Web aplikacija za simulaciju natjecateljske lige

Marincel, Bruno

Undergraduate thesis / Završni rad

2021

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:200:602417

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-20**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Bruno Marincel

Web aplikacija za simulaciju natjecateljske lige

Završni rad

Osijek, 2021.

Sadržaj

1. UVOD 1
1.1 Zadatak završnog 2
2. SLIČNA RJEŠENJA 3
2.1 Fantasy Premier League 3
2.2 UEFA Champions League Fantasy 4
2.3 Fantasy Bundesliga 5
2.4 La liga fantasy 6
2.5 Najboljih jedanaest 7
3. ALATI ZA IZRADU 9
3.1 Što je HTML 9
3.2 Što je PHP 9
3.3 Što je phpMyAdmin 10
3.4 Što je WAMP 11
3.5 Što je MVC 12
4. STRUKTURA APLIKACIJE 14
4.1 Instalacija aplikacije 14
4.2 Izrada korisničkog računa 17
4.3 Prijava 19
4.4 Postavljanje ekipe 21
4.5 Glavna stranica 24
4.6 Admin 25
5. ZAKLJUČAK 33
LITERATURA 34
SAŽETAK 35
ABSTRACT 36
PRILOG 37

1. UVOD

Cilj rada je bio napraviti web aplikaciju za nogometnu natjecateljsku ligu. Aplikacija je za nogometne ljubitelje, povremene pratitelje i one željne zabave. Pri ulasku u aplikaciju potrebno je napraviti korisnički račun i dodijeliti ime ekipi, te se prijaviti. Kada je to napravljeno korisnik aplikacije daje ime svojoj ekipi, te potom dobiva na izbor igrače jedne lige, od kojih na kraju odabire svojih idealnih jedanaest igrača. Prilikom izbora igrača treba biti pažljiv jer svaki korisnik dobiva sumu od četrdeset milijuna eura. Svaki igrač ima svoju cijenu i ona ovisi o kvaliteti i formi igrača, na primjer ako je igrač koji se nalazi na poziciji napadača bio najbolji strijelac lige, cjenovno će biti najskuplji napadač, isto to vrijedi i za veznjake i za obrambene i za golmane. Cijene igrača su uzete sa svjetski najpoznatije njemačke stranice „Transfermarkt [1]“, zadužene za sve informacije vezane uz nogomet, golove, rezultate, statistiku, transfere igrača, novosti, raspored kola i ono što je najvažnije tržišnu vrijednost igrača. Kada korisnik odabere svojih idealnih jedanaest, naravno u zadanom budžetu, igra može započeti. Cilj je skupiti što više bodova u sezoni. Bodovi se skupljaju tako da aplikacija prati stvarne događaje, odnosno utakmice i tako dodjeljuje bodove pojedinim igračima. Administrator pogleda i unese ishod utakmice i sve parametre koji dodjeljuju bodove igračima. Ako je utakmica završila rezultatom 2:0 i igrač koji se nalazi u korisnikovoj idealnoj ekipi je dao gol, korisnik na tom igraču dobiva bodove. Isto tako ako je igrač asistirao na utakmici, a nalazi se u korisnikovih idealnih jedanaest, dobiva bodove na tom igraču.

Bodovanje izgleda ovako :

- Napadač
    - Za postignuti gol igrač dobiva četiri bodova
    - Za asistenciju igrač dobiva tri boda
- Veznjak
    - Za postignuti gol igrač dobiva pet bodova
    - Za asistenciju igrač dobiva tri boda
- Obrambeni
    - Za postignuti gol igrač dobiva pet bodova
    - Za asistenciju igrač dobiva tri boda
    - Za ne primljen gol igrač dobiva četiri boda

- Golman
    - Za postignut gol igrač dobiva deset bodova
    - Za asistenciju igrač dobiva šest bodova
    - Za ne primljen gol igrač dobiva četiri boda
- Za sve igrače
    - Ako dobiju žuti karton oduzima im se jedan bod
    - Ako dobiju crveni karton oduzima im se dva boda
    - Ako nisu na klupi dobiju dva boda

Postoji trideset i dva kola, odnosno deset ekipa u ligi od kojih svaka ekipa igra protiv svake i to četiri puta. Nakon svakog kola bodovi se dodaju korisniku. Pobjednik je onaj korisnik koji je nakon trideset i dva kola uspio skupiti najviše bodova.

1.1 Zadatak završnog

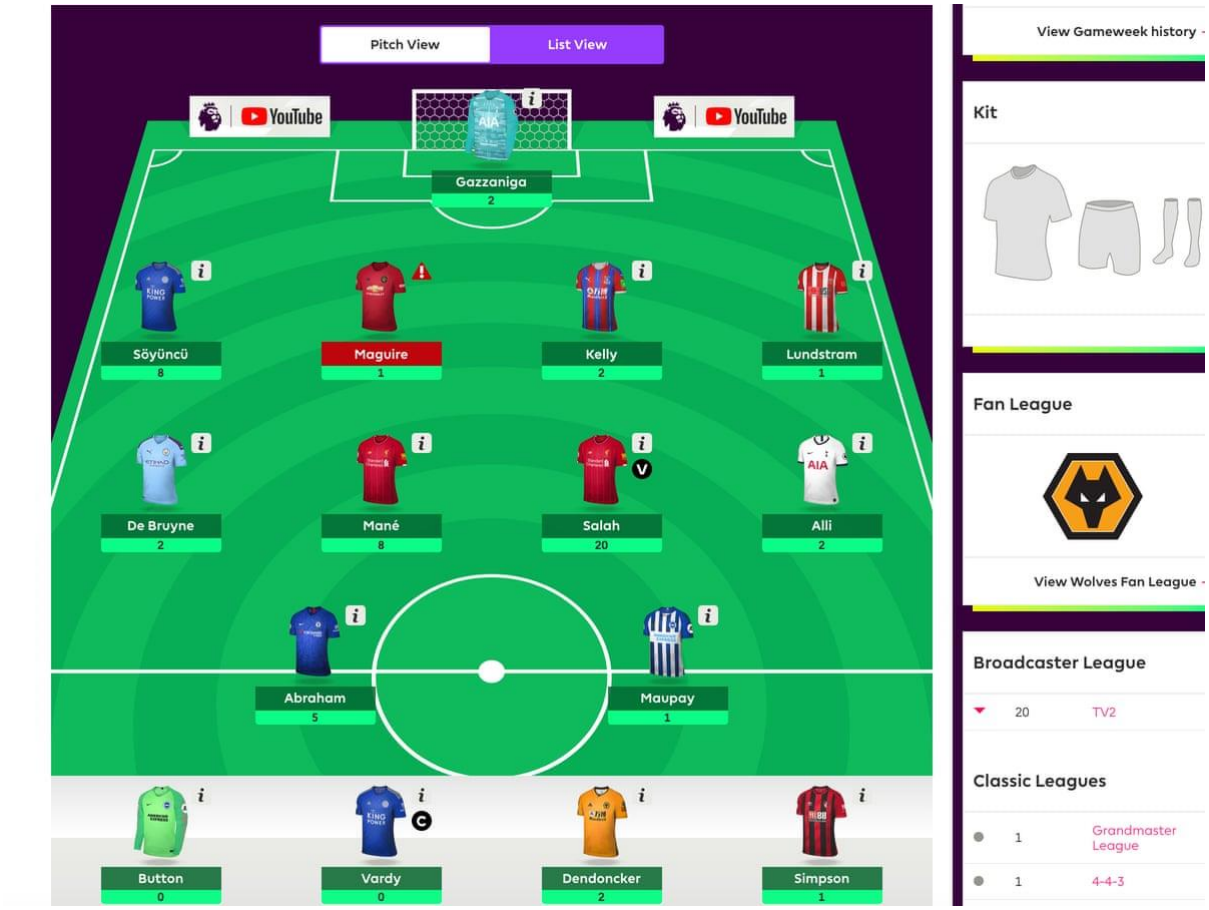
Opisati način rada natjecateljske lige i parametre koji se određuju za pojedini sportski klub i igrača. Opisati tehnologije koje će se koristiti za izradu web aplikacije. Navesti funkcionalnosti koje će imati web aplikacija. Izraditi bazu podataka koju će koristiti web aplikacija. Izraditi web aplikaciju i opisati postupak izrade.

2. SLIČNA RJEŠENJA

Najpopularnije aplikacije na svijetu, kao što su "Fantasy Premier League" [2], službena stranica prve Engleske nogometne lige i "UEFA Champions League Fantasy" [3] , službene stranice UEFA-ine Lige prvaka, također rade po sličnom principu kao i ova web aplikacija. Najveća razlika je što je ova aplikacija namijenjena domaćem tržištu, odnosno prvoj Hrvatskoj nogometnoj ligi [4]. Cilj je približiti domaću ligu i povećati njezino praćenje kako bi na taj način liga postala kvalitetnija i zabavnija. Do prošle sezone je bila aktivna stranica „Najboljih 11“ [5] , koja je bila vrlo slična ovoj, no bila je puna raznih nepravilnosti i grešaka, te je koristila "Flash Player" za kojeg je podrška prestala u prosincu 2020. godine, nakon 20 godina korištenja.

2.1 Fantasy Premier League

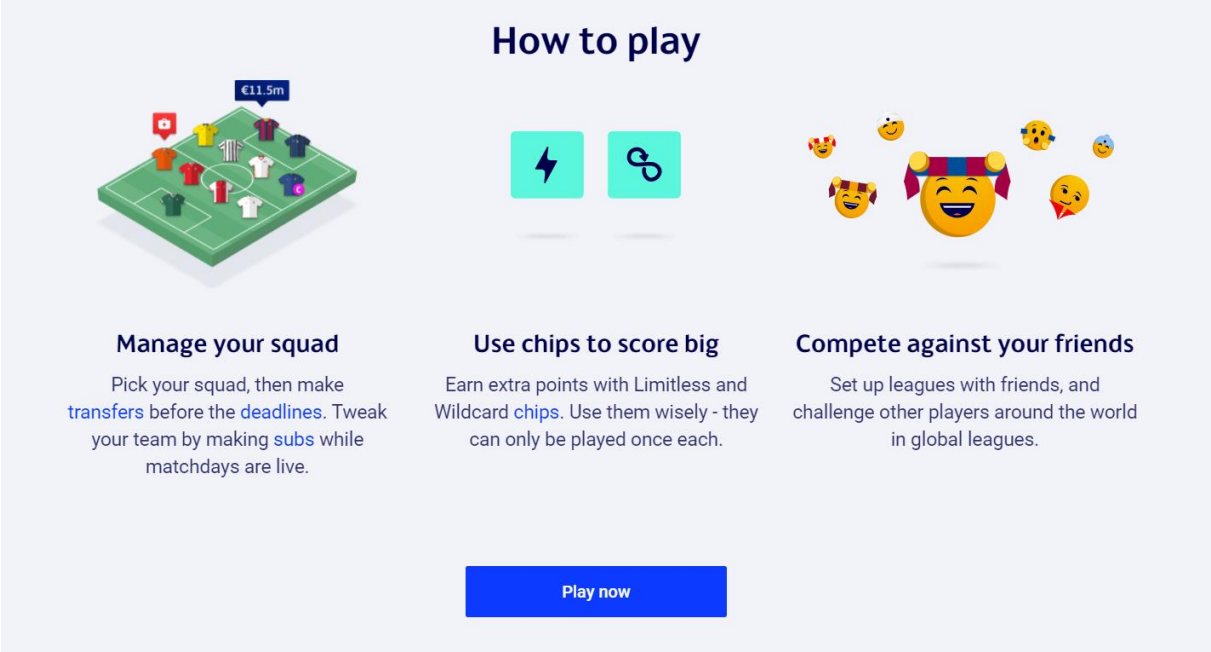
Ovo je službena stranica za natjecateljsku ligu prve Engleske nogometne lige. Postoji web i smartphone aplikacija, koje su jako slične, te imaju jako malo vizualnih razlika. Izgled web aplikacije vidi se na slici 2.1, iz koje se vidi da sadrži prikaz korisnikovih igrača, prikaz bodova posljednjeg kola, prikaz srednje vrijednosti bodova svih korisnika, prikaz maksimalnog broja bodova postignut tog kola, prikaz rednog broja na ljestvici i slične informacije.



