplikacije su *Aptana Studio 3*, Notepad++ 7 i *code editor* na stranici https://byet.host/, a operacijski sustav je *Microsoft Windows 10.*

Kako bi mogli napraviti prijavu i registraciju korisnika za *web* aplikaciju tj. fitnes centar, potrebna je baza korisnika, u kojoj bi registrirali (unosili korisnike) i nakon toga omogućili postojećim korisnicima prijavu. Za kreiranje baze podataka, korištena je najpopularnija baza među *web* aplikacijama – *MySQL*.

5.1. Postavljanje *web* aplikacije *lokalno* i općenito o bazama

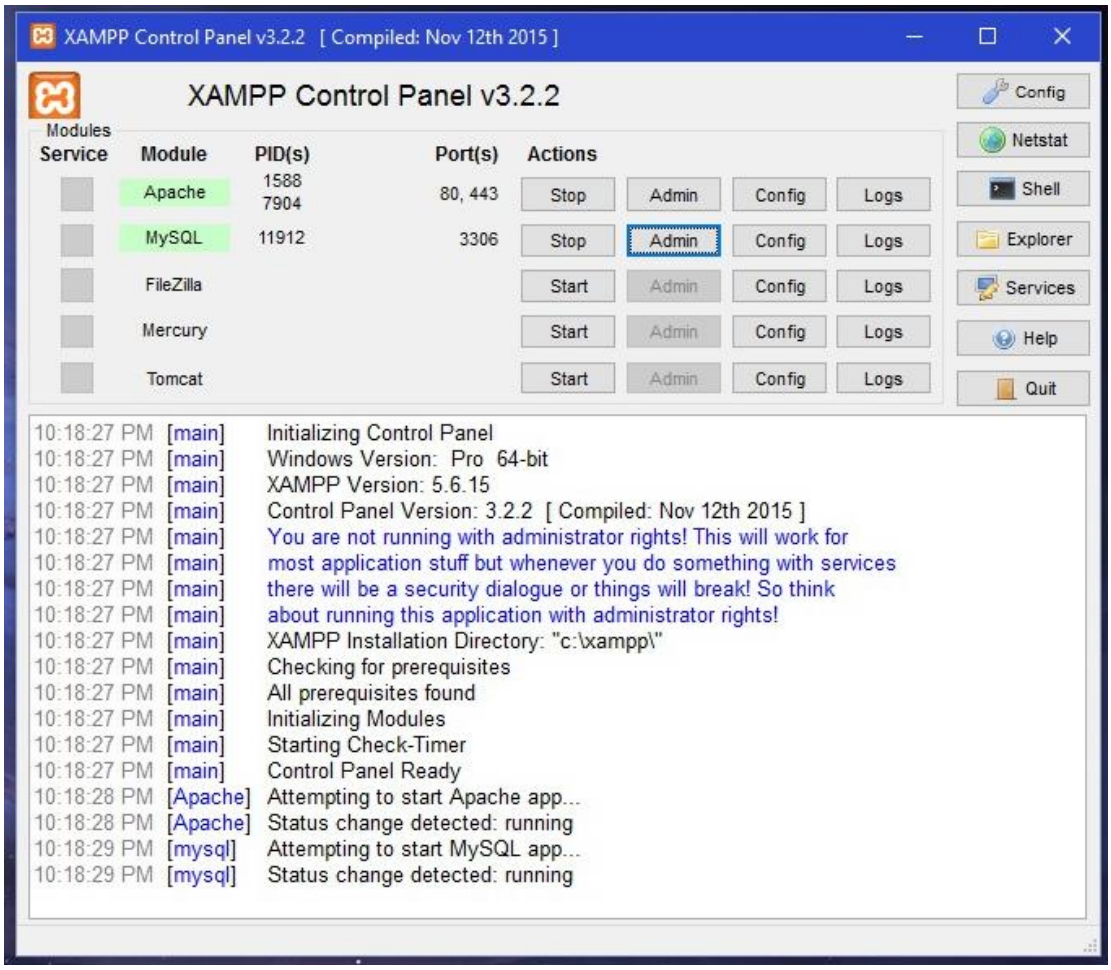
MySQL predstavlja relacijsku bazu podataka. [3] Relacijski model podrazumijeva određenu strukturu podataka pohranjenu u tablice. [3] Baza može imati jednu ili više tablica. Kod baze imamo upite (eng. *query),* a to su pitanja ili zahtjevi koje postavljamo bazi podataka. [3] Postavljanjem upita, mi iz baze tražimo neku određenu informaciju i automatski dobijamo povratnu informaciju (rezultat). Upiti se obavljaju sa *SQL* jezikom. *MySQL* baza se najčešće koristi sa *PHP* jezikom. Prvo se treba stvoriti baza da bi mogli uspostaviti konekciju s njom i da bi ju mogli koristiti pomoću *PHP-a*.

Za završni rad, *lokalno*, stvorena je baza pod nazivom "fitnes" (Sl.5.1.), a ona sadrži tablicu koja se zove "clan". Sa slike 5.1. možemo vidjeti da se prvo poziva kod koji briše bazu fitnes ako ona postoji, a ako ne postoji, stvara se nova baza pod tim imenom i koristi se. Nakon toga stvaramo tablicu "clan" koja se sastoji od redaka: *id* koji je primarni ključ (jedinstveni broj tj. vrijednost za svakog korisnika, počinje od 1, povećava se automatski za 1 svaki put kad se stvori novi korisnik i taj broj mu se dodjeljuje), ime, prezime, korisničko ime, email, password i da li je članarina plaćena. *Int* i *varchar* su tipovi podataka od kojih se sastoje ime, prezime itd. gdje *int* označava cjelobrojni broj, a *varchar* da se radi o *stringu* (slovima, tekstu). *Not null* znači da te vrijednosti ne smiju biti nula tj. prazne.



Slika 5.1. Baza fitnes

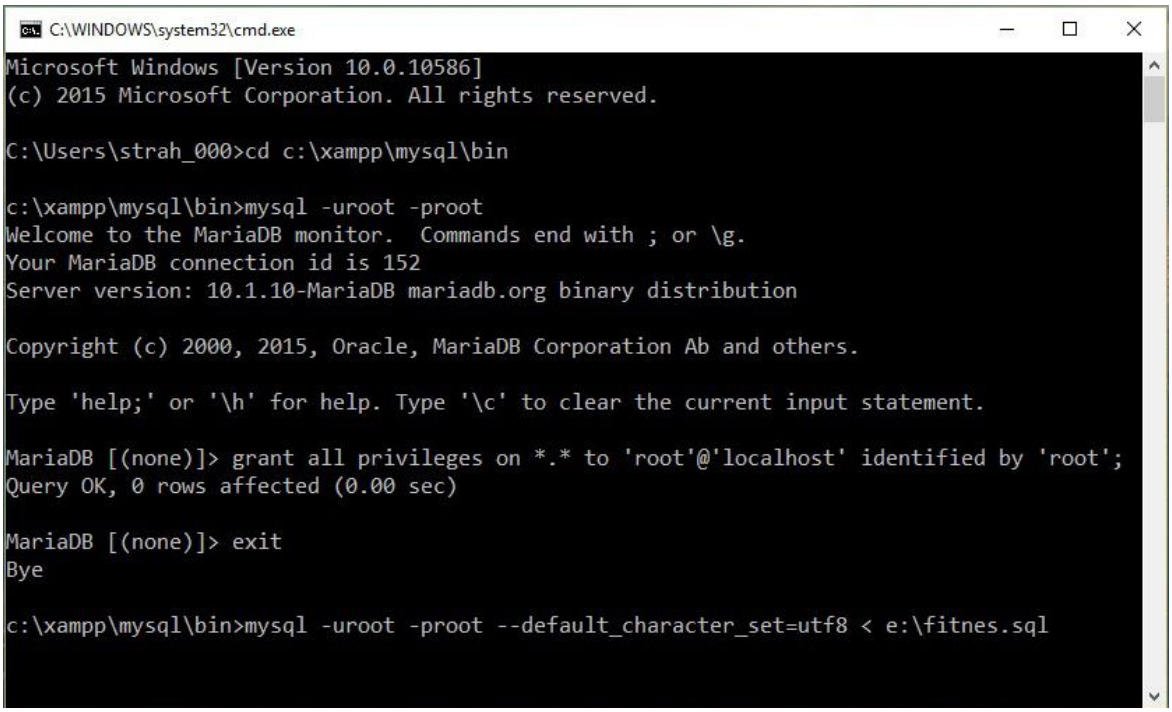
Nakon što smo kreirali bazu potrebno je skinuti i instalirati *XAMPP* alat (Sl.5.2.) tj. serverski paket koji u sebi sadrži *Apache* i *MySQL* server, kao i *MariaDB* bazu i skriptni jezik *PHP* (sadrži interpreter za skripte napisane u *PHP* jeziku). [4] *XAMPP* je namijenjen za upotrebu u lokalnoj mreži, a ne kao *web* server. [4]



Slika 5.2. XAMPP alat

Kada je *XAMPP* instaliran, pokreće se *MySQL* server, automatski je stvoren ugrađeni korisnik "root" koji je admin na računalu i unosi se nova baza (fitnes) na lokalni server pomoću *Windows Command Prompt-a* – CMD (Sl.5.3.) koristeći naredbe:

1. C:\Users\strah_000>cd c:\xampp\mysql\bin - pozicioniramo se u *MySQL*
2. c:\xampp\mysql\bin>mysql -uroot -proot - postavljamo korisnika "root" i password "root"
3. mysql>grant all privileges on *.* to 'root'@'localhost' identified by 'root' - dajemo privilegije (ovlasti) korisniku "root" na *localhost-u* (*lokalnom* serveru), to znači da on ima sve ovlasti, apsolutnu kontrolu i uvid u cijeli server sa svim njegovim bazama, te može brisati, dodavati i izmijeniti korisnike.
4. mysql>exit - izlazimo iz *MySQL-a*
5. c:\xampp\mysql\bin>mysql -uroot -proot < e:\fitnes.sql - unosimo bazu "fitnes" koja se nalazi na "e:" direktoriju računala



Slika 5.3. CMD

Ako sve prođe bez grešaka, baza je uspješno napravljena i stavljena na lokalni server. Nakon toga, treba se pozicionirati u direktorij gdje smo instalirali *XAMPP* i otvoriti xampp/htdocs. Taj direktorij predstavlja „root“ direktorij našeg *web* servera. Kako bi pristupili tom direktoriju u adresnu liniju pretraživača se upisuje:

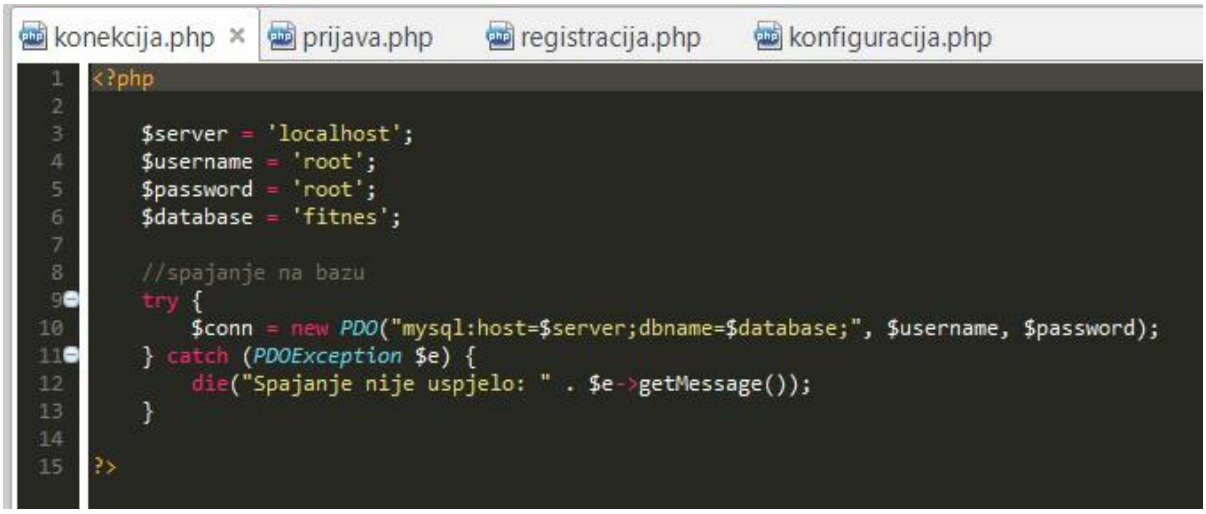
http://localhost

Preporučuje se da se obriše čitav sadržaj *htdocs* direktorija radi lakšeg prijenosa novih datoteka i organizacije. U *htdocs* ide direktorij „fitnes“ koji sadrži sve datoteke za *web* aplikaciju tj. završni, a pristupa mu se tako da se u adresnu liniju pretraživača upiše:

http://localhost/fitnes

Međutim, ne smijemo zaboraviti da moramo uključiti *Apache* i *MySQL* u *XAMPP* alatu kako bi smo mogli vidjeti *lokalni* server tj. *web* aplikaciju „ XL *fitness* centar“. *Apache* mora biti uključen jer *PHP* neće raditi bez njega zato što *PHP* je *server-side* (serverski) skriptni programski jezik koji zahtjeva server za rad, a *Apache* je lokalni *web* server.

