Model i programska podrška za rangiranje studenata pri upisu u studentski dom

Podunavac, Ivor

Master's thesis / Diplomski 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:200:975329

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij

MODEL I PROGRAMSKA PODRŠKA ZA RANGIRANJE STUDENATA PRI UPISU U STUDENTSKI DOM

Diplomski rad

Ivor Podunavac

Osijek, 2017.

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

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 12.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Ivor Podunavac
Studij, smjer:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D 696 R, 09.10.2014.
OIB studenta:	49074498720
Mentor:	Izv.prof.dr.sc. Dominika Crnjac Milić
Sumentor:	Izv. prof. dr. sc. Krešimir Nenadić
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Alfonzo Baumgartner
Član Povjerenstva:	Izv. prof. dr. sc. Krešimir Nenadić
Naslov diplomskog rada:	Model i programska podrška za rangiranje studenata pri upisu u studentski dom
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak diplomskog rada:	Zadatak ovog rada je izraditi web aplikaciju koja bi omogućila prijavu potencijalnih kandidata za korištenje smještaja u studentske domove, te rangiranje prijavljenih na osnovi čega bi bila izrađena izlazna lista budućih korisnika smještaja uz određene pripadajuće mjesečne naknade. Rangiranje bi bilo izvršeno na osnovi prosjeka ocjena iz srednje škole ili fakulteta, ECTS bodova, godina fakulteta, materijalnog stanje,itd. (sumentor: Krešimir Nenadić)
Prijedlog ocjene pismenog dijela ispita (diplomskog 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:	12.09.2017.

	Potpis:
	Datum: 12.09.2017.

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 28.09.2017.

Ime i prezime studenta:	Ivor Podunavac
Studij:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D 696 R, 09.10.2014.
Ephorus podudaranje [%]:	17%

Ovom izjavom izjavljujem da je rad pod nazivom: **Model i programska podrška za rangiranje studenata pri upisu u studentski dom**

izrađen pod vodstvom mentora Izv.prof.dr.sc. Dominika Crnjac Milić

i sumentora Izv. prof. dr. sc. Krešimir Nenadić

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:

Ivor Podunavac

SADRŽAJ

1. UVOD 1
1.1. ZADATAK DIPLOMSKOG RADA 2
2. ANALIZA RADA STUDENTSKIH CENTARA I PREDNOSTI RADA PUTEM APLIKACIJE 3
3. KORIŠTENE TEHNOLOGIJE 5
3.1. OPISNI JEZIK HTML – HYPERTEXT MARKUP LANGUAGE 5
3.1.1. Inačice HTML-a 5
3.1.2. Struktura HTML dokumenta 6
3.2. STILSKI JEZIK CSS – CASCADING STYLE SHEETS 7
3.2.1. Sintaksa CSS-a 7
3.3. SKRIPTNI JEZIK PHP – HYPERTEXT PREPROCESSOR 8
3.3.1. Sintaksa PHP-a 9
3.4. MYSQL SUSTAV ZA UPRAVLJANJE BAZAMA PODATAKA 9
3.5. BOOTSTRAP RAZVOJNI OKVIR 10
3.6. SKRIPTNI PROGRAMSKI JEZIK JAVASCRIPT 11
4. IZRADA WEB APLIKACIJE 12
4.1. BAZA PODATAKA – SKUP SVIH TABLICA KOJE SE KORISTE U APLIKACIJI 12
4.2. REGISTRACIJA KORISNIKA USLUGE 13
4.3. PRIJAVA NA WEB APLIKACIJU 15
4.4. KORISNIČKI REŽIM RADA NA CENTRALNOM DIJELU APLIKACIJE 17
4.4.1. Unos osobnih podataka 18
4.4.2. Postavljanje dokumenata na aplikaciju 20
4.4.3. Stranica „ Smještaj“ 22
4.5. REZULTATI UPISA 22
4.6. MOGUĆNOSTI RADA ADMINISTRATORA U APLIKACIJI 26
4.6.1. Postavljanje opcija 26
4.6.2. Provjera podataka u Listi prijavljenih kandidata 27
4.6.3. Uvid u Rang listu prijavljenih kandidata 29
5. ZAKLJUČAK 30
LITERATURA 31
SAŽETAK 32
ŽIVOTOPIS 34
PRILOZI 35

1. UVOD

Tema ovog diplomskog rada je izrada web aplikacije koja omogućava prijavu te rangiranje studenata pri upisu u studentski dom. Rad se sastoji od dva dijela: praktični dio koji podrazumijeva izradu web aplikacije te dokumentacija o izrađenoj aplikaciji.

Danas je korištenje web aplikacija dostupno svima, na svakom računalu ili mobilnom telefonu stoga ne čudi značajan napredak u razvoju istih. One su postale neizostavan oblik zadovoljstva svojim korisnicima. Pristupa im se putem jednog od dostupnih internet preglednika.

Kroz cijeli mjesec srpanj studenti imaju mogućnost prijave u studentski dom. Studenti prve i viših godina studija prijavnice predaju osobno, dok studenti koji će tek upisati prvu godinu studija prijavnice mogu poslati i poštom. Dakle, ukoliko se želi upisati u dom, prvo je potrebno prijavnicu podići u domu, te je ispuniti i priložiti uz ostale tražene dokumente. U prijavnici se ispunjavaju sljedeći podaci: ime i prezime studenta, naziv visokog učilišta, ukupan broj stečenih ECTS bodova, broj stečenih ECTS bodova u tekućoj akademskoj godini, jesu li položeni svi kolegiji u tekućoj akademskoj godini te pripada li studijski program znanstvenom polju koje se vodi kao deficitarno na nacionalnoj razini. Kroz ovaj rad i web aplikaciju, svi ti traženi uvjeti su pristupačni na jednom mjestu i prijava može biti gotova kroz par minuta uz nekoliko klikova mišem.

U drugom dijelu rada obrađena je analiza rada studentskih centara na temelju dosadašnjeg načina rada kod upisivanja studenata u domove. Nakon toga opisane su tehnologije koje su potrebne za izvršenje istoga. U četvrtom dijelu rada je slikama i objašnjenjem pojedinih dijelova koda prikazan način stvaranja aplikacije te način njezine primjene, a na kraju i željeni rezultat radi kojeg je stvarana aplikacija.

Radom je dana analiza dosadašnjeg načina rada za apliciranje na smještaj u studentske domove, te kontroliranje radnih učinaka vezanih za obradu podataka u dosadašnjem načinu prijave za smještaj u studentski dom. Ukazane su dobrobiti korištenja ovakve aplikacije koja se može pokazati kao značajna potpora IT-ja u poslovanju učeničkih i studentskih domova. Za razliku od dosadašnje prijave u studentske domove, ovo je mnogo pristupačnije, jeftinije i odrađuje se u kratkom vremenskom roku. Dovoljno je imati pristup internetu, te se kroz koju minutu završi cjelokupna prijava.

1.1. Zadatak diplomskog rada

Zadatak ovog rada je izraditi web aplikaciju koja bi omogućila prijavu potencijalnih kandidata za korištenje smještaja u studentskim domovima, te rangiranje prijavljenih na osnovi čega bi bila izrađena izlazna lista budućih korisnika smještaja uz određene pripadajuće mjesečne naknade. Rangiranje bi bilo izvršeno na osnovi prosjeka ocjena iz srednje škole ili fakulteta, ECTS bodova, godina fakulteta, materijalnog stanje, itd.

2. ANALIZA RADA STUDENTSKIH CENTARA I PREDNOSTI RADA PUTEM APLIKACIJE

U Hrvatskoj trenutno postoji 16 studentskih domova smještenih u devet gradova. Najmanji broj ležaja ima studentski dom u Slavonskom Brodu – 117, a najveći je studentski dom „Stjepan Radić“ u Zagrebu s 4014 kreveta. Zbrojem svih dostupnih kreveta u šesnaest domova dobiva se ukupan kapacitet od 11709. To je broj studenata koji svake godine stanuju u domovima diljem Hrvatske. Ukupan broj prijava za upis u studentski dom svake godine kreće se oko 35-40% iznad dostupnog kapaciteta. Svakog ljeta u Republici Hrvatskoj između 15 i 17 tisuća studenata prijavljuje se za upis u dom, noseći razne dokumente na adresu doma u kojeg žele biti upisani, neovisno o mjestu iz kojeg moraju ići na upis.

Kroz rad je dana analiza i rješenje prijave za smještaj u studentske domove u gradu Osijeku.

Pravo sudjelovanja u natječaju imaju redoviti studenti koji su hrvatski državljani i državljani EU-a s prijavljenim boravkom u Republici Hrvatskoj koji imaju odobren status stranca na stalnome boravku u Republici Hrvatskoj, a upisani su na Sveučilištu J. J. Strossmayera u Osijeku na sveučilišnim i stručnim studijima (sadašnji redoviti studenti i redoviti studenti koji će upisati prvu godinu studija akademske godine 2015./2016.) i studenti s invaliditetom poslijediplomskih sveučilišnih studija.

Pravo sudjelovanja u natječaju nemaju redoviti studenti kojima je mjesto prebivališta u Osijeku. Pravo na subvencionirani smještaj ne mogu ostvariti: studenti koji su ostvarili manje od 18 ECTS bodova u ak. god. 2015./2016 i manje od 45 ECTS bodova prosječno u prethodnim godinama studiranja osim studenata koji upisuju prvu godinu studija i studenata invalida od 1.do 5. kategorije invalidnosti, studenti koji nisu upisani u statusu redovitoga studenta u ak. god. 2015./2016., studenti koji mijenjaju studij (bez završetka studija) i na drugome studiju ponovno upisuju prvu godinu, ako se prijave s prosjekom ocjena iz srednje škole.[1]

Trenutno studentski domovi u Republici Hrvatskoj ne koriste ovakav način upisivanja studenata kakav je prikazan ovim radom. Dakle, svi studenti koji idu na upis trebaju donijeti tražene podatke o sebi na mjesto koje studentski centar predloži, a većinom se to odnosi na sam studentski dom. Tu se javlja problem što studenti iz udaljenih mjesta moraju trošiti novac i vrijeme. Mogućnost smještaja prijaviteljima ne mora na kraju biti osigurana, jer je kapacitet smještaja ograničen. Nakon što to i naprave prilaganjem svih potrebnih dokumenata, kreće proces upisivanja. Tada zaposlenici studentskog centra prepisuju u svoj program s mnoštva dokumenata redom tražene podatke. Upravo se tu vidi razlika u efikasnosti i ekonomičnosti između dosadašnjeg načina upisa i upisa predloženog ovim radom. Posao koji rade zaposlenici, mogli bi

raditi i studenti iz svoga doma, a zaposlenici bi samo trebali provjeriti dokumente koje su im studenti poslali kroz aplikaciju kako ne bi bilo lažiranja pojedinih podataka u prijavitelju (prosjeci, invalidnost, razina studija, itd.).

3. KORIŠTENE TEHNOLOGIJE

U radu su korištene tehnologije: Cascading Style Sheets (CSS) i HyperText Markup Language (HTML) za prikaz izgleda web aplikacije, Hypertext Preprocessor (PHP) skriptni jezik pomoću kojeg su izrađeni algoritmi za računanje bodova te samo računanje bodova, JavaScript za manipulaciju na strani korisnika (npr. klikom na neko dugme ne osvježi se stranica i ne šalju se podaci na server, nego se operacija izvodi na strani korisnika/klijenta) i sustav za upravljanje bazom podataka (MySQL).