Sl. 2.1 Prikaz službene web aplikacije za prvu englesku ligu. (izovr: www.theguardian.com)

2.2 UEFA Champions League Fantasy

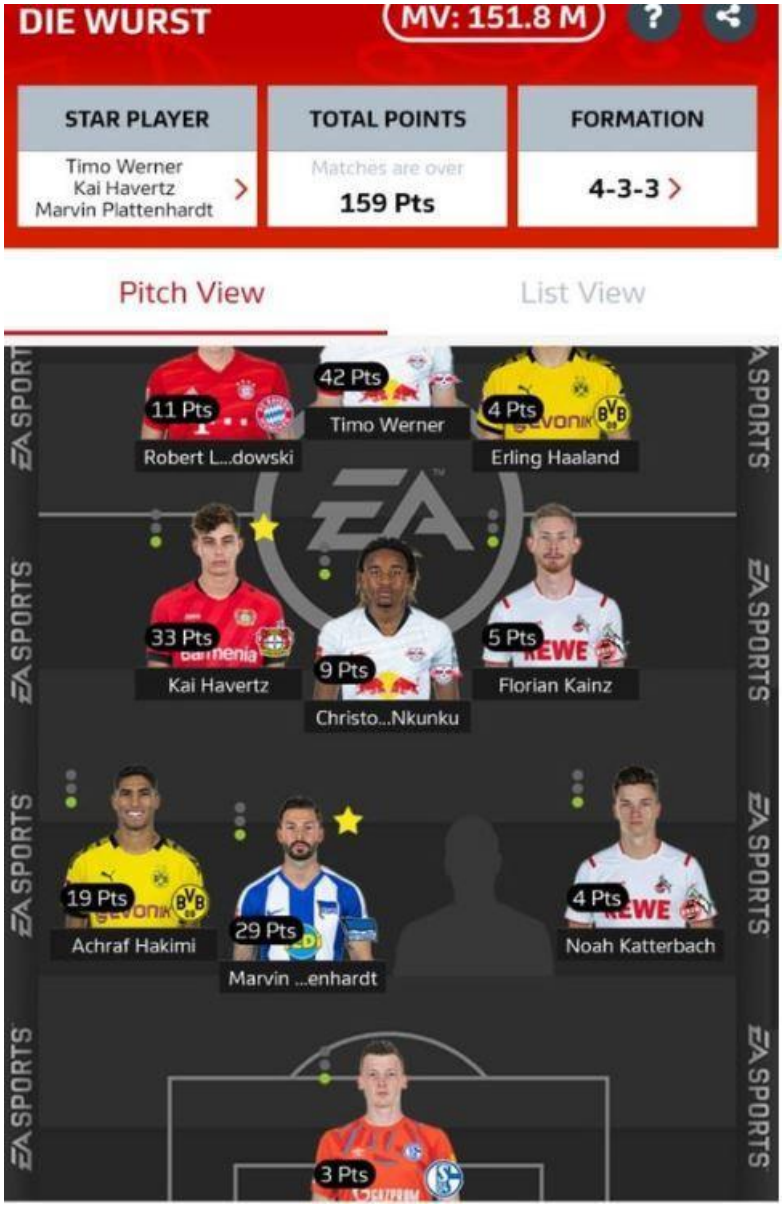
Ovo je službena aplikacija lige prvaka. Princip je isti kao kod „Fantasy Premier League“. Korisnik odabire igrače i na temelju tih igrača skuplja bodove. Aplikacija je dostupna u mobilnoj i web verziji. Na slici 2.2 su vidljiva pravila za igre, koji se nalaze na njihovoj službenoj stranici.



Sl. 2.2 Pravila igra za „UEFA Champions League Fantasy“. (izvor: gaming.uefa.com)

2.3 Fantasy Bundesliga

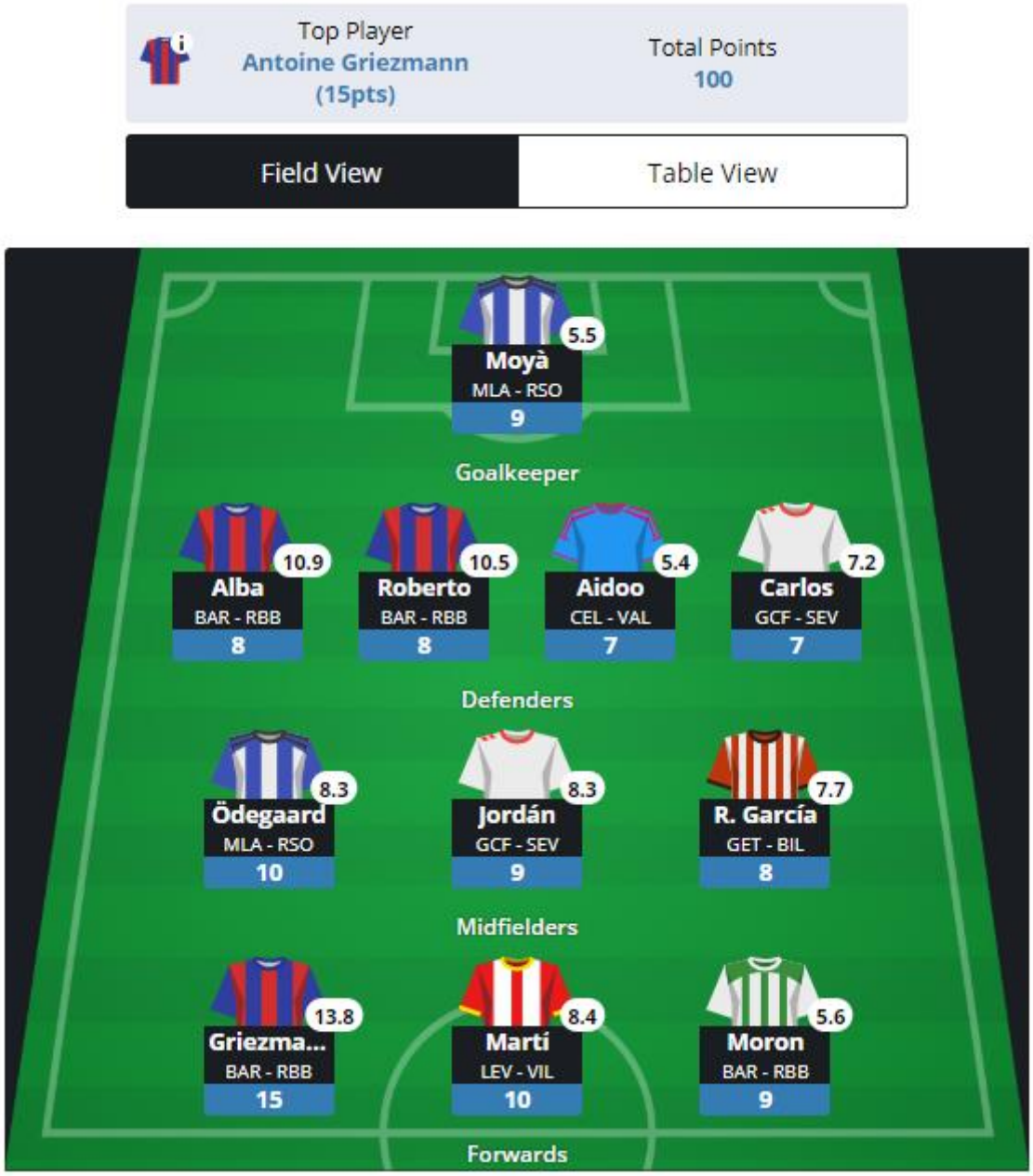
Fantasy Bundesliga je službena aplikacija namijenjena njemačkom tržištu. Vizualno puno bolje izgleda jer su prikazane slike igrača umjesto dresova kao u prethodne dvije aplikacije. Na stranici se nalaze informacije o: statistikama nekih igrača, sveukupni bodovi, i formaciji, te bodovi skupljeni na pojedinačnim igračima. To je vidljivo na slici 2.3.



Sl. 2.3 Prikaz nasumično odabrane ekipe i prikaz njihovih bodova. (izovr: www.fantasyfootballgeek.co.uk)

2.4 La liga fantasy

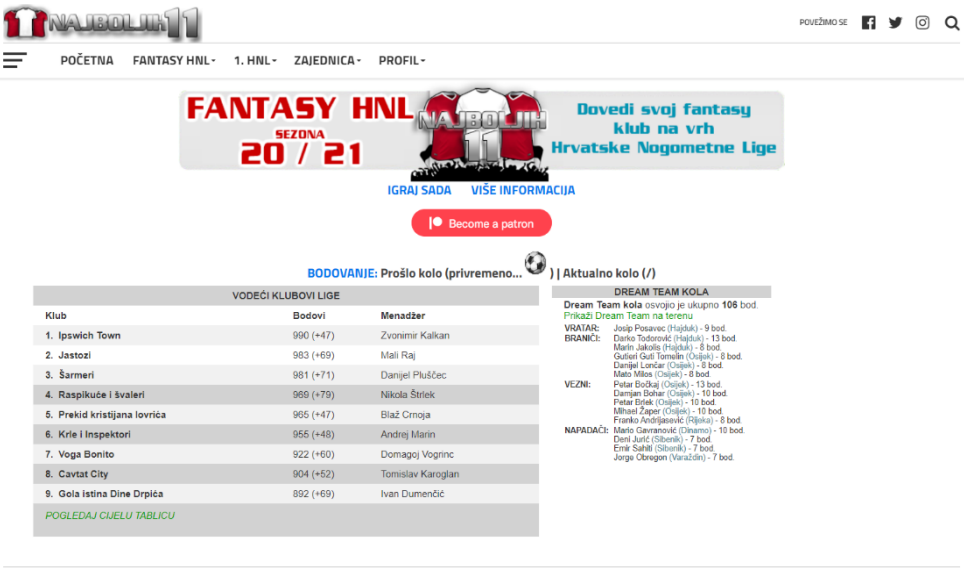
Španjolska verzija natjecateljske lige koja je vrlo slična gore navedenim aplikacijama. Sučelje izgleda najjednostavnije od ostalih rješenja, odnosno sličnih aplikacija. Vizualan prikaz aplikacije je na slici 2.4. Prikazana je korisnikova ekipa s njihovim brojem bodova, sveukupnim brojem bodova i najboljim igračem tog kola.



Sl. 2.4 Prikaz sučelja za Španjolsku natjecateljsku ligu. (izvor: draftgym.com)

2.5 Najboljih jedanaest

Najboljih jedanaest je web aplikacija koja su napravili obožavatelji. Vizualno najlošije izgleda kao što se može vidjeti na slici 2.5. Na slici se vidi da u aplikaciji postoji navigacijska traka, tablica vodećih korisnika i tablica igrača koji su za neko kolo skupili najviše bodova. Stranica je puna grešaka i jako sporo radi.



Sl. 2.5 Web aplikacija „Najboljih 11“. (izvor: www.najboljih11.com)

3. ALATI ZA IZRADU

U ovom poglavlju će pobliže biti nabrojani i opisani svi potrebni alati koji su bili korišteni pri izradi ovog rada. Alati, tehnologije i programski jezici koji su korišteni su: HTML, PHP, phpMyAdmin te WAMP.

3.1 Što je HTML

HTML (eng. HyperText Markup Language) je standardni jezik za označavanje dokumenata dizajniranih za prikaz u web pregledniku. Pomoći mu mogu tehnologije poput CSS-a i skriptni jezik kao što je JavaScript. Web preglednici primaju HTML dokumente s web poslužitelja ili iz lokalne pohrane i generiraju ih u multimedijske web stranice. HTML stranica je napravljena od HTML elemenata. Kod HTML konstrukcija, slike i drugi objekti poput interaktivnih obrazaca mogu se ugraditi u prikazanu stranicu. HTML pruža sredstvo za stvaranje strukturiranih dokumenata označavanjem strukturne semantike za tekst kao što su naslovi, odlomci, liste, poveznice, citati i druge stavke. HTML elementi su označeni oznakama ili tagovima, napisanim pomoću znakova veće je i manje je. Svaki dokument u pravilu treba izgledati identično ostalim, ekstenzija mu je „.html“ ili „.htm“ i u sebi sadržava tri osnovne oznake, a to su: „<html></html>“, „<body></body>“ i „<footer></footer>“. Oznake kao što su „<img></img>“ i „<input></input>“ izravno uvode sadržaj na stranicu. Ostale oznake poput „ “ okružuju i pružaju informacije o tekstu dokumenta i mogu uključivati druge oznake kao podelemente. Preglednici ne prikazuju HTML oznake, ali ih koriste za tumačenje sadržaja stranice. HTML može ugrađivati programe napisane u skriptnom jeziku kao što je JavaScript, što utječe na ponašanje i sadržaj web stranice. Uključivanje CSS-a definira izgled sadržaja.