Datoteka „konekcija.php“ (Sl.5.4.) sadrži postavke za spajanje baze sa serverom, gdje smo definirali ime servera, korisnika i *password* kao i naziv baze. Nakon toga, pomoću *try{}catch{}* metode se spajamo na bazu i ona izbacuje grešku ako se nismo uspješno spojili.

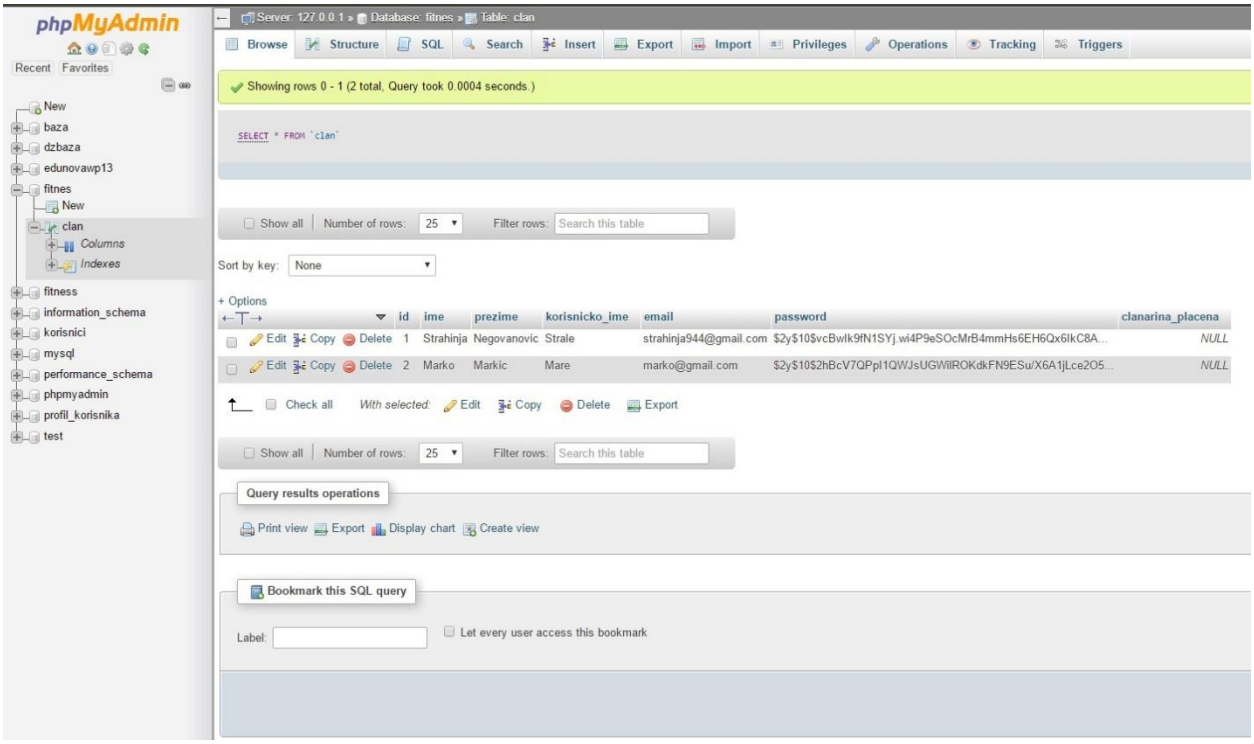


Slika 5.4. Datoteka konekcija.php

Nakon što smo se uspješno spojili na bazu i ako je sve prošlo bez grešaka, možemo ukucati stranicu u adresnu liniju pretraživača:

http://localhost/phpmyadmin/

phpMyAdmin (Sl.5.5.) je besplatan alat napisan u *PHP-u* koji služi za upravljanje *MySQL* bazama podataka. Pomoću njega, mi možemo vidjeti sve dostupne (napravljene baze), njihovu strukturu, dodavati, brisati i mijenjati korisnike u bazi, te vidjeti registrirane korisnike sa njihovim podacima. S lijeve strane se nalazi popis baza i njihovih tablica, a u sredini registrirani korisnici za „fitnes“ bazu.



Slika 5.5. phpMyAdmin

5.2. Postavljanje *hosting-a* i stvaranje nove baze

Dok smo radili *lokalno web* aplikaciju, *XAMPP* nam je služio kao server (jer u sebi sadrži *Apache* server) i baza (sadrži *MySQL* bazu). Da bi stranicu mogli vidjeti "*online"* pomoću poveznice na tu *web* aplikaciju, potreban nam je *hosting*. Odlučio sam postaviti stranicu *online* pomoću *byet.host hosting-a* koji je besplatan. *Hosting* sadrži *Cpanel* tj. upravljačku ploču pomoću koje se upravlja sadržajem *hosting-a*. Naša *web* aplikacija će biti učitana tek kada se prebace (eng. *upload*-aju) sve datoteke u mapu od upravljačke ploče. Kada smo prebacili datoteke *web* aplikacije na *byet.host* stranicu, *XAMPP* nam više nije bio potreban jer *host* stranica sve sadrži kao i *XAMPP*.

Moramo stvoriti novu bazu podataka jer baza "fitnes" radi samo *lokalno.* Datoteka fitness.sql (P.5.10. i P.5.11.) sadrži *SQL* upit kojim se stvara nova, veća baza podataka i kreiraju 4 tablice: clan, clanarina, status_clanarine i galerija i pomoću kojeg se odmah u bazu ubacuju podaci pomoću *insert* oznake. Za pregled baze podataka na *byet.host* stranici, koristi se *phpMyAdmin* sučelje pomoću kojeg se mogu vidjeti promjene na bazi i gdje se može baza uređivati. Moju *web* aplikaciju je moguće vidjeti na ovoj poveznici: http://fitneswp13.byethost15.com/?i=1

5.3. Spajanje na bazu

Datoteka konfiguracija.php (P.5.12. i P.5.13.) je datoteka koja se učitava prilikom otvaranja svake *PHP* datoteke *web* aplikacije "XL *fitness* centar" jer sam tako napisao. Ova datoteka sadrži različite postavke kao npr. postavke za hrvatski i engleski jezik, naslov i putanju *web* aplikacije, postavke za spajanje na bazu podataka i ispis greške ako spajanje nije uspjelo.

Datoteka konekcija.php (Sl.5.6.) služi za spajanje na novu bazu podataka kada je to potrebno, a uključi se pomoću *PHP* oznake *require* ili *include.*

Datoteka fitness.sql (P.5.10. i P.5.11.) sadrži sve postavke tj. *SQL* upit kojim se stvara konačna baza *fitness* za *host* stranicu.

```php
<?php

    $server = 'sql201.byethost15.com';
    $username = 'b15_18879166';
    $password = 'fitneshosting';
    $database = 'b15_18879166_fitnes';

    //spajanje na bazu
    try {
        $conn = new PDO("mysql:host=$server;dbname=$database;", $username, $password);
    } catch (PDOException $e) {
        die("Spajanje nije uspjelo: " . $e->getMessage());
    }

?>
```

Slika 5.6. Datoteka konekcija.php

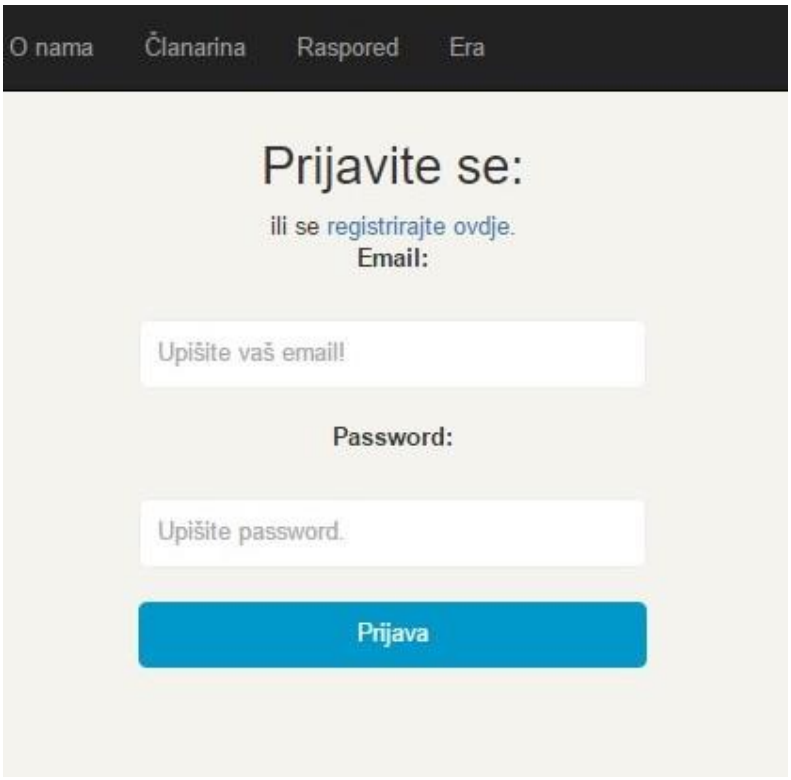
5.4. Prijava i registracija – forme i datoteke

Kada govorimo o prijavi i registraciji korisnika, prvo se mora napraviti forma u *HTML-u* (Sl.5.7.), gdje će se moći upisivati ime, prezime, email, password itd. Forma se u *HTML-u* označava sa oznakom *form, action* označava o kojoj se skripti radi, a *method = „post“* znači da će primiti korisnikov unos i neće ga prikazati u adresnoj traci pretraživača. <input type> označava polje koje se treba unijeti, ako je <input type=“email“>, tu će se unositi email itd. Svakom tom polju dajemo ime (eng. *name*) kako bi poslije mogli pomoću *PHP-a* koristiti ta imena za prijavu i registraciju odnosno ta imena su nam potrebna kako bi provjeravati da li su polja prazna, da li je korisnik registriran itd. *Placeholder* označava što će pisati u polju dok korisnik ne klikne na njega i počne nešto pisati. <input type=“submit“> je gumb koji ima vrijednost „prijava“ (na njemu piše prijava) i koji kada se pritisne šalje tu formu *PHP* skripti koja to šalje u bazu.

```
<form action="prijava.php" method="post">
    <fieldset>
        <label for="email" class="lemail">Email:</label><input id="email" type="email"  placeholder="Upišite vaš email!" name="email"/>
        <label for="password" class="lpassword">Password:</label><input id="password" type="password" placeholder="Upišite password." name="password" />
        <input type="submit" value="Prijava" name="prijava">
    </fieldset>
</form>
```

Slika 5.7. *HTML* forma za prijavu

Izgled prijave na stranici:



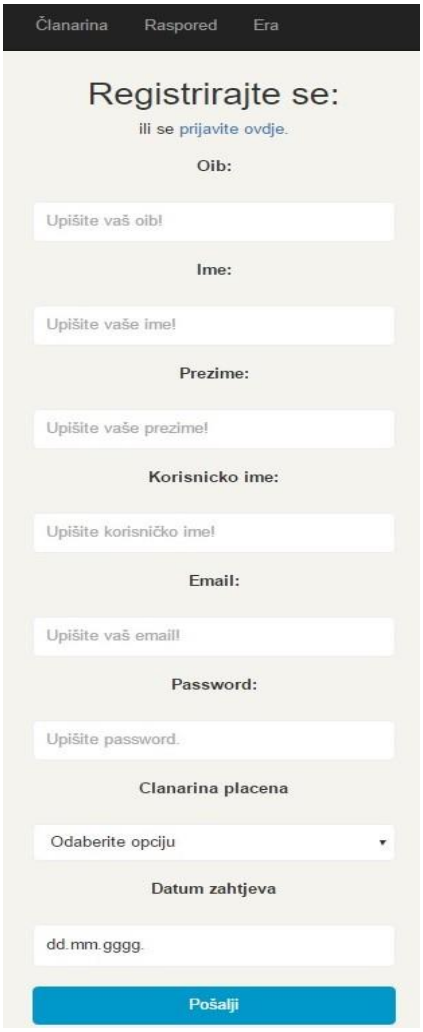
Slika 5.8. Prijava

Dio za registraciju korisnika sadrži sve isto kao i prijava, samo ima više polja za unijeti.

```
<form action="registracija.php" method="post">
    <fieldset>
        <label for="oib" class="lime">Oib:</label><input id="oib" type="text" placeholder="Upišite vaš oib!" name="oib"/>
        <label for="ime" class="lime">Ime:</label><input id="ime" type="text" placeholder="Upišite vaše ime!" name="ime"/>
        <label for="prezime" class="lprezime">Prezime:</label><input id="prezime" type="text" placeholder="Upišite vaše prezime!" name="prezime"/>
        <label for="korisnicko_ime" class="lkorisnik">Korisnicko ime:</label><input id="korisnicko_ime" type="text" placeholder="Upišite korisničko ime!" name="korisnicko_ime"/>
        <label for="email" class="lemail">Email:</label><input id="email" type="email" placeholder="Upišite vaš email!" name="email"/>
        <label for="password" class="lpassword">Password:</label><input id="password" type="password" placeholder="Upišite password." name="password" />

        <label for="clanarina_placena" class="lplaceno">Clanarina placena</label>
                <select id="clanarina_placena" name="clanarina_placena">
                    <option value="0">Odaberite opciju</option>
                    <option value="Da">Da</option>
                    <option value="Ne">Ne</option>
                </select>

        <label for="datum_zahtjeva" class="ldatum">Datum zahtjeva</label><input type="date" pattern="date" name="datum_zahtjeva" id="datum_zahtjeva"/>

        <input type="submit" name="registracija">
    </fieldset>
</form>
```

Slika 5.9. Forma za registraciju korisnika

Izgled registracije na stranici:



Slika 5.10. Registracija

Datoteka registracija.php (P.5.14.) sadrži kod za unos članova u bazu. *session_start()* je *PHP* funkcija koju *PHP* poziva da se pokrene nova "sjednica" (eng. *session)* i na taj način zapamti registrirani tj. prijavljeni korisnik, jer bez ove funkcije, nikad ne bismo imali prijavljenog korisnika jer *web* stranica ne zna da li je netko prijavljen ili nije, a sa ovom funkcijom zna. [5] Sa "if(isset($_SESSION…))" linijom koda se postavlja upit da li je "sjednica" postavljena tj. da li je korisnik registriran (da li je prijavljen), ako je, odvedi ga na početnu stranicu. Require izjava uključuje datoteku kojom se spajamo na bazu. Stvara se varijabla "message" kako bi mogli kasnije ispisati poruku ako je došlo do greške prilikom registracije. If (!empty($_POST['ime'] itd.) petlja provjerava da li je korisnik unio u polja ime, prezime i dr., i ako je, petlja unosi te parametre u bazu (pomoću varijable "sql" koja koristi SQL naredbu (*insert* upit) za unos podataka u bazu: "INSERT INTO clan (ime, prezime…) values (:ime, :prezime,..)). BindParam unosi ime u parametar ime i ostale dijelove, a password se "*hash-ira*" (sakriva nasumično generiranim znakovima) pomoću funkcije "password_hash". Zadnja if petlja provjerava da li su podaci dobro unešeni u bazu tj. da li je izjava "izvedena" i ako je, preusmjerava korisnika na stranicu za prijavu (datoteka prijava.php), ako nije, ispisuje poruku da je došlo do greške.

Datoteka prijava.php (P.5.15.) provjerava da li je član već prijavljen, ako je, ne može pristupiti ovoj stranici. Isto tako provjerava da li je član unio nešto u polja, ako je, provjeri u bazi da li podaci uneseni u polju odgovaraju podacima u bazi, ako odgovaraju, prijava člana je uspješna. Kada se član prijavi, ova datoteka ga preusmjerava na datoteku index.php (početna stranica). Sve dok član nije prijavljen, on vidi datoteku pocetna.php (P.5.3.).

Kada se član prijavi, isto tako se može i odjaviti, a to se postiže pomoću datoteke odjava.php (Sl.5.11.) gdje se *session* briše i *web* aplikacija više ne pamti korisnika. U odjavi, imamo *PHP* kod u kojem isključujemo "sjednicu" i poništavamo je, kako bi se korisnik mogao odjaviti i preusmjerava ga se na početnu stranicu, jer isključivanjem "sjednice", *web* stranica tj. skripta ne može upamtiti da je korisnik prijavljen i stoga dolazi do odjave.

```php
<?php

    session_start();

    session_unset();

    session_destroy();

    header("Location: ../../index.php");

?>
```

Slika 5.11. odjava.php

Index stranice “XL *fitness* centar” (P.5.16.) sadrži if petlju koja provjerava da li je korisnik prijavljen i ako je, sprema “rezultat” u varijablu *user,* a ako korisnik nije prijavljen, vraća ga na početnu stranicu.

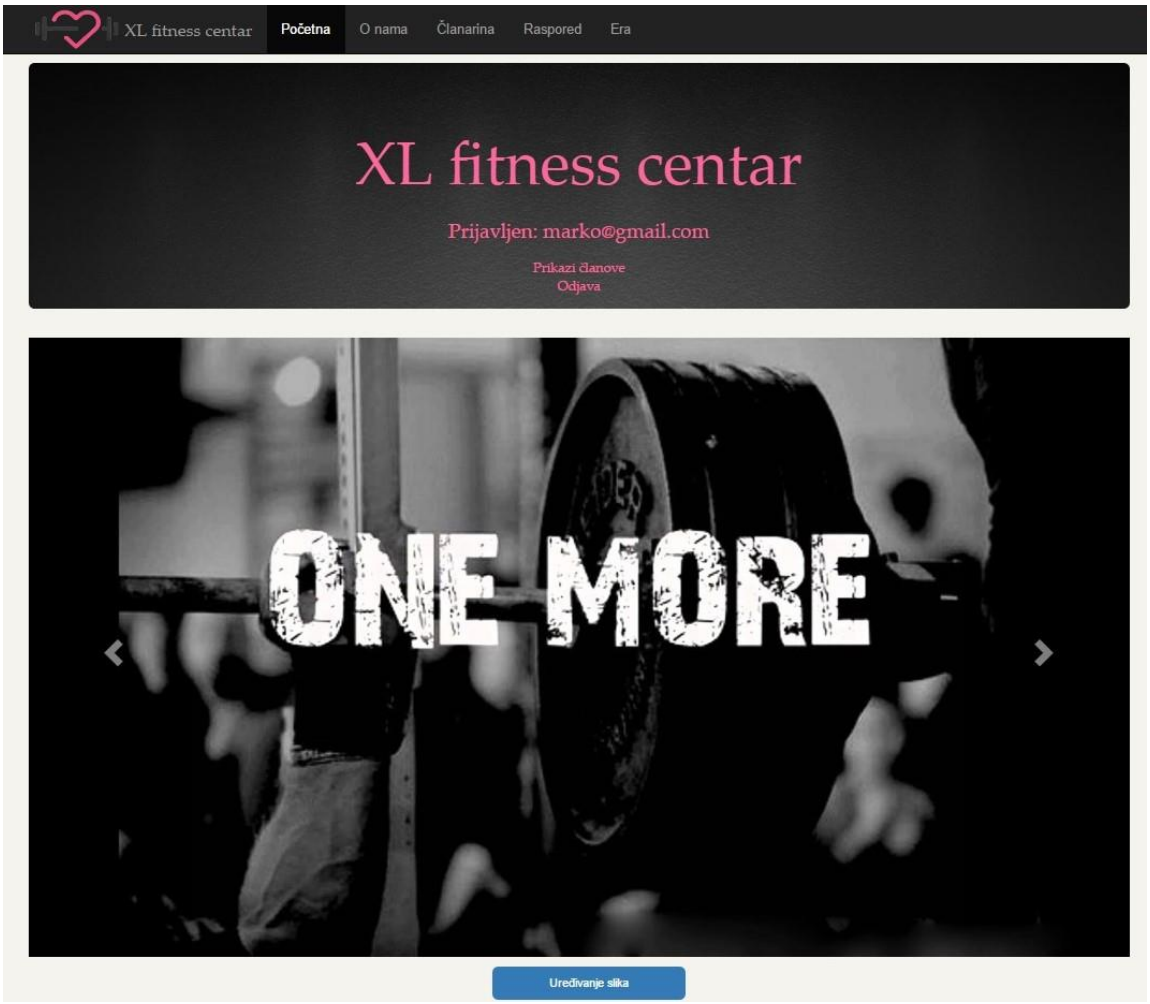
Drugi dio datoteke index.php (P.5.17) provjerava da li je korisnik prijavljen, ako je, da li ima ID = 1, ako ima, prijavljen je *admin* i on može vidjeti datoteke clanovi.php (P.5.19.) i prikaz.php (P.5.20. i P.5.21.), ako član nema ID = 1, radi se o članu koji samo može vidjeti datoteku korisnici.php (P.5.18. i Sl.5.13.) gdje su prikazani svi aktivni članovi (ali ne sa svim detaljima). Ako je korisnik prijavljen, ispisat će se kod naslova (Sl.5.12.) koji je korisnik prijavljen i on će imati opciju tj. link na datoteku “korisnici.php” gdje će moći vidjeti sve registrirane korisnike. A ako je prijavljen *admin,* onda će se na početnoj stranici ispisati da je prijavljen *admin* (Sl.5.14.) i pojavit će se dvije dodatne opcije, a to su: prikaži članove (Sl.5.15.) i prikaz svih tablica (Sl.5.16. i Sl.5.17.).

Datoteka korisnici.php (P.5.18.) dostupna je samo registriranim članovima, gdje imamo provjeru da li je član prijavljen i ako je, da mu se prikaže tablica članova sa njihovim detaljima (ali ne svim).

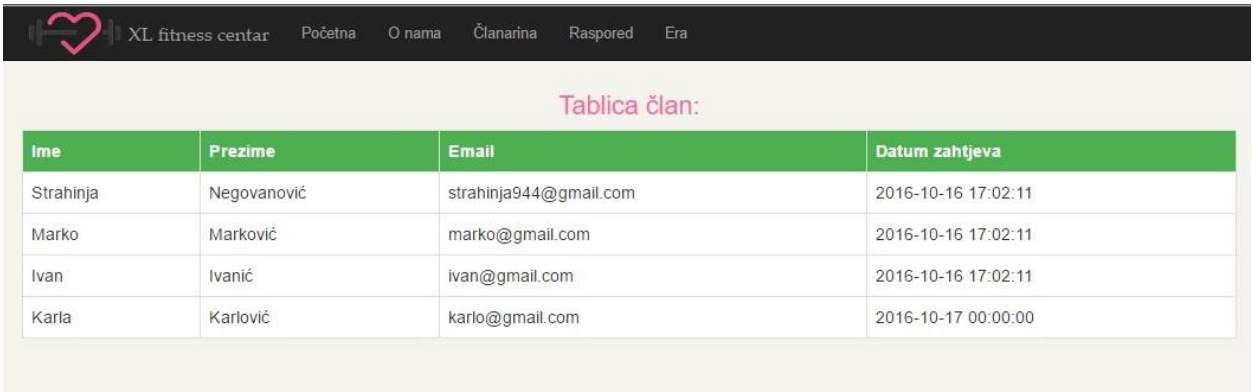
Datoteka clanovi.php (P.5.19.) je dostupna samo *adminu*, a pomoću nje je *adminu* prikazana tablica “clan” sa svim detaljima o aktivnim članovima (id, ime, prezime itd. kao i da li je članarina plaćena za određenog člana i kada). Početak *PHP koda* se odnosi na spajanje s bazom i odabir svih korisnika, gdje će se poredati u tablici po šifri tj. *id-u*. *PHP* kod gdje se nalazi *foreach* petlja će ispisati sva korisnikova svojstva jer *foreach* petlja „hvata“ vrijednost iz baza, a *echo* $red -> id, $red -> ime itd. naredbe će ispisati tu vrijednost.

