# Web aplikacija za vođenje malonogometnog turnira

**Kovač, Krunoslav**

**Undergraduate thesis / Završni rad**

**2020**

*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:200:585567

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2024-04-25**

*Repository / Repozitorij:*

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

**Stručni studij**

# WEB APLIKACIJA ZA VOĐENJE MALONOGOMETNOG TURNIRA

**Završni rad**

**Krunoslav Kovač**

**Osijek, 2020.**

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

**Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju**

**Osijek, 07.09.2020.**

**Odboru za završne i diplomske ispite**

# Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

| | |
|---|---|
| **Ime i prezime studenta:** | Krunoslav Kovač |
| **Studij, smjer:** | Preddiplomski stručni studij Elektrotehnika, smjer Informatika |
| **Mat. br. studenta, godina upisa:** | AI4556, 26.09.2019. |
| **OIB studenta:** | 96325301807 |
| **Mentor:** | Robert Šojo |
| **Sumentor:** | |
| **Sumentor iz tvrtke:** | |
| **Predsjednik Povjerenstva:** | Marina Peko |
| **Član Povjerenstva 1:** | Robert Šojo |
| **Član Povjerenstva 2:** | dr.sc. Ivana Hartmann-Tolić |
| **Naslov završnog rada:** | Web aplikacija za vođenje malonogometnog turnira |
| **Znanstvena grana rada:** | **Programsko inženjerstvo (zn. polje računarstvo)** |
| **Zadatak završnog rada** | Kreirati web aplikaciju za vođenje malonogometnog turnira u kojoj treba postojati organizator koji je glavni administrator i trener momčadi odnosno registrirani korisnik koji vrši prijavu malonogometnog kluba te gost koji je neregistrirani korisnik web aplikacije. Organizator kreira turnire, datum i mjesto održavanja, kotizacije, nagrade. Treneri vrše prijavu malonogometnog kluba, šalje se obavijest organizatoru oko potvrde prijavljenih. Gost samo ima uvid u popis kreiranih turnira te u prikaz rezultata i poredak natjecanja. |
| **Prijedlog ocjene pismenog dijela ispita (završnog rada):** | Izvrstan (5) |
| **Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:** | Primjena znanja stečenih na fakultetu: 3 bod/boda<br>Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda<br>Jasnoća pismenog izražavanja: 3 bod/boda<br>Razina samostalnosti: 3 razina |
| **Datum prijedloga ocjene mentora:** | 07.09.2020. |

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

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

# IZJAVA O ORIGINALNOSTI RADA

**Osijek, 14.09.2020.**

| | |
|---|---|
| **Ime i prezime studenta:** | Krunoslav Kovač |
| **Studij**: | Preddiplomski stručni studij Elektrotehnika, smjer Informatika |
| **Mat. br. studenta, godina upisa:** | AI4556, 26.09.2019. |
| **Turnitin podudaranje [%]:** | 10 |

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za vođenje malonogometnog turnira**

izrađen pod vodstvom mentora Robert Šojo

i sumentora

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:

# SADRŽAJ

1. UVOD ..... 1
1.1. Zadatak završnog rada ..... 1
2. MALONOGOMETNI TURNIRI ..... 2
2.1. Sistem natjecanja kup ..... 2
2.2. Sistem natjecanja na izbacivanje ..... 2
3. KORIŠTENE TEHNOLOGIJE ZA IZRADU WEB APLIKACIJE ..... 3
3.1. HTML ..... 3
3.2. CSS ..... 4
3.3. PHP ..... 5
3.4. Bootstrap ..... 6
3.5. JavaScript ..... 6
4. IMPLEMENTACIJA WEB STRANICE ..... 8
4.1. Baza podataka ..... 8
4.2. Registracija i prijava ..... 10
4.3. Korisnici aplikacije ..... 12
5. PREGLED WEB APLIKACIJE ..... 17
5.1. Početna stranica ..... 17
5.2. Pregled, rezultati i kreiranje turnira ..... 19
5.3. Završnica turnira ..... 22
6. ZAKLJUČAK ..... 25
LITERATURA ..... 26
SAŽETAK ..... 27
ABSTRACT ..... 28
ŽIVOTOPIS ..... 29
PRILOZI ..... 30

# 1. UVOD

U ovome završnom radu zadatak je napraviti web aplikaciju za vođenje malonogometnih turnira. Malonogometni turniri su vrlo popularni diljem cijeloga svijeta stoga ova web aplikacija može biti od velike pomoći organizatorima te sudionicima kako bi jednostavnije pratili svoje rezultate te ujedno i termine u kojim bi trebali igrati vlastite utakmice. Kako trenutno gotovo svaka osoba ima dostupan pametni telefon, aplikacija je zamišljena da bude pogodna za prikaz na malim ekranima.

Malonogometni turniri se igraju već dugo vremena kroz povijest te se obično održavaju tokom ljetnih i zimskih praznika. Igraju se gotovo u svakom gradu i selu.

Nadalje u poglavlju „Malonogometni turniri“ objasnit će se koncept malonogometnih turnira odnosno sistemi natjecanja gdje postoji dva tipa koja su sistem na izbacivanje i kup. Zatim u poglavlju „Korištene tehnologije za izradu web aplikacije“ pobliže će se objasniti tehnologije koje se koriste za izradu web aplikacije (HTML, CSS, PHP, JavaScript, Bootstarp) te kroz poglavlje „Implementacija web stranice“ biti će objašnjena i sama implementacija u kojoj se opisuje funkcionalnost i primjena te rad web aplikacije.

## 1.1. Zadatak završnog rada

Kreirati web aplikaciju za vođenje malonogometnog turnira u kojoj treba postojati organizator koji je glavni administrator i trener momčadi odnosno registrirani korisnik koji vrši prijavu malonogometnog kluba te gost koji je neregistrirani korisnik web aplikacije. Organizator kreira turnire, datum i mjesto održavanja, kotizacije, nagrade. Treneri vrše prijavu malonogometnog kluba, šalje se obavijest organizatoru oko potvrde prijavljenih. Gost samo ima uvid u popis kreiranih turnira te u prikaz rezultata i poredak natjecanja.

# 2. MALONOGOMETNI TURNIRI

Nogomet je jedan od najpopularnijih sportova na svijetu. Milijuni ljudi se bave nogometom profesionalno, ali i amaterski. Nogometni turniri se uglavnom organiziraju kao malonogometni turniri ili futsal turniri. Razlika u njima je da se kod malonogometnih turnira igra sistem 5+1 (5 igrača i golman), dok se kod futsala igra 4+1 (4 igrača i golman).

Pravila uglavnom ovise o organizatorima kako zamišljaju svoj turnir. U većini slučajeva turniri se igraju „kup“ sistemom gdje je prva faza natjecanja grupe, pa zatim faza izbacivanja (engl. *knockout*), dok se u nekim turnirima odmah igra faza izbacivanja.

## 2.1. Sistem natjecanja kup

U sistemu natjecanja kup postoje dvije faze igranja. Prva faza je igranje u grupama gdje svaka ekipa igra protiv svake. Pobjedom ekipa ostvaruje 3 boda, neriješenim rezultatom bodovi se dijele (po 1 bod svakoj ekipi), a izgubljenom utakmicom ekipa ne dobiva bodove. U prvoj fazi ekipe se natječu za prolazak grupe kako bi mogli nastaviti svoje natjecanje u drugu fazu. Druga faza je faza izbacivanja u koju prolaze ekipe s najbolje ostvarenim rezultatima u grupama, a broj ekipa koji prolazi grupu obično ovisi o organizatoru. U drugoj fazi grupe se „križaju“ gdje, npr. iz svake grupe prolazi po dvije ekipe, a igra ekipa koja je bila prva u svojoj grupi protiv ekipe koja je bila druga u drugoj grupi. Nadalje će se detaljno objasniti druga faza natjecanja.

## 2.2. Sistem natjecanja na izbacivanje

U sistemu natjecanja na izbacivanje nema bodovanja, postoje samo pobjednik i gubitnik. To je zapravo eliminacijski tip turnira gdje su gubitnici odmah eliminirani iz turnira, a gdje u sljedećem krugu igraju pobjednici jedni protiv drugih sve dok ne ostanu samo dvije ekipe, odnosno do finala turnira nakon čega je pobjednik ujedno i prvak turnira. Utakmice se najčešće u ovome sistemu generiraju ždrijebom (izvlačenjem parova). Primjerice ukoliko je na turniru 16 ekipa, tada će se krenuti od osmine finala i igrat će se maksimalno četiri runde.

# 3. KORIŠTENE TEHNOLOGIJE ZA IZRADU WEB APLIKACIJE

Korištene tehnologije za razvoj web aplikacije su opisni jezici (HTML, CSS), razvojni okvir Bootstrap, skriptni i programski jezici (JavaScript, PHP) te MySQL za upravljanje bazom podataka. Za njihovo pisanje nije potrebno razvojno okruženje, ali zbog lakšeg pisanja i jasnijeg pregleda kôda koristi se Visual Studio Code razvijen od Microsotfa.