3.1. Opisni jezik HTML – HyperText Markup Language

HTML dolazi od engleskih riječi HyperText Markup Language. Hypertext je dio teksta koji ima ulogu poveznice (linka), a Markup Language označava način unosa izgleda informacija unutar dokumenta. HTML je jednostavan za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i tome što je od početka bio zamišljen kao besplatan i tako dostupan svima. Prikaz hipertekst dokumenta omogućuje web preglednik.[2] Osnovna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertext dokument. Pri tome se nastoji da taj dokument izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ. HTML nije programski jezik niti su ljudi koji ga koriste programeri. On služi za opis hipertekstualnih dokumenata.

3.1.1. Inačice HTML-a

U početcima Interneta, jedini način prijenosa informacije do korisnika bio je tekst. Prva inačica HTML nastala je 1990. godine. Pojavom CSS-a, HTML-u je vraćena prvotna namjena – označavanje, tj. određivanje, sadržaja, a ne njegovog izgleda. Time je u potpunosti omogućeno odvajanje sadržaja od dizajna. [3]

Novije inačice HTML-a su:

HTML 4.01	1999.
XHTML 1.0	2000.
HTML5	2012.
XHTML5	2013.

3.1.2. Struktura HTML dokumenta

HTML datoteke su zapravo obične tekstualne datoteke, ekstenzija im je *.html* ili *.htm*. Svaki HTML dokument sastoji se od osnovnih građevnih blokova - HTML elemenata. Svaki HTML element sastoji se od para HTML oznaka . Također, svaki element može imati i atribute kojim se definiraju svojstva tog elementa. Na početku HTML dokumenta preporučljivo je postaviti <!DOCTYPE> element, kojim se označava DTD (engl. *Document Type Definition*), čime se definira točna inačica standarda koja se koristi za izradu HTML dokumenta. [2]

S <html> elementom označava se početak HTML dokumenta. Unutar <html> elementa nalazi se <head> element, te <body> element. Element <head> predstavlja zaglavlje HTML dokumenta u kojemu se najčešće specificiraju jezične značajke HTML dokumenta kao i sam naslov stranice. Pomoću određenih HTML elemenata unutar zaglavlja dodaju se i stilska obilježja stranice, bila ona direktno ugrađena ili dodana kao referenca na vanjsku CSS datoteku. Unutar zaglavlja često se još definiraju i skripte kreirane u JavaScript jeziku. U <body> elementu kreira se sadržaj HTML dokumenta, odnosno, stranice koju on reprezentira. Svaka HTML oznaka počinje znakom < (manje od), a završava znakom > (više od). Zatvarajuća HTML oznaka kreira se na isti način kao i otvarajuća, ali se prije završnog znaka < dodaje kosa crta /.



Sl.3.1. Struktura jednostavnog HTML dokumenta

3.2. Stilski jezik CSS – Cascading Style Sheets

CSS je kratica od engleskog naziva Cascading Style Sheets. Radi se o stilskom jeziku, koji se rabi za opis prezentacije dokumenta napisanog pomoću *markup* (HTML) jezika. [4]
Kako se web razvijao, tako su u HTML ubacivani elementi za definiciju prezentacije, ali je dovoljno brzo uočena potreba za stilskim jezikom koji će HTML osloboditi potrebe prikazivanja sadržaja i njegovog oblikovanja (čemu danas služi CSS). CSS-om se uređuje sam izgled i raspored stranice. Pomoću CSS-a definira se kako će se neki HTML element prikazati (slike, slova, linkovi itd.). Stilovi se obično spremaju u posebne datoteke izvan HTML dokumenta. Odvajanje CSS-a od HTML-a poboljšava dostupnost sadržaja, pruža veću fleksibilnost i kontrolu kod prikazivanja obilježja. Omogućava se dijeljenje istog stila oblikovanja između više html dokumenata kreiranjem .css datoteke koja se pozove u svakom html dokumentu.

3.2.1. Sintaksa CSS-a

Sintaksa CSS-a se razlikuje od sintakse HTML-a. Sastoji se samo od tri dijela:

```
1  selektor {
2      svojstvo : vrijednost;
3  }
```

Selektor je html element kojem se želi dodijeliti vrijednost svojstva. Svojstvo je točno predefinirana oznaka koja govori koje svojstvo se mijenja, a vrijednost je stil koji se dodjeljuje svojstvu. Svaki selektor može imati više svojstava. Svojstvo i vrijednost su odvojeni dvotočkom i nalaze se unutar vitičastih zagrada. Iza vrijednosti svojstva nalazi se graničnik točka-zarez koji odvaja svojstva. Višestruke vrijednosti jednog svojstva odvojene su zarezom, a ako vrijednost svojstva ima više od jedne riječi stavlja se unutar dvostrukih navodnika.[5]

```css
.bg-img
{
    position:absolute;
    width:100%;
    height:100%;
    background-size: cover;
    background-repeat: no-repeat;
    -webkit-filter: blur(3px);
    -moz-filter: blur(3px);
    -o-filter: blur(3px);
    -ms-filter: blur(3px);
    filter: blur(2px);
}

.stari
{
    background-image: url('stari.gif');
}

.novi
{
    background-image: url('novi.png');
}

#forma
{
    margin-top:10%;
}
```

Sl. 3.2. Primjer CSS datoteke

3.3. Skriptni jezik PHP – Hypertext Preprocessor

Kratica PHP dolazi od skraćenice Hypertext Preprocessor. To je *open-source* skriptni jezik za dinamičko generiranje HTML koda. Open-source u gornjoj definiciji znači da svatko tko želi može skinuti izvorne PHP kodove pisane u C-u i, ukoliko ih razumije, može ih mijenjati po svojoj volji te dodavati nove funkcije i funkcionalnosti PHP-u. PHP je jedan od najnaprednijih i najkorištenijih poslužiteljskih skriptnih tehnologija danas u upotrebi.[6]

Svojom je sintaksom sličan mnogim drugim sličnim jezicima. Ima istoznačne (iste po sintaksi i funkcionalnosti) funkcije kao i neki drugi jezici kao što su C ili Perl. U prijevodu, to znači da se jedna radnja može izvršiti korištenjem više različitih funkcija.

3.3.1. Sintaksa PHP-a

Cijeli PHP kod nalazi se između `<?php` i `?>` oznaka. Oznaka `<?php` je početak PHP programa, a `?>` predstavlja kraj programa, te se može smjestiti bilo gdje u .php dokumentu. Osim `<?php` za početak PHP blok koda može se koristiti i kraći oblik `<?` , ali to nije preporučljivo jer postoje još neki skriptni jezici koji započinju blok koda na taj način.

U većini slučajeva PHP datoteka sadrži i HTML tagove zbog oblikovanja podataka koji se ispisuju na ekranu kada se npr. posjeti neka stranica. Moguće je vidjeti i PHP datoteke koje ne sadrže HTML tagove već su u njima sadržane nekakve PHP funkcije, klase, vrijednosti varijabli, polja itd.

Dakle, kako bi server uspješno preveo PHP kod u HTML datoteka mora imati nastavak **.php** jer sa nekom drugom ekstenzijom PHP kod neće biti izvršen.

Svaki red PHP koda, odnosno set instrukcija mora završiti s (**;**) točka zarezom. [7]

```php
<?php
include 'conn.php';
$passRepErr=$usernameErr=$passErr=$emptyErr=$registered="";
$korIme=strip_tags($_POST['username']);
$ime=strip_tags($_POST['ime']);
$prezime=strip_tags($_POST['prezime']);
$lozinka=strip_tags($_POST['pwd']);
$lozinkaRep=strip_tags($_POST['pwdRep']);
echo $korIme + "<br />" +$ime;
```

Sl 3.3. Primjer PHP koda

3.4. MySQL sustav za upravljanje bazama podataka

MySQL je besplatan, *open source* sustav za upravljanje bazom podataka. *Open-source* oblik govori da je korištenje dozvoljeno svakome radi svojih potreba. Svaki korisnik Interneta može se služiti s MySQL-om. SQL znači „structured query language“ – strukturni jezik za pretraživanje standardiziran za pristupanje bazama podataka. MySQL baze su relacijskog tipa, koji se pokazao kao najbolji način skladištenja i pretraživanja velikih količina podataka i u biti predstavljaju osnovu svakog informacijskog sustava, tj. temelj svakog poslovnog subjekta koji svoje poslovanje bazira na dostupnosti kvalitetnih i brzih informacija. Osnovi element koji se pohranjuje u bazi naziva se entitet, entitet može biti bilo što: osoba, neki objekt, događaj, služba u nekoj organizaciji i sl., dakle stvari iz stvarnog života o kojima se želi čuvati informacije. Drugi važan pojam u teoriji baza podataka jeste relacija.[8] Kao što u stvarnom životu postoje određeni

međusobni odnosi između dvije ili više osoba, događaja i sl. tako se i u bazama podataka mogu pojaviti određeni odnosi ili relacije između raznih entiteta, koji se na odgovarajući način predstavljaju unutar same baze.

Svaka MySQL baza može imati nekoliko korisnika koji joj mogu pristupiti, a svaki korisnik ima predefinirane mogućnosti za rad, odnosno ovlasti. Ovakav pristup uz dobra podešenja, znatno umanjuje mogućnost greške. MySQL kao sistem može raditi na mnogo različitih operacijskih sustava.[9]

3.5. Bootstrap razvojni okvir

Bootstrap je besplatan razvojni okvir otvorenog koda za izradu web stranica i aplikacija. Sadrži predloške pisane u HTML-u i CSS-u za razne tipove komponenti korisničkog sučelja, kao i različite JavaScript ekstenzije.[10] Dakle to je razvojni okvir za izradu web aplikacija i stranica.

Bio je najveći projekt na GitHub-u. Web dizajnerima omogućuje jednostavnu i brzu izradu responzivne web stranice, tj. stranice koje se automatski prilagođavaju na različite veličine zaslona, od malih kod pametnih telefona do velikih zaslona. Sadrži predloške pisane u HTML-u i CSS-u za razne tipove komponenti korisničkog sučelja, kao i različite JavaScript ekstenzije. Kompatibilan je sa svim vrstama modernih web preglednika. Bootstrap dokumenti (JavaScript, CSS) se mogu uključiti u HTML dokument pomoću linka na CDN (Content Delivery Network) ili linkom na lokalni bootstrap dokument na računalu.[11] Njegova najveća prednost je ta što u sebi ima ugrađen set alata i biblioteka za kreiranje fleksibilnih web obrazaca sa svim pripadajućim elementima. Neke od osnovnih prednosti su što olakšava i ubrzava razvoj, omogućava jednostavniji razvoj web formi za prikazivanje sadržaja na mobilnim telefonima i uređajima, jednostavan za upotrebu itd. Nedostaci su mu što velik broj aplikacija koje koriste bootstrap vizualno podsjećaju jedna na drugu. Mora se učitati svaki put kod učitavanja stranice, pa to dovodi do sporijeg učitavanja same web stranice.

Može se zaključiti da sama upotreba bootstrap-a nije neophodna, ali programeru može pomoći prilikom razvoja web aplikacija.