Datoteka prikaz.php (P.5.20. i P.5.21.) je vidljiva samo *admin-u*. Kada se *admin* prijavi, dostupna mu je stranica prikaz svih tablica (Sl.5.14. i Sl.5.15.) gdje može vidjeti sve tablice u bazi i uređivati ih (promijeniti, obrisati, unijeti). Ispod svake tablice se nalazi gumb koji ga preusmjerava na unos, npr. ako se radi o tablici “clanarina”, *poveznica* ga preusmjerava na unos nove članarine. Isto tako, u tablicama se nalaze *link-ovi* promijeni i obriši, koji ga preusmjeravaju na promjenu detalja o odabranoj tablici ili na brisanje odabranog reda u tablici.

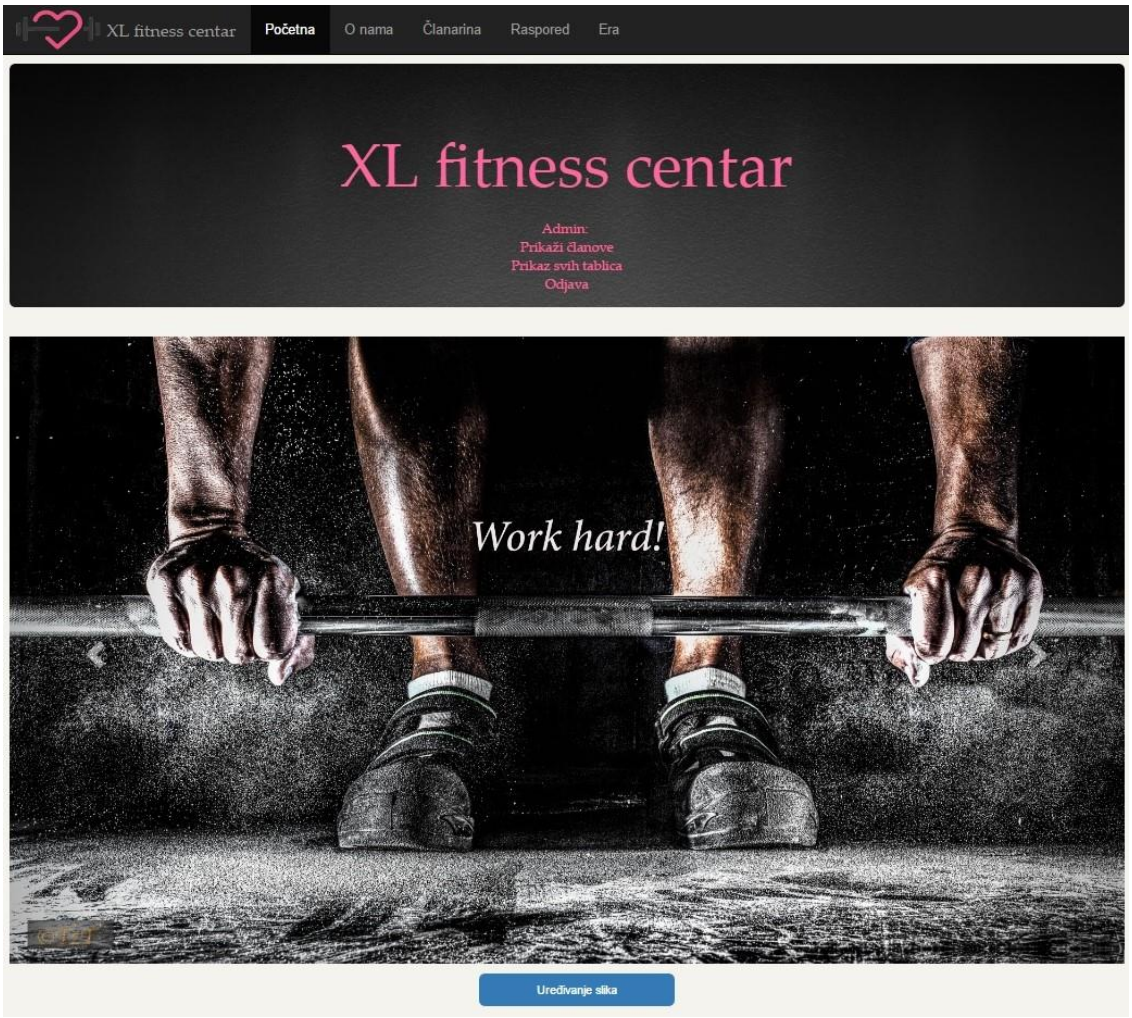
Datoteka insertc.php (P.5.22. i P.5.23.) sadrži formu za *insert* nove članarine. Ista takva forma se nalazi u insertsc.php datoteci, ali u toj datoteci se pišu svojstva tablice status_clanarine umjesto clanarine (P.5.22.). Ova datoteka također sadrži *SQL insert* upit za tablicu clanarina, gdje isti takav upit, ali prilagođen tablici status_clanarina, se nalazi u datoteci insertsc.php.



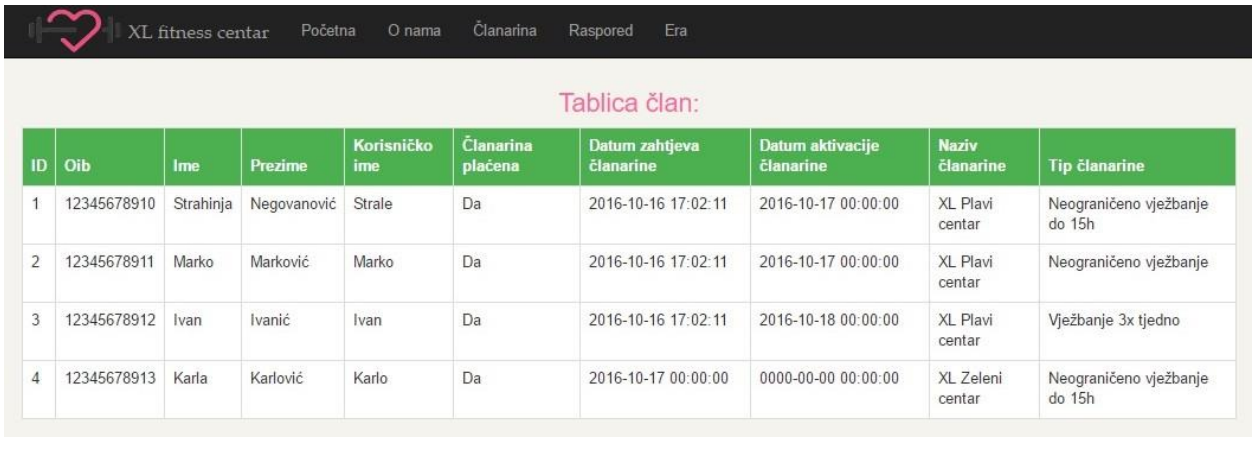
Slika 5.12. Prijavljen korisnik



Slika 5.13. Korisnici.php

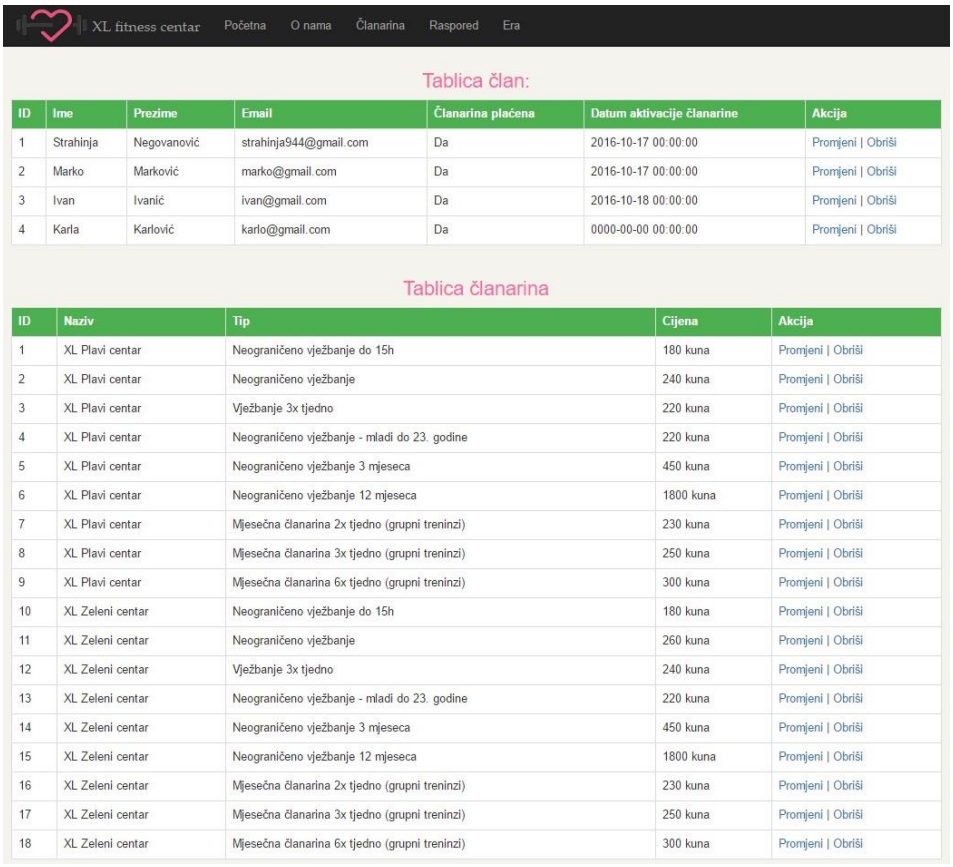


Slika 5.14. Prijavljen *admin*

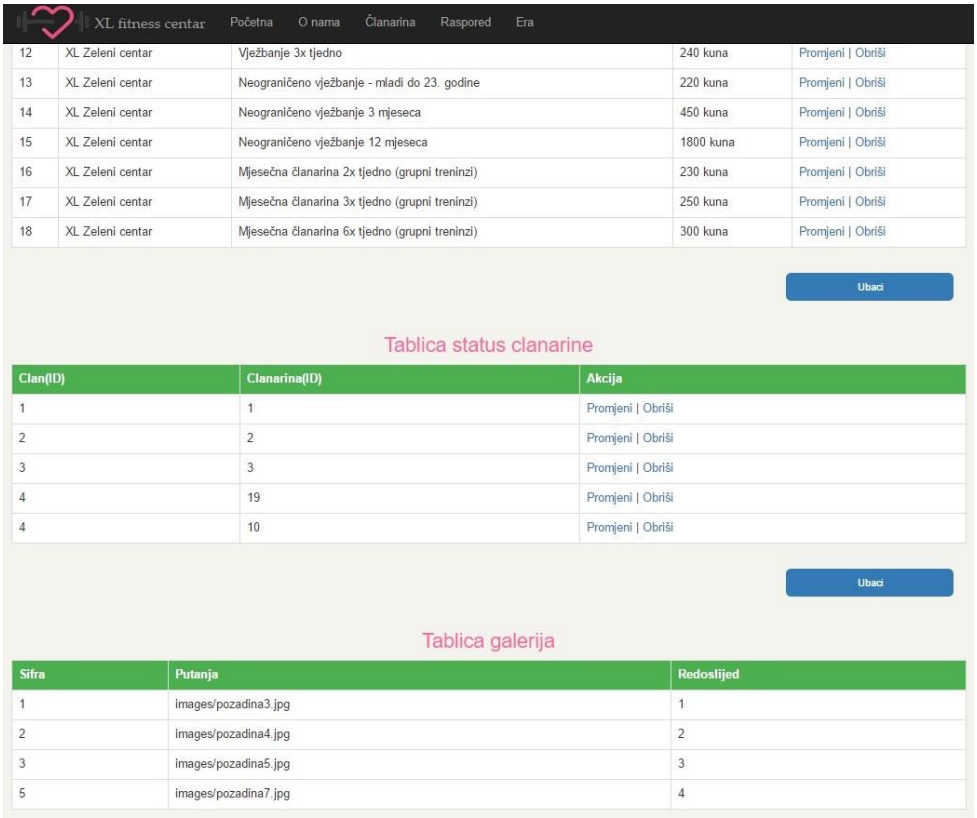


ID	Oib	Ime	Prezime	Korisničko ime	Članarina plaćena	Datum zahtjeva članarine	Datum aktivacije članarine	Naziv članarine	Tip članarine
1	12345678910	Strahinja	Negovanović	Strale	Da	2016-10-16 17:02:11	2016-10-17 00:00:00	XL Plavi centar	Neograničeno vježbanje do 15h
2	12345678911	Marko	Marković	Marko	Da	2016-10-16 17:02:11	2016-10-17 00:00:00	XL Plavi centar	Neograničeno vježbanje
3	12345678912	Ivan	Ivanić	Ivan	Da	2016-10-16 17:02:11	2016-10-18 00:00:00	XL Plavi centar	Vježbanje 3x tjedno
4	12345678913	Karla	Karlović	Karlo	Da	2016-10-17 00:00:00	0000-00-00 00:00:00	XL Zeleni centar	Neograničeno vježbanje do 15h