## 3.1. HTML

HTML (engl. *Hypertext Markup Language*) je opisni jezik odnosno prezentacijski jezik koji se koristi za kreiranje web stranica te ga je vrlo lako naučiti [1]. Hypertext je tekst koji sadrži linkove na druge tekstualne dokumente odnosno stranice (npr. Osijek predstavlja link koji će nas odvesti na stranicu grada Osijeka). HTML dokument je zapravo tekstualna datoteka koja imaju ekstenziju .html. Kada se HTML dokument učita u web preglednik, on postaje objekt koji je korijenski čvor HTML dokumenta. Ti dokumenti se stvaraju pomoću HTML jezika. Sadržaj datoteka odnosno elementi se sastoje od oznaka (engl. *tag*) koje okružuju određeni sadržaj kao što je <html> koji predstavlja početak HTML dokumenta te <head>, <body> itd. Slika 3.1. predstavlja jednostavan primjer korištenja ključnih elemenata koji tvore HTML dokument.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Primjer</title>
5  </head>
6  <body>
7      <!--SADRŽAJ STRANICE-->
8      <h1>NASLOV</h1>
9      <p>Paragraf</p>
10 </body>
11 </html>
```

**Slika 3.1.** *Jednostavan HTML primjer.*

Slikom 3.2. prikazan je izgled navedenog primjera na web pregledniku.

**Slika 3.2.** *Izgled primjera na web pregledniku.*

Najnovija verzija je HTML5 koja je ujedno i korištena za izradu web aplikacije. Prvi put je objavljen u javnom obliku 2008. godine. U listopadu 2014. ažuriran je standard te dobiva status „W3C preporuke“. Dodane su mnoge nove značajke kao što je uključivanje multimedijskih i grafičkih sadržaja, novi dodani elementi <video>, <audio> i <canvas> [2].

## 3.2. CSS

CSS (engl. *cascading style sheets*) je stilski jezik. Koristi se za prezentaciju odnosno za uređivanje te oblikovanje HTML dokumenta, tj. izgleda web stranice. Na slikama 3.3. i 3.4. prikazan je jednostavan primjer CSS datoteke te kako taj primjer izgleda unutar web preglednika. CSS je glavna tehnologija za razvoj web stranica uz HTML i JavaScript. CSS ima ekstenziju .css. [3] Dobra praksa je odvajati CSS i HTML dokumente ponajviše zbog pregleda kôda i lakšeg održavanja. Pomoću CSS-a možemo oblikovati izgled web stranice pogodan za prikaz na mobilnim uređajima. Kako su danas mobilni uređaji dostupni gotovo svima i kako se koriste gotovo u svim generacijama za lakšu komunikaciju putem interneta, kupovanju raznih proizvoda, praćenju novosti u cijelom svijetu, korištenje u poslovne svrhe, web stranice je potrebno prilagoditi mobilnom prikazu, odnosno web stranicu je potrebno napraviti responzivnom, a za to će se koristiti razvojno okruženje Bootstrap.

```
1   body{
2       background-color: rgb(0, 0, 0);
3   }
4
5   h1{
6       color: rgb(255, 255, 255);
7   }
8
9   p{
10      color: rgb(255, 255, 255);
11      text-decoration: underline;
12  }
```

**Slika 3.3.** *Jednostavan CSS primjer.*

**Slika 3.4.** *Izgled primjera na web pregledniku.*

## 3.3. PHP

PHP (engl. *Hypertext preprocessor*) je programski odnosno skriptni jezik čija je namjena izrada dinamičkih web stranica i aplikacija. Izvršava se na poslužiteljskoj strani te je zasnovan na C sintaksi. PHP kôd nije vidljiv klijentu. Osnovne mogućnosti PHP kôda su generiranje HTML kôda te spremanje istih zapisa na disk. Također može se koristiti i za povezivanje odnosno komunikaciju s bazom podataka. Sadrži mnoštvo ugrađenih funkcija. U ovome radu korišten je MySQL kao baza podataka. PHP je korišten za funkcionalnosti kao što su prijava i registracija korisnika, upis rezultata odnosno jasnije rečeno korišten je za svaku funkcionalnost vezanu uz spremanje u bazu podataka. <?php označava početak PHP kôda te završava s ?> kao što je prikazano na slici 3.5. te izgled istoga na slici 3.6.

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <title>Primjer</title>
5   </head>
6   <body>
7       <!--SADRŽAJ STRANICE-->
8       <h1>NASLOV</h1>
9       <p>Paragraf</p>
10      <?php
11          echo "Hello World!";
12      ?>
13  </body>
14  </html>
```

**Slika 3.5.** *Jednostavan PHP primjer.*

**Slika 3.6.** *Izgled primjera na webu pregledniku.*

## 3.4. Bootstrap

Bootstrap je alat otvorenog kôda (engl. *open source*) za razvoj web aplikacija koji sadrži predloške temeljene na CSS-u i JavaScriptu. Bootstrap je front-end razvojni okvir (engl. *framework*) koji omogućuje responzivan dizajn za optimalniji prikaz web stranice na mobilnim uređajima te uvelike olakšava razvoj web aplikacija. Sadrži CSS i JavaScript temeljene predloške kojima se dizajniraju obrasci, tipke, navigacija i ostale komponente sučelja. Drugim riječima Bootstrap je zbirka CSS klasa i JavaScript funkcija. Na slici 3.7. linija 8 prikazuje dodavanje Bootstrap-a u HTML stranicu.

```
3  <head>
4      <title>FUTSAL TURNIRI</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <link rel="stylesheet" href="moj.css">
7      <!-- Latest compiled and minified CSS -->
8      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
9      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
10 </head>
```

**Slika 3.7.** *Dodavanje CSS Bootstrap framework-a.*

## 3.5. JavaScript

JavaScript je objektni skriptni programski jezik koji se primjenjuje na webu (podržava ga većina web preglednika). Dizajniran je za stvaranje online aplikacija koje povezuju objekte i resurse na klijentima i poslužiteljima. Programerima omogućuje dinamičku promjenu i dodavanje novog sadržaja. Skripta se izvršava naredbu po naredbu bez prevođenja (engl. *compile*) programa.

JavaScript se najčešće koristi za dodavanje interaktivnosti web stranicama, kao što su animacija, dijaprojekcija, provjera valjanosti obrazaca te još mnogo toga. Sadrži *math* objekt koji omogućuje izvođenje matematičkih zadataka nad brojevima kao što je potenciranje, korjenovanje, zaokruživanje brojeva, generiranje slučajnih itd. Slika 3.8. prikazuje jednostavan primjer u kojem se klikom na tipku mijenja tekst paragrafa gdje na slici 3.9. je prikazan primjer unutar web preglednika.

```html
<!DOCTYPE html>
<html>
<head>
    <title>Primjer</title>
</head>
<body>
    <!--SADRŽAJ STRANICE-->
    <h1>NASLOV</h1>
    <p id="demo">Paragraf</p>
    <?php
        echo "Hello World!";
    ?>
    <button type="button" onclick='ChangeText()'>Click Me!</button>

<script>
    function ChangeText(){
        document.getElementById("demo").innerHTML = "Hello JavaScript!"
    }
</script>
</body>
</html>
```

**Slika 3.8.** *Jednostavan JavaScript primjer.*

**Slika 3.9.** *Izgled primjera u web pregledniku prije i nakon klika na tipku.*

# 4. IMPLEMENTACIJA WEB STRANICE

Za implementaciju web stranice potrebno je kreiranje baze podataka u koju se spremaju svi podaci kao što je primjerice registracija korisnika, tada se u bazu podataka spremaju podaci o korisniku (ime, prezime, korisničko ime, lozinka itd.).

## 4.1. Baza podataka

U radu se koristi lokalna baza podataka koju pruža XAMPP koji je namijenjen za upotrebu u lokalnoj mreži. Prije svega služi za kreiranje servera za potrebe testiranja.

Naziv lokalne baze podataka je „futsal“. Slika 4.1. predstavlja kôd koji služi kao spoj na bazu podataka „futsal“.

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "futsal"; //Database name

$conn = mysqli_connect($servername, $username, $password, $dbname);
mysqli_set_charset($conn,"utf8");
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

**Slika 4.1.** *Datoteka spoj.php koja služi za spajanje na bazu podataka.*

Slika 4.2. prikazuje tablicu „korisnici“ koja sadržava potrebne podatke koje korisnik unosi prilikom registracije na web stranicu te koji ujedno i predstavljaju zasebni tip podataka.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | k_ime | varchar(100) | utf32_croatian_mysql561_ci | | No | None | | | Change Drop More |
| 3 | ime | varchar(100) | utf32_croatian_mysql561_ci | | No | None | | | Change Drop More |
| 4 | prezime | varchar(100) | utf32_croatian_mysql561_ci | | No | None | | | Change Drop More |
| 5 | email | varchar(100) | utf32_croatian_mysql561_ci | | No | None | | | Change Drop More |
| 6 | lozinka | varchar(100) | utf32_croatian_mysql561_ci | | No | None | | | Change Drop More |
| 7 | uloga | varchar(100) | utf32_croatian_mysql561_ci | | No | None | | | Change Drop More |