3.6. Skriptni programski jezik JavaScript

JavaScript je najpopularniji skriptni programski jezik. To je jezik koji je nastao s ciljem dodavanja interaktivnosti HTML stranicama. Podržavaju ga svi poznatiji Internetski preglednici (Mozilla, Internet Explorer, Netscape, Opera). Stvorila ga je kompanija Netscape 1996. godine kako bi se postigla interaktivnost i dinamičnost web stranica. JavaScript je interpreter, što znači da se skripta izvršava odmah naredbu po naredbu, bez prethodnog prevođenja cijelog programa i kreiranja izvršne datoteke. Dakle, ne prevodi se u strojni jezik, tj. binarni kod.

Zbog karakteristike da se izvršava naredbu po naredbu, JavaScript se izvodi na strani korisnika. Javascript se izvodi na strani klijenta i izvodi se naredbu po naredbu bez prethodnog prevođenja cijelog programa i kreiranja izvršne datoteke. JavaScript kod se može pozvati iz HTML dokumenata tako da se nalazi unutar tagova <script> i </script> u head ili body sekciji, ili se može pozvati vanjski JavaScript dokument putem linka.[13] Ona omogućuje pretvaranje dinamičkog teksta u HTML stranicu, kao što je ispis vrijednosti određene varijable, također omogućuje reagiranje na događaje(klikom na određeno dugme), manipulaciji HTML elemenata, validiranje podataka (npr. kod prijave kako bi se smanjio opseg provjere na strani servera) ili kreiranje kolačića kako bi se spremile informacije o korisnikovom računalu. [12]

Biblioteka za JavaScript je jQuery. Ona radi na više sustava (eng. cross-platform), npr. Linux 32-bitnoj ili Windows 64-bitnoj arhitekturi. Sadrži zadatke koji zahtijevaju veći broj naredbi JavaScripta i omotava ih u metode koje se mogu pozvati jednom linijom koda. JQuery biblioteka obuhvaća funkcionalnosti kao što su HTML/DOM manipulacija, CSS manipulacija, efekti, animacije i metode HTML događaja, AJAX. Tri najbitnije stvari koje su izdvojile JQuery u odnosu na konkurenciju su korištenje CSS selektora, nenametljiv JavaScript i jednostavnost upotrebe. JQuery koristi identične CSS selektore koje web dizajneri koriste za dizajniranje web stranica.[14] AJAX (asinkroni JavaScript i XML) omogućuje dohvaćanje podataka s poslužitelja bez osvježavanja stranice. Podatke je moguće dohvaćati samo s poslužitelja s kojeg je dohvaćena stranica. AJAX je skupina međusobno povezanih tehnologija koje se koriste na strani klijenta za razvoj sinkronih i asinkronih web aplikacija. Uporabom AJAX-a, web aplikacije mogu slati i primati podatke sa servera asinkrono, u pozadini bez promjene trenutnog rada i prikaza stranice.

```
$.ajax({
        type: "POST",
        data: {"id": id },
        url:"prihvatiPrijavu.php",
        success:function() {
            location.reload();
        }
});
```

Sl.3.4. Struktura AJAX poziva u jQuery-u.

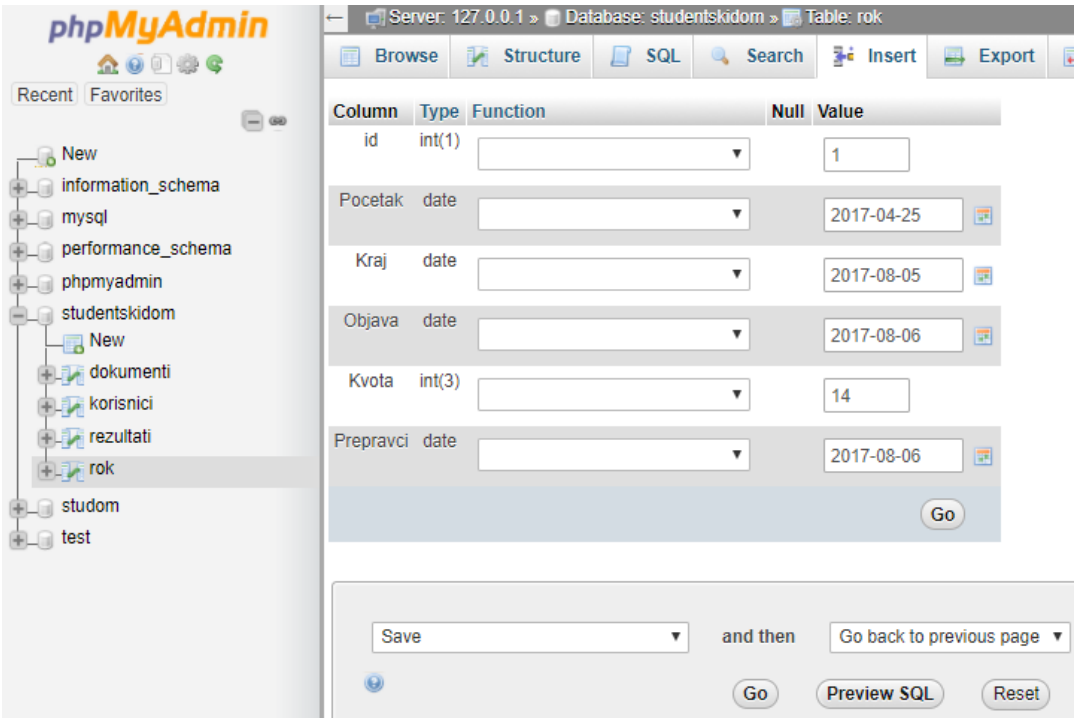
4. IZRADA WEB APLIKACIJE

Nakon što su u prošlom poglavlju objašnjene tehnologije pomoću kojih je ovaj rad napravljen, u ovom poglavlju je opisan način realizacije same web aplikacije. U ovoj aplikaciji postoje dva načina režima rada, a to su korisnički i administratorski režim rada. Svaki od tih načina je podrobnije opisan u sljedećim poglavljima.

4.1. Baza podataka – skup svih tablica koje se koriste u aplikaciji

Baza podataka se tvori u svrhu pohranjivanja i dohvaćanja podataka iz aplikacije. Dohvaćaju se podaci iz tablica koje su smještene u bazi podataka, kako bi se mogli ispisati na aplikaciji (lista čekanja, rang lista). Baza podataka se zove *studentskidom* i u njoj se nalaze 4 tablice (korisnici, rok, rezultati, dokumenti). Tablice *rezultati* i *dokumenti*, svaka ima jedan stupac koji je strani ključ koji referira, odnosno pokazuje na stupac ID u tablici Korisnici. Strani ključ osigurava integritet baze podataka u smislu da se redak koji je primarni ključ iz jedne tablice ne smije obrisati ako on pokazuje na redak u drugoj tablici. Primarni ključ (ID) može biti samo jedan u tablici. Stupac koji ima primarni ključ ne može biti prazan i mora biti jedinstven. Slogovi, tj. redci u tablicama su autoinkrementirani po primarnom ključu (ID) što znači da se kod unosa novog retka u tablicu, *id* povećava za 1. Stupac *idKor* u tablici rezultati je strani ključ koji pokazuje na tablicu korisnici i govori od kojeg korisnika je pojedina prijava. Isto vrijedi i za dokumente, od kojeg korisnika je pojedini dokument.

Postavljanje opcija se provodi u datoteci *postavi.php*. Kada administrator odredi rokove za prijavu, objavu rezultata i kvotu, oni se unose u tablicu *rok*. Za početak i kraj prijave te objavu i prepravke tip podataka je *date*, a za kvotu je *integer*.



Sl.4.1. Tablica *rok* u bazi podataka

Uz ovu tablicu, u bazi se još nalazi tablice *rezultati*, *korisnici* i *dokumenti*. U *rezultati* se spremaju podaci o studentu (razina studija, fakultet, bodovi za pojedine kategorije) te njegov ID (povećava se za jedan kod dodavanja nove prijave) i *idKor*.

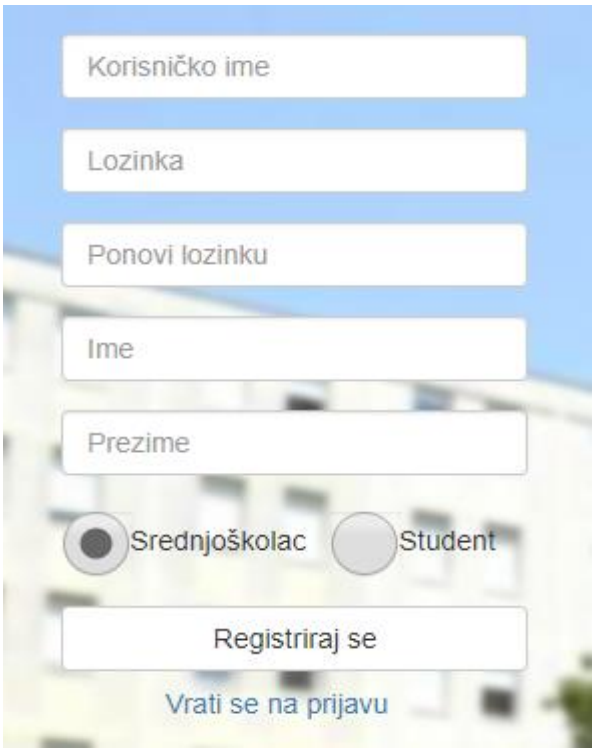
U tablici *korisnici* se nalazi id, ime i prezime, zaporka koja je kriptirana i jedinstveni kod koji je korisnik dobio prilikom prijave kako se prilikom objave rezultata u tablici ne bi znalo čija je pojedina prijava osim osobe čija je prijava.

Tablica *dokumenti* sadrži nazive dokumenata koje je korisnik priložio kako bi se administrator uvjerio u vjerodostojnost podataka. Svaki dokument ima naziv upravo po kodu kojeg je korisnik dobio prilikom prijave.

4.2. Registracija korisnika usluge

Na početku samoga korištenja aplikacije svaki njezin korisnik mora obaviti registraciju. Nakon što pravilno ispuni potrebne podatke, automatski će biti registriran te se tada može prijaviti i koristiti danom aplikacijom. Prilikom registracije svaki korisnik dobije jedinstveni ID koji je za svaku novu registraciju za jedan broj veći, te kod/šifru koja se dobije funkcijom *uniqid()*. Ta funkcija generira niz znakova na temelju trenutnog vremena sustava. Kod registracije korisnik odabire svoj status – srednjoškolac(maturant) ili student. To je bitno radi kasnijeg računanja

bodova, ako je student prosjek mu se dijeli s ukupnim prosjekom svih studenata iste godine, a ako je srednjoškolac s ukupnim prosjekom svih srednjoškolaca.



Sl.4.2. Registracija na aplikaciju

U PHP skripti *registracija.php* sama registracija realizirana je na sljedeći način. *Strip_tags* uklanja HTML znakove iz teksta. Linijom *$korIme=strip_tags($_POST['ime']);* se sprema tekst iz polja *ime* u varijablu *korIme*. To vrijedi za sve ostale varijable u kodu ispod.

```php
<?php
include 'conn.php';
$passRepErr=$usernameErr=$passErr=$emptyErr=$registered="";
$korIme=strip_tags($_POST['username']);
$ime=strip_tags($_POST['ime']);
$status=$_POST['status'];
$prezime=strip_tags($_POST['prezime']);
$lozinka=strip_tags($_POST['pwd']);
$lozinkaRep=strip_tags($_POST['pwdRep']);
```

Sl.4.3. Spremanje podataka iz polja u varijable

Ako su zadovoljeni svi uvjeti iz priloženog koda sa slike 4.4., može se izvršiti registracija. S naredbom *uniqid* kreira se jedinstveni kod temeljen na trenutno vremenu (jedinstveni kod korisnika). Linijom *$hash = password_hash($lozinka, PASSWORD_BCRYPT);* kriptira se zaporka i pretvara u slučajni skup znakova koji vidi administrator u bazi podataka. Linijom ispod unosi se korisnika u tablicu *korisnici*.