Slika 5.15. *Admin*: prikaz članova



Slika 5.16. *Admin*: prikaz svih tablica prvi dio



Slika 5.17. *Admin*: prikaz svih tablica drugi dio

Admin i registrirani članovi dobijaju pristup uređivanju slika tj. *slider-a.* Kada se prijave, pojavi im se *link* za uređivanje slika (datoteka uploadslika.php, slika 5.18.). Na toj poveznici se slike prvo odabiru, *upload-aju* na disk pomoću datoteka *upload.php* (Sl.5.19.), potom ubace u bazu pomoću datoteke insertimage.php (Sl.5.20.) i prikažu u tablici galerija.

```
<form action="upload.php" method="post" enctype="multipart/form-data" class="upload">
    <p>Dodajte sliku:</p>
    <input type="file" name="fileToUpload" id="fileToUpload" align="center">
    <br />
    <input type="submit" value="Upload Image" name="submit" class="slika2">
</form>

<a href="insertimage.php" class="slika1"><button type="button" class="btn btn-sm btn-primary slika">Ubaci sliku u bazu</button></a>
<a href="../../index.php" class="slika1"><button type="button" class="btn btn-sm btn-primary slika3">Odustani</button></a>
<br /> <br /> <br />
<h3 class="sve_tablice">Tablica galerija</h3>
     <table class="table table-bordered">

        <thead>
        <tr>
            <th>Sifra</th>
            <th>Putanja</th>
            <th>Redoslijed</th>
            <th>Akcija</th>
        </tr>
        </thead>

        <tbody>

        <?php

        try{

            $izraz = $veza->query("select * from galerija");
            $izraz->execute();
            $rezultati=$izraz->fetchAll(PDO::FETCH_OBJ);
        ?>

        <?php foreach ($rezultati as $red2): ?>
            <tr>
                <td><?php echo $red2->id ?></td>
                <td><?php echo $red2->putanja ?></td>
                <td><?php echo $red2->redoslijed ?></td>
                <td><a href="promjeni.php?id=<?php  echo $red2->id ?>">Promjeni</a> | <a href="obrisisliku.php?id=<?php  echo $red2->id ?>">Obriši</a></td>
            </tr>
        <?php endforeach; ?>

        <?php } catch(PDOException $e){
```

Slika 5.18. Datoteka uploadslika.php

Datoteka uploadslika.php (Sl.5.18.) sadrži formu za dodavanje slika i *SQL* upit za prikazivanje tablice galerija sa linkom na promjeni.php i obrisisliku.php gdje se mogu mijenjati stavke u tablici i obrisati slike.

Datoteka upload.php (Sl.5.19.) sadrži napisane *PHP* funkcije pomoću kojih se slika *upload-a*, samo je bitno na početku odabrati direktorij gdje će se slike ubaciti.

```php
<?php
    $target_dir = "../../images/";
    $target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
    $uploadOk = 1;
    $imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
    // Check if image file is a actual image or fake image
    if(isset($_POST["submit"])) {
        $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
        if($check !== false) {
            echo "File is an image - " . $check["mime"] . ".";
            $uploadOk = 1;
        } else {
            echo "File is not an image.";
            $uploadOk = 0;
        }
    }

    // Check if file already exists
    if (file_exists($target_file)) {
        echo "Sorry, file already exists.";
        $uploadOk = 0;
    }

    // Check file size
    if ($_FILES["fileToUpload"]["size"] > 2000000) {
        echo "Sorry, your file is too large.";
        $uploadOk = 0;
    }

    // Allow certain file formats
    if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" && $imageFileType != "gif" ) {
        echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
        $uploadOk = 0;
    }

    // Check if $uploadOk is set to 0 by an error
    if ($uploadOk == 0) {
        echo "Sorry, your file was not uploaded.";
    // if everything is ok, try to upload file
    } else {
        if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
            echo "The file ". basename( $_FILES["fileToUpload"]["name"]). " has been uploaded.";
            header('Location: uploadslika.php');
        } else {
            echo "Sorry, there was an error uploading your file.";
        }
        }
```

Slika 5.19. Datoteka upload.php

Datoteka insertimage.php (Sl.5.20.) sadrži *HTML* formu za unos slike u bazu i *SQL insert* upit u bazu kojim se slika unosi u bazu.

```php
<?php

    session_start();

    require '../../konekcija.php';
    $message = '';

    if(!empty($_POST['putanja']) && !empty($_POST['redoslijed'])):


        $sql = "INSERT INTO galerija (putanja, redoslijed) VALUES (:putanja, :redoslijed)";
        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':putanja', $_POST['putanja']);
        $stmt->bindParam(':redoslijed', $_POST['redoslijed']);

        if( $stmt->execute() ):
            $message = 'Ubacena slika uspjesno!';
            header("Location: ../../index.php");
        else:
            $message = 'Neuspjelo dodavanje slike!';
        endif;

    endif;

?>

<!DOCTYPE html>
<html lang="hr">
  <head>
    <?php include_once '../../predlozak/head.php'; ?>
  </head>

  <body>
    <?php include_once '../../predlozak/izbornik.php'; ?>

  <div class="container">
    <form action="insertimage.php" method="post">
        <fieldset>
            <label for="putanja" class="lime">Putanja:</label><input id="putanja" type="text" placeholder="Upišite putanju npr. images/pozadina5.jpg!" name="putanja"/>
            <label for="redoslijed" class="lprezime">Redoslijed:</label><input id="redoslijed" type="text" placeholder="Upišite broj redoslijeda, zadnji je 3!" name="redoslijed"/>
            <input type="submit" name="registracija">
        </fieldset>
    </form>
```

Slika 5.20. Datoteka insertimage.php

HTML forma za promjenu svojsta tablice galerije (Sl.5.21.) je ista za sve tablice (clan, clanarina, status_clanarine) gdje se samo kod drugih tablica stavlja svojstvo koje pripada njima u formi. Npr. na slici 5.21. Svojsva galerije su putanja i redoslijed, kod clana su oib, ime, prezime itd.

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"] ?>">
    <label for="putanja" class="labela">Putanja</label>
        <input type="text" id="putanja" name="putanja" value="<?php echo $entitet->putanja; ?>" />
        <?php if(strlen($entitet->putanja)==0): ?>
        <span class="alert label">Upisati putanju!</span>
        <?php endif; ?>

                <label for="redoslijed" class="labela">Redoslijed</label>
        <input type="number" id="redoslijed" name="redoslijed" value="<?php echo $entitet->redoslijed; ?>" />
        <?php if(strlen($entitet->redoslijed)==0): ?>
        <span class="alert label">Upisati redoslijed!</span>
        <?php endif; ?>

        <input type="hidden" name="sifra" value="<?php echo $entitet->id; ?>" />
        <input type="submit" value="Spremi" />
</form>
```

Slika 5.21. promjeni.php *HTML* forma

Datoteka promjeni.php (Sl.5.22.) je ista za sve tablice također, samo ima drugačiji naziv, npr. za člana je promjenic.php itd. Isto tako, kod unutar promjeni.php, se primjenjuje i na druge tablice, samo se stavljaju njihova svojstva umjesto svojstva galerije kao što je prikazano na slici 5.21. Datoteka promjeni.php ima dva *SQL* upita, prvi odabire sve iz tablice pomoću ID-a, a drugi upit radi *update* tj. promjenu gdje je ID trenutno postavljen (tražen).

```
<?php include_once '../../konfiguracija.php';

    if($_GET){
        $izraz = $veza->prepare(
        "select * from galerija where id=:id");
        $izraz->execute($_GET);
        $entitet=$izraz->fetch(PDO::FETCH_OBJ);
    }else{

        if(strlen($_POST["putanja"])>0){
        try
        {

            $izraz = $veza->prepare(
            "update galerija set putanja=:putanja, redoslijed= :redoslijed where id=:id");
            $izraz->bindParam(':putanja', $_POST["putanja"], PDO::PARAM_STR);
            $izraz->bindParam(':redoslijed', $_POST["redoslijed"], PDO::PARAM_INT);
            $izraz->bindParam(':id', $_POST["sifra"], PDO::PARAM_INT);
            $izraz->execute();
        }

         catch(PDOException $pe)
        {
            die($pe->getMessage());
        }

        if($izraz)
            header("location: uploadslika.php");
        else
            echo "Pogreška";

        }else{
            $entitet=new stdClass();
            $entitet->sifra=$_POST["id"];
            $entitet->naziv=$_POST["putanja"];
            $entitet->naziv=$_POST["redoslijed"];
        }

    }

?>
```

Slika 5.22. promjeni.php

Datoteka obrisisliku.php (Sl.5.23.) provjerava da li je član prijavljen, ako je, slijedi *SQL* upit kojim se briše sve iz tablice galerija gdje je postavljen tj. tražen ID i preusmjerava člana na uploadslika.php. Kod napisan u datoteci obrisisliku.php je isti za sve tablice u bazi tj. za sve datoke koje imaju u nazivu "obrisi", samo je *SQL* upit kod svih tih datoteka različit i prilagođen nazivu tražene tablice.

```php
<?php include_once '../../konfiguracija.php';