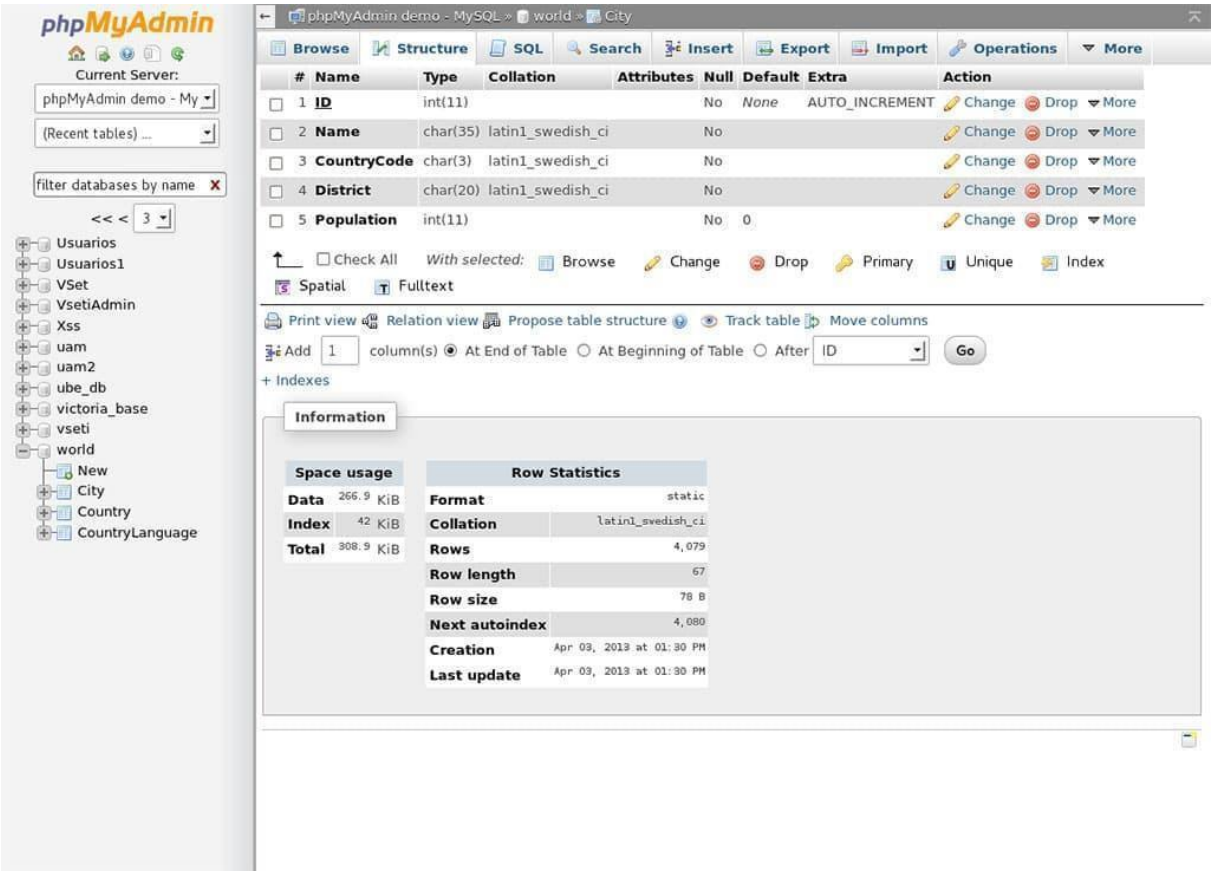
3.2 Što je PHP

PHP ili *Hypertext PreProcessor* je skriptni poslužiteljski programski jezik odnosno skriptni programski jezik za generiranje HTML dokumenta, tj. koda. To znači svatko može skinuti i koristiti PHP kodove sa službene stranice[5], da ih je moguće mijenjati, prilagođavati te dodavati nove funkcije. Iz tih razloga PHP je baš zbog toga jedan od najkorištenijih programskih jezika za stvaranje Internet stranica. PHP omogućuje kreiranje HTML stranica sa svim dinamičkim sadržajem prije nego li je stranica poslana klijentu. PHP koristi template,

odnosno predloške s kojima je omogućeno popunjavanje stranice. Klijent jedino može vidjeti HTML kod, dok je skripta ili sami kod ne vidljivi dio stranice. Sintaksa samog PHP je vrlo slična drugim programskim jezicima kao što je „C++“.

3.3 Što je phpMyAdmin

Web aplikacija i softverski alat, razvijen u PHP-u jeziku za potrebe administriranja „MySQL“ sistema baze podataka u web preglednicima. Aplikacija phpMyAdmin pruža usluge: kreiranje i uklanjanje baze podataka, izvođenje operacija na tablicama i poljima, izvršavanje SQL upita, te upravljanje ključevima, kao što je prikazano na slici 3.1. Zbog široke palete usluga, phpMyAdmin je jedna od najpopularnijih aplikacija i vrlo jednostavan alata za administriranje „MySQL“ baze podataka.



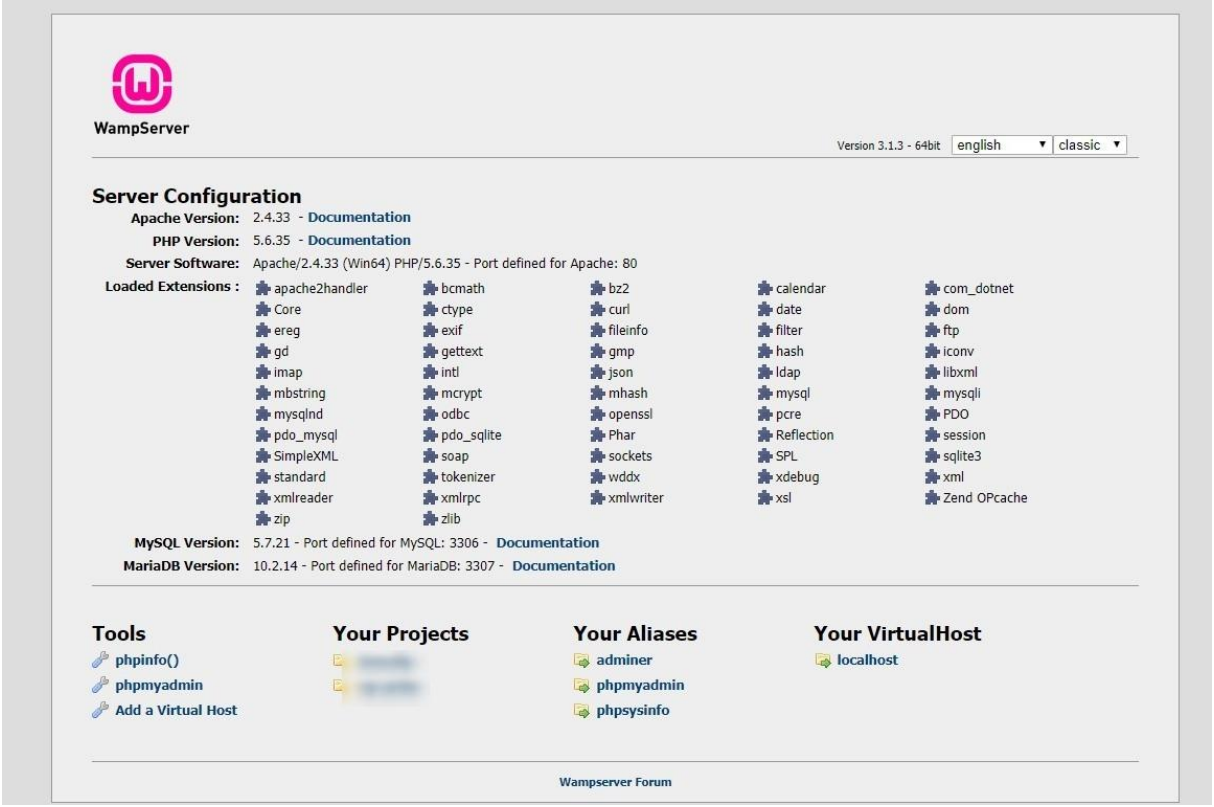
Sl. 3.1 Vjerodostojan prikaz phpMyAdmin aplikacije.

3.4 Što je WAMP

To je softverski paket [6] koji služi za simulaciju internet server na računalu. Pomoću njega se provjerava funkcionalnost internet servera ukoliko se želi napraviti neka stranica ili prezentacija koja će se postaviti na internet. Glavni dijelovi Wamp servera su:

- W - operativni sustav windows
- A - Apache web server koji omogućava da se na računalu istovremeno nalazi lokalni server i korisnik (browser)
- MySQL- sistem koji upravlja racionalnim bazama podataka
- PHP - skriptni jezik koji pomaže u izradi dinamičkog web sadržaja

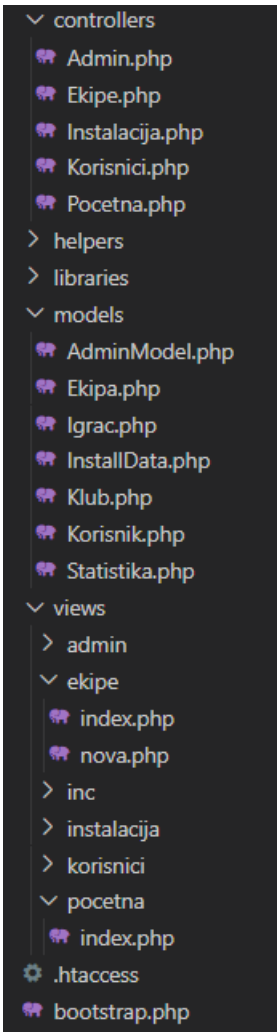
Dodatni programi phpMyAdmin, napravljen u PHP-u koji služi za modificiranje i kreiranje baze podataka također mogu biti uključeni u paketu. Vizualni prikaz WAMP konfiguracije je na slici 3.2.



Sl. 3.2 Wamp configuracije

3.5 Što je MVC

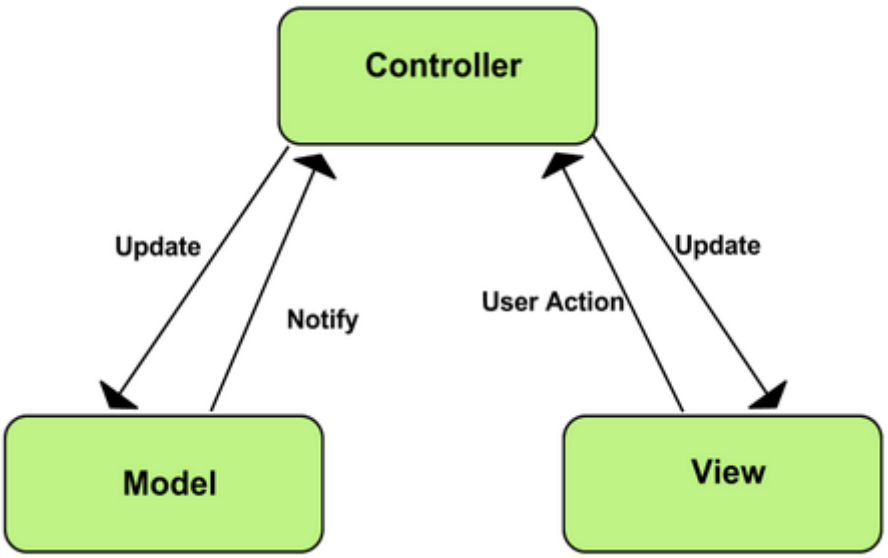
Model View Controller ili MVC [6] je arhitektura web aplikacije koja se koristi u softverskom inženjerstvu za organizaciju dijelova softverske aplikacije. Dijeli se u tri grupe komponenti: model, view ili pogled i controller ili kontroler. Na slici 3.3 je vidljiv prikaz izrađenog projekta.