**Slika 4.2.** *Tablica korisnici.*

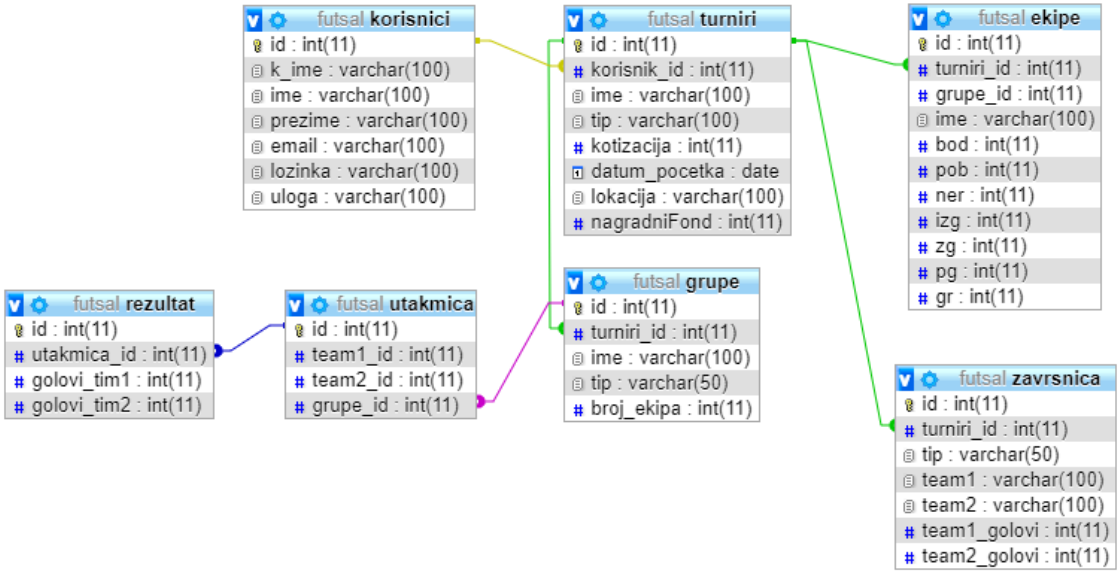
U tablici 4.1. prikazan je popis svih tablica u bazi podataka. Svaka tablica sadrži svoj primarni ključ.

| Naziv tablice | Kratki opis tablice |
|---|---|
| -korisnici | Sadrži podatke o korisniku:<br>-korisničko ime,<br>-ime i prezime,<br>-email,<br>-lozinku,<br>-ulogu (admin/korisnik). |
| -turniri | Sadrži podatke o turniru:<br>-ime,<br>-tip,<br>-kotizaciju,<br>-datum početka,<br>-lokaciju,<br>-nagradni fond. |
| -ekipe | Sadrži podatke o ekipama prijavljenim na turnir:<br>-ime,<br>-broj pobjeda, izgubljenih i neriješenih utakmica,<br>-broj zabijenih i primljenih golova. |
| -grupe | Sadrži podatke o grupama:<br>-ime,<br>-tip (grupa/playoff),<br>-broj ekipa. |
| -utakmica | Sadrži podatke:<br>-id ekipa koji se susreću,<br>-primarni ključ tablice grupe. |
| -rezultat | Sadrži podatke:<br>-broj golova ekipa koji se susreću. |

| | |
|---|---|
| -zavrsnica | Sadrži podatke o završnici nakon faze grupa:<br>-tip (četvrtfinale/polufinale/finale),<br>-imena ekipa,<br>-broj golova ekipa. |

**Tablica 4.1.** *Naziv i kratki opis tablica baze podataka.*

Slika 4.3. prikazuje vezu između tablica u bazi podataka.




**Slika 4.3.** *Veze između tablica baze podataka.*

## 4.2. Registracija i prijava

Slika 4.4. prikazuje obrazac registracije realiziran u PHP-u. Skripta provjerava da li je tipka koja ima svoj ID („registracija“) pritisnuta i ako je pritisnuta, dohvaća unesene podatke te prije nego što unese podatke u bazu podataka odnosno tablicu korisnici provjerava da li je korisničko ime zauzeto. Nakon unosa podatak u tablicu korisnik dobiva informaciju da je uspješno registriran. Lozinka je kriptirana MD5 kriptografskom *hash* funkcijom koja prihvaća vrijednost bilo koje duljine, a vraća rezultat fiksne duljine (128 bita) [4].

```php
<?php
    require_once("spoj.php");
    session_start();
    $error = "";

    if(isset($_POST['registracija'])){
    if($_SERVER['REQUEST_METHOD'] == 'POST'){

        $name = $_POST['ime_prezime'];
        $name_lastname = split_name($name);
        $firstName = $name_lastname[0];
        $lastName = $name_lastname[1];
        $username = $_POST['k_ime'];
        $email = $_POST['email'];
        $password = $_POST['lozinka'];
        $password = md5($password);
        $role = $_POST['uloga'];

        $sql = "SELECT * FROM korisnici WHERE k_ime = '$username'";
        $result = $conn->query($sql);
        if (mysqli_num_rows($result) > 0){
            $error = "Korisničko ime je već zauzeto.";
        }else{
            $sql = "INSERT INTO korisnici (ime, prezime, email, k_ime, lozinka, uloga) VALUES ('$firstName', '$lastName', '$email', '$username',
            '$password', '$role')";
            if ($conn->query($sql) == TRUE){
                    echo "<script type='text/javascript'>alert('Uspješno ste se registrirali!');</script>";
                    header("Location: prijava.php");
                }else{
                echo $conn->error;
            }
        }
    }
  }
?>
```

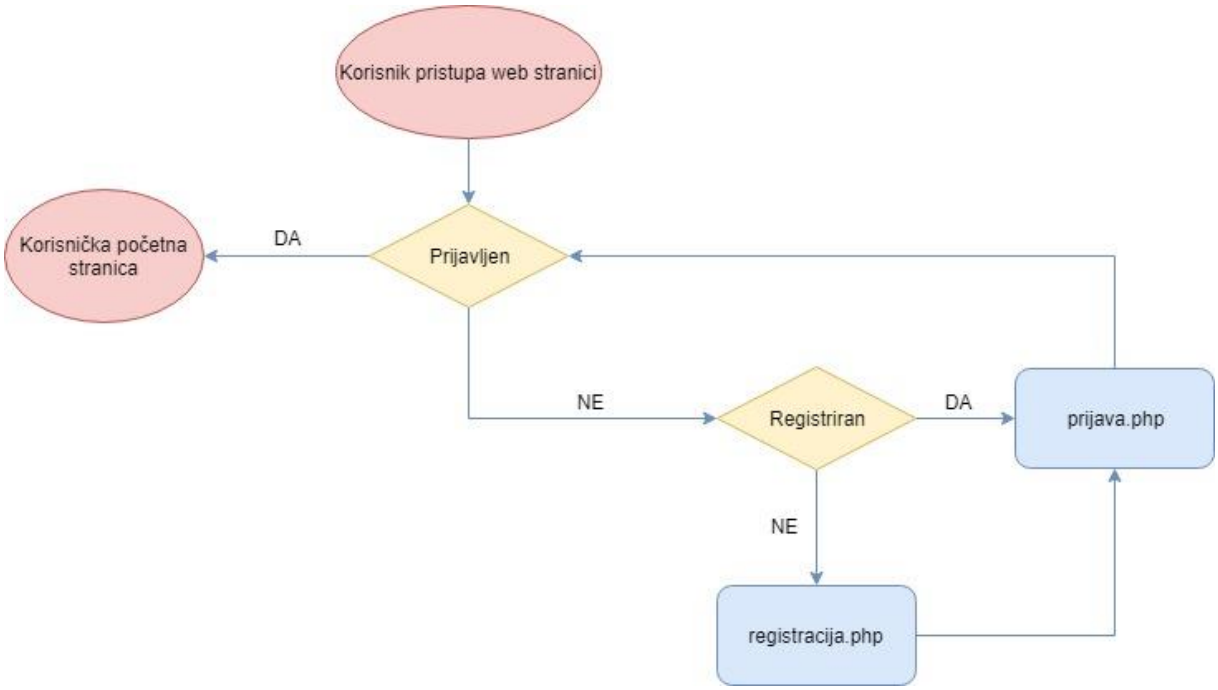
**Slika 4.4.** *Skripta za registraciju korisnika registracija.php.*

Slika 4.5. prikazuje obrazac za prijavu korisnika. Prilikom prijave korisnika, dohvaćaju se podaci (korisničko ime i lozinka) iz tablice korisnici koji se podudaraju s unesenim podacima. Ukoliko su podaci krivo uneseni ili korisnik nema registriran račun korisnik će biti obavješten.

```php
<?php
    require_once("spoj.php");
    session_start();
    $error = "";

    if(isset($_POST['prijava'])){
        //Username and password sent from form

        $k_ime = $_POST['k_ime'];
        $lozinka = md5($_POST['lozinka']);

        //Dohvati sve iz tablice korisnici koji se podudaraju s poslanom formom
        $query = "SELECT * FROM korisnici WHERE k_ime = '$k_ime' AND lozinka = '$lozinka'";

        $result = mysqli_query($conn, $query);

        if(mysqli_num_rows($result) == 1) {
            $row = mysqli_fetch_assoc($result);
            $_SESSION['id'] = $row['id'];
            $_SESSION['k_ime'] = $k_ime;
            $_SESSION['ime'] = $row['ime'];
            $_SESSION['prezime'] = $row['prezime'];
            $_SESSION['uloga'] = $row['uloga'];
            header("Location: index.php");
        } else {
                session_unset();
                session_destroy();
                $error = "Pogrešno korisničko ime ili lozinka!";
        }
    }

    mysqli_close($conn);
?>
```

**Slika 4.5.** *Skripta za prijavu korisnika prijava.php.*

Slika 4.6. prikazuje dijagram toka za prijavu korisnika. Kada korisnik pristupa web stranici provjerava se da li je korisnik prijavljen, ukoliko je prijavljen korisnik je preusmjeren na korisničku početnu stranicu, a ukoliko nije, dolazi do provjere da li je korisnik registriran. Ako korisnik nije registriran preusmjerava se na stranicu za registraciju te nakon obavljene registracije, korisnik je preusmjeren na prijavu na koju je ujedno preusmjeren ukoliko je već registrirani korisnik, zatim se ponovo vraća na provjeru da li je korisnik prijavljen.