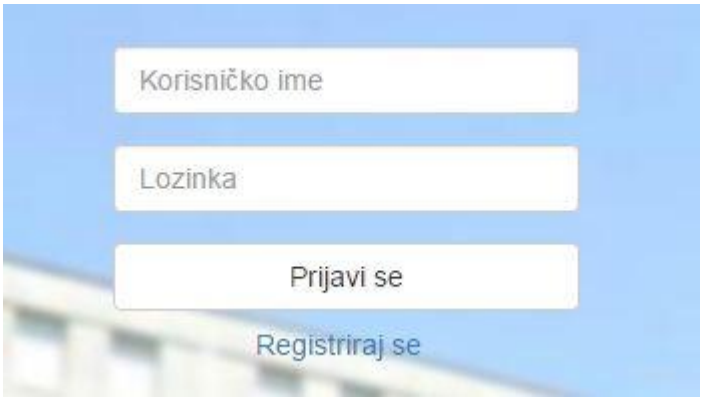
```php
if(isSubmitClicked() && notEmptyFields($korIme, $ime, $prezime, $lozinka, $lozinkaRep) && passChar($lozinka)
   && passLength($lozinka) && passMatch($lozinka, $lozinkaRep) && usernameLength($korIme) && !userExists($conn, $korIme))
{
        $kod = uniqid();
        $hash = password_hash($lozinka, PASSWORD_BCRYPT);
        mysqli_query($conn, "INSERT INTO korisnici(KorIme, Kod, Lozinka, Ime, Prezime, Status, Role) VALUES ('$korIme', '$kod', '$hash', '$ime', '$prezime', '$
            status', NULL)");
        $registered="Uspješno ste registrirani";

}
```

Sl.4.4. Unos korisnika u tablicu *korisnici* u bazi podataka.

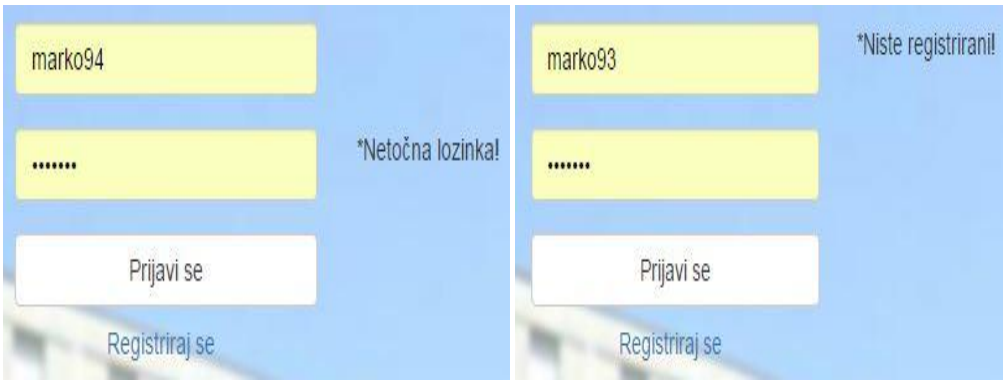
4.3. Prijava na web aplikaciju

Postupak prijave je realiziran putem PHP-a. Otvaranjem početne stranice prikazuje se okvir u kojem, ukoliko je korisnik već registriran, može popuniti tražene podatke te se prijaviti. Ako korisničko ime i zaporka odgovaraju podacima koje je korisnik unio prilikom registracije, on će biti prijavljen na aplikaciju te će imati mogućnost ispunjavanja potrebnih parametara za upis u dom.



Sl. 4.5. Prijava na aplikaciju

Ako korisnik netočno upiše korisničko ime dobit će obavijest kako taj korisnik nije registriran, tj. to ime nije zabilježeno u bazi. U drugom slučaju, ako se pogriješi kod zaporke dobit će obavijest o netočnoj zaporci.



Sl. 4.6. Pogrešno unesena zaporka ili korisničko ime

Prilikom prijave korisnika, svaki korisnik dobije svoju sjednicu. Svaka sjednica se razlikuje po imenu korisnika koji se prijavio. Svaki korisnik koji pristupa bilo kojoj stranici u aplikaciji mora proći provjeru o tome postavljena li je sjednica. Ukoliko nije postavljena sjednica, aplikacija se preusmjerava na *Login* stranicu.

U priloženom kodu koji se nalazi u PHP skripti *login.php* s *include* se uključuje PHP skripta *conn.php* jer se tamo nalazi kod za spajanje na bazu, inače se ne može odraditi prijava. Prva funkcija provjerava jesu li neka polja u obrascu za prijavu prazna. Druga funkcija provjerava je li korisnik kliknuo na dugme Spremi, a *isset* služi za provjeru ima li *Post* vrijednost. U *if* bloku se provjerava jesu li popunjena sva polja te ako je ispravna zaporka i korisničko ime ulazi se u petlju. Sa *session_start()* pokreće se sjednica, a s *$_SESSION['name'] = $korIme* linijom postavljamo varijablu sjednice i pridružujemo joj vrijednost imena korisnika koji se prijavio. Ako se prijavio administrator otvorit će se stranica *upisudom.php*, a ukoliko je to bio korisnik otvorit će se *podaci.php*, a u toj skripti se ispunjavaju obrasci za prijavu.

```php
<?php
include 'conn.php';
$usernameErr=$passErr=$emptyErr=$registered="";
$korIme=strip_tags($_POST['username']);
$lozinka=strip_tags($_POST['pwd']);

function notEmptyFields($korIme, $lozinka)
{
    if(!empty($korIme) && !empty($lozinka))
    {
        return true;
    }
    return false;
}
function isSubmitClicked()
{
    if(isset($_POST['login']))
    {
        return true;
    }

    return false;
}
if(isSubmitClicked() && !notEmptyFields($korIme, $lozinka))
{
    $passErr="*Popunite sva polja!";
}
if(isSubmitClicked() && userExists($conn, $korIme) && passCorrect($conn, $korIme, $lozinka))
{
    session_start();
    $_SESSION['name'] = $korIme;
    if(userExists($conn, $korIme) == "admin")
    {
        header("Location:upisudom.php");
    }
    else
    {
        header("Location:podaci.php");
    }
}
```

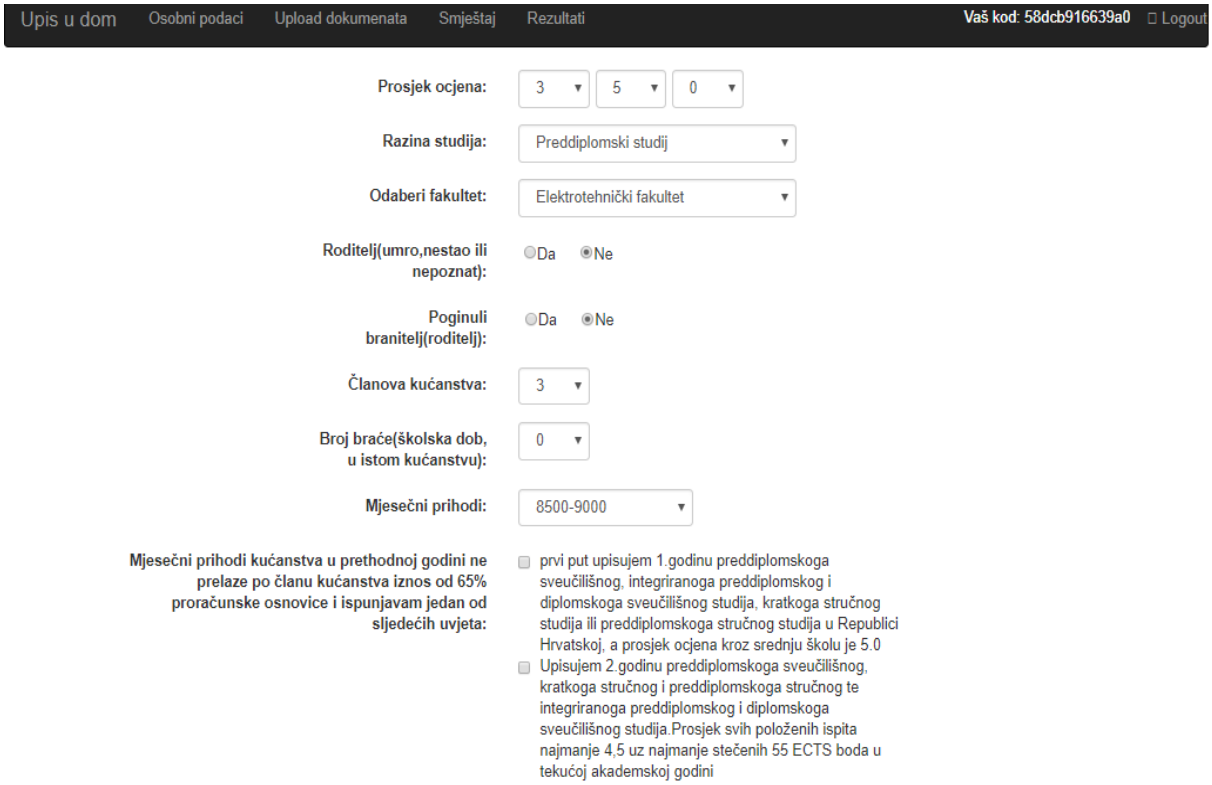
Sl.4.7. *'Login.php'* datoteka

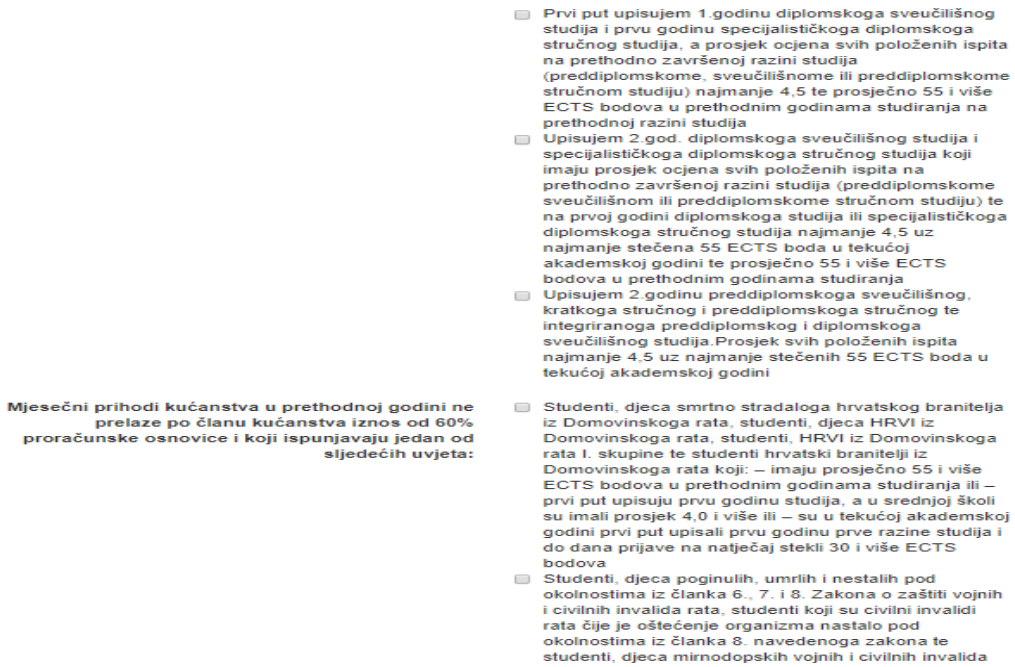
4.4. Korisnički režim rada na centralnom dijelu aplikacije

Nakon prijave koja je bila uspješna, otvara se glavna stranica u kojoj korisnik, tj. student ima mogućnost otvoriti neku od određenih stranica. Klikom na *Osobni podaci* ispunjava najvažniji obrazac, potom ima na raspolaganju *Upload dokumenata* za postavljanje osobnih dokumenata. Opcija *Smještaj* daje informacije o smještaju u oba studentska doma i na kraju opciju *Rezultati* gdje može vidjeti konačnu rang listu prijavljenih studenata.