    if (!isset($_GET["id"])  ){
        header("location:" . $putanjaAPP);
    }


    $izraz = $veza->prepare(
    "delete from galerija where id= :id");
    $izraz->execute($_GET);
    header("location: uploadslika.php" );
```

Slika 5.23. obrisisliku.php

6. Zaključak

Izrada dinamične *web* aplikacije za organizaciju rada fitnes centra mi je bilo zadovoljavaće iskustvo i izazov jer sam svoje osnovno znanje *HTML-a, CSS-a, JavaScript-e, PHP-a i MySQLa* ovim završnim radom doveo na veću razinu i napravio po mom mišljenju lijepu i funkcionalnu *web* aplikaciju koja može pomoći fitnes centrima u njihovoj organizaciji.

Web aplikacija je dinamična, prilagođena za sve rezolucije monitora i mobitele, izgleda lijepo, ima napravljen grafički prikaz rasporeda i članarine, kontakt formu gdje se može poslati dodatno pitanje, savjet itd. vlasniku putem *mail-a*. Također ima napravljenu potpuno funkcionalnu prijavu i registraciju gdje imamo dva korisnička profila koji se razlikuju po ovlastima: *admina* i korisnika. *Admin* može brisati, dodavati i mijenjati informacije o korisniku, a korisnik može vidjeti normalno stranicu i pregledati aktivne korisnike.

Dijelovi stranice kao što su prijava i registracija, kao i raspored i članarina, ne moraju nužno biti korišteni samo za fitnes centre. Ti dijelovi se mogu koristiti i za druge *web* stranice gdje se koristi prijava i registracija (npr. za forume, fakultete, ankete itd.) i gdje je potreban nekakav raspored (npr. za školu) i cijenik (npr. za restoran).

Svaka *web* aplikacija se može u nekom području unaprijediti pa tako i ova moja. Dizajn i izgled stranice uvijek može biti bolji, tako da ako bi ovu *web* aplikaciju prepravio dobar dizajner, ona bi imala još ljepši dizajn i izgled. Vijesti na početnoj stranici i dio "o nama" bi se trebali prilagoditi korisničkim željama kako bi imale smisao. Raspored i članarina sadrže izmišljene podatke, stoga bi također trebali sadržavati podatke prema korisničkim željama ili željama vlasnika nekog fitnes centra. Kontakt forma može imati više polja za unos, kao i forme za prijavu i registraciju. I na kraju, kod *web* aplikacije bi mogao biti kraće napisan i promijenjen kako tehnologije napreduju, nove funkcije zamjenjuju stare, ako se neke oznake više ne koriste itd.

Literatura:

[1] http://getbootstrap.com/getting-started/

[2] http://www.w3schools.com/

[3] https://www.codecademy.com/learn

[4] https://www.apachefriends.org/index.html

[5] http://php.net/manual/en/

[6] K. Tatroe, P. MacIntyre, R. Lerdorf - PHP programiranje, Dobar plan, Zagreb, 2015.

[7] https://www.udemy.com/courses/

[8] https://developer.mozilla.org/hr/docs/Web/HTML/Element

[9] https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

[10] http://devdocs.io/javascript/

[11] http://dev.mysql.com/doc/

Popis upotrijebljenih oznaka:

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

PHP - PHP: Hypertext Preprocessor

SQL – Structured Query Language

MySQL – My Structured Query Language

XAMPP – Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P)

MariaDB – Maria Database

jQuery – JavaScript Query

Int - integer

Varchar – Variable Character Field

Id – identifier, identity, identification

Sažetak

Tijekom rada na *web* aplikaciji, prijava i registracija članova su mi predstavljali najveći izazov. Kod prijave i registracije korisnika, lozinka mi je stvarala najveći problem tj. metoda zaštite (eng. *hash*). Trebao sam odabrati metodu *hash-iranja* lozinke (zaštitu) i odlučio sam se za metodu *MD5*. Međutim, istraživajući na internetu, saznao sam da je ta metoda *deprecated* (ne koristi se više) i umjesto ove metode, odlučio sam koristiti *BCRYPT* metodu. Problem je bio taj što sam prilikom registracije dodavao članove sa *BCRYPT* metodom, a u bazi podataka sam unio 2 člana sa *MD5* metodom i zbog toga mi nije radila prijava za ta 2 člana. U bazu podataka sam unio 2 člana sa *MD5* metodom jer nisam znao kako ubaciti *BCRYPT* metodu, a za *MD5* metodu postoji oznaka MD5 u *SQL* jeziku, pomoću koje je moguć unos *hash-a* u bazu. Drugim riječima, u bazu se stavi oznaka "*md5*" i lozinka korisnika, dok za *BCRYPT* metodu ne postoji takav način dodavanja. *BCRYPT* metoda se unosi u bazu tako što se se prvo *hashira* vaša lozinka pomoću ove metode u dužini do 12 znakova, nakon čega se taj *hash* ubaci u bazu i na taj način prijava bude moguća.

Ključne riječi:

HTML, CSS, PHP, MySQL, SQL, baza, *phpMyAdmin, XAMPP, JavaScript, jQuery*, fitnes, fitnes centar, XL fitnes centar, *web* aplikacija, *web*, aplikacija, *Bootstrap, framework*, raspored, članarina, prijava, registracija, *admin*, korisnik, forma, tablica, era dijagram, *hosting*.

Responsive web application for fitness center

Abstract

While working on a web application, my biggest challenge was to make user login and registration. I had most problems with choosing appropriate password hash because if you want to save password safely, you need to use hash. Firstly, I have chosen MD5 method. However, exploring the Internet I've found out that this MD5 method is deprecated (no longer in use) and instead, I've decided to use BCRYPT method. The main problem was during user registration, where I was adding members with BCRYPT method but in database two users were added with MD5 method and because of that, user login wasn't working (methods weren't same). I've added two users with MD5 method in database because I didn't knew how to insert BCRYPT method. If you want to use MD5 password method, there is a MD5 tag in SQL language for it and you need to put that „md5“ tag in database for users password. But, for BCRYPT method, that kind of a tag doesn't exist. In order to insert BCRYPT method in the database, firstly you must hash your password with this method (password length is around 12 characters) after which the method is inserted into the database and thus user login is possible and working.

Keywords:

HTML, CSS, PHP, MySQL, SQL, database, phpMyAdmin, XAMPP, JavaScript, jQuery, fitness, fitness center, XL fitness center, web application, web, application, Bootstrap, framework, membership, login, registration, admin, user, form, table, er diagram, hosting.

Životopis:

Strahinja Negovanović rođen je 17.09.1994. u Vukovaru, gdje je završio osnovnu i srednju školu jezičnog smjera. Student je stručnog studija na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, smjer Informatika. Dodatnu edukaciju je stekao u školi Edunova, gdje je dobio zvanje „Web programer“. Od stranih jezika zna engleski gdje je dobar u čitanju, pisanju i govoru.

_____

Potpis studenta

Prilozi

```
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- ikonica u pregledniku -->
<link rel="icon" href="ikonice/logo1.ico">
<title><?php echo $naslovAPP; ?></title>
<!-- Bootstrap core CSS -->
<link href="<?php echo $putanjaAPP; ?>bootstrap/css/bootstrap.min.css" rel="stylesheet">
<!-- Custom styles for this template -->
<link href="<?php echo $putanjaAPP; ?>css/stil.css" rel="stylesheet">
<link href="<?php echo $putanjaAPP; ?>css/animate.css" rel="stylesheet">
```

Prilog 2.1. Datoteka head.php

```
<!-- div koji se sastoji od logo slike, imena i izbornika -->
<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a href="index.php" class="pull-left"><img src="/images/logo1.jpg" alt="" height="50" width="100"></a>
      <a class="navbar-brand" href="index.php">XL fitness centar</a>
    </div>

    <!-- izbornik -->
    <div id="navbar" class="collapse navbar-collapse">
      <ul class="nav navbar-nav">
          <li <?php

              if($_SERVER["PHP_SELF"]== $putanjaAPP . "index.php")
              {
                echo "class=\"active\"";
              }

              ?>><a href="<?php echo $putanjaAPP; ?>index.php">Početna</a></li>
        <li <?php

              if($_SERVER["PHP_SELF"]== $putanjaAPP . "onama.php")
              {
                echo "class=\"active\"";
              }

              ?>><a href="<?php echo $putanjaAPP; ?>onama.php">O nama</a></li>
        <li <?php

              if($_SERVER["PHP_SELF"]== $putanjaAPP . "clanarina.php")
              {
                echo "class=\"active\"";
              }

              ?>><a href="<?php echo $putanjaAPP; ?>clanarina.php">Članarina</a></li>
        <li <?php

              if($_SERVER["PHP_SELF"]== $putanjaAPP . "raspored.php")
              {
                echo "class=\"active\"";
```

Prilog 2.2. Datoteka izbornik.php

```php
<div class="container">

    <div class="jumbotron">
        <div class="naslov"><h1 class="animated tada"><?php echo $naslovAPP; ?></h1>
          <a href="CRUD/clan/prijava.php" class="unos"><button type="button" class="btn btn-sm btn-primary gumb">Prijava</button></a>
          <a href="CRUD/clan/registracija.php" class="unos"><button type="button" class="btn btn-sm btn-primary gumb">Registracija</button></a>
        </div>
    </div>

        <div id="carousel-example-generic" class="carousel slide" data-ride="carousel">

        <div class="carousel-inner" role="listbox">
              <div class="item active">
                    <img src="images/pozadina1.jpg" alt="First slide" />
              </div>

              <?php

                try{
                    $izraz = $veza->query("select putanja from galerija order by redoslijed");
                    $izraz->execute();
                    $rezultati=$izraz->fetchAll(PDO::FETCH_OBJ);
              ?>

              <?php foreach ($rezultati as $red3): ?>

                    <div class="item">
                        <img src="<?php echo $red3->putanja ?>"/>
                    </div>
              <?php endforeach; ?>

              <?php }   catch(PDOException $e){
                    d($e);
                }
              ?>
        </div>

        <a class="left carousel-control" href="#carousel-example-generic" role="button" data-slide="prev">
          <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
          <span class="sr-only">Previous</span>
        </a>
        <a class="right carousel-control" href="#carousel-example-generic" role="button" data-slide="next">
          <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
          <span class="sr-only">Next</span>
        </a>
      </div>
```

Prilog 2.3. Datoteka pocetna.php

```php
<!-- Bootstrap core JavaScript
================================================== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="../../assets/js/vendor/jquery.min.js"><\/script>')</script>
<script src="<?php echo $putanjaAPP; ?>bootstrap/js/bootstrap.min.js"></script>
```

Prilog 2.4. Datoteka skripte.php

```css
body {
    padding-top: 60px;
    background-color: #F5F3EE;
}

.era{
    padding-top: 20px;
    padding-bottom: 302px;
    text-align: center;
    margin-bottom: 6px;
}

.era2{
    padding-top: 50px;
}

hr {
    width: 45%;
}

.lead {
    text-align:justify;
}

.headerline {
    border: 1px solid #e5e5e5;
    margin-top: 1em;
}

.navbar-header {
    font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;
}

.footer {
    background-color: #222;
    color: white;
    height: 70px;
    width: 100%;
    text-align: center;
}

.jumbotron {
    padding-bottom: 1em;
    text-align: center;
    background-image: url("../images/pozadina2.jpg");
    background-size: cover;
```

Prilog 2.5. Datoteka stil.css

```php
<?php include_once 'konfiguracija.php'; ?>
<!DOCTYPE html>
<html lang="hr">
  <head>
    <?php include_once 'predlozak/head.php'; ?>
  </head>

  <body>
    <?php include_once 'predlozak/izbornik.php'; ?>

  <div class="container">
    <h2>Članarina u XL fitnes centrima:</h2>
    <div class="table-responsive">
     <table class="table table-bordered">
            <thead>
              <tr>
                <th>Tip</th>
                <th>XL1</th>
                <th>XL2</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <td>Neograničeno vježbanje do 15h</td>
                <td>180 kuna</td>
                <td>180 kuna</td>
              </tr>
              <tr>
                <td>Neograničeno vježbanje</td>
                <td>240 kuna</td>
                <td>260 kuna</td>
              </tr>
              <tr>
                <td>Vježbanje 3x tjedno</td>
                <td>220 kuna</td>
                <td>240 kuna</td>
              </tr>
              <tr>
                <td>Neograničeno vježbanje - mladi do 23. godine</td>
                <td>220 kuna</td>
                <td>220 kuna</td>
              </tr>
              <tr>
                <td>Neograničeno vježbanje 3 mjeseca</td>
                <td>450 kuna</td>
                <td>450 kuna</td>
```

Prilog 3.6. Datoteka clanarina.php

```php
<?php include_once 'konfiguracija.php'; ?>

<!DOCTYPE html>
<html lang="hr">
  <head>

    <?php include_once 'predlozak/head.php'; ?>
    <script src="http://cdn.alloyui.com/3.0.1/aui/aui-min.js"></script>
    <link href="http://cdn.alloyui.com/3.0.1/aui-css/css/bootstrap.min.css" rel="stylesheet"></link>

  </head>

  <body style="background-color: #F5F3EE;">

    <?php include_once 'predlozak/izbornik.php'; ?>

    <h2 align="center">Raspored fitness grupa:</h2>
    <div class="container">
        <div id="wrapper">
            <div id="myScheduler">
                <script>
                    YUI().use(
                    'aui-scheduler',
                    function(Y) {
                      var events = [
                        {
                          color: "#ff5fc8",
                          content: 'S-Dance',
                          endDate: new Date(2016, 9, 17, 11),
                          startDate: new Date(2016, 9, 17, 9)
                        },
                        {
                          color: "#ff5fc8",
                          content: 'S-Dance',
                          endDate: new Date(2016, 9, 20, 20),
                          startDate: new Date(2016, 9, 20, 18)
                        },
                        {
                          color: "#ff5fc8",
                          content: 'S-Dance',
                          endDate: new Date(2016, 9, 21, 10),
                          startDate: new Date(2016, 9, 21, 8)
                        },
                        {
                          color: "#ff5fc8",
                          content: 'Lift and Pump',
```

Prilog 3.7. Datoteka raspored.php