**Slika 4.6.** *Dijagram toka za prijavu.*

## 4.3. Korisnici aplikacije

Postoje tri vrste korisnika aplikacije: organizator (administrator), trener (korisnik) i gost. Svaki od tih korisnika ima zasebne mogućnosti na web stranici.

Administratori odnosno organizatori imaju mogućnost pregleda svih kreiranih turnira te mogućnost kreiranja i brisanja vlastitih malonogometnih turnira. Administrator za kreiranje mora ispuniti potrebne podatke kao što su datum i mjesto održavanja, kotizacija, nagrade itd. Administrator ima mogućnost unošenja i uređivanja rezultata. Kreira završnicu odabirom sistema (četvrtfinale, polufinale, finale).

Registrirani korisnik ima mogućnost prijavljivanja na određeni turnir te pregled svih otvorenih turnira i pregled poretka i rezultata utakmica.

Gost ili neregistrirani korisnik ima mogućnost pregleda svih turnira te pregled poretka i rezultata utakmica, ali nema mogućnost prijave na iste ili kreiranje novih turnira.

Logika kod kreiranja turnira je ta da se dohvate svi podaci uneseni od strane korisnika koji se zatim unose u tablicu turniri iz koje se kasnije dohvaćaju podaci za prikaz turnira korisnicima. Implementacija kôda prikazana je slikom 4.7.

```php
<?php
    require_once("spoj.php");
    session_start();
    $error = "";

    if(isset($_POST['kreiraj'])){
    if($_SERVER['REQUEST_METHOD'] == 'POST'){
        if (session_status() == PHP_SESSION_NONE) {
            session_start();
        }
        $ime = $_POST['ime'];
        $tip = $_POST['tip'];
        $kotizacija = $_POST['kotizacija'];
        $datum_pocetka = $_POST['datumPocetka'];
        $lokacija = $_POST['lokacija'];
        $fond = $_POST['nagradniFond'];


        $sql = "SELECT * FROM turniri WHERE ime = '$ime'";
        $result = $conn->query($sql);

        if (mysqli_num_rows($result) > 0){
            $error = "Ime turnira je zauzeto.";
        }else{
            $korisnik_id = $_SESSION['id'];
            $sql = "INSERT INTO turniri (korisnik_id, ime, tip, kotizacija, datum_pocetka, lokacija, nagradniFond)
                                VALUES ('$korisnik_id','$ime', '$tip', '$kotizacija', '$datum_pocetka', '$lokacija', '$fond')";
            if ($conn->query($sql) == TRUE){
                    echo "<script type='text/javascript'>alert('Uspješno ste kreirali turnir!');</script>";
                }else{
                echo $conn->error;
            }
        }
    }
}
?>
```

**Slika 4.7.** *Logika kod kreiranja turnira.*

Nakon što korisnik (trener) nakon unesenih potrebnih podataka pritisne gumb za prijavu ekipe, tada se iz tablice *ekipe* dohvaćaju sve ekipe koje imaju jednako ime s onim u tablici i id turnira s odabranim turnirom. Taj dio služi za provjeru postoji li već unutar tablice *ekipe* ime ekipe za odabrani turnir. Zatim se dohvaća id korisnika koji je kreirao turnir odnosno e-mail administratora turnira na koji se korisnik želi prijaviti. Nakon uspješne prijave u tablicu *ekipe* se upisuju uneseni podaci, a administratoru se šalje obavijest o novoj prijavi ekipe funkcijom *mail()*, dok se korisniku pojavljuje prozor s obavijesti da je ekipa uspješno prijavljena. Kôd je prikazan na slici 4.8.

```php
<?php
    require_once("spoj.php");
    $error = "";

    if(isset($_POST['prijava_ekipe'])){
    if($_SERVER['REQUEST_METHOD'] == 'POST'){
        $ime = $_POST['ime'];
        $id = $_GET['id'];

        $sql = "SELECT * FROM ekipe WHERE ime = '$ime' AND turniri_id = '$id'";
        $result = $conn->query($sql);

        $select_id = $conn->query("SELECT korisnik_id FROM turniri WHERE id = '$id'");
        $row=mysqli_fetch_array($select_id);
        $korisnik_id = $row[0];

        $select_korisnik = $conn->query("SELECT email FROM korisnici WHERE id='$korisnik_id'");
        $getEmail = mysqli_fetch_array($select_korisnik);

        $to = $getEmail[0];
        $email_subject = "Nova prijava ekipe";
        $msg = "Obavijest o novoj prijavi ekipe na vaš turnir. Ime nove prijavljene ekipe je: $ime ";

        if (mysqli_num_rows($result) > 0){
            $error = "Ime ekipe je zauzeto.";
        }else{
            $sql = "INSERT INTO ekipe (turniri_id, grupe_id, ime) VALUES ('$id',0,'$ime')";
            if ($conn->query($sql) == TRUE){
                    mail($to,$email_subject,$msg);
                    echo "<script type='text/javascript'>alert('Uspješno ste prijavili ekipu!');</script>";
                }else{
                echo $conn->error;
            }
        }
    }

}
?>
```

**Slika 4.8.** *Logika kod prijavljivanja ekipe na turnir*

Logika generiranja utakmice implementirana je rekurzivnom funkcijom *generirajUtakmice()* koja kao parametre prima broj ekipa, ID ekipa, ID grupe te konekciju, a zatim se ti podaci unose u tablicu utakmica odnosno formiraju se utakmice ukoliko je zadovoljen uvjet da je broj ekipa veći od 1. Kôd funkcije *generirajUtakmice()* je prikazan na slici 4.9.

```php
function generirajUktamice( $n, $team_ids, $id_grupe, $conn ) {

    if($n <= 1){
        return 1;
    } else {
        $n = $n-1;
        $m = $n-1;
        mysqli_query($conn,"INSERT INTO utakmica (team1_id, team2_id, grupe_id) VALUES ('$team_ids[$m]', '$team_ids[$n]','$id_grupe') ");
        $m = $n-2;
        if($m > -1){
            mysqli_query($conn,"INSERT INTO utakmica (team1_id, team2_id, grupe_id) VALUES ('$team_ids[$n]', '$team_ids[$m]','$id_grupe') ");
            $m = $n-3;
            if($m > -1){
                mysqli_query($conn,"INSERT INTO utakmica (team1_id, team2_id, grupe_id) VALUES ('$team_ids[$n]', '$team_ids[$m]','$id_grupe') ");
            }
            $m = $n-4;
            if($m > -1){
                mysqli_query($conn,"INSERT INTO utakmica (team1_id, team2_id, grupe_id) VALUES ('$team_ids[$n]', '$team_ids[$m]','$id_grupe') ");
            }
        }
    }

    return generirajUktamice($n, $team_ids, $id_grupe, $conn);
}
```

**Slika 4.9.** *Funkcija generirajUtakmice()*

Logika zapisa rezultata prikazana je slikom 4.10. U tablicu rezultat zapisuje se ID utakmice te golovi ekipa, a zatim se u tablicu ekipe upisuju podaci (bodovi, broj pobjeda, gol razlika itd.) onim ekipama kojima su ID ekipe iz tablice ekipe i ID ekipe iz tablice rezultat jednaki.

```
$result = $conn->query("INSERT INTO rezultat (utakmica_id, golovi_tim1, golovi_tim2) VALUES ('$utakmice_id','$golovi_tim1','$golovi_tim2')");
if($golovi_tim1 > $golovi_tim2){
    $sq = "UPDATE ekipe
        SET bod=bod+3, pob=pob+1, zg=zg+'$golovi_tim1', pg=pg+'$golovi_tim2', gr=zg-pg
        WHERE id = (SELECT team1_id FROM utakmica WHERE id = '$utakmice_id')";
    $ekipe = $conn->query($sq);

    $sq = "UPDATE ekipe
        SET izg=izg+1, zg=zg+'$golovi_tim2', pg=pg+'$golovi_tim1', gr=zg-pg
        WHERE id = (SELECT team2_id FROM utakmica WHERE id = '$utakmice_id')";
    $ekipe = $conn->query($sq);

}else if($golovi_tim1 < $golovi_tim2){
    $sq = "UPDATE ekipe
        SET izg=izg+1, zg=zg+'$golovi_tim1', pg=pg+'$golovi_tim2', gr=zg-pg
        WHERE id = (SELECT team1_id FROM utakmica WHERE id = '$utakmice_id')";
    $ekipe = $conn->query($sq);

    $sq = "UPDATE ekipe
        SET bod=bod+3, pob=pob+1, zg=zg+'$golovi_tim2', pg=pg+'$golovi_tim1', gr=zg-pg
        WHERE id = (SELECT team2_id FROM utakmica WHERE id = '$utakmice_id')";
    $ekipe = $conn->query($sq);
}else{
    $sq = "UPDATE ekipe
        SET bod=bod+1, ner=ner+1, zg=zg+'$golovi_tim1', pg=pg+'$golovi_tim2', gr=zg-pg
        WHERE id = (SELECT team1_id FROM utakmica WHERE id = '$utakmice_id')";
    $ekipe = $conn->query($sq);

    $sq = "UPDATE ekipe
        SET bod=bod+1, ner=ner+1, zg=zg+'$golovi_tim2', pg=pg+'$golovi_tim1', gr=zg-pg
        WHERE id = (SELECT team2_id FROM utakmica WHERE id = '$utakmice_id')";
    $ekipe = $conn->query($sq);
```