4.4.1. Unos osobnih podataka

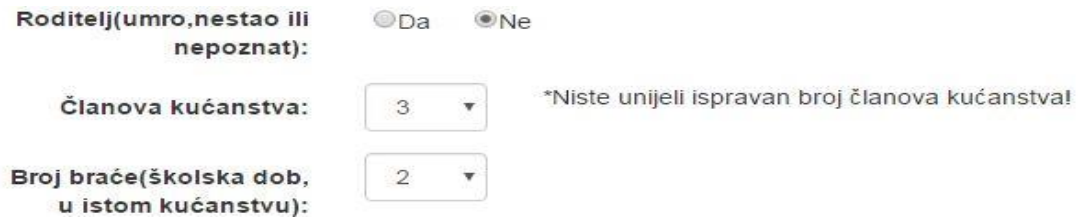
U ovom dijelu aplikacije na stranici *Osobni podaci* prikazuje se obrazac u koji se unose podaci o prosjeku ocjena u dosadašnjem školovanju, razini studija, fakultetu, roditelju (ima li oba ili samo jednoga), članovima kućanstva, broju braće ili sestara koji su školske dobi te o mjesečnim prihodima roditelja. Dakako, tu su i uvjeti kojima se potvrđuje izravan upis u dom (student nema oba roditelja, roditelj poginuo u ratu, a da mu je prosjek preko 4.0, itd.). Ti uvjeti se kasnije dokazuju na stranici *Upload dokumenata* o čemu više u sljedećem poglavlju.





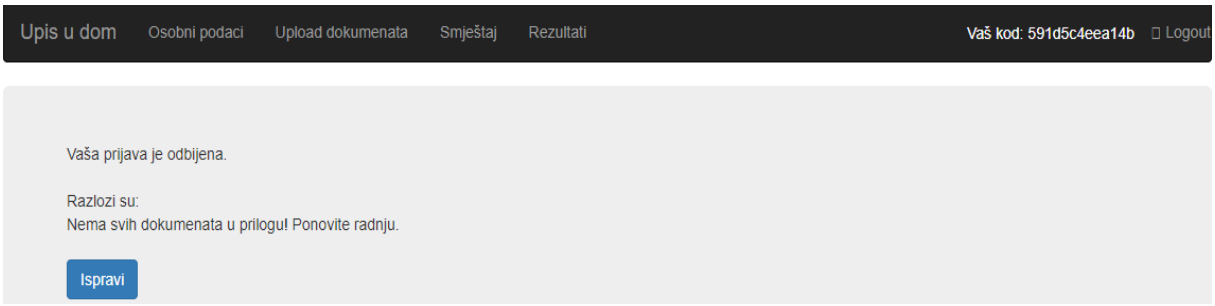
Sl.4.8. Osobni podaci studenta

Student može mijenjati svoje podatke sve dok ne završi rok za prijavu. Također, u kodu je napravljeno da aplikacija odmah izbaci grešku ukoliko je broj članova kućanstva npr. tri, a korisnik pokuša unijeti broj braće 2, a ima oba roditelja.



Sl.4.9. Pogrešan unos članova kućanstva ili broja braće

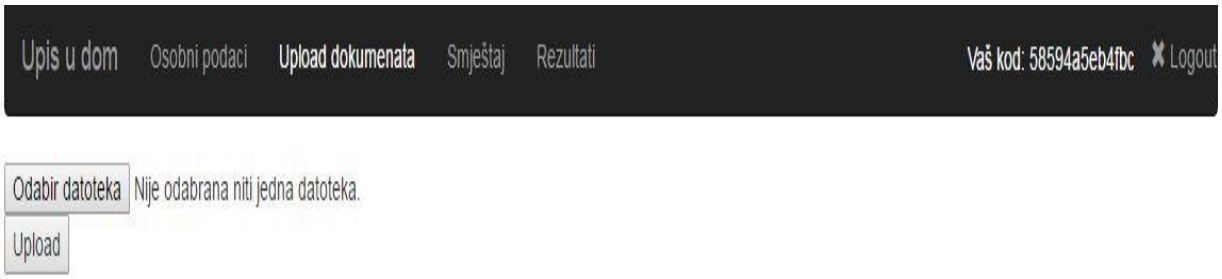
Ako se dogodi slučaj da prijava nije bila valjana, u periodu između kraja roka prijave i vremena za prepravke, korisnik može ponovno ispuniti svoje obrasce i postaviti tražene dokumente. To izgleda ovako:



Sl.4.10. Obavijest korisniku za neispravnu prijavu

4.4.2. Postavljanje dokumenata na aplikaciju

Kako bi provjerili podatke koje je student naveo, postoji i opcija *Upload dokumenata*. Osim provjere navedenih podataka, također treba priložiti prijavnicu u kojoj se između ostalog, navode i podaci o ECTS bodovima kako bi administrator mogao odbiti korisnika ukoliko ne zadovoljava kriterije koje je postavilo Ministarstvo znanosti, obrazovanja i sporta. Tu student postavlja dokument u PDF-u preko kojeg u digitalnom obliku predaje tražene podatke o sebi (domovnica, preslike indeksa, svjedodžbe itd.).



Sl.4.11. Odabir dokumenta i postavljanje istog

Sljedeći dio koda je iz dokumenta *upload.php*. Funkcija *isClicked* služi za provjeru je li stisnuto dugme *Upload*. Ispod funkcije *checkSize* se provjerava veličina pojedinog dokumenta. Datoteke ne smiju biti veće od 2 MB. S funkcijom *getTmpName* dohvaća se ime *temp*-a, tj. folder u kojem se dokumenti nalaze u trenutku odabira i prije postavljanja.

```php
function isClicked()
{
    if(isset($_POST['submit']))
    {
        return true;
    }

    return false;
}

function checkSize($files)
{
    foreach ($files['name'] as $position => $fileName)
    {
        if($files['size'][$position] > 2097152)
        {
            return false;
        }

        return true;
    }
}

function getTmpName($files)
{
    foreach ($files['name'] as $position => $fileName)
    {
        $fileTmp[] = $files['tmp_name'][$position];
    }

    return $fileTmp;
}
```

Sl.4.12. *Upload.php* datoteka

Sa funkcijom getExt dobivamo ekstenziju dokumenata koji se postavljaju. Funkcija *extAllowed* provjerava je li ekstenzija dokumenata koji će se postaviti PDF ili JPG, jer su oni jedini dopušteni. Funkcija *uploadExists* provjerava je li korisnik već postavio dokumente. Ukoliko je postavljeno, vraća *True*, u suprotnom *False*.

```php
function getExt($files)
{
    foreach ($files['name'] as $position => $fileName)
    {
        $fileExt[] = strtolower(end(explode('.', $fileName)));
    }

    return $fileExt;
}

function extAllowed($files, $allowed)
{
    $fileExt = getExt($files);

    foreach ($fileExt as $fileExt)
    {
        if(!in_array($fileExt, $allowed))
        {
            return false;
        }
    }

    return true;
}

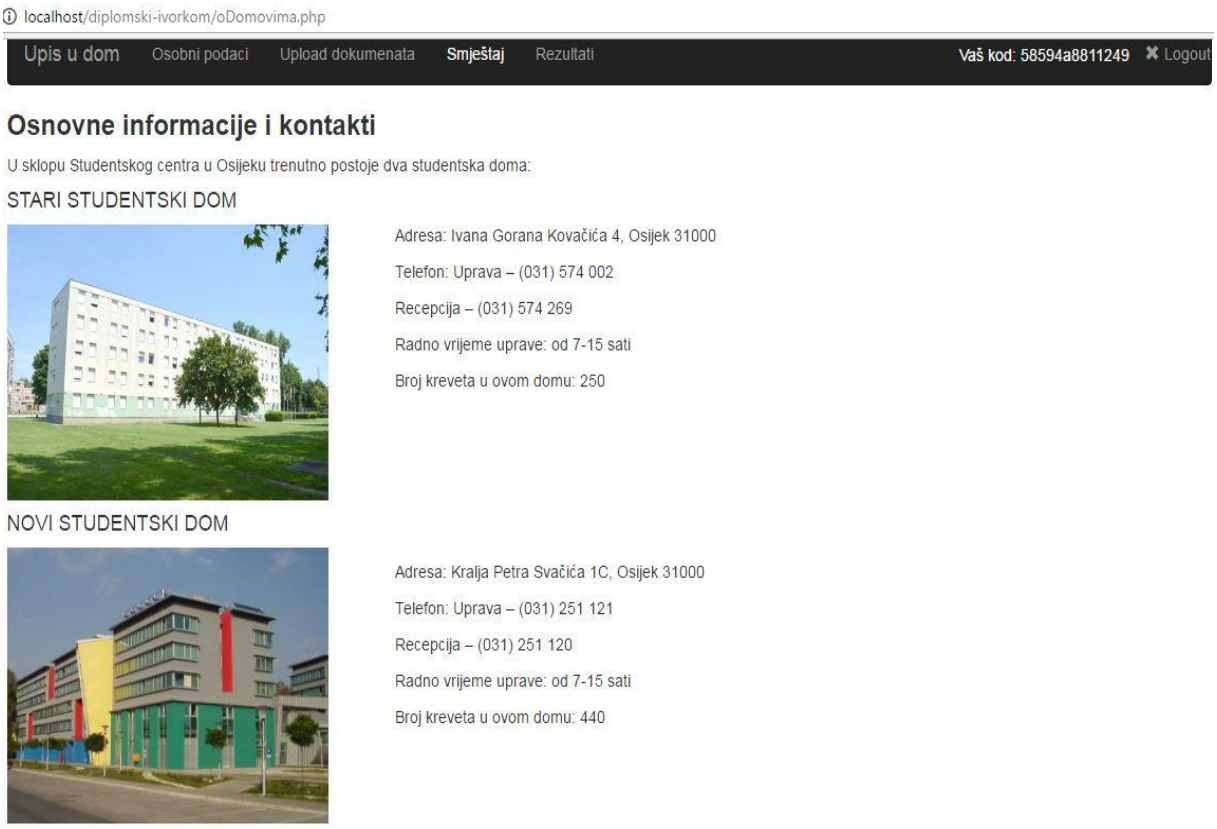
function uploadExists($conn, $idKor)
{
    $result = mysqli_query($conn, "SELECT * FROM dokumenti WHERE idKor='$idKor'");
    $numRows = mysqli_num_rows($result);
    if($numRows > 0)
    {
        return true;
    }

    return false;
}
```

Sl.4.13. *Upload.php* datoteka

4.4.3. Stranica „ Smještaj“

Uz prijavu i ispunjavanje osobnih podataka te postavljanje dokumenata, korisnik na stranici ima mogućnost pogledati osnovne informacije o oba studentska doma u Osijeku. Stari studentski dom raspolaže s 250 slobodnih mjesta, a novi s 440. Dakle, jednostavnom računicom broj studenata koji će imati pravo na upis u studentski dom je 690. Uz podatke o broju kreveta u domovima, može naći i telefonske brojeve oba studentska doma, adrese, kao i radno vrijeme.