```
                {
                  color: "#ff5fc8",
                  content: 'Puls',
                  endDate: new Date(2016, 9, 20, 11),
                  startDate: new Date(2016, 9, 20, 9)
                },
                {
                  color: "#ff5fc8",
                  content: 'Puls',
                  endDate: new Date(2016, 9, 21, 13),
                  startDate: new Date(2016, 9, 21, 12)
                }
              ];

              var agendaView = new Y.SchedulerAgendaView();
              var dayView = new Y.SchedulerDayView();
              var weekView = new Y.SchedulerWeekView();
              var monthView = new Y.SchedulerMonthView();
              var eventRecorder = new Y.SchedulerEventRecorder();

              new Y.Scheduler(
                {
                  activeView: weekView,
                  boundingBox: '#myScheduler',
                  date: new Date(2016, 9, 16),
                  eventRecorder: eventRecorder,
                  items: events,
                  render: true,
                  views: [dayView, weekView, monthView, agendaView]
                }
              );
            }
          );

    </script>
```

Prilog 3.8. Datoteka raspored.php drugi dio

```
function validateForm() {
        var x = document.forms["myForm"]["ime"].value;
        if (x == null || x == "") {
        alert("Morate napisati ime!");
        return false;
        }

        var x = document.forms["myForm"]["prezime"].value;
        if (x == null || x == "") {
        alert("Morate napisati prezime!");
        return false;
        }

        var x = document.forms["myForm"]["email"].value;
        if (x == null || x == "") {
        alert("Morate napisati e-mail!");
        return false;
        }

        var x = document.forms["myForm"]["email"].value;
        var atpos = x.indexOf("@");
        var dotpos = x.lastIndexOf(".");
        if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length) {
            alert("Nije valjana e-mail adresa");
            return false;
        }

        var x = document.forms["myForm"]["tekst"].value;
        if (x == null || x == "") {
        alert("Morate napisati pitanje, jer bez pitanja, kontakt dio nema smisla. :)");
        return false;
        }


    }
```

Prilog 4.9. Datoteka validacija.js

```
drop database if exists fitness;
create database fitness charset utf8 COLLATE utf8_croatian_ci;
use fitness;

create table clan(
id int not null primary key auto_increment,
oib char(11) not null,
ime varchar(255) not null,
prezime varchar(255) not null,
korisnicko_ime varchar(255) not null,
email varchar(255) not null,
password varchar(255) not null,
clanarina_placena char(2) not null,
datum_zahtjeva TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
datum_aktivacije datetime
);

create table clanarina(
id int not null primary key auto_increment,
naziv varchar (255) not null,
tip varchar (255) not null,
cijena varchar (50)
);

create table status_clanarine (
clan int not null,
clanarina int not null
);

create table galerija (
id int not null primary key auto_increment,
putanja varchar (255) not null,
redoslijed int not null
);

alter table status_clanarine add foreign key(clan) references clan(id);
alter table status_clanarine add foreign key(clanarina) references clanarina(id);

insert into clan (oib, ime, prezime, korisnicko_ime, email, password, clanarina_placena, datum_zahtjeva, datum_aktivacije)
values ('12345678910','Strahinja','Negovanović', 'Strale', 'strahinja944@gmail.com', '$2a$10$6v2LiSWqCSCrr/CVdRmXpunClLCJ.hLqI8NakbyyVbheLt8DyCLGa', 'Da', CURRENT_TIMESTAMP, '2016-10-17'),
('12345678911','Marko','Marković','Marko','marko@gmail.com', '$2a$10$aHpJUA5K01cS1yiFPy2gJ.aTGdHqJZVUe5mItPVhMt37t.CQGmhsq', 'Da', CURRENT_TIMESTAMP, '2016-10-17'),
('12345678912','Ivan','Ivanić','Ivan','ivan@gmail.com', '$2a$10$DUGCllBbCOhJD6XyIivHeuaNmTUSqS1Yag6VaUF3cZqjvROT6tCum', 'Da', CURRENT_TIMESTAMP, '2016-10-17');

insert into clanarina (id, naziv, tip, cijena)
values (1, 'XL Plavi centar', 'Neograničeno vježbanje do 15h', '180 kuna'),
(2, 'XL Plavi centar', 'Neograničeno vježbanje', '240 kuna'),
```

Prilog 5.10. Datoteka fitness.sql prvi dio

```
insert into clan (oib, ime, prezime, korisnicko_ime, email, password, clanarina_placena, datum_zahtjeva, datum_aktivacije)
values ('12345678910','Strahinja','Negovanović', 'Strale', 'strahinja944@gmail.com', '$2a$10$6v2LiSWqCSCrr/CVdRmXpunClLCJ.hLqI8NakbyyVbheLt8DyCLGa', 'Da', CURRENT_TIMESTAMP, '2016-10-17'),
('12345678911','Marko','Marković','Marko','marko@gmail.com', '$2a$10$aHpJUA5K01cS1yiFPy2gJ.aTGdHqJZVUe5mItPVhMt37t.CQGmhsq', 'Da', CURRENT_TIMESTAMP, '2016-10-17'),
('12345678912','Ivan','Ivanić','Ivan','ivan@gmail.com', '$2a$10$DUGCllBbCOhJD6XyIivHeuaNmTUSqS1Yag6VaUF3cZqjvROT6tCum', 'Da', CURRENT_TIMESTAMP, '2016-10-17');

insert into clanarina (id, naziv, tip, cijena)
values (1, 'XL Plavi centar', 'Neograničeno vježbanje do 15h', '180 kuna'),
(2, 'XL Plavi centar', 'Neograničeno vježbanje', '240 kuna'),
(3, 'XL Plavi centar', 'Vježbanje 3x tjedno', '220 kuna'),
(4, 'XL Plavi centar', 'Neograničeno vježbanje - mladi do 23. godine', '220 kuna'),
(5, 'XL Plavi centar', 'Neograničeno vježbanje 3 mjeseca', '450 kuna'),
(6, 'XL Plavi centar', 'Neograničeno vježbanje 12 mjeseca', '1800 kuna'),
(7, 'XL Plavi centar', 'Mjesečna članarina 2x tjedno (grupni treninzi)', '230 kuna'),
(8, 'XL Plavi centar', 'Mjesečna članarina 3x tjedno (grupni treninzi)', '250 kuna'),
(9, 'XL Plavi centar', 'Mjesečna članarina 6x tjedno (grupni treninzi)', '300 kuna'),
(10, 'XL Zeleni centar', 'Neograničeno vježbanje do 15h', '180 kuna'),
(11, 'XL Zeleni centar', 'Neograničeno vježbanje', '260 kuna'),
(12, 'XL Zeleni centar', 'Vježbanje 3x tjedno', '240 kuna'),
(13, 'XL Zeleni centar', 'Neograničeno vježbanje - mladi do 23. godine', '220 kuna'),
(14, 'XL Zeleni centar', 'Neograničeno vježbanje 3 mjeseca', '450 kuna'),
(15, 'XL Zeleni centar', 'Neograničeno vježbanje 12 mjeseca', '1800 kuna'),
(16, 'XL Zeleni centar', 'Mjesečna članarina 2x tjedno (grupni treninzi)', '230 kuna'),
(17, 'XL Zeleni centar', 'Mjesečna članarina 3x tjedno (grupni treninzi)', '250 kuna'),
(18, 'XL Zeleni centar', 'Mjesečna članarina 6x tjedno (grupni treninzi)', '300 kuna');

INSERT INTO `status_clanarine` (`clan`, `clanarina`) VALUES ('1', '1'), ('2', '2'), ('3', '3');

insert into galerija (putanja, redoslijed)
values ('images/pozadina3.jpg', 1),
('images/pozadina4.jpg', 2),
('images/pozadina5.jpg', 3);
```

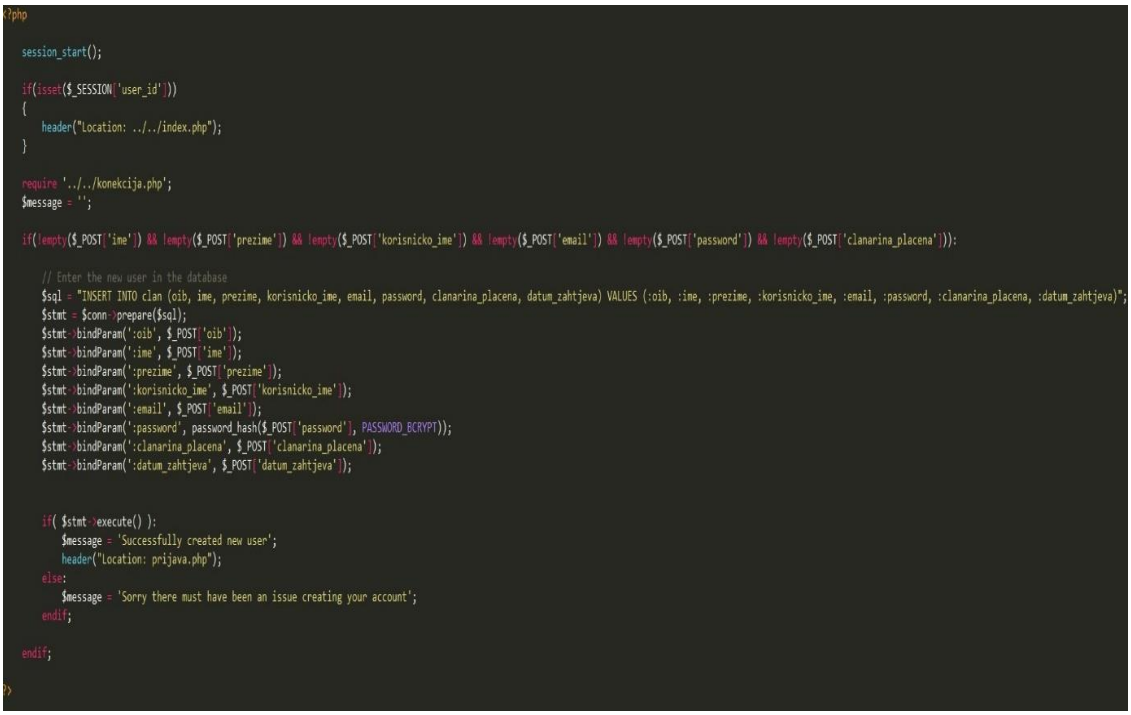
Prilog 5.11. Datoteka fitness.sql drugi dio

```php
<?php

class Postavke
{
    public static $HRVATSKI_JEZIK=1;
    public static $ENGLESKI_JEZIK=2;
}

switch ($_SERVER["SERVER_NAME"])
{
    case 'localhost':
        $putanjaAPP="/Fitness/";
        break;
    case 'byethost32.com':
        $putanjaAPP="/fitnesswp13.byethost32.com/";
        break;
    default:
        $putanjaAPP="/";
    break;
}

$naslovAPP = "XL fitness centar";
$razvoj=$_SERVER["SERVER_NAME"]=="localhost";

include_once 'funkcije.php';

$mysql_host = "sql201.byethost15.com";
$mysql_database = "b15_18879166_fitnes";
$mysql_user = "b15_18879166";
$mysql_password = "fitneshosting";

try{
$veza = new PDO(
"mysql:dbname=" .   $mysql_database . //1. parametar PDO
";host=" .          $mysql_host     . //1. parametar PDO
";charset=utf8"     /*1. parametar PDO */   ,
$mysql_user             , //2. parametar PDO
$mysql_password         , //3. parametar PDO
array(                  //4. parametar PDO
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,//4. parametar PDO
    PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"//4. parametar PDO
    )//4. parametar PDO
);

}catch(PDOException $e){
```

Prilog 5.12. Datoteka konfiguracija.php prvi dio

```php
$mysql_host = "sql201.byethost15.com";
$mysql_database = "b15_18879166_fitnes";
$mysql_user = "b15_18879166";
$mysql_password = "fitneshosting";

try{
$veza = new PDO(
"mysql:dbname=" .   $mysql_database . //1. parametar PDO
";host=" .          $mysql_host     . //1. parametar PDO
";charset=utf8"     /*1. parametar PDO */   ,
$mysql_user             , //2. parametar PDO
$mysql_password         , //3. parametar PDO
array(                  //4. parametar PDO
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,//4. parametar PDO
    PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8"//4. parametar PDO
    )//4. parametar PDO
);