**Slika 4.10.** *Logika zapisa rezultata*

Logika kod prikaza poretka je vrlo jednostavna. Dohvaća se ID turnira i ID grupe nakon čega se iz tablice ekipe dohvaćaju sve ekipe gdje su njihovi strani ključevi (turniri_id i grupe_id) jednaki onima koji su dohvaćeni. Ekipe su poredane silazno najprije po bodovima, a zatim po gol razlici. Implementacija kôda prikazana je slikom 4.11.

```
<tbody>
    <?php
        $id_turnira = $_GET["id"];
        if(isset($_GET['id_grupe'])){
            $id_grupe = $_GET['id_grupe'];
        }

        if(empty($id_grupe)){
            $res=mysqli_query($conn,"SELECT id FROM grupe WHERE turniri_id='$id_turnira'");
            $row=mysqli_fetch_array($res);
            $id_grupe = $row['id'];
        }

        $query = "SELECT * FROM ekipe WHERE turniri_id = '$id_turnira' AND grupe_id ='$id_grupe' ORDER BY bod DESC, gr DESC";
        $result=mysqli_query($conn,$query);

        while($row=mysqli_fetch_array($result)):
    ?>
<tr>
    <td scope="row"><?php echo $row["ime"]?></td>
    <td><?php echo $row["bod"]?></td>
    <td><?php echo $row["pob"]?></td>
    <td><?php echo $row["ner"]?></td>
    <td><?php echo $row["izg"]?></td>
    <td><?php echo $row["zg"]?></td>
    <td><?php echo $row["pg"]?></td>
    <td><?php echo $row["gr"]?></td>
</tr>

<?php endwhile;?>
</tbody>
```

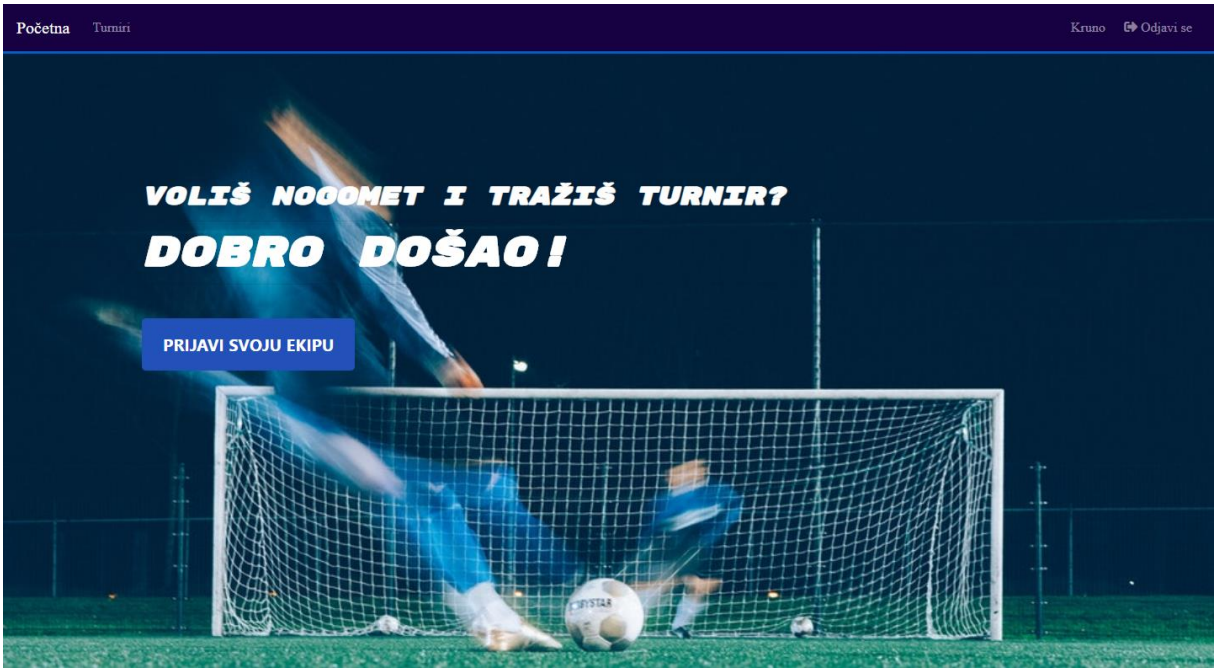
**Slika 4.11.** *Logika kod prikaza poretka turnira*

# 5. PREGLED WEB APLIKACIJE

U ovome poglavlju bit će opisan rad te kroz nekoliko slika prikazan dizajn i izgled web aplikacije unutar preglednika.

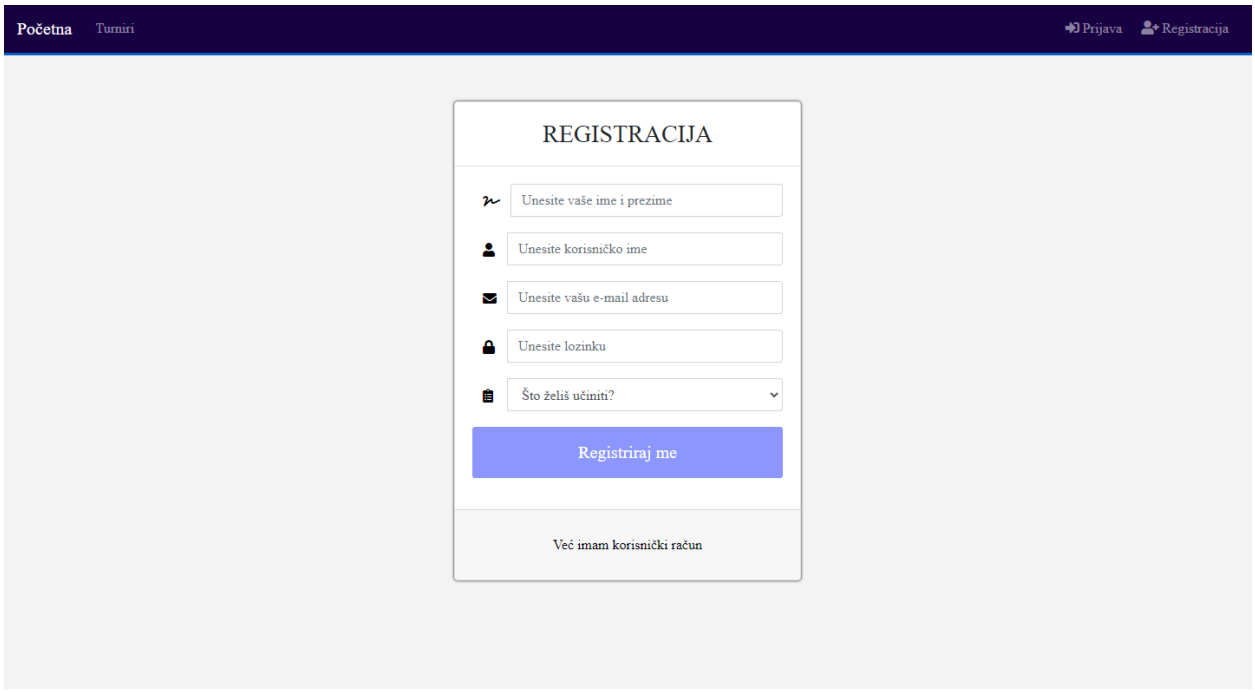
## 5.1. Početna stranica

Slika 5.1. predstavlja izgled početne stranice koja je prikazana korisnicima koji pristupaju web stranici. Na početnoj stranici nalazi se navigacijska traka koja nudi opcije kao što je pregled turnira te registraciju i prijavu korisnika.