Sl.4.14. Izgled stranice *Smještaj*

4.5. Rezultati upisa

Objava rezultata bila bi očekivana u aplikaciji nekoliko dana nakon krajnjeg roka za prijavu. U međuvremenu korisnici bi mogli ponoviti svoje prijave ukoliko im je nešto nedostajalo, pa im je administrator odbio prijavu (više u sljedećem poglavlju). U tablici će se naći svi prijavljeni studenti, ali samo prvih 440 studenata na listi imat će pravo upisa u novi studentski dom, a preostalih 250 u stari dom.

Bodovi za prosjek računaju se tako da skripta *prosjekIzracun.php* uzima sve prijave koje se nalaze u tablici *rezultati* za studente koji su na istoj razini studija i koji su isti fakultet, te sumira

stupac prosjek za sve prijavljene studente koji su na istom fakultetu i razini studija, a dijeli ih s brojem prijava te se tako dobije ukupni prosjek na toj razini studija i tom fakultetu. Poslije toga se u *while* petlji prijave iteriraju, tj. ide se redom po prijavama i dijeli se prosjek pojedinog prijavljenog studenta s ukupnim prosjekom ocjena za tu razinu studija i taj fakultet i na kraju pomnoži sa 1000 te zaokruži na tri decimalne znamenke. Tako se dobivaju bodovi za svakog pojedinog prijavljenog studenta. Potom se zbrajaju bodovi iz svake kategorije za svakog korisnika i umeću se u tablicu rezultati.

Na temelju deset deficitarnih zanimanja koja su određena na nacionalnoj razini, student odabirom na to zanimanje dobiva 200 bodova.

Ukoliko korisnik nema jednog roditelja, dobiva 600 bodova, a ako je dijete poginulog branitelja dobije dodatnih 200.

Po svakom bratu ili sestri zbraja mu se dodatnih 150 bodova. Također je u prijavi potrebno navesti broj članova obitelji. Ako po jednom članu mjesečni prihod obitelji iznosi manje od 500kn, korisnik dobiva maksimalnih 750 bodova. Broj bodova se smanjuje sve dok iznos po članu ne pređe 65% proračunske osnovice. Dakle kada je prihod od 1900kn pa do 65% proračunske osnovice(2162kn) dobit će minimalnih 75 bodova, a kada je iznos veći, dobit će 0 bodova.

Na temelju kriterija za izravan upis, svaki korisnik može odabrati jedan od ponuđenih uvjeta te ako je odabrani kriterij potvrdio dokumentom kojeg je priložio prilikom prijave, biti će izravno upisan neovisno o broju bodova.

U PHP skripti *prosjekIzracun.php* (sl.4.15.) računaju se bodovi na temelju prosjeka svakog studenta. Nakon što program izračuna broj bodova za prosjek, tada i zbroji sve pojedine kategorije kako bi se dobio ukupan broj bodova za rang listu.

Funkcija *prosSum($conn)* iterira kroz sve ponuđene fakultete i razine studija te računa bodove za prosjek za studente koji su na istoj razini studija i fakultetu. Također kod izračuna bodova za prosjek ponovo zbroji sve bodove, jer su bodovi za prosjek prije izračuna bili 1000 za svakog. S petljom *foreach* se prolazi kroz sve elemente niza *razinaStudij*, a zatim prolazi kroz sve elemente niza *fakultet* (prvo se izvršava unutarnji *foreach*). Potom naredba *COUNT* daje broj redova tablice *rezultati* u kojima je ispunjen uvjet da je stupac '*Faks'* jednak trenutnom elementu u nizu u *foreach* petlji *fakultet* i stupac *razinaStudij* jednak trenutnom elementu u nizu *razinaStudij*. Dakle, *COUNT* daje broj redova prijava koje su na istom fakultetu i razini studija, *SUM* (prosjek) zbraja prosjeke svih prijava koji su na istoj razini studija i na istom faksu. Uz sve to, mora biti zadovoljeno da je *Valid* jednako 1, što znači da prijavu mora prihvatiti administrator aplikacije. Zatim samo gleda po stupcu *Srednja* je li Da ili Ne, ako je Da znači da je srednjoškolac i onda računa bodove za prosjek u odnosu na sve srednjoškolce i poslije ažurira ukupan broj bodova isto kao i kod fakulteta.

Sljedeći dio koda nalazi se u skripti *prosjekIzracun.php,* a služi za računanje bodova za prijavitelje u studentski dom koji su isključivo studenti.

```php
<?php
function prosSum($conn)
{
    $razinaStudij = array("Preddiplomski", "Stručni", "Diplomski", "Integrirani");
    $fakultet = array("Elektrotehnički", "Medicinski", "Građevinski", "Filozofski", "Poljoprivredni", "Pravni",
  "Ekonomski", "Prehrambeno tehnološki", "Odjel za kulturologiju", "Odjel za fiziku", "Odjel za matematiku",
  "Odjel za kemiju", "Odjel za biologiju", "Fakultet za odgojne i obrazovne znanosti", "Umjetnička akademija");

    foreach($razinaStudij as $razina)
    {
        foreach($fakultet as $faks)
        {
            $results = mysqli_query($conn, "SELECT COUNT(*) AS count, SUM(Prosjek) AS sum FROM rezultati "
                    . "WHERE Faks='$faks' AND razinaStudij='$razina' AND Valid=b'1'");

            $row1 = mysqli_fetch_assoc($results);
            $sum = $row1['sum'];
            $count = $row1['count'];
            $prosUk = $sum/$count;

            $query = mysqli_query($conn, "SELECT * FROM rezultati WHERE razinaStudij='$razina' AND Faks='$faks' AND Valid=b'1'");

            while($row2=mysqli_fetch_assoc($query))
            {
                $prosjek = $row2['Prosjek'];
                $idKor = $row2['idKor'];

                $prosBod = round(1000*($prosjek/$prosUk), 3);
                $roditelj = $row2['Roditelj'];
                $defic = $row2['Deficitarna'];
                $braca = $row2['Braca'];
                $prihod = $row2['mjPrihod'];
        $pogBranitelj = $row2['pogBranitelj'];

                $ukupno = $prosBod + $prihod + $defic + $roditelj + $braca + $pogBranitelj;

                mysqli_query($conn, "UPDATE rezultati SET prosBod='$prosBod', Ukupno='$ukupno' WHERE idKor='$idKor' AND Valid=b'1'");
            }
        }
    }
  $resultsSrednja = mysqli_query($conn, "SELECT COUNT(*) AS count, SUM(Prosjek) AS sum FROM rezultati "
                    . "WHERE Srednja='Da' AND Valid=b'1'");
  $row1Srednja = mysqli_fetch_assoc($resultsSrednja);
  $sumSrednja = $row1Srednja['sum'];
  $countSrednja = $row1Srednja['count'];
  $prosUkSrednja = $sumSrednja/$countSrednja;

  $querySrednja = mysqli_query($conn, "SELECT * FROM rezultati WHERE Srednja='Da' AND Valid=b'1'");

  while($row2Srednja=mysqli_fetch_assoc($querySrednja))
  {
    $prosjekSrednja = $row2Srednja['Prosjek'];
    $idKorSrednja = $row2Srednja['idKor'];

    $prosBodSrednja = round(1000*($prosjekSrednja/$prosUkSrednja), 3);
    $roditeljSrednja = $row2Srednja['Roditelj'];
    $deficSrednja = $row2Srednja['Deficitarna'];
    $bracaSrednja = $row2Srednja['Braca'];
    $prihodSrednja = $row2Srednja['mjPrihod'];
    $pogBraniteljSrednja = $row2Srednja['pogBranitelj'];

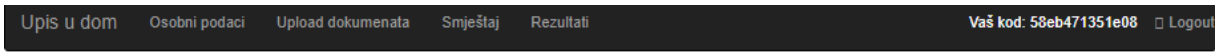
    $ukupnoSrednja = $prosBodSrednja + $prihodSrednja + $deficSrednja + $roditeljSrednja + $bracaSrednja + $pogBraniteljSrednja;

    mysqli_query($conn, "UPDATE rezultati SET prosBod='$prosBodSrednja', Ukupno='$ukupnoSrednja' WHERE idKor='$idKorSrednja' AND Valid=b'1'");
  }
}
prosSum($conn);
```

Sl.4.15. *prosjekIzracun.php* datoteka

Sve prijave koje administrator nije pregledao imaju stupce *Pregledan* i *Valid* iz tablice *rezultati* postavljene na „0“. Kada administrator prihvati njihove prijave, *Pregledan* i *Valid* se postavljaju na „1“. Ako administrator odbije, ostaju na „0“ i neće se naći u tablici *Rezultati.* Korisnici tada imaju mogućnost nekoliko dana izmijeniti svoju prijavu, te se uz priložene dokumente koji potkrjepljuju odabrane kriterije, mogu naći na konačnoj tablici *Rezultati.*

Na sljedećem primjeru prikazan je izgled tablice rezultata te upisanih i ne upisanih studenata. Radi lakšeg pregleda u tablici će biti 28 studenata, a najboljih 14 će imati pravo upisa. Korisnicima se kod ispisa rezultata neće pojaviti imena i prezimena nego su rezultati poredanima po bodovima i izravnom upisu, a sebe će prepoznati po svom jedinstvenom kodu.