Sl. 3.3 Izrađeni projekt

Model mora omogućiti pristup podatcima za pregled ili prikupljanje i zapisivanje, a poveznica je između komponente View-a ili pogled i komponente Controller ili kontrolera. Jedan od važnijih aspekata modela je taj da je "slijep". Pod "slijep" se misli da model nema vezu ili uvid u to što se događa s podatcima koji se prosljeđuju iz kontrolera ili pogleda, niti model to zanima. Modelova jedina svrha je obrada podataka koji se nalaze u njemu ili pripremi podatke koje treba

proslijediti ostalim dijelovima. Pogled je taj koji je zadužen za prikaz podataka zatraženih od modela i isporučivanje istih krajnjem korisniku. Tradicionalno u web aplikacijama izrađenih pomoću MVC-a, pogled je dio sustava u kojem se generira i prikazuje HTML. Pogled također reagira na korisnikove akcije, zatim iz tih akcija nastavlja interakciju s kontrolerom. Jednostavan primjer toga je gumb koji generira pogled koji korisnik klikne i pokrene neku radnju u kontroleru. Svrha kontrolera je upravljanje podatcima koje korisnik unosi ili šalje i sukladno s modelom ih ažurira. Bez korisnikovih interakcija kontroler nema svrhu. To je jedini dio s kojim bi korisnik trebao vršiti interakciju. Za kontroler se može reći i da je sakupljač informacija koji ih zatim prosljeđuje modelu da bude organiziran za pohranu i ne sadrži nikakvu drugu logiku osim potrebne za prikupljanje unosa. Na slici 3.3. je vidljiv dijagram toka MVC koncepta.



Sl. 3.3 Dijagram prikaz MVC koncepta.

4. STRUKTURA APLIKACIJE

U ovom poglavlju bit će opisana i napisana sva uputstva i koraci korištenja aplikacija. Struktura je podijeljena u šest koraka ili dijelova; instalacija aplikacije, izrada korisničkog računa, prijava, postavljanje ekipe, glavna stranica i admin. Svaki od ovih koraka ili dijelova o radu aplikacije, te kako se korisnik može njom ispravno služiti je detaljno opisan u ovome poglavlju.

4.1 Instalacija aplikacije

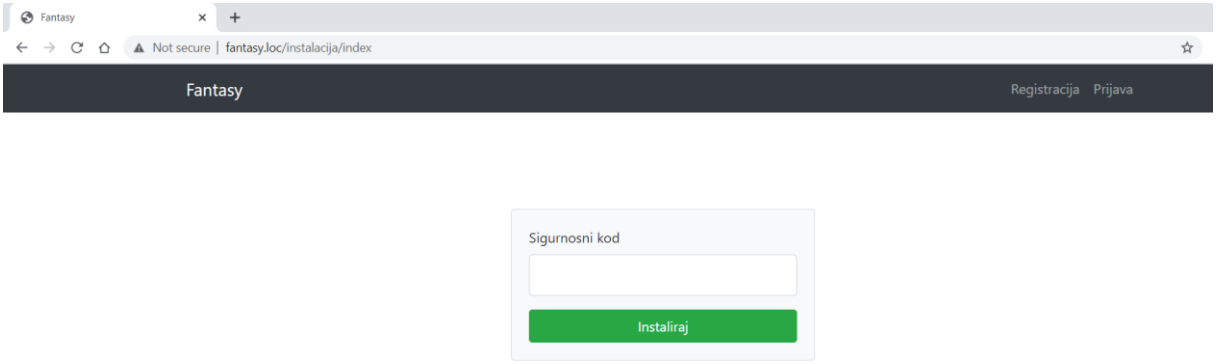
Prije početka rada s aplikacijom, potrebno je uključiti WAMP za aktivaciju emulatora „Apache“ web server na lokalnom računalu. Kada je WAMP uključen, potrebno je podesiti neke postavke aplikacije.

Nakon uspješno postavljenog WAMP-a, u URL (engl. Uniform Resource Locator) se unosi ruta do projekta „http://fantasy.loc/“, koja preusmjeri ili odvede do „app/controllers/Pocetna“. Tamo se prvo izvršava funkcija „__construct()“ u kojoj se provjerava postojanje korisnika, odnosno je li korisnik prijavljen. Ako postoji korisnik koji je prijavljen, vrši se preusmjeravanje na „app/controllers/Admin::index()“ gdje se provjerava je li korisnik koji je prijavljen admin ili nije. Ako prijavljeni korisnik nije admin radi se preusmjeravanje na *login page* „app/controllers/Korisnici:prijava()“. Tamo se prvo obavi provjera postoji li prethodno instalirana aplikacija. Utvrdili se ne postojanje, aplikacija preusmjerava na „app/controllers/Instalacija::indeks()“. Funkcija „__construct()“ u klasi Instalacija šalje na klasu u modelu „InstallData“, gdje se nalaze svi podatci o klubovima i igračima o Prvoj hrvatskoj nogometnoj ligi, te gdje se kreira, instalira i postavlja baza i svi potrebni podatci u nju. U funkciji „index()“, koja je prikazana na slici 4.1, provjerava se je li prijavljeni korisnik admin, ako je, aplikacija preusmjerava na url „fantasy.loc/instalacija/instalacija“, te ako je došlo do nekakve pogreške odnosno ako korisnik nije admin, neka renderira view „instalacija/index“.

```
public function index()
{
    if($_SESSION["prijavljeni_korisnik_id"] == 1) redirect("instalacija/instalacija");
    $this->view("instalacija/index");
}
```

Sl. 4.1 Prikaz funkcije „index()“.

U funkciji „index()“ se nalazi „$this->view("instalacija/index");“. S tom metodom se izvršava skripta „app/views/Instalacija/index.php“ u kojoj se nalazi HTML / JAVASCRIPT / CSS + PHP. Stranica instalacije „fantasy.loc/instalacija“ vidljiva je na slici 4.2.

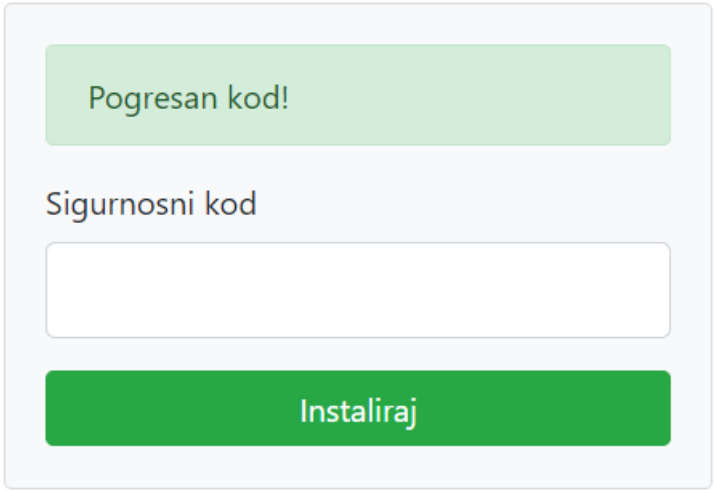


Sl. 4.2 Prikaz view-a „instalacija/indeks“.

Sami HTML je podijeljen u tri datoteke:

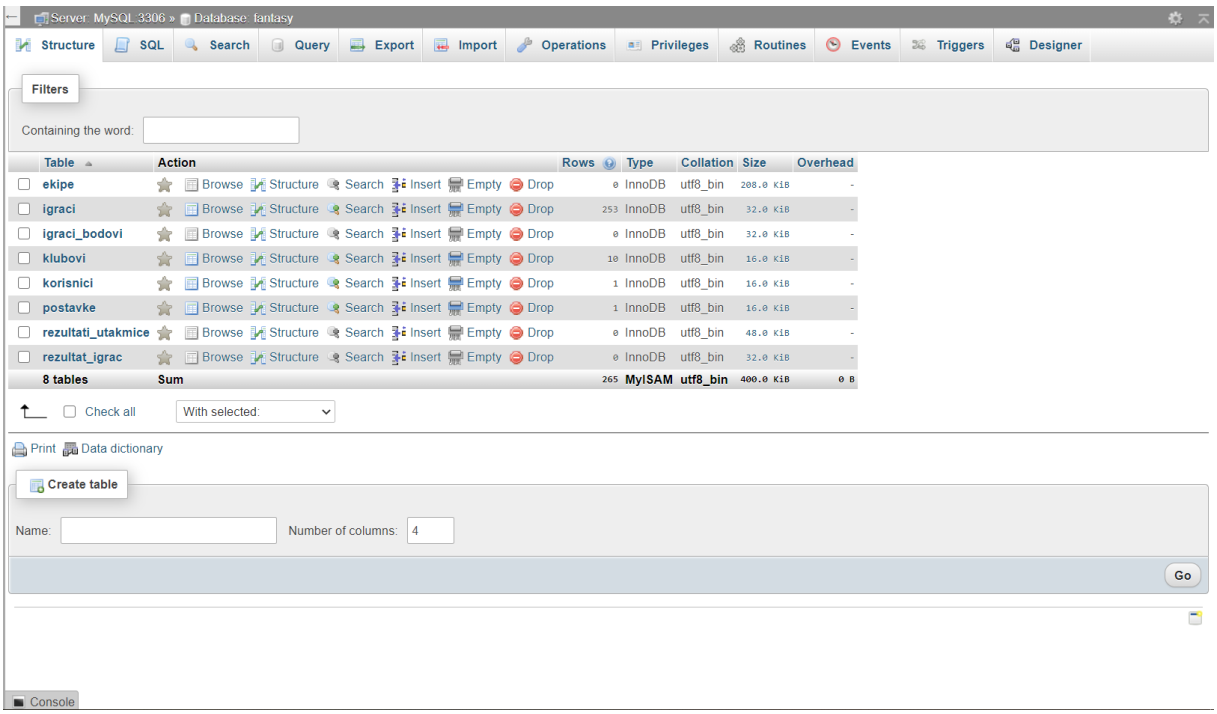
- „app/views/inc/header.php“
- „app/views/inc/footer.php“
- „app/views/inc/navbar.php“

Svaka view skripta u sebi sadrži osnovnu strukturu HTML dokumenta uz PHP skriptne funkcije. Za uspješnu instalaciju potrebno je upisati ispravan sigurnosni kod, odnosno kod prethodno definiran u postavkama same aplikacije. Ako je obrazac sigurnosnog koda netočan ili ako je krivo napisan korisniku će se iznad obrasca za unos sigurnosnog koda pojaviti poruka da je kod pogrešno napisan, vidljivo na slici 4.3.

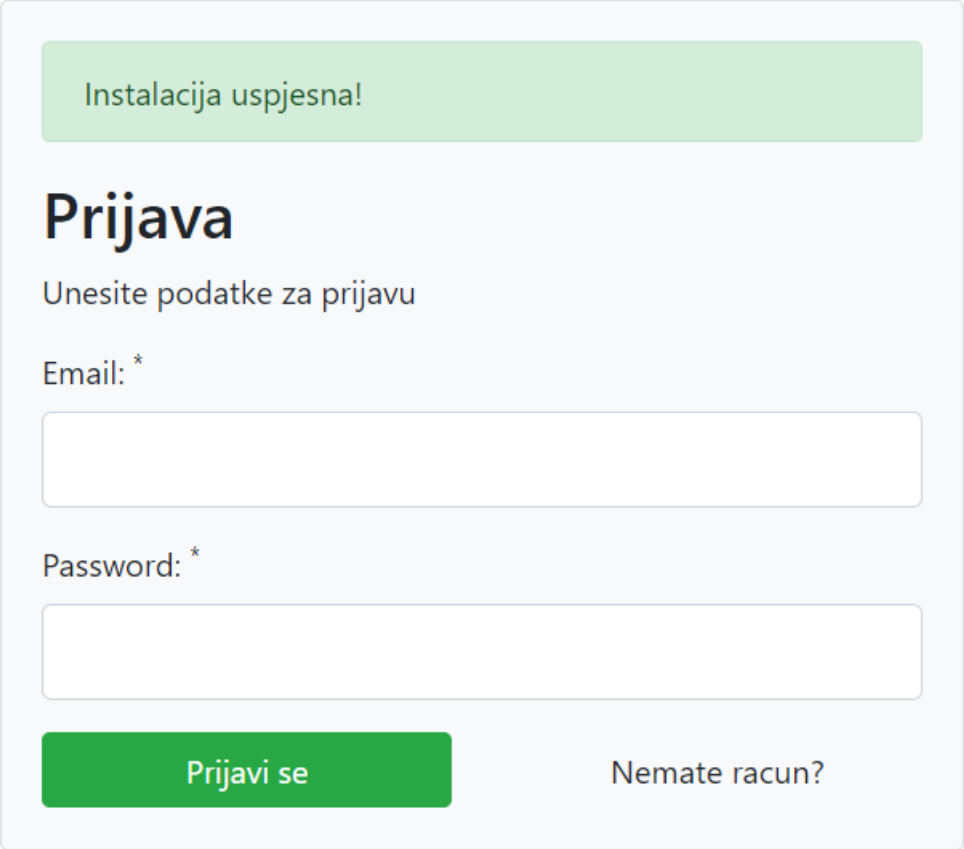


Sl. 4.3 Vizualni prikaz poruke „Pogresan kod!“.