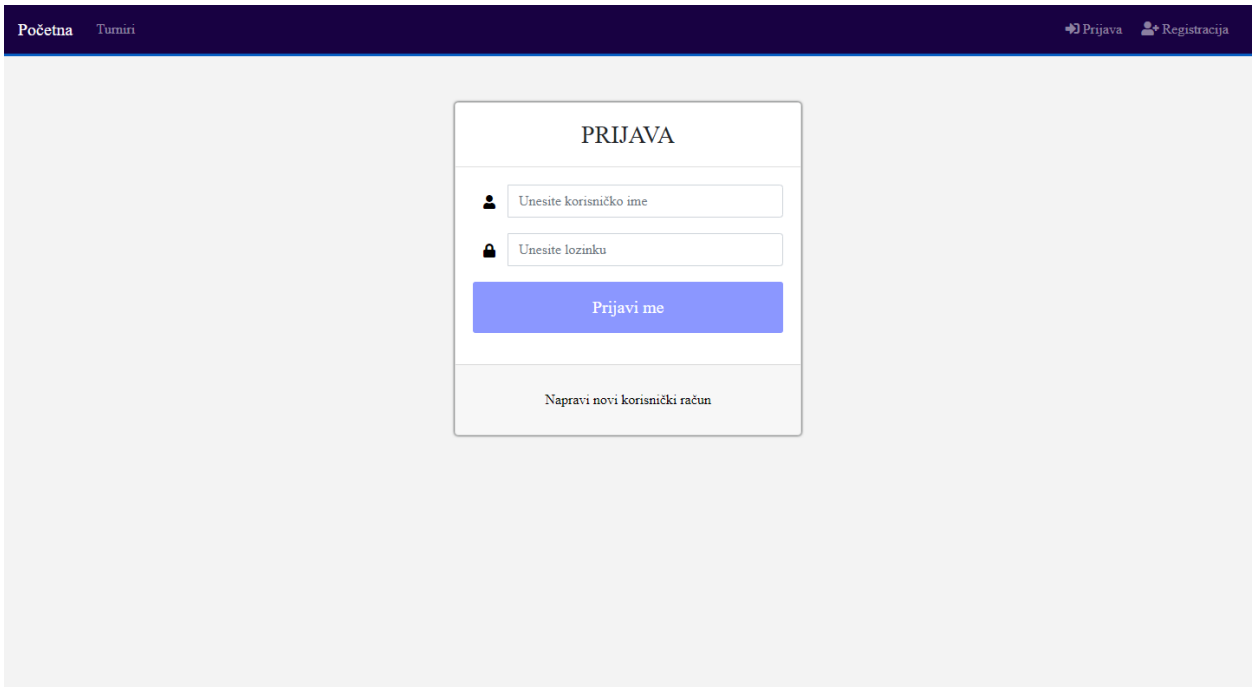

**Slika 5.1.** *Izgled početne stranice.*

Na slici 5.2. prikazan je izgled forme za registraciju. Kako bi se korisnik mogao registrirati, potrebno je unijeti sve tražene podatke jer su sva polja obavezna. Polja: ime i prezime, željeno korisničko ime, e-mail adresu, zatim lozinku i na kraju u polju „Što želiš učiniti?“ iz izbornika odabire jednu od opcija „Organizirati turnir“ ili „Prijaviti se na turnir“.




**Slika 5.2.** *Prikaz forme registracija*

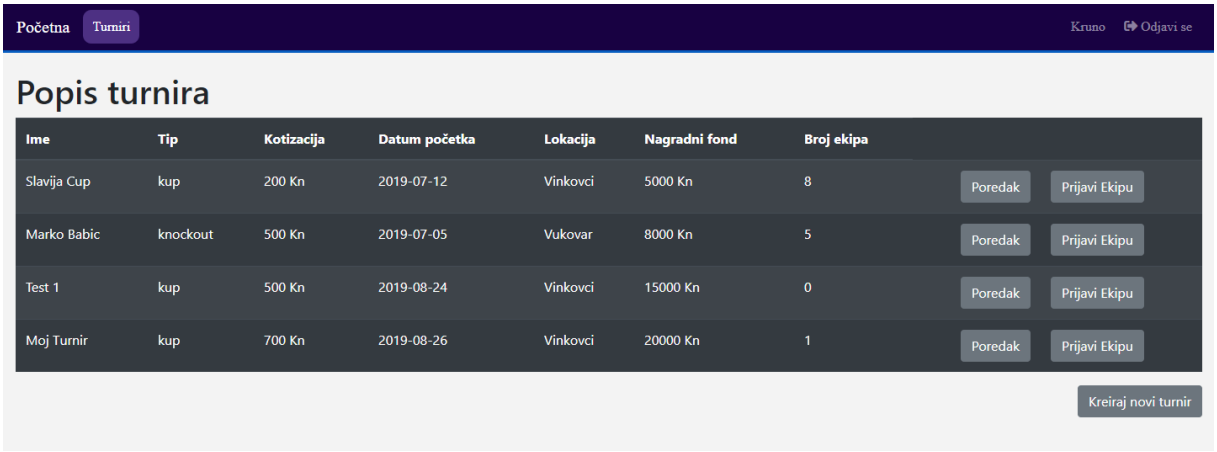
Ukoliko je korisnik uspješno izvršio registraciju tada za prijavu u web aplikaciju unosi korisničko ime i lozinku koju je odabrao prilikom registracije. Izgled forme za prijavu prikazan je slikom 5.3.



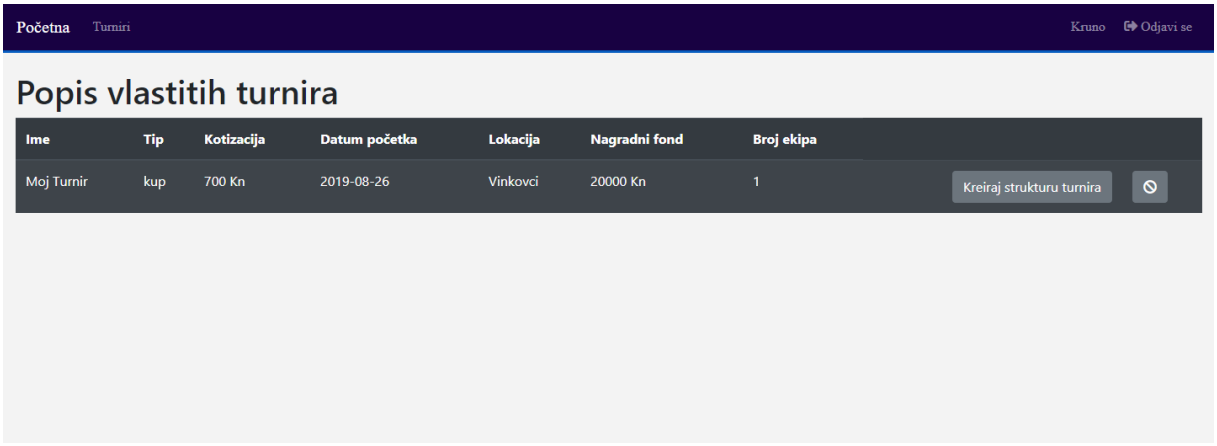

**Slika 5.3.** *Prikaz forme prijava*

## 5.2. Pregled, rezultati i kreiranje turnira

Odabirom linka „*Turniri*“ na navigacijskoj traci lijevo, otvara se stranica koja prikazuje popis svih kreiranih turnira, mogućnost prijave na neki od turnira ukoliko je korisnik prijavljen kao korisnik (trener) te mogućnost pregleda poretka turnira (poredak u grupama, završnica, rezultati itd.) prikazanim na slici 5.4. dok klikom na vlastito ime na navigacijskoj traci desno otvara se popis turnira kreiranih od strane prijavljenog korisnika (ukoliko je korisnik prijavljen kao organizator) te se na tome popisu nudi opcija brisanja turnira slika 5.5. Ekipe se na turnir prijavljuju klikom na tipku „Prijavi ekipu“ zatim se otvara stranica na kojoj je potrebno unošenje ime ekipe nakon čega se korisniku otvara prozor upozorenja u kojemu ga upituje da li je siguran da želi prijaviti ekipu. Potvrdom korisnik je uspješno prijavio ekipu na odabrani turnir nakon čega organizator turnira dobiva obavijest na email o novoj prijavi ekipe.