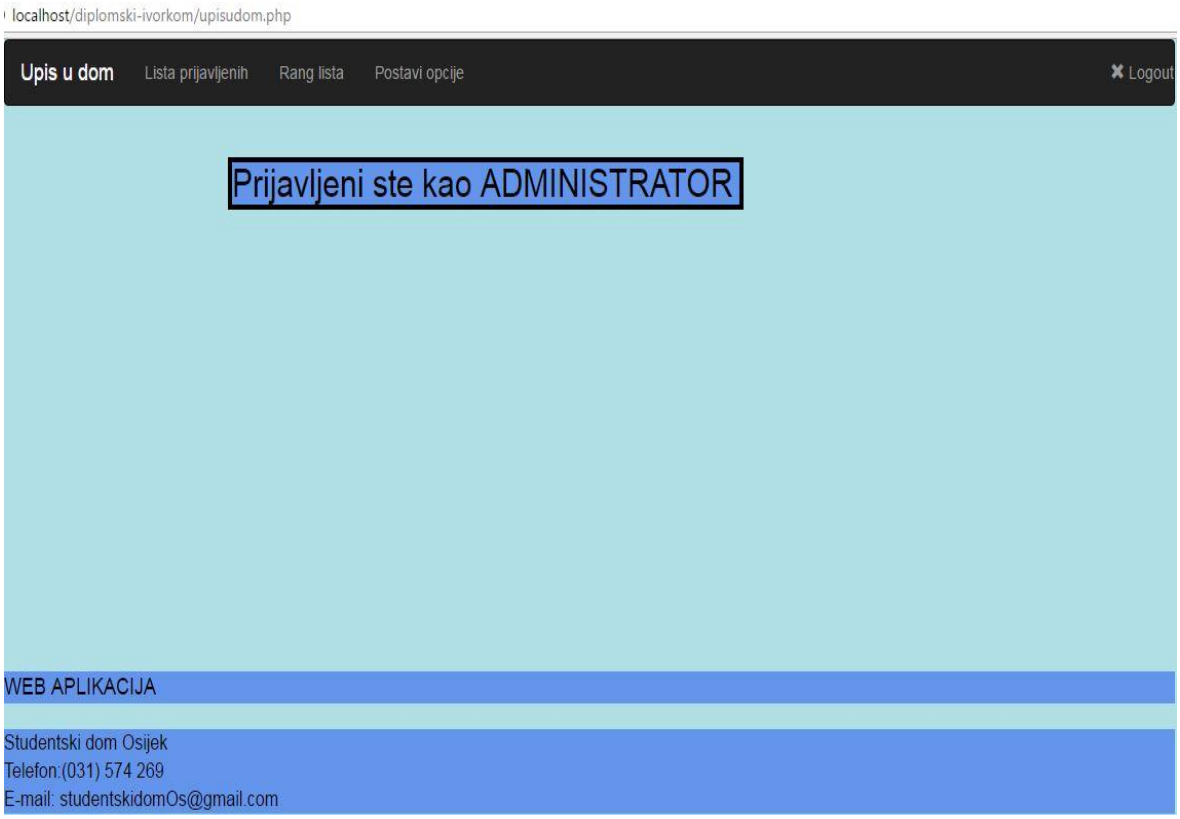


	Kod	Prosjek	Deficitarna zanimanja	Bez roditelja	Poginuli branitelj(roditelj)	Braće	Prihodi	Ukupno	Izravan upis	Status
1	58e784af85e4c	784.586	0	600	0	150	0	1534.586	Da	PRIMLJEN
2	5877ddd298a86	1108.483	200	0	0	150	0	1458.483	Da	PRIMLJEN
3	58e4d208ae2b8	1000	0	0	0	0	375	1375	Da	PRIMLJEN
4	58c9969382a89	1089.727	0	600	0	300	525	2514.727	Ne	PRIMLJEN
5	590af750bf6e7	1000	200	600	0	150	225	2175	Ne	PRIMLJEN
6	58e68f087bd6d	924.126	200	600	0	150	225	2099.126	Ne	PRIMLJEN
7	58eb471351e08	884.634	200	0	0	600	375	2059.634	Ne	PRIMLJEN
8	58eb4777d811f	1000	0	600	0	150	300	2050	Ne	PRIMLJEN
9	58eb48e067bbc	1095.261	200	0	0	450	300	2045.261	Ne	PRIMLJEN
10	58c7fc3e0ae21	1260.504	0	0	0	300	375	1935.504	Ne	PRIMLJEN
11	58e4e0828b395	856.444	200	600	200	0	0	1856.444	Ne	PRIMLJEN
12	5877d05852815	1082.569	200	0	0	150	375	1807.569	Ne	PRIMLJEN
13	58e783d0771a4	846.339	0	600	200	150	0	1796.339	Ne	PRIMLJEN
14	58b6a0c075100	1000	0	600	0	150	0	1750	Ne	PRIMLJEN
15	58c7bbe2a98f6	893.157	0	600	200	0	0	1693.157	Ne	NIJE PRIMLJEN
16	5877cf4120383	1172.529	200	0	0	300	0	1672.529	Ne	NIJE PRIMLJEN

Sl.4.16. Tablica konačnih rezultata upisa

4.6. Mogućnosti rada administratora u aplikaciji

Administrator se na aplikaciju prijavljuje kao i ostali korisnici, unošenjem korisničkog imena i zaporke koji su već definirani u bazi. U samom kodu postoji posebna funkcija koja dohvaća samo administratora iz tablice *korisnici* u bazi podataka. Tako se vrši autorizacija i prepoznaje se administrator.



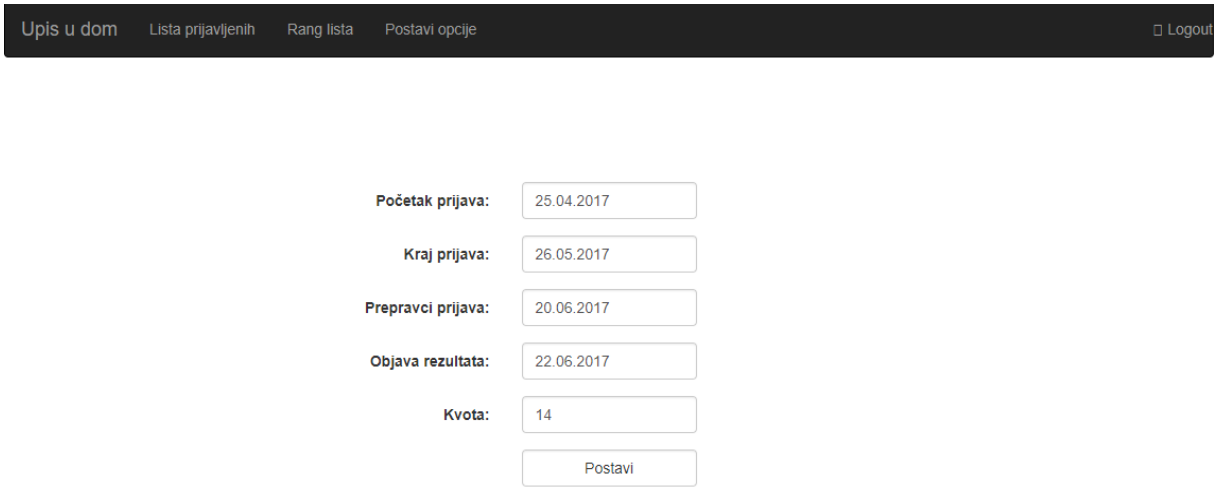
Sl.4.17. Početna stranica prilikom prijave administratora

4.6.1. Postavljanje opcija

Prema datumu navedenom u natječaju iz novina, administrator na stranici *Postavi opcije* postavlja datume za početak i kraj prijave, za prepravke koje će moći korisnici obavljati ako im administrator odbije prijavu, objavu rezultata te kvotu slobodnih mjesta u domu, tj. broj studenata koji će biti upisani.

Nakon što je student ispunio sve svoje obveze postavljajući tražene dokumente te je ispunio obrazac o osobnim podacima, ostalo mu je čekati da završi rok za prijavu. Polja moraju biti

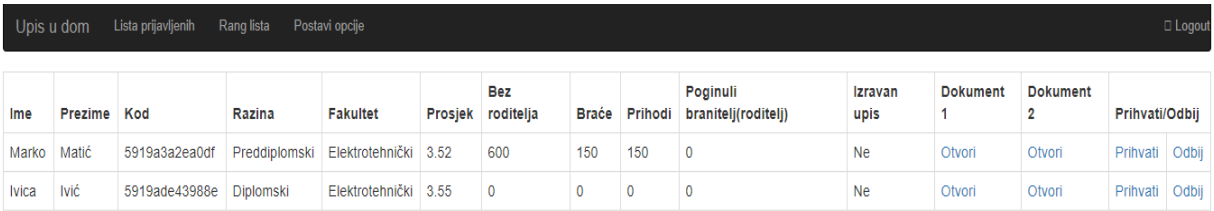
postavljena u vremenskom slijedu, početak prijava ne može biti prije kraja, prepravci prije objave. Kod kvote, smije se unijeti samo *integer*, ako se unese nešto drugo, javi grešku.



Sl.4.18. Administrator postavlja datume za rokove prijave

4.6.2. Provjera podataka u Listi prijavljenih kandidata

Ova lista može se protumačiti kao nekakva lista čekanja. Administratoru se ovdje u tablici pojavljuju studenti kojima on još nije prihvatio prijavu. Ovdje pregledava njihovu prijavu te priložene dokumente. Ukoliko sve smatra valjanim stišće na *Prihvati*, u suprotnom *Odbij*.
U slučaju da klikne *Odbij*, administratoru iskoči prozorčić u koji unosi tekst gdje navodi što nije valjalo kod prijave. Kada se korisnik prijavi na aplikaciju dolazi mu obavijest te ima mogućnost izmjene svojih podataka do roka koji je postavio administrator. Potom administrator opet može prihvatiti ili odbiti prijavu.

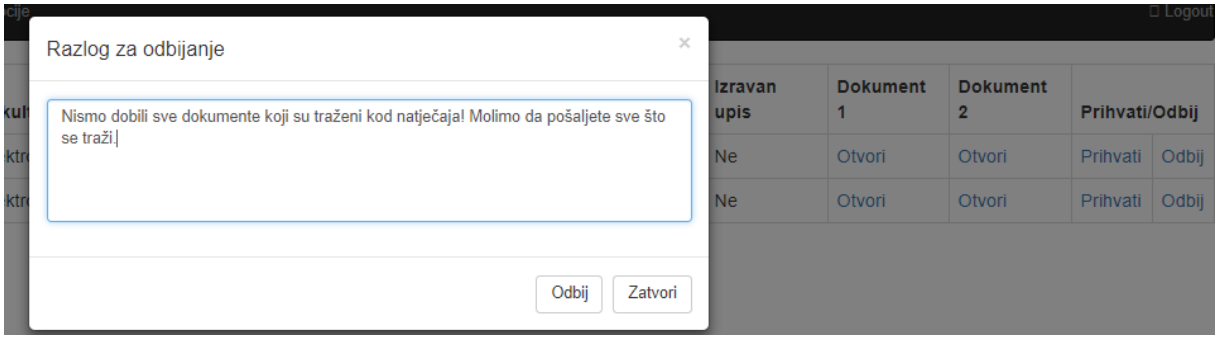


Ime	Prezime	Kod	Razina	Fakultet	Prosjek	Bez roditelja	Braće	Prihodi	Poginuli branitelj(roditelj)	Izravan upis	Dokument 1	Dokument 2	Prihvati/Odbij	
Marko	Matić	5919a3a2ea0df	Preddiplomski	Elektrotehnički	3.52	600	150	150	0	Ne	Otvori	Otvori	Prihvati	Odbij
Ivica	Ivić	5919ade43988e	Diplomski	Elektrotehnički	3.55	0	0	0	0	Ne	Otvori	Otvori	Prihvati	Odbij

Sl. 4.19. Lista prijavljenih – Prihvati/Odbij

Ako administrator klikne na *Odbij* pojavljuje se ovaj prozorčić. Tekstom upozorava korisnika što treba promijeniti. Pojavljivanje prozorčića je napravljeno u Javascriptu. Prozorčić je Bootstrap modal. Lista čekanja je nastala upitom na bazu kojim su povezane tri tablice (rezultati, korisnici, dokumenti). Taj upit sadrži naredbu JOIN koja povezuje tablice po nekom uvjetu, u

ovom slučaju id-evi korisnika moraju biti isti. Također, u listi se pojavljuju samo oni reci koji imaju stupac *Pregledan* postavljen na „0“.



Sl.4.20. Tekst koji unos administrator kao razlog odbijanja prijave.