Kada je kod upisan i kliknut gumb „Instaliraj“, u bazi će se generirati sve potrebne tablice i podatci koji su potrebni za ispravan rad ove aplikacije. Rezultat ovog rada vidljiv na slici 4.4. Nakon generiranja podataka, korisnik je preusmjeren na stranicu za prijavu korisnika, prikazano na slici 4.5.



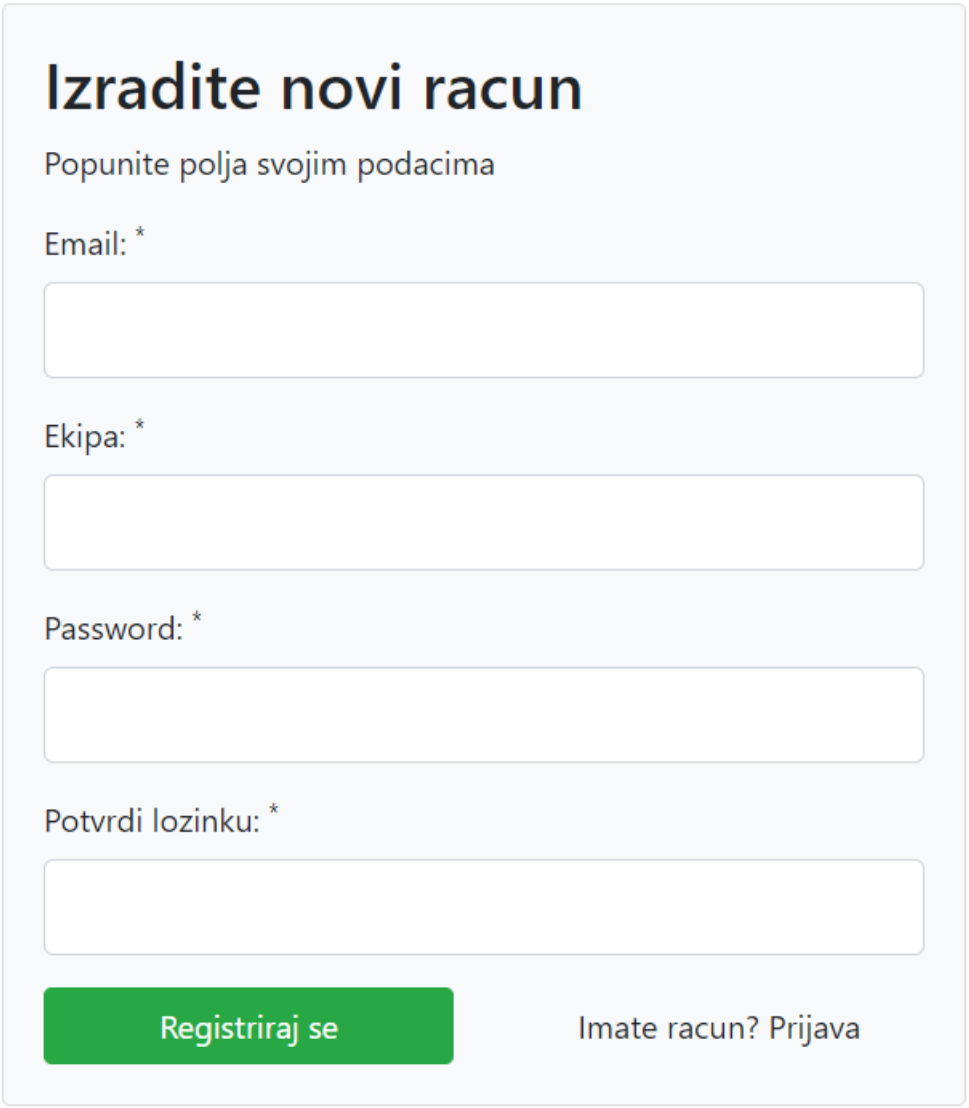
Sl. 4.4 Prikaz pune baze podataka



Sl. 4.5 View „korisnici/prijava“ sa porukom „Instalacija uspjesna!“.

4.2 Izrada korisničkog računa

Prije prijave, ako ne postoji račun, korisnik ga prvo mora napraviti. Izrada računa ili registracija je moguća klikom na jedan od 2 gumba: „Registracija“ gumb, koji se nalazi na vrhu stranice u navigacijskoj traci ili na gumb „Nemate račun?“. Klikom na jedan od ta dva gumba, korisnik će biti preusmjeren na stranicu rute „fantasy.loc/korisnici/registracija“, na kojoj se nalazi forma za izradu novog računa. Prikaz stranice s praznom formom je vidljiv na slici 4.6.



Sl. 4.6 Forma za registraciju.

Sva polja moraju biti ispravno popunjena kako ne bi došlo do neke pogreške. Na primjer u polje ispod Email oznake, korisnik treba upisati valjanu adresu elektroničke pošte, inače će vratiti poruku „Pogresan mail format!“, odnosno unese li korisnik email adresu koja se već koristi i s kojom je već netko prije ostvario registraciju. Vratit će poruku „Email već postoji!“. Postupak provjere validacije vidljiv je na slici 4.7.

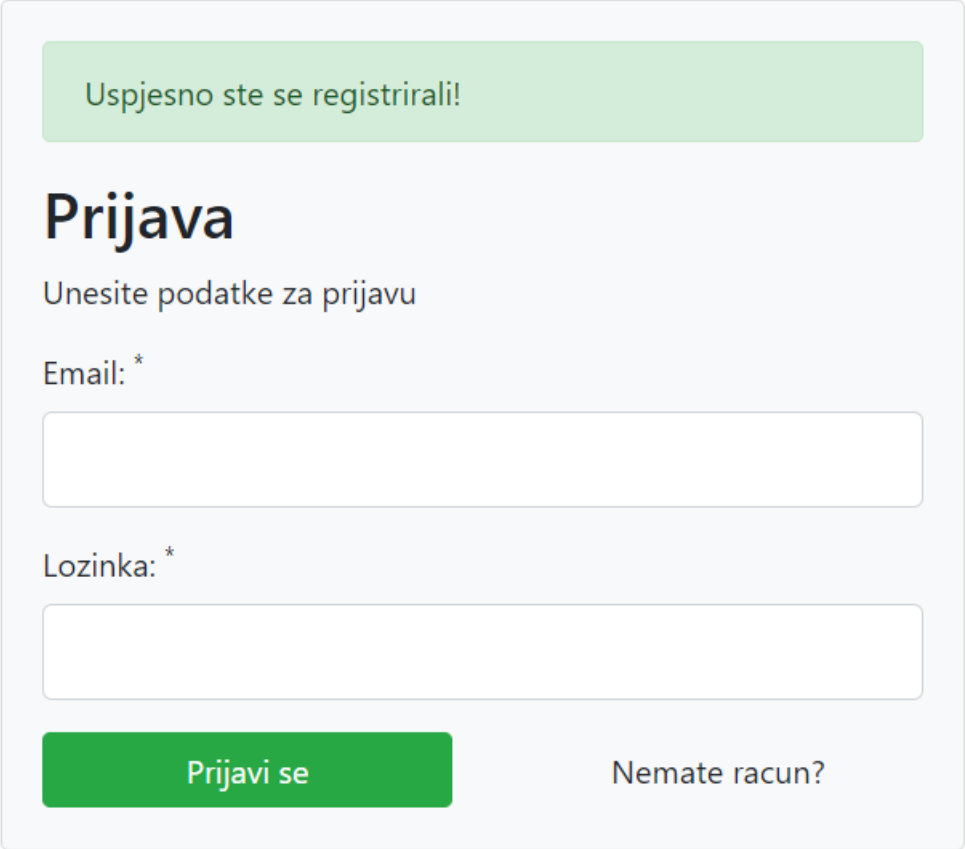
```php
if (empty($podaci["email"])) {
    $podaci["email_err"] = "Unesite email!";
} elseif (!preg_match("^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$^", $podaci["email"])) {
    $podaci["email_err"] = "Pogresan mail format!";
} else {

    if ($this->korisnikModel->pretragaKorisnikaEmail($podaci["email"])) {
        $podaci["email_err"] = "Email vec postoji!";
    }
}
```

Sl. 4.7 Postupak validacije email polja.

Aplikacija sprema upisane podatke u varijable, te vrši završnu provjeru i kriptira zaporku kako bi korisniku bila osigurana njegova privatnost, to jest kako ne bi došlo do krađe podataka. Kada korisnik sve podatke ispravno napiše i klikne gumb „Registriraj se“, ti se podatci šalju POST zahtjevom na isti URL „fantasy.loc/korisnici/registracija“, no razlika je što se u ovom slučaju funkcija izvrši na drugi način, odnosno detektira se POST zahtjev i na osnovu POST podataka i modela, ti se podatci spremaju u bazu. POST podatci su podatci koje je korisnik upisao u polja: „Email“, „Ekipa“, „Lozinka“, i „Potvrdite lozinku“.

4.3 Prijava

Nakon što korisnik uspješno popuni obrazac za registraciju, te klikne gumb „Registriraj se“ , bit će preusmjeren, odnosno vraćen na stranicu rute „fantasy.loc/korisnici/prijava“ na kojoj će biti view s potpuno praznim poljima za unos teksta i porukom kako je registracija korisnika uspješna, to jest, pisat će „Uspjesno ste se registrirali!“. Poruka je vidljiva na slici 4.8.



Sl. 4.8 Forma prijave nakon registracije.

Prijava radi vrlo slično kao i registracija. Korisnik treba upisati „Email“ i „Lozinku“ koju je upisao pri registraciji. Ukoliko to korisnik ne napravi i pritisne gumb „Prijavi se“, ispisat će se poruke greške. Dio koda u koji je zadužen za to nalazi se na slici 4.9.

```
if (empty($podaci["email"])) {
    $podaci["email_err"] = "Please enter email!";
}

if (empty($podaci["lozinka"])) {
    $podaci["lozinka_err"] = "Please enter password!";
}

if (!$this->korisnikModel->pretragaKorisnikaEmail($podaci["email"])) {
    $podaci["email_err"] = "No user found!";
}
```

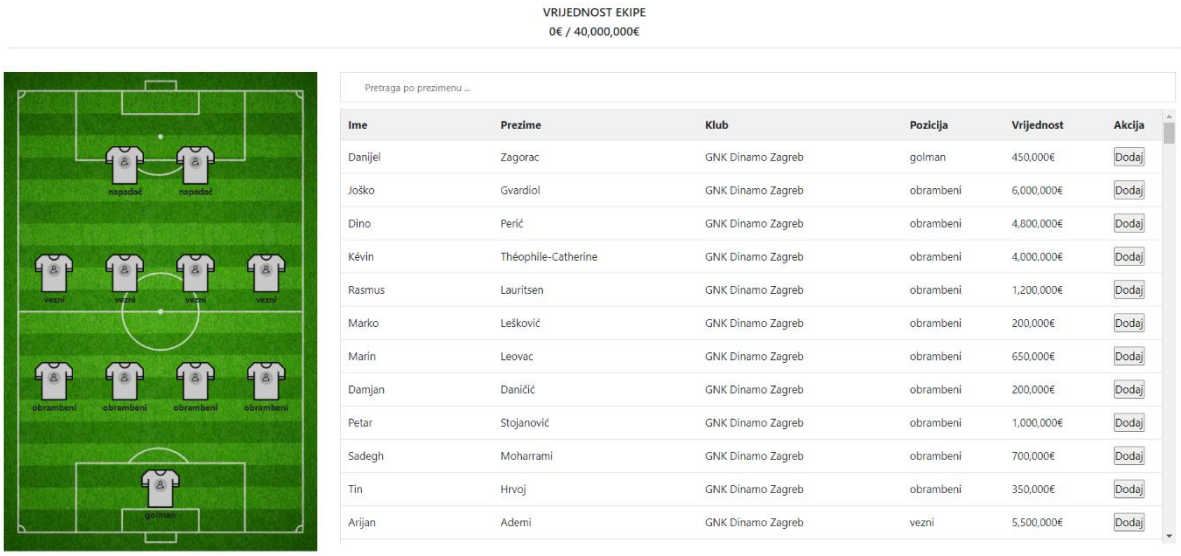
Sl. 4.9 Provjera valjanosti prilikom prijave u kodu.