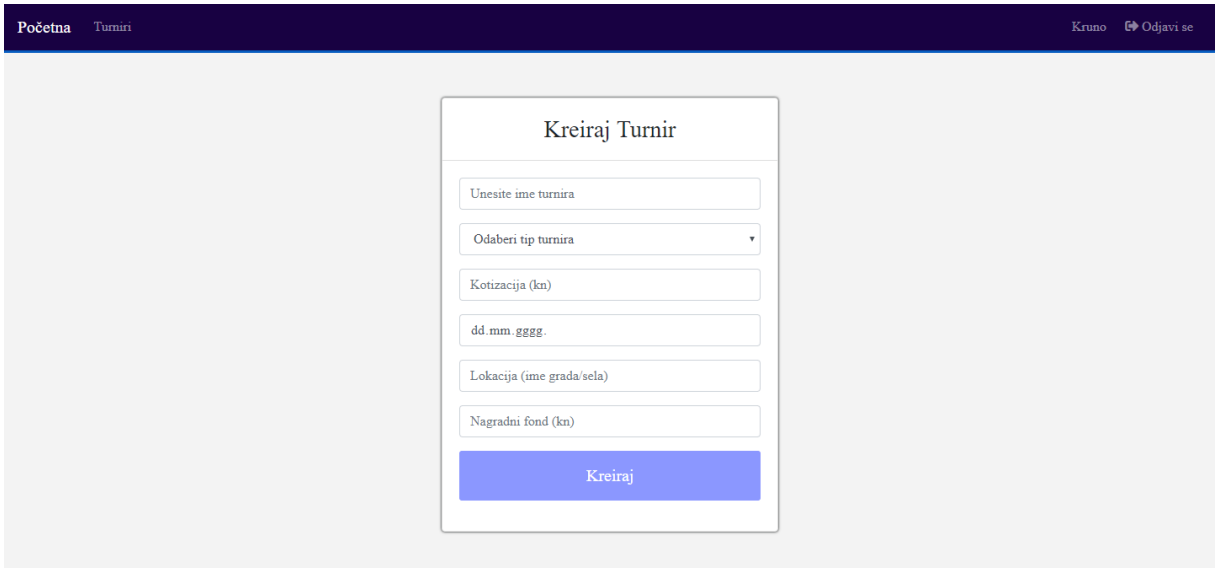



**Slika 5.4.** *Popis svih turnira.*




**Slika 5.5.** *Popis turnira kreiranih od prijavljenog korisnika.*

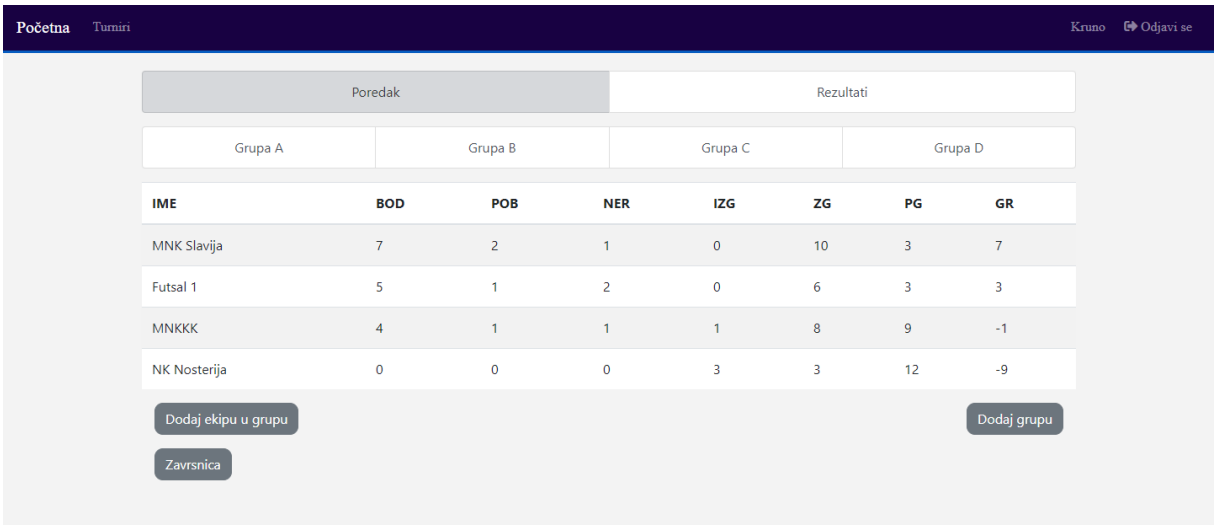
Slika 5.6. predstavlja obrazac za kreiranje novoga turnira, prikazane su potrebne informacije koje je potrebno ispuniti za kreiranje turnira.




**Slika 5.6.** *Obrazac za kreiranje novog turnira.*

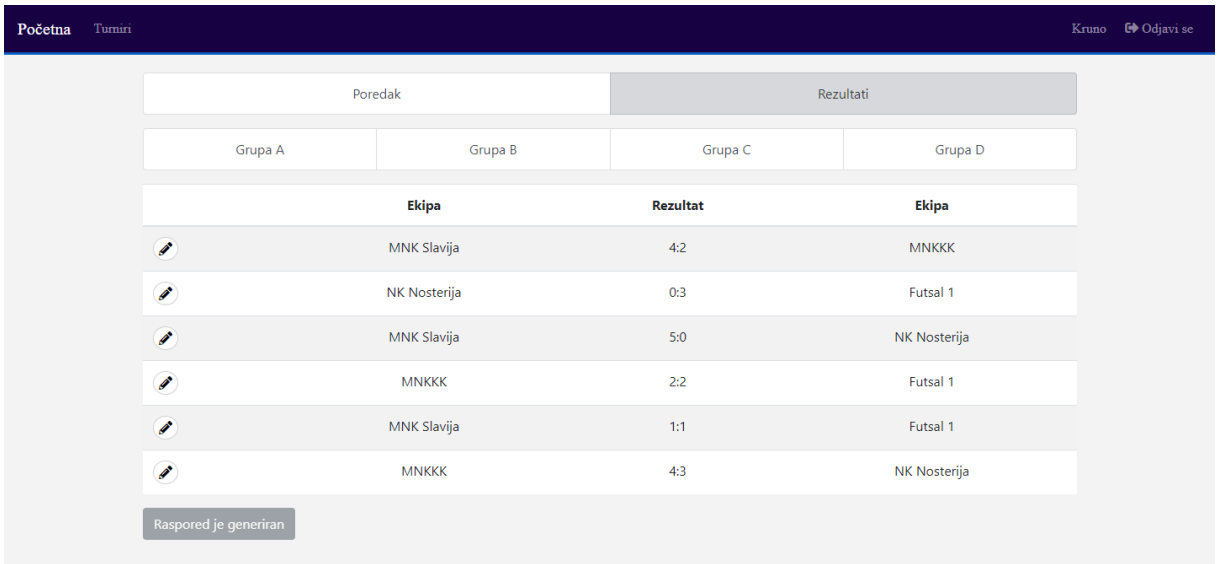
Nakon prijava ekipa, administrator odnosno organizator vlastitog turnira, ima mogućnost kreiranja novih grupa te raspoređivanja ekipa po grupama po svojoj volji.

Sistem grupa se odvija na način da svaka ekipa igra protiv svake, što znači da će prema slici 5.7. biti ukupno 6 utakmica u grupi (1-2, 1-3, 1-4, 2-3, 2-4, 3-4). Bodovanje izgleda ovako: Pobjeda donosi 3 boda, neriješena utakmica donosi 1 bod te izgubljena utakmica ne donosi bodove. U poretku najprije se gleda broj bodova ekipa, a ukoliko je broj bodova jednak kod dvije ili više ekipa, gleda se bolja gol razlika (gr).

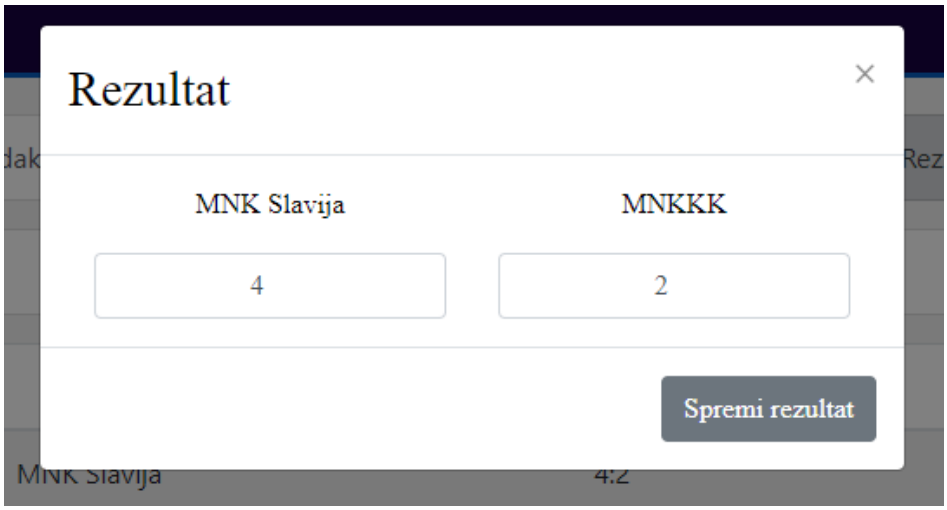



**Slika 5.7.** *Poredak ekipa.*

Uz pregled poretka postoji mogućnost praćenja rezultata utakmica (slika 5.8.). Kada organizator popuni grupu s određenim brojem ekipa, tada ima mogućnost generirati raspored utakmica. Klikom na tipku raspored je generiran automatski. Klikom na predviđenu tipku za uređivanje, otvara se modalni oblik (slika 5.9.) u kojemu organizator unosi rezultat odigrane utakmice (rezultat je moguće uređivati naknadno).