Pomoću JavaScript programskog jezika obrađena je ova Lista čekanja. Svaki redak liste čekanja ima svoj *id*. Klikom na *Odbij* ili *Prihvati* se dohvaća taj *id*. Klikom na prihvati prosljeđuje se *id* retka skripti *prihvati prijavu.php*. Klikom na odbij se prikazuje modalni prozor što se vidi na slici 4.14. Ako se skripta uspješno izvrši, osvježi se stranica (*success* funkcija).
Ako se ništa ne upiše za razlog odbijanja dodaje se crveni obrub. Ukoliko je nešto napisano, onda se *id* retka i razlog odbijanja prosljeđuje skripti *odbijPrijavu.php*. Kada se skripta uspješno izvrši, osvježi se stranica. Zadnja tri reda služe u svrhu resetiranja prozora za odbijanje. Dakle, kada se zatvori prozor za odbijanje, prozor se resetira kako ne bi ostali obrubi ili stari tekst.

```
var id;
$(".odbij").click(function(){
    id = $(this).attr('id');
    $('#modal').modal('show');
});

$(".prihvati").click(function(){
    id = $(this).attr('id');

    $.ajax({
            type: "POST",
            data: {"id": id },
            url:"prihvatiPrijavu.php",
            success:function() {
                location.reload();
            }
    });

});
```

```
$(document).on("click", "#odbijRazlog", function(event){
    var razlog = $("#razlog").val();

    if(!razlog){
        $("#razlog").css('border-color', 'red');
    }else{
        $.ajax({
            type: "POST",
            data: {"id": id, "razlog": razlog},
            url:"odbijPrijavu.php",
            success:function() {
                location.reload();
            }
        });
    }
});

$(".modal").on("hidden.bs.modal", function(){
    $("#razlog").val("");
    $("#razlog").css('border-color', '');
});
```

Sl.4.21. Kod u datoteci *listaCekanja.php*.

4.6.3. Uvid u Rang listu prijavljenih kandidata

Završetkom roka prijave, na ovoj stranici prikazuje se rang lista u obliku tablice. Ovdje će se naći svi studenti kojima je administrator prihvatio prijavu. Na temelju izračuna bodova (vidi podnaslov Rezultati) za prosjek, deficitarna zanimanja, broj braće i sestara, prihodima u obitelji ili izravan upis, radi se ova rang lista studenata. Administrator je ranije postavio kvotu za upis koja je glasila na 14. Izgled je sljedeći:

Upis u dom Lista prijavljenih Rang lista Postavi opcije Logout

	Kod	Naziv	Prosjek	Deficitarna zanimanja	Bez roditelja	Poginuli branitelj(roditelj)	Braće	Prihodi	Ukupno	Izravan upis
1	58e784af85e4c	Ivan Ibriks	784.586	0	600	0	150	0	1534.586	Da
2	5877ddd298a86	Luka Mandić	1108.483	200	0	0	150	0	1458.483	Da
3	58e4d208ae2b8	Hrvoje Filajdić	1000	0	0	0	0	375	1375	Da
4	58c9969382a89	Frano Zoljan	1089.727	0	600	0	300	525	2514.727	Ne
5	590af750bf6e7	Mirko Ivša	1000	200	600	0	150	225	2175	Ne
6	58e68f087bd6d	Jurica Ljevaković	924.126	200	600	0	150	225	2099.126	Ne
7	58eb471351e08	Ante Šimunović	884.634	200	0	0	600	375	2059.634	Ne
8	58eb4777d811f	Marko Šimunović	1000	0	600	0	150	300	2050	Ne
9	58eb48e067bbc	Kristijan Tutić	1095.261	200	0	0	450	300	2045.261	Ne
10	58c7fc3e0ae21	Mirko Matić	1260.504	0	0	0	300	375	1935.504	Ne
11	58e4e0828b395	Matija Gostimir	856.444	200	600	200	0	0	1856.444	Ne
12	5877d05852815	Dušan Zorić	1082.569	200	0	0	150	375	1807.569	Ne
13	58e783d0771a4	Branimir Zelić	846.339	0	600	200	150	0	1796.339	Ne
14	58b6a0c075100	Sebastijan Žilić	1000	0	600	0	150	0	1750	Ne
15	58c7bbe2a98f6	Dražen Mrnj	893.157	0	600	200	0	0	1693.157	Ne
16	5877cf4120383	Antonio Sušac	1172.529	200	0	0	300	0	1672.529	Ne
17	58e7537d002b7	Marko Abramović	1316.419	200	0	0	0	0	1516.419	Ne
18	58dcb916639a0	alen frankopan	910.273	0	0	0	300	225	1435.273	Ne
19	58ee08fe9a4e4	Marko Markić	1190.043	200	0	0	0	0	1390.043	Ne
20	590218bdbca2d	Ranko Renić	1035.073	200	0	0	150	0	1385.073	Ne
21	5877cf05885e4	Matko Vujnovac	827.471	200	0	0	0	300	1327.471	Ne
22	5901d4b840846	Bruno Topolski	1087.362	0	0	0	150	0	1237.362	Ne
23	5901d587f0bd4	Domagoj Ajman	1000	0	0	0	150	0	1150	Ne
24	59021d3427366	Marija Mamić	932.025	200	0	0	0	0	1132.025	Ne
25	59011fcbbb323	Siniša Marki	929.392	200	0	0	0	0	1129.392	Ne
26	5909d42fc737e	Antun Domankušić	929.392	200	0	0	0	0	1129.392	Ne
27	5900ff17b8870	Luka Azimut	926.759	200	0	0	0	0	1126.759	Ne
28	5877d1dbdfe75	Mateo Tokić	917.431	200	0	0	0	0	1117.431	Ne

Sl.4.22. Konačna rang lista

5. ZAKLJUČAK

Aplikacija čija je izrada i funkcionalnost objašnjena ovim radom nudi jednostavno korištenje, s obzirom da je korisnicima dovoljan bilo kakav web preglednik i pristup Internetu. Ova aplikacija stvarana je korištenjem različitih tehnologija kao što su CSS, HTML, PHP, JavaScript, a za upravljanje bazom podataka korišten je MySQL. Dakle, u vrijeme kada je Internet dostupan svima i na svakom mjestu događa se visoki rast broja web aplikacija. Iz tog razloga ova aplikacija ima svijetlu budućnost te može biti korisna svim studentskim domovima, a uz male preinake i učeničkim domovima. Korisnost korištenja ovakve aplikacije može biti višestruka, kao što je prijava studenata na daljinu što bi im značilo značajne uštede vezane za novac i vrijeme s obzirom da fizički ne bi morali biti prisutni pri prijavi u njima udaljenim gradovima, kontrola rada ljudskih resursa u studentskim centrima, minimiziranje pogrešaka koje su moguće pri ručnom unosu prikupljenih podataka kandidata koji su aplicirali za smještaj u računalni sustav. Ukoliko bi došlo do promjena u zakonskom okviru koje se odnose na prijave i uvjete koji trebaju biti ispunjeni, korisnici bi putem aplikacije bili pravovremeno informirani te bi svoje prijave obavili valjano i u predviđenom roku.

LITERATURA

[1] http://www.stucos.unios.hr/natjecaj/natjecaj_domovi_2015_2016.pdf [02.06.2017.]

[2] https://www.mojwebdizajn.net/skriptni-jezici/vodic/html/uvod-u-html.aspx [17.06.2016.]

[3] http://www.mojwebdizajn.net/skriptni-jezici/vodic/html/uvod-u-html.aspx [17.06.2016.]

[4] https://www.w3.org/Style/CSS/Overview.en.html, [18.06.2016.]

[5] http://www.mojwebdizajn.net/skriptni-jezici/vodic/css/css-syntax.aspx [18.06.2016.]

[6] http://php.com.hr/77 [18.06.2016.]

[7] http://www.hdonweb.com/programiranje/php-sintaksa [18.06.2016.]

[8] M.E. Davis, J.A. Phillips, Learning PHP and MySQL, O'Reilly Media, Inc., Sebastopol, 2007. [31.05.2016.]

[9] http://razno.sveznadar.info/10-doc-PDF/2_MySQL.pdf [20.06.2016.]

[10] http://blog.falcon-tech.rs/2015/04/22/sta-je-bootstrap/sta-je-bootstrap/ [15.5.2017.]

[11] https://www.w3schools.com/bootstrap/ [15.5.2017.]

[12] http://www.znanje.org/knjige/computer/JavaScript/2010/osnove.htm [15.5.2017.]

[13] M., Haverbeke, Eloquent JavaScript: A Modern Introduction to Programming, William Pollock, San Francisco, 2011. [01.6.2017]

[14] J., Ducket, JavaScript and JQuery: Interactive Front-End Web Development, John Wiley & Sons, Inc., Indiana, 2014. [01.6.2017]

SAŽETAK

Ovim diplomskim radom je dan model i web aplikacija za upis studenata u studentski dom. Aplikacija ima dva režima rada, a to su korisnički i administratorski. Svaki student prilikom prijave na aplikaciju treba navesti podatke o prosjeku ocjena u dosadašnjem školovanju, razini studija, fakultetu, roditelju (ima li oba ili samo jednoga), članovima kućanstva, broju braće ili sestara koji su školske dobi te o mjesečnim prihodima roditelja. Također, postoje i uvjeti kojima se potvrđuje izravan upis u dom.

Nakon što je prošao krajnji rok za prijavu, administrator može prihvatiti ili odbiti prijave, koje korisnik opet može mijenjati do određenog roka. Na kraju slijedi konačna objava rezultata.

Za razliku od dosadašnje prijave u studentske domove, ovo je mnogo pristupačnije, jeftinije i ne uzima gotovo ništa vremena. Dovoljno je imati pristup Internetu, te se kroz koju minutu završi cjelokupna prijava za smještaj u studentski dom.

Ključne riječi: student, studentski dom, web aplikacija, Internet, korisnik, administrator

ABSTRACT

A model and software support for ranking students during enrolment in student dormitory

A model and a web application for student enrolments in student dormitory are given with this thesis. The application has two ways of working; one is intended for users and the other one for administrators. Every student when applying must give data on GPA (academic achievement at school), level of education, university, parents (both or just one), household members, number of brothers and sisters and their school age and monthly income of parents. Also, there are extra conditions which can confirm direct enrolment in student dormitory.

After a deadline for applying, the administrator can accept or decline applications, which user can change again until a certain deadline. In the end follows final announcement of results. On the contrary to previous applications for student dormitory, this is much more accessible, cheaper and it takes only few minutes for applying. It is enough to have Internet access and the whole registration for student dormitory ends in just few minutes.

Key words: student, student dormitory, web application, Internet, user

ŽIVOTOPIS

Ivor Podunavac rođen je 18.02.1993. u Zagrebu. Odrasta i živi u Lipiku gdje završava osnovnu školu s odličnim uspjehom. Poslije osnovne škole upisuje srednju školu, Tehničar za računalstvo u Kutini. Tamo boravi u učeničkom domu, te školu završava s vrlo dobrim uspjehom. Također postiže uspjeh na razini Sisačko - Moslavačke županije u obliku prvoga mjesta na natjecanju iz Kemije.

Godine 2011. upisuje preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. U roku završava prve tri godine, a potom na istom fakultetu upisuje diplomski studij Procesnog računarstva.

Svoje slobodno vrijeme upotpunjava baveći se sportom, igranjem nogometa u klubu „NK Lipik 1925“. Godine 2016. proglašen za najboljeg sportaša grada Lipika.

PRILOZI

Na CD-u priloženom uz diplomski rad nalaze se:

Dokumenti:

- Model i programska podrška za rangiranje studenata pri upisu u studentski dom.docx
- Model i programska podrška za rangiranje studenata pri upisu u studentski dom.pdf

Datoteke:

-Izvorni kodovi aplikacije