}catch(PDOException $e){

//d($e);
switch ($e->getCode()) {
    case 2002:
        $g= "Ne mogu se spojiti na bazu (Da li je server pokrenut i dobro navedena adresa?)";
        break;
    case 1045:
        $g= "Ne mogu se spojiti na bazu (Neispravno korisničko ime ili lozinka)";
        break;
    case 1049:
        $g= "Ne mogu se spojiti na bazu (Ime baze neispravno)";
        break;
    default:
        $g= "Ne mogu se spojiti na bazu (Kontaktirajkte programera)";
        break;
}
header("location: greska.php?g=" . $g);
}
```

Prilog 5.13. Datoteka konfiguracija.php drugi dio



```php
<?php

   session_start();

   if(isset($_SESSION['user_id']))
   {
       header("Location: ../../index.php");
   }

   require '../../konekcija.php';
   $message = '';

   if(!empty($_POST['ime']) && !empty($_POST['prezime']) && !empty($_POST['korisnicko_ime']) && !empty($_POST['email']) && !empty($_POST['password']) && !empty($_POST['clanarina_placena'])):

       // Enter the new user in the database
       $sql = "INSERT INTO clan (oib, ime, prezime, korisnicko_ime, email, password, clanarina_placena, datum_zahtjeva) VALUES (:oib, :ime, :prezime, :korisnicko_ime, :email, :password, :clanarina_placena, :datum_zahtjeva)";
       $stmt = $conn->prepare($sql);
       $stmt->bindParam(':oib', $_POST['oib']);
       $stmt->bindParam(':ime', $_POST['ime']);
       $stmt->bindParam(':prezime', $_POST['prezime']);
       $stmt->bindParam(':korisnicko_ime', $_POST['korisnicko_ime']);
       $stmt->bindParam(':email', $_POST['email']);
       $stmt->bindParam(':password', password_hash($_POST['password'], PASSWORD_BCRYPT));
       $stmt->bindParam(':clanarina_placena', $_POST['clanarina_placena']);
       $stmt->bindParam(':datum_zahtjeva', $_POST['datum_zahtjeva']);


       if( $stmt->execute() ):
          $message = 'Successfully created new user';
          header("Location: prijava.php");
       else:
          $message = 'Sorry there must have been an issue creating your account';
       endif;

   endif;

?>
```

Prilog 5.14. Datoteka registracija.php

```php
<?php

    session_start();

    if(isset($_SESSION['user_id']))
    {
        header("Location: ../../index.php");
    }

    require '../../konekcija.php';

    if(!empty($_POST['email']) && !empty($_POST['password'])):

        $records = $conn->prepare('SELECT id, email, password FROM clan WHERE email= :email');
        $records->bindParam(':email', $_POST['email']);
        $records->execute();
        $results = $records->fetch(PDO::FETCH_ASSOC);

        $message = '';

        if(count($results) > 0 && password_verify($_POST['password'], $results['password']))
        {
            $_SESSION['user_id'] = $results['id'];
            header("Location: ../../index.php");
        }
        else
        {
            $message = 'Ne odgovaraju podaci za prijavu.';
        }

    endif;

?>
```

Prilog 5.15. Datoteka prijava.php

```php
<?php

    session_start();
    require 'konekcija.php';

    if(isset($_SESSION['user_id']))
    {
        $records = $conn->prepare('SELECT id, email, password FROM clan WHERE id= :id');
        $records->bindParam(':id', $_SESSION['user_id']);
        $records->execute();
        $results = $records->fetch(PDO::FETCH_ASSOC);

        $user = NULL;

        if(count($results)> 0)
        {
            $user = $results;
        }
    }

?>
```

Prilog 5.16. Datoteka index.php prvi dio

```php
<!DOCTYPE html>
<html lang="hr">
  <head>
    <?php include_once 'predlozak/head.php'; ?>
  </head>

  <body>
    <?php include_once 'predlozak/izbornik.php'; ?>

    <?php if(!empty ($user)): ?>

      <div class="container">

        <div class="jumbotron">
         <div class="naslov"><h1 class="animated tada"><?php echo $naslovAPP; ?></h1></div>

         <div align="center">
            <?php if($_SESSION['user_id'] == 1):  ?>
                Admin:
            <?php else: ?>
            <p>Prijavljen: <?= $user['email']; ?></p>
            <?php endif;    ?>
            <?php if($_SESSION['user_id'] == 1):  ?>
            <br />
            <a href="clanovi.php" class="odjava">Prikaži članove</a><br />
            <a href="prikaz.php" class="odjava">Prikaz svih tablica</a><br />
            <?php else: ?>
            <a href="korisnici.php" class="odjava">Prikazi članove</a><br />
            <?php endif;    ?>
            <a href="CRUD/clan/odjava.php" class="odjava">Odjava</a>
         </div>

        </div>
```

Prilog 5.17. Datoteka index.php drugi dio

```php
<?php
    require_once ('konekcija.php');
    require_once ('konfiguracija.php');
    include 'predlozak/head.php';

    $sql = 'SELECT * FROM clan ORDER BY id';
    $result = $conn->query($sql,PDO::FETCH_ASSOC);
    $users = $result->fetchAll();

    $i = 0;
?>
<!DOCTYPE html>
<head>
    <?php include_once 'predlozak/head.php'; ?>
</head>
<body class="prikaz">
    <?php include_once 'predlozak/izbornik.php'; ?>
    <div class="container">
        <h3 class="sve_tablice">Tablica član:</h3>
        <table class="table table-bordered">
            <thead>
            <tr>
                <th>Ime</th>
                <th>Prezime</th>
                <th>Email</th>
                <th>Datum zahtjeva</th>
            </tr>
            </thead>
            <tbody>
            <?php
            try{

                $izraz = $veza->query("select * from clan");
                $izraz->execute();
                $rezultati=$izraz->fetchAll(PDO::FETCH_OBJ);
            ?>
            <?php foreach ($rezultati as $red): ?>
                <tr>
                    <td><?php echo $red -> ime ?></td>
                    <td><?php echo $red -> prezime ?></td>
                    <td><?php echo $red -> email ?></td>
                    <td><?php echo $red -> datum_zahtjeva ?></td>
                </tr>
            <?php endforeach; ?>

            <?php } catch(PDOException $e){
```

Prilog 5.18. Datoteka korisnici.php

```php
<div class="container">

    <h3 class="sve_tablice">Tablica član:</h3>
    <table class="table table-bordered">
        <thead>
        <tr>
            <th>ID</th>
            <th>Oib</th>
            <th>Ime</th>
            <th>Prezime</th>
            <th>Korisničko ime</th>
            <th>Članarina plaćena</th>
            <th>Datum zahtjeva članarine</th>
            <th>Datum aktivacije članarine</th>
            <th>Naziv članarine</th>
            <th>Tip članarine</th>
        </tr>
        </thead>
        <tbody>

        <?php

        try{

        $izraz = $veza->query("SELECT clan.id, clan.oib, clan.ime, clan.prezime, clan.korisnicko_ime, clan.clanarina_placena, clan.datum_zahtjeva, clan.datum_aktivacije, clanarina.naziv,
                        clanarina.tip FROM clan, status_clanarine, clanarina WHERE clan.id = status_clanarine.clan AND status_clanarine.clanarina = clanarina.id order by clan.id ");
            $izraz->execute();
            $rezultati=$izraz->fetchAll(PDO::FETCH_OBJ);
        ?>

        <?php foreach ($rezultati as $red): ?>
        <tr>
            <td><?php echo $red -> id ?></td>
            <td><?php echo $red -> oib ?></td>
            <td><?php echo $red -> ime ?></td>
            <td><?php echo $red -> prezime ?></td>
            <td><?php echo $red -> korisnicko_ime ?></td>
            <td><?php echo $red -> clanarina_placena ?></td>
            <td><?php echo $red -> datum_zahtjeva ?></td>
            <td><?php echo $red -> datum_aktivacije ?></td>
            <td><?php echo $red -> naziv ?></td>
            <td><?php echo $red -> tip ?></td>
        </tr>
       <?php endforeach; ?>

        <?php } catch(PDOException $e){
```

Prilog 5.19. Datoteka clanovi.php

```php
<div class="container">
         <h3 class="sve_tablice">Tablica član:</h3>
         <table class="table table-bordered">

             <thead>
             <tr>
                 <th>ID</th>
                 <th>Ime</th>
                 <th>Prezime</th>
                 <th>Email</th>
                 <th>Članarina plaćena</th>
                 <th>Datum aktivacije članarine</th>
                 <th>Akcija</th>
             </tr>
             </thead>

             <tbody>

             <?php

             try{

                 $izraz = $veza->query("select * from clan");
                 $izraz->execute();
                 $rezultati=$izraz->fetchAll(PDO::FETCH_OBJ);
             ?>

             <?php foreach ($rezultati as $red): ?>
                 <tr>
                     <td><?php echo $red -> id ?></td>
                     <td><?php echo $red -> ime ?></td>
                     <td><?php echo $red -> prezime ?></td>
                     <td><?php echo $red -> email ?></td>
                     <td><?php echo $red -> clanarina_placena ?></td>
                     <td><?php echo $red -> datum_aktivacije ?></td>
                     <td><a href="CRUD/clan/promjeniclana.php?id=<?php  echo $red->id ?>">Promjeni</a> | <a href="CRUD/clan/obrisi.php?id=<?php  echo $red->id ?>">Obriši</a></td>
                 </tr>
             <?php endforeach; ?>

             <?php } catch(PDOException $e){
                 d($e);
                 }
             ?>

             </tbody>
         </table>
```

Prilog 5.20. Datoteka prikaz.php prvi dio

```
<h3 class="sve_tablice">Tablica članarina</h3>
<table class="table table-bordered">

      <thead>
      <tr>
          <th>ID</th>
          <th>Naziv</th>
          <th>Tip</th>
          <th>Cijena</th>
          <th>Akcija</th>
      </tr>
      </thead>

      <tbody>

      <?php

      try{

          $izraz = $veza->query("select * from clanarina order by id");
          $izraz->execute();
          $rezultati=$izraz->fetchAll(PDO::FETCH_OBJ);
      ?>

      <?php foreach ($rezultati as $red2): ?>
          <tr>
              <td><?php echo $red2 -> id ?></td>
              <td><?php echo $red2 -> naziv ?></td>
              <td><?php echo $red2 -> tip ?></td>
              <td><?php echo $red2 -> cijena ?></td>
              <td><a href="CRUD/clanarina/promjeni.php?id=<?php  echo $red2->id ?>">Promjeni</a> | <a href="CRUD/clanarina/obrisiclanarinu.php?id=<?php  echo $red2->id ?>">Obriši</a></td>
          </tr>
      <?php endforeach; ?>

      <?php } catch(PDOException $e){
          d($e);
          }
      ?>
    </tbody>
   </table>

  <div class="ubaci"><a href="CRUD/clanarina/insertc.php"><button type="button" class="btn btn-sm btn-primary gumb2">Ubaci</button></a></div>
```

Prilog 5.21. Datoteka prikaz.php drugi dio

```
  <form action="insertc.php" method="post">
      <fieldset>
          <label for="naziv" class="lime">Naziv clanarine:</label><input id="naziv" type="text" placeholder="Upišite naziv clanarine!" name="naziv"/>
          <label for="tip" class="lime">Tip clanarine:</label><input id="tip" type="text" placeholder="Upišite tip clanarine!" name="tip"/>
          <label for="cijena" class="lime">Cijena clanarine:</label><input id="cijena" type="text" placeholder="Upišite cijenu clanarine!" name="cijena"/>
          <input type="submit" name="clanarina">
      </fieldset>
  </form>
</div>
```

Prilog 5.22. *HTML* forma za datoteku insertc.php

```php
<?php

    session_start();

    require '../../konekcija.php';

    $message = '';

    if(!empty($_POST['naziv']) && !empty($_POST['tip']) && !empty($_POST['cijena'])):

        // Enter the new user in the database
        $sql = "INSERT INTO clanarina (naziv, tip, cijena) VALUES (:naziv, :tip, :cijena)";
        $stmt = $conn->prepare($sql);
        $stmt->bindParam(':naziv', $_POST['naziv']);
        $stmt->bindParam(':tip', $_POST['tip']);
        $stmt->bindParam(':cijena', $_POST['cijena']);


        if( $stmt->execute() ):
            $message = 'Successfully created new user';
            header("Location: ../../prikaz.php");
        else:
            $message = 'Sorry there must have been an issue creating your account';
        endif;

    endif;

?>
```

Prilog 5.23. Datoteka insertc.php