Kada korisnik unese „Email“ i „Lozinku“ i klikne na gumb „Prijavi se“, vrši se provjera valjanosti podataka iz obrasca. Ako validacija podataka prođe, kreira se nova sjednica

korisnika, te se korisnika preusmjeri na početnu stranicu, odnosno stranicu rute „fantasy.loc/pocetna/index“ gdje se vrši provjera prijave, odnosno je li trenutni korisnik koji je prijavljen admin, je li korisnik prijavljen i je li korisnikova ekipa potpuna. Ukoliko je korisnik po prvi puta prijavljen, odnosno ako nije admin i ako mu ekipa nije potpuna, bit će preusmjeren na stranicu rute „fantasy.loc/ekipe/nova“.

4.4 Postavljanje ekipe

Nakon prijave korisnika, aplikacija će provjeriti je li ekipa korisnika potpuna, odnosno ima li korisnik svojih idealnih jedanaest igrača. Ukoliko nema, korisnik će biti preusmjeren na „fantasy.loc/ekipe/nova“ gdje se nalazi sučelje za dodavanje igrača u ekipu, vidljivo na slici 4.10.



Sl. 4.10 Prikaz stanice za pravljenje ekipe.

Na vrhu stranice se nalazi oznaka u kojoj se nalazi informacija o budžet ekipe, odnosno koliko svaki korisnik može potroši prilikom odabiranja igrača. Tu se vrši provjera kako korisnik ne bi prekoračio zadani budžet, te ukoliko ga prijeđe, pojavit će se poruka „Nije moguće dodati igrača jer vrijednost ekipe mora biti maksimalno 40,000,000€!“.

S lijeve strane nalazi se slika terena, s praznim pozicijama odnosno praznim dresovima, koji se prilikom dodavanja novog igrača zamjene s dresom kluba odakle igrač dolazi, odnosno za koji klub igra. Slika, te sama mreža igrača na terenu odrađena je kroz CSS, vidljivo na slici 4.11

```
.frames {
    width: 90%;
    height: 90%;
    display: grid;
    grid-template-columns: 1fr 1fr 1fr 1fr;
    grid-template-rows: 1fr 1fr 1fr 1fr;
    grid-template-areas:
    ".        napad1   napad2   ."
    "srednji1 srednji2 srednji3 srednji4"
    "obrana1  obrana2  obrana3  obrana4"
    ".        golman   golman   .   ";
}
```

Sl. 4.11. Prikaz dijela CSS-a zaduženog za prikaz mreže igrača.

Dio stranice zauzima popis svih igrača koji su spremljeni u bazu, te se ovdje prikazani u obliku tablice. Tablica se sastoji od 6 stupaca;

- Ime (u kojem se nalazi ime igrača)
- Prezime (u kojem se nalazi prezime igrača)
- Klub (u kojem se nalazi klub za koji igrač igra)
- Pozicija (u kojem se nalazi pozicija na kojoj igrač igra)
- Vrijednost (u kojem se nalazi vrijednost igrača)
- Akcija (u kojem se nalazi gumb za dodavanje igrača)

Ukoliko korisnik želi dodati određenog igrača, akcija koju treba napraviti je kliknuti gumb „Dodaj“. Ako je dostupna pozicija u ekipi odabranog igrača, u bazu se dodaje igrač i ažurira se vrijednost ekipe, tj. smanji se budžet za onoliko koliko igrač košta. Kada korisnik pritisne gumb, prikazana mu je stranica „fantasy.loc/ekipe/nova“ s dodanom slikom dresa na već spomenutu mrežu. Umjesto gumba „Dodaj“ prikazuje gumb „Ukloni“ u ćeliji tablice upravo dodanog igrača. Za lakše pretraživanje igrača iznad tablice svih igrača nalazi se polje za unos teksta s ugrađenom tražilicom koja pretražuje po prezimenima igrača, što će korisniku uvelike olakšati korištenje aplikacijom. Pretraživanje po prezimenu odrađeno je kroz JavaScript, prikazanom na slici 4.12.

```javascript
function myFunction() {
    var input, filter, table, tr, td, i, txtValue;
    input = document.getElementById("myInput");
    filter = input.value.toUpperCase();
    table = document.getElementById("myTable");
    tr = table.getElementsByTagName("tr");

    for (i = 0; i < tr.length; i++) {
        td = tr[i].getElementsByTagName("td")[1];
        if (td) {
            txtValue = td.textContent || td.innerText;
            if (txtValue.toUpperCase().indexOf(filter) > -1) {
                tr[i].style.display = "";
            } else {
                tr[i].style.display = "none";
            }
        }
    }
}
```

Sl. 4.12 Cijeli JavaScript dio za pretraživanje igrača po prezimenu.

Igrača je moguće ukloniti pritiskom na gumb „Ukloni“. Pritiskom na gumb serveru se šalje GET zahtjev sa „id“-em igrača kojeg korisnik želi ukloniti. Ukoliko korisnik pritisne na gumb, prikaže mu se stranica „fantasy.loc/ekipe/nova“ s porukom kako je odabrani igrač iz njegove ekipe uklonjen, te se umanji vrijednost, tj. budžet ekipe i ažurira se baza podataka. Funkcija zadužena za to je prikazana na slici 4.14.

```php
public function ukloniIgraca($igracId)
{
    $mojaEkipa = $this->ekipaModel->mojaEkipa();
    $vrijednostIgraca = $this->igracModel->vrijednostIgracId($igracId);
    if(!$this->ekipaModel->provjeriIgracDodan($igracId)){
        flash("ukloni", "Igrač kojeg želite ukloniti se ne nalazi u vašoj ekipi!");
        redirect("ekipe/nova");
    }
    if($this->ekipaModel->ukloniIgraca($igracId)){
        $vrijednostEkipe = $mojaEkipa->vrijednost - $vrijednostIgraca;
        $this->ekipaModel->azurirajVrijednostEkipe($vrijednostEkipe);
        flash('uklonjen', "Uklonili ste igrača!");
        redirect('ekipe/nova');
    }

}
```

Sl. 4.14 Prikaz funkcije za uklanjanje igrača

4.5 Glavna stranica

Kada je korisnik uspješno napravio svoju ekipu, bit će automatski preusmjeren na „fantasy.loc/pocetna“. Na stranici su korisniku vidljivi ekipa koju je sastavio i pet tablica:

- Ljestvica (Poredak ekipa)
- Rezultati zadnjeg kola
- Top deset igrača
- Zadnje kolo
- Sva kola

Unutar tablice Ljestvica (Poredak ekipa) nalazi se svi registrirani korisnici, s informacijom o njihovom imenu ekipe, te ukupnom broju bodova. Ta je tablica vidljiva na slici 4.15. Korisnici u toj tablici, poredani su od najbolje ekipe prema najlošijoj. U drugoj tablici, nalazi se informacija o posljednje dodanim utakmicama u posljednje odigranom kolu. Unutra se nalazi imena domaćina koje su to kolo igrali, gostiju, te krajnji ishod utakmice, tj. rezultat. Unutar treće tablice nalazi se lista top deset igrača, top deset u smislu koji igrač ima najveći broj bodova. Unutar tablice se nalazi ime i prezime igrača, te ostvareni sveukupni bodovi u sezoni. U četvrtoj tablici se nalazi informacije o broju bodova korisnikove ekipe iz zadnjeg kola. Informacije su raspoređene u 3 stupca: ime, prezime i bodovi. Posljednja tablica je vrlo slična četvrtoj, gotovo identična, no jedina razlika između, te dvije tablice je ta što se ne gledaju bodovi iz prošlog kola nego broj bodova pojedinog igrača u cijelom prvenstvu, tj. svim do tad odigranim kolima.

LJESTVICA

Poredak ekipa

#	Ekipa	Ukupno bodova
1	Test_Ekipa	41
4	FCLosi	30
2	Nova_Ekipa	24
5	FCDobri	23
3	Dobri	0

Sl. 4.15 Vizualni prikaz tablice poretka ekipa

Slično poput stranice sa sastavljanjem ekipe nalazi se slika terena sa svim igračima dodanim u ekipu, gdje je vidljivo njihovo ime i prezime, te grb kluba na dresu iz kojeg dolaze, odnosno za koji igraju.

Na stranici je također vidljiva oznaka u kojoj se nalazi informacija o sveukupnom broju bodova korisnikove ekipe. Pravila igre ili sistem bodovanje u obliku teksta je vidljiv na glavnoj stranici. Na stranici se također nalaze i neki zanimljivi podatci vezani uz Prvu hrvatski nogometnu ligu pod rubrikom „Jeste li znali“.

Ukoliko se korisnik želi odjaviti to mu je omogućeno klikom na gumb u navigacijskoj traci gdje korisnik mora odabrati gumb „Odjava“. Sama odjava je pobliže objašnjena u poglavlju Admin.

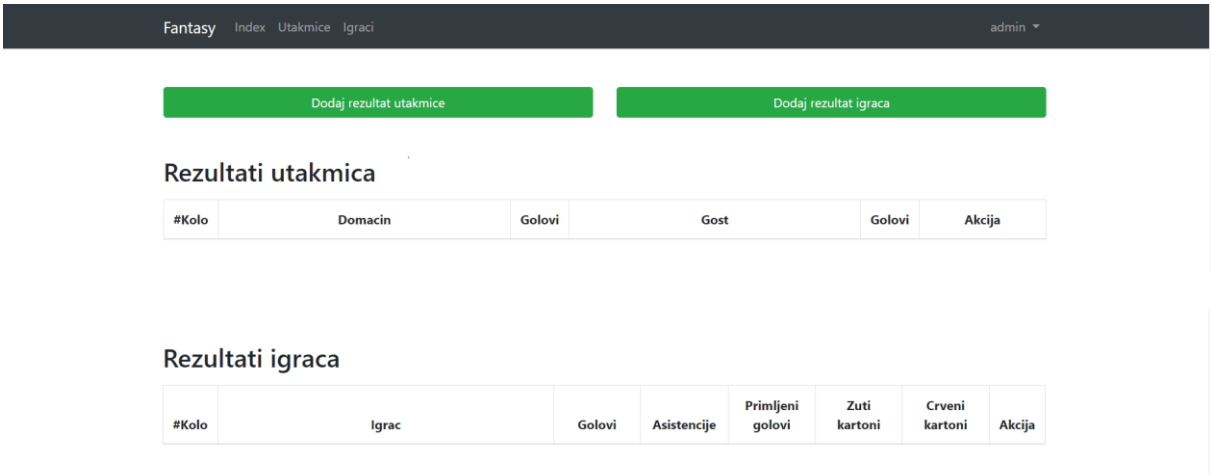
4.6 Admin

Prilikom instalacije aplikacije, kreira se korisnik imena „admin“, s lozinkom koja je definirana u klasi „InstallData“.

```php
private function dodajKorisnike()
{
    $lozinkaAdmin = password_hash('admin', PASSWORD_DEFAULT);
    $this->db->query(
        "INSERT INTO korisnici (email, lozinka, ekipa) VALUES
        ('admin@admin.admin', '$lozinkaAdmin', 'admin' )"
    );

    if($this->db->execute()){
        return true;
    } else {
        return false;
    }
}
```

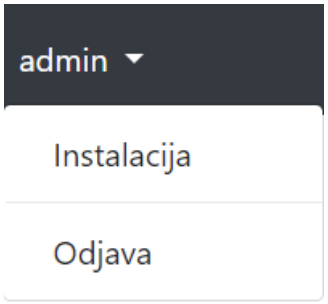
Sl. 4.16 Kreiranje admin korisnika

Slika 4.16 prikazuje funkciju za kreiranje korisnika admin. Za prikaz sučelja za dodavanje rezultata utakmica i igrača potrebno je na stranici za prijavu „http://fantasy.loc/korisnici/prijava“, unijeti podatke za prijavu admin korisnika, te se potom u klasi „Admin“ provjerava valjanost. Preko modela iz baze, uzimaju se podatci potrebni za prikaz dvije tablice koje se mogu vidjeti na view-u „fantasy.loc/admin/index“, te je prikazan na slici 4.17.



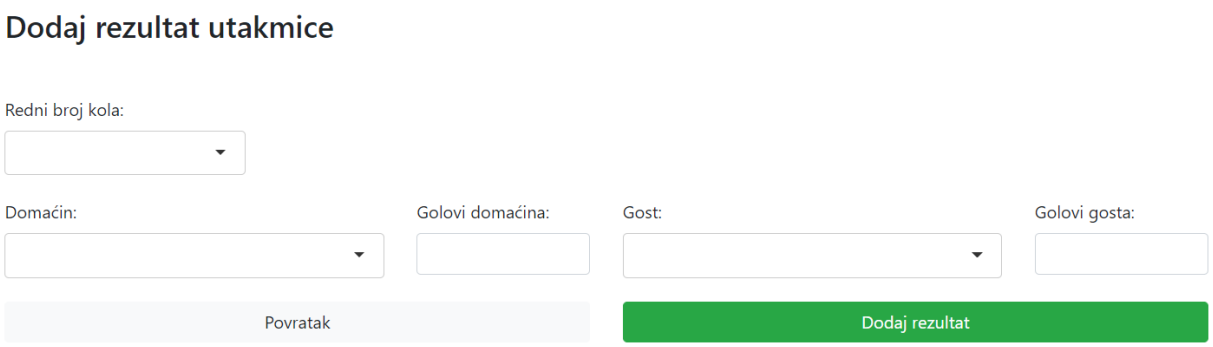
Sl. 4.17 Admin sučelje za dodavanje rezultata.

Prilikom prijave admin korisnika, vidljive su promjene u navigacijskoj traci. Uklonjena su dva gumba, odnosno poveznice, „Registracija“ i „Prijava“, te su dodana tri nova: „Index“, koji ima putanju do „fantasy.loc/admin/index“, potom „Utakmice“ i „Igrači“. Svaka od ove tri poveznice vodi do zadane tablice, ovisno o tome koji gumb korisnik odabere. Dodan je i padajući izbornik na kojem je moguća ponovna instalacija aplikacije ili odjava korisnika admin koji je vidljiv na slici 4.18.



Sl. 4.18 Prikaz sadržaja padajućeg izbornika

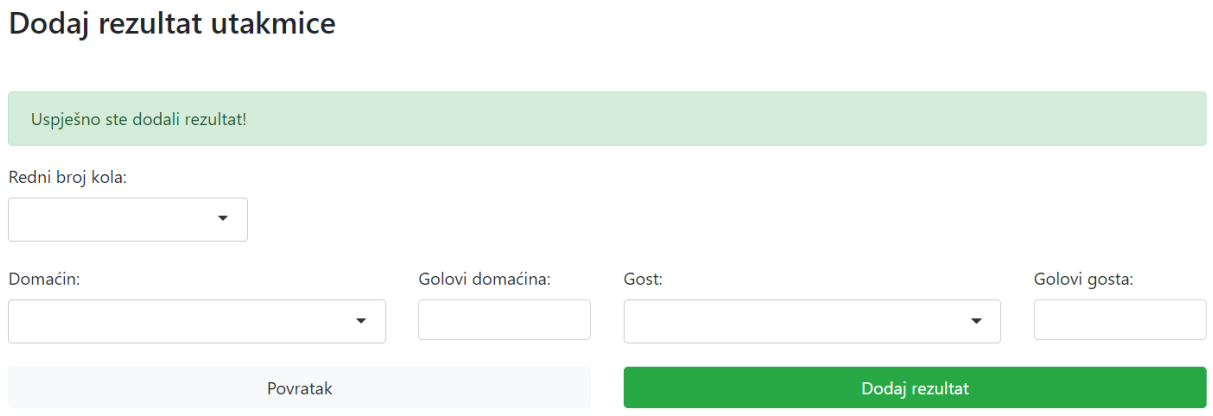
Ukoliko admin korisnik pritisne na gumb „Dodaj rezultat utakmice“, preusmjerava ga se na rutu „fantasy.loc/admin/dodajrezultat“. Funkcija za dodavanje rezultata utakmice radi isto kao i instalacija, registracija, te prijava, a to znači da ukoliko se pristupi kroz URL učitava se stranica praznog obrasca, koja je vidljiva na slici 4.19.



Sl. 4.19 Forma za dodavanje rezultata jedne utakmice

Na stranici se nalaze tri polja za unos teksta s padajućim izbornicima, a to su: „Broj kola“, „Domaćin“ i „Gost“. U prvom polju admin korisnik unosi broj kola, u drugom polju unosi domaći tim utakmice dok u trećem unosi gostujući tim utakmice. Na stranici se također nalazi dva polja predodređena za unos postignutih golova domaćina i golova gostiju. Korisnikovim pritiskom na gumb „Dodaj rezultat“ šalje se POST zahtjev na isti URL s podatcima upisanim u formu. Ukoliko dođe do pogrešno ili netočno napisane forme, korisniku će se prikazati isti view samo s dodatnom porukom u kojem je polju pogriješio.

Korisnikovim ispravnim unosom prikaže se view kontrolera Admin za unos novog rezultata utakmice s porukom o uspješno dodanoj utakmici i praznoj formi za dodavanje utakmice. Poruka je vidljiva na slici 4.20.



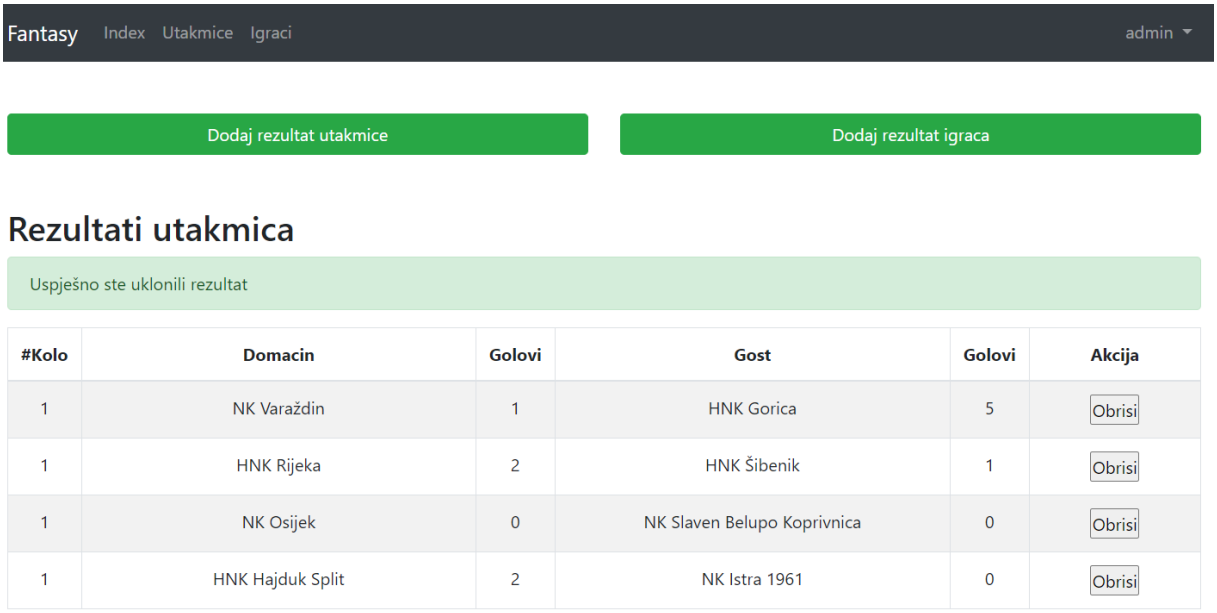
Sl. 4.20 Prikaz stranice nakon uspješno dodane utakmice

Ukoliko korisnik poželi upisati još nekoliko rezultata utakmica, morat će paziti na podudaranje s nekim od prijašnjih unosa jer će biti prikazana poruka „Rezultat s odabranom/im ekipom/ama već postoji!“. Provjera se obavlja u funkciji prikazanoj na slici 4.21.

```php
public function provjeriPostojanostEkipa($podaci)
{
    $this->db->query("SELECT * FROM rezultati_utakmice
    WHERE (domacin = :domacin OR gost = :domacin OR domacin = :gost OR gost =:gost)
    AND kolo_broj = :kolo_broj");
    $this->db->bind(":domacin", $podaci['domacin']);
    $this->db->bind(":gost", $podaci['gost']);
    $this->db->bind(":kolo_broj", $podaci['kolo_broj']);
    $row = $this->db->single();
    if($this->db->rowCount() > 0){
        return true;
    } else {
        return false;
    }
}
```

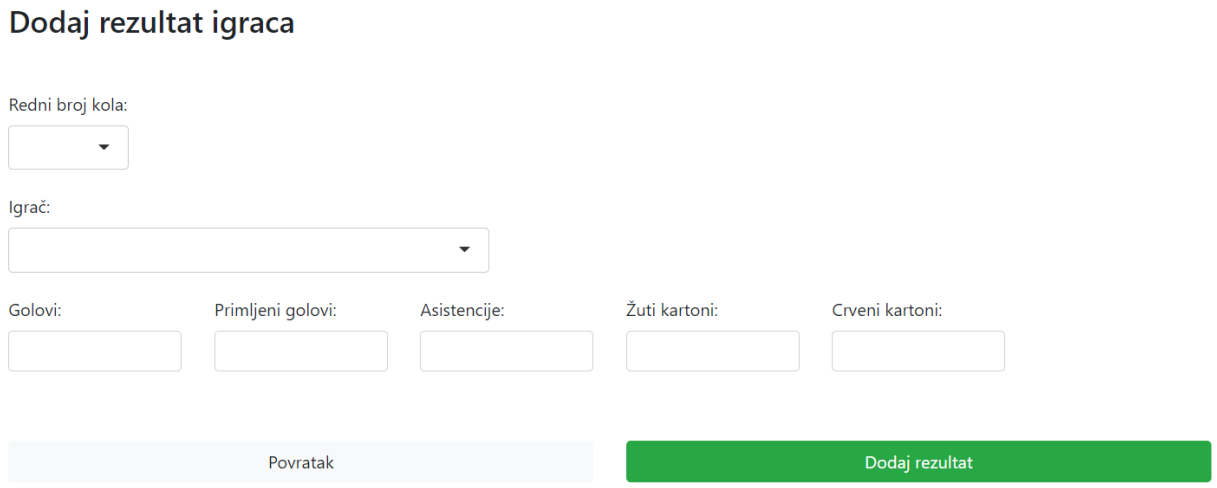
Sl. 4.21 Prikaz funkcije za provjeru postojanosti upisanih podataka u bazi

Nakon uspješno dodanih rezultata utakmice, klikom na gumb „Povratak“, korisnika se preusmjerava na rutu „http://fantasy.loc/admin/index“, sa svim dodanim utakmicama koji su prikazane u redovima u tablici „Rezultati utakmica“. Dodane rezultate je moguće i poništiti odnosno obrisati. U svakom redu se nalazi gumb s poveznicom u kojem piše „Ukloni“. Ukoliko korisni odabere gumb „Ukloni“ šalje se GET zahtjev na URL „fantasy.loc/admin/uklonirezultat/id“, odnosno dinamički je napisan „id“ samog zapisa iz baze podataka, npr. prvi dodani rezultat ima „id=1“, drugi ima „id=2“ i tako dalje. Ukoliko dođe do nekakve pogreške, tj. da ne postoji rezultat utakmica tog „id“-a, ispisati će se poruka „Rezultat nije uklonjen, pokušajte ponovno!“. Ukoliko postoji rezultat određenog „id“-a bit će obrisan, te će biti prikazana poruka o uspješno uklonjenom rezultatu, koja je vidljiva na slici 4.22.



Sl. 4.22 „Admin“ sučelje s porukom o uspješno uklonjenim rezultatom

Pritiskom admin korisnika na gumb „Dodaj rezultat igraca“ preusmjerava ga se na rutu „fantasy.loc/admin/dodajrezultatigraca“. Funkcija za dodavanje rezultata igrača radi na istom principu kao i funkcija za dodavanje rezultata utakmice. Ukoliko pristupi kroz URL bit će prikazana stranica praznog obrasca. Stranica praznog obrasca se nalazi na slici 4.23.



Sl. 4.23 Prikaz stranice nakon uspješno dodanog igrača

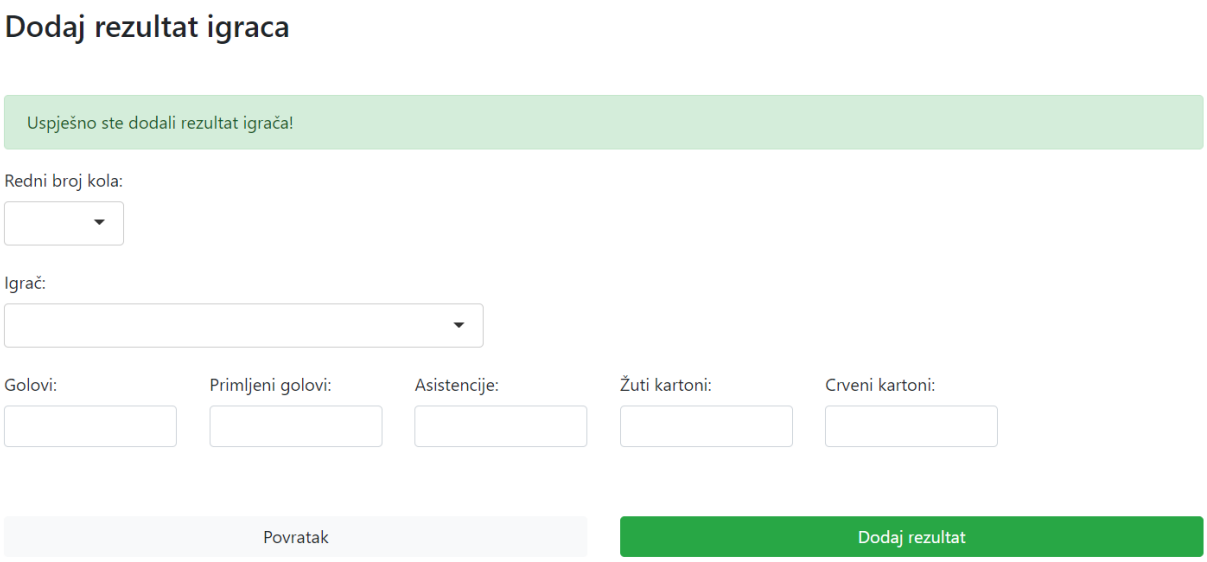
Stranica je strukturirana tako da se na njoj nalazi sedam polja za unos teksta od kojih dva polja u sebi sadrže padajući izbornik i tražilicu za lakše pretraživanje određenog igrača i odabiranje rednog broja kola. Ostalih pet polja je zaduženo za upis koliko je odabrani igrač zabio golova, koliko je njegova ekipa primila golova, koliko je postigao asistencija, žutih, te crvenih kartona. Kao i kod dodavanja utakmica, sve mora biti pravilno i točno upisano jer će inače admin korisnika dočekati poruka s greškom koju napravi prilikom unosa. Ukoliko admin korisnik pritisne na gumb „Dodaj igrača“, poslat će se POST zahtjev na isti URL s podatcima upisanim u obrazac. Dođe li do pogrešno napisanog obrasca, admin korisniku će se prikazivati isti view s porukom na kojem je polju za unos pogriješio.

Prilikom unosa, funkcija za dodavanje rezultata igrača će napraviti izračun za dodavanje bodova upisanim igračima, ovisno o poziciji koju igraju i upisanim parametrima. Naposljetku se ti izračunati bodovi spremaju u bazu podataka. Prikazanoj funkciji na slici 4.24, nije moguće „ručno“ pristupiti, nego se poziva unutar klase za dodavanje rezultata igrača.

```php
private function izracunajBodoveEkipa()
{
    $ekipe = $this->ekipaModel->sveEkipe();
    foreach ($ekipe as $ekipa){
        $podaci = [];
        foreach ($ekipa as $key => $igrac){
            if(
                $key == "id" ||
                $key == "korisnik_id" ||
                $key == "bodovi" ||
                $key == "golman" ||
                $key == "vrijednost"
            ) continue;
            $podaci["bodovi_ekipe"] .= $this->igracModel->igracBodovi($igrac);

        }
        $podaci["ekipa"] = $ekipa->id;
        if($podaci["bodovi_ekipe"]){
            $this->ekipaModel->azuirajBodoveEkipe($podaci);
        }
        unset($podaci);
    }
}
```

Sl. 4.24 Prikaz funkcije za izračunavanje bodova

Ukoliko je sve valjano, odnosno ukoliko admin korisnik napravi ispravan unos, prikazuje se view kontrolera za unos novog rezultata igrača s porukom o uspješno dodanom rezultatu igrača, te stranica praznog obrasca za dodavanje igrača vidljivoj na slici 4.25.

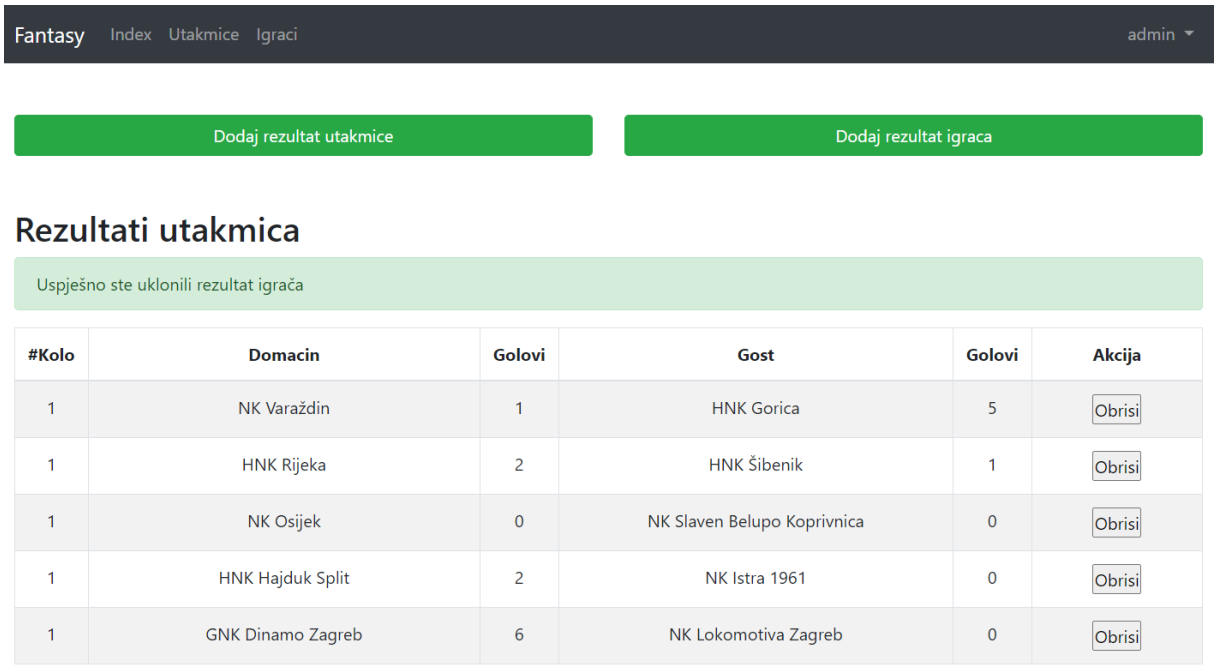


Sl. 4.25 Prikaz stranice nakon uspješno dodanog igrača

#Kolo	Igrac	Golovi	Asistencije	Primljeni golovi	Zuti kartoni	Crveni kartoni	Akcija
1	Mislav Oršić	1	1	0	0	0	Obrisi
1	Robert Murić	1	0	1	0	0	Obrisi
1	Kristian Dimitrov	1	0	0	0	0	Obrisi
1	Mile Škorić	0	0	0	0	1	Obrisi
1	Mario Gavranović	1	0	0	0	0	Obrisi
1	Lovro Majer	2	1	0	0	0	Obrisi
1	Denis Kolinger	0	0	6	1	0	Obrisi
1	Kemal Osmanković	0	0	6	1	0	Obrisi
1	Petar Stojanović	0	0	0	1	0	Obrisi
1	Luka Ivanušec	1	0	0	0	0	Obrisi
1	Komnen Andrić	1	0	0	0	0	Obrisi

Sl. 4.26 Prikaz tablice nakon uspješno dodanih igrača

Prikazana tablica o uspješno dodanim igračima se nalazi na slici 4.26. Uspješnim dodavanjem rezultata igrača, odabirom na gumb „Povratak“, admin korisnika se preusmjerava na rutu „http://fantasy.loc/admin/index“, sa svim prethodno dodanim igračima koji su prikazani u drugoj tablici odnosno tablici zvanoj „Rezultati igrača“. Ukoliko se admin korisnik odluči obrisati rezultat pojedinog igrača, to je moguće napraviti klikom na gumb u redu gdje se željeni igrač nalazi. Gumb „Ukloni“ za razliku od gumba u tablici „Rezultati utakmica“, ne šalje GET zahtjev, nego šalje POST zahtjev kao u slučaju kod instalacije, registracije, te prijave. Na ruti „fantasy.loc/admin/uklonirezultatigraca“ se šalje POST zahtjev, s podacima. Uklanjanjem određenog igrača se ažuriraju se i sami bodovi na tom igraču, tj. ukoliko je uklonjen jedan igrač, umanjit će se ostvareni bodovi na tom igraču ili svesti na nula bodova, te će se korisniku prikazati „http://fantasy.loc/admin/index“ s porukom o uspješno obrisanom rezultatu igrača.



Sl. 4.27 Ruta „fantasy.loc/admin/index“ s porukom o uspješnom uklonjenom rezultatom igrača

Slika 4.27. prikazuje stranicu s porukom o uspješno uklonjenom igraču. Ukoliko nešto pođe po zlu, korisnika će se obavijestiti porukom „Rezultat ne postoji!

5. ZAKLJUČAK

Cilj izrade ove aplikacije je prije svega zabava, te pomalo i znatiželja. Ova web aplikacija je osmišljena za igrače željne natjecanja i pobjeda, te sve one koji žele pokazati svoje nogometno znanje i vještine u vođenju svog "idealnog" tima kojeg su imali priliku sami formirati. Osim toga cilj joj je približiti Hrvatsku nogometnu ligu prosječnim korisnicima koji ju vjerno prate i imaju svoju viziju pobjedničke momčadi izvan pravog nogometnog terena. Aplikacija je napisana je u programu Visual Studio Code, te su korišteni HTML, PHP, JavaScript, CSS i drugi. Aplikacijom je vrlo lako upravljati, te je namijenjena korisnicima svih uzrasta.

LITERATURA

[1] „*https://www.transfermarkt.com*“, službena web stranica Transfermarketa

[2] „*https://fantasy.premierleague.com*“, službena web stranica Fantasy Premier lige

[3] „*https://gaming.uefa.com/en/uclfantasy*“, službena web stranica UEFA fantasy lige

[4] „*https://hns-cff.hr/natjecanja/hrvatski-telekom-prva-liga*“, službena web stranica Hrvatski Telekom Prve lige

[5] „*https://www.najboljih11.com*“, web stranica domaće natjecateljske lige

[6] „*https://www.php.net/docs.php*“ , službena web stranica PHP dokumentacije

[7] „*https://www.wampserver.com/en*“, službena web stranica WAMP-a

SAŽETAK

U radu je opisana funkcionalnost web aplikacije za simulaciju nogometne lige. Navedene su tehnologije i jezici koji su se koristili tijekom izrade web aplikacije. Objašnjen je postupak izrade aplikacije korištenjem MVC programskog okvira te je dan pregled kako se koristi aplikacija. Opisan je način rada aplikacije. Način rada, korištenje aplikacije i postupak izrade potkrijepljen je slikovnim materijalom. Aplikacija je funkcionalna i spremna za korištenje.

Ključne riječi: CSS, HTML, JavaScript, MVC, PHP, simulacija nogometne lige, Web simulacija

ABSTRACT

WEB APPLICATION FOR COMPETITION LEAGUE SIMULATION

The paper describes the functionality of a web application for football league simulation. The technologies and languages used during the development of the web application are listed. The process of creating an application using the MVC program framework is explained and an overview of how to use the application is given. The method of work and the production process are supported by pictorial material. The application is functional and ready to use.

Keywords: CSS, football league simulation, HTML, JavaScript, MVC, PHP, Web simulation

PRILOG