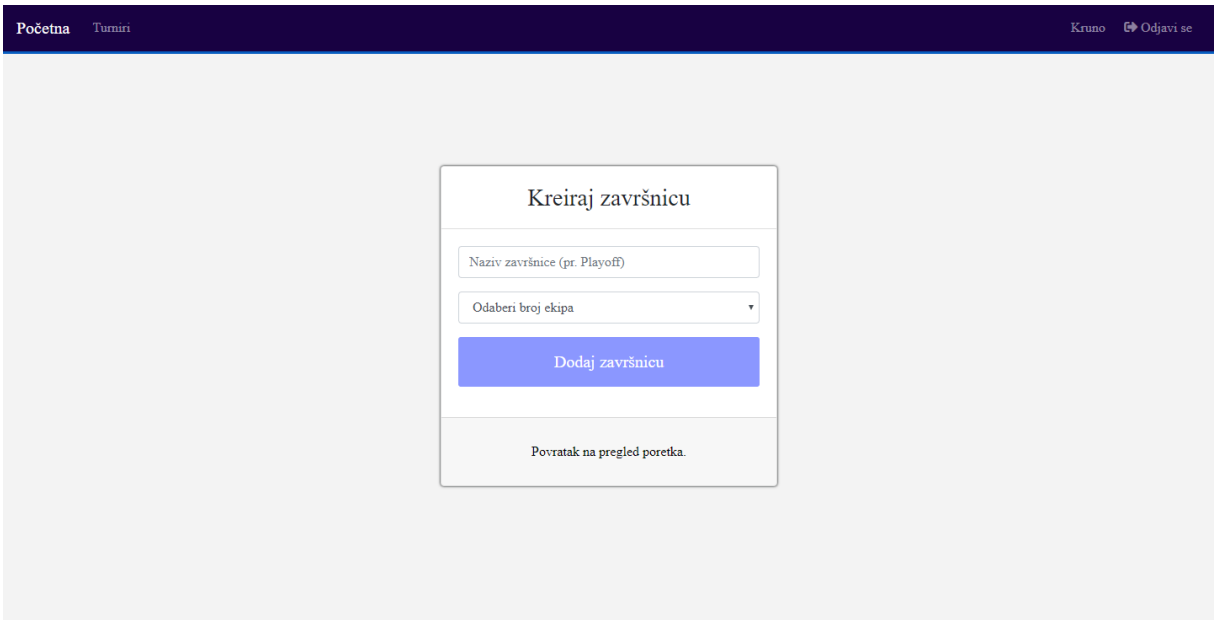

**Slika 5.8.** *Rezultati utakmica.*



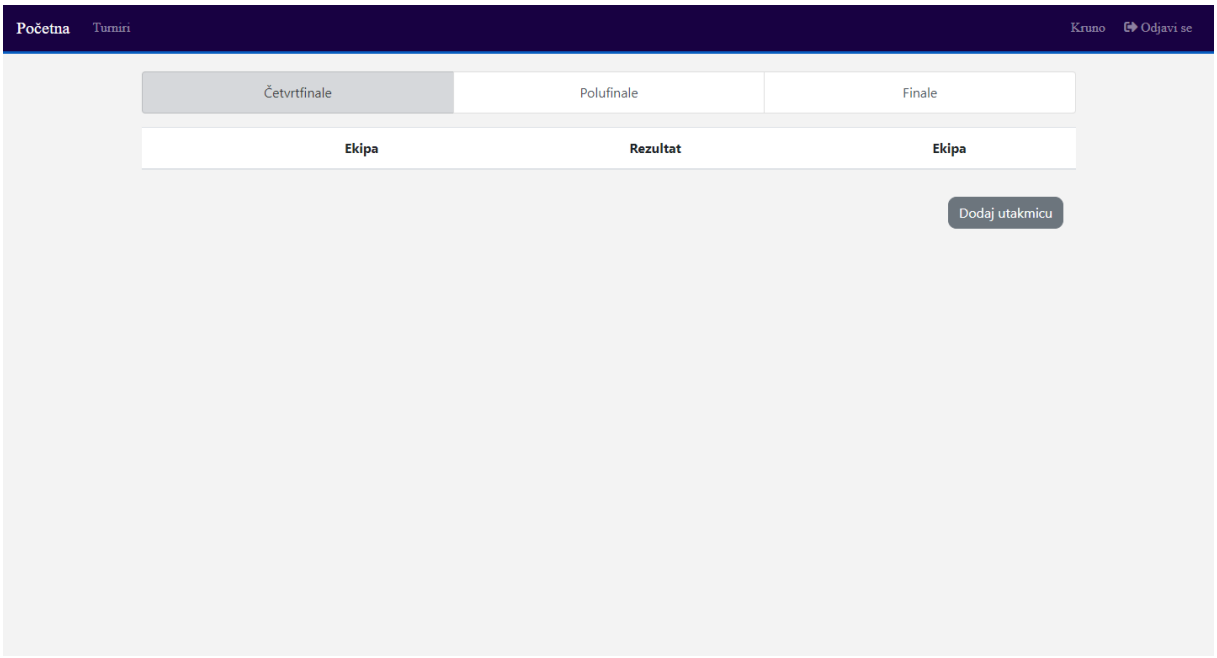

**Slika 5.9.** *Modalni oblik unosa rezultata utakmice.*

## 5.3. Završnica turnira

Nakon grupne faze natjecanja slijedi završnica. Najprije potrebno je kreirati novu završnicu. Klikom na tipku završnica otvara se stranica prikazana slikom 5.10. u kojoj je potrebno unijeti naziv te odabrati tip završnice odnosno broj ekipa (finale koje broji dvije ekipe, polufinale četiri, četvrtfinale osam itd.) nakon čega se ovisno o odabranom tipu završnice otvara predviđena stranica. Primjer odabira četvrtfinala prikazan je slikom 5.11. U završnici se utakmice dodaju „ručno“ kako bi organizatori imali slobodu odabira parova.

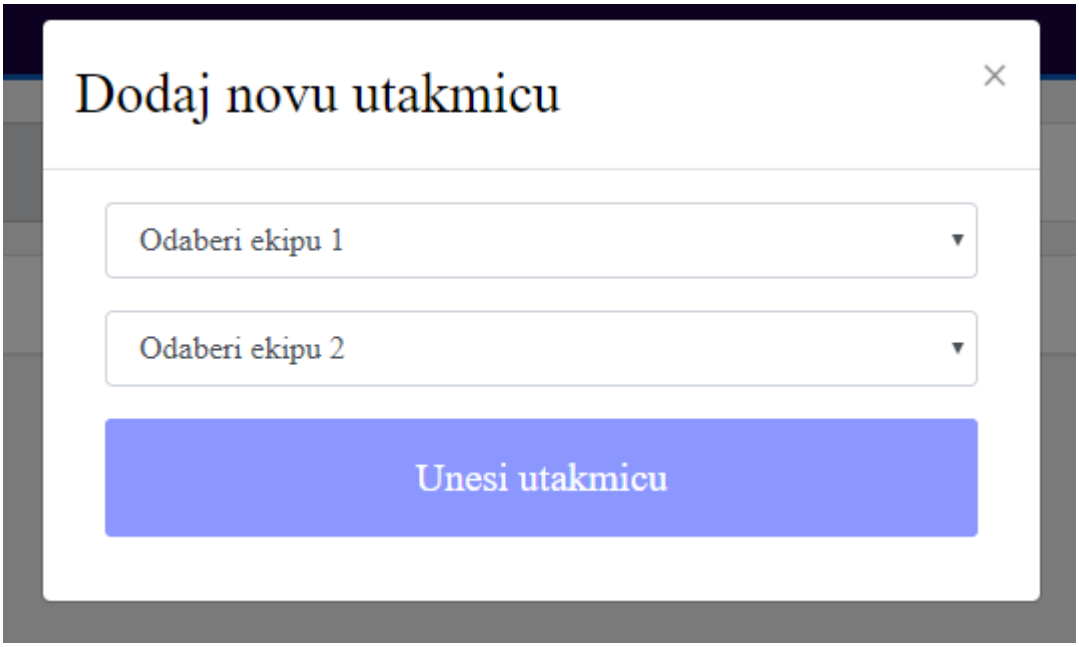



**Slika 5.10.** *Kreiranje završnice turnira.*




**Slika 5.11.** *Završnica turnira.*

Slika 5.12. prikazuje unošenje utakmice u završnicu. Otvaranjem izbornika prikazan je popis svih ekipa prijavljenih na turnir gdje organizator odabire jednu ekipu (ekipa 1) te protivnika (ekipa 2). Maksimalan broj utakmica koji se može dodati u četvrtfinale je 4, nakon toga organizator više nema mogućnost dodavanja novih utakmica, ali ima mogućnost uređivanja već postojećih rezultata. Slika 5.13. prikazuje rezultate dodanih utakmica u završnicu.




**Slika 5.12.** *Modalni oblik za dodavanje utakmice u završnicu.*

**Slika 5.13.** *Pregled rezultata utakmica završnice.*

# 6. ZAKLJUČAK

Web aplikacija za vođenje malonogometnih turnira kako i sam naziv govori, olakšava vođenje malonogometnih turnira organizatorima, a samim time i trenerima odnosno ekipama koje traže turnire da imaju pregled nadolazećih turnira na jednome mjestu. U radu su pobliže opisani tipovi i koncept malonogometnih turnira. Aplikacija omogućava ovisno o načinu na koji se korisnik registrirao (kao organizator ili kao trener) da kreira turnir ili se pridruži nekom od već postojećih. Aplikacija organizatorima omogućava unos ili uređivanje rezultata te naravno i uvid u poredak dok korisnici registrirani kao trener nisu u mogućnosti izmjenjivati rezultate. Gost na stranici samo ima mogućnost pregleda postojećih turnira i rezultata istih, ali nema prava kreirati ili biti sudionik na jednom od njih. Također opisane su tehnologije koje se koriste za izradu web aplikacije kao što je HTML koji se koristi za raspoređivanje elemenata i CSS koji se koristi za uređivanje izgleda web stranice uz Bootstrap razvojni okvir koji je korišten kako bi web aplikacija bila responzivna. PHP/MySQL koji je korišten za upravljanje bazom podataka te JavaScript za dinamičke promjene i dodavanje sadržaja na web stranicu. Omogućeno je praćenje rezultata natjecanja svim korisnicima.

# LITERATURA


[1] https://www.w3schools.com/html/default.asp (pristupljeno 27.05.2019.)

[2] I. Hickson, D. Hyatt, "HTML5: A vocabulary and associated APIs for HTML and XHTML". W3.org. W3C. 22 January 2008. dostupno na:

https://www.w3.org/TR/2008/WD-html5-20080122/ pristupljeno 28.05.2019.

[3] https://www.w3schools.com/css/css_intro.asp (pristupljeno 27.05.2019.)

[4] Rivest, R. (1992). The MD5 message-digest algorithm. dostupno na:

https://searchsecurity.techtarget.com/definition/MD5 pristupljeno 05.06.2019.

# SAŽETAK

**Naslov:** Web aplikacija za vođenje malonogometnog turnira

Zadatak završnog rada je napraviti web aplikaciju za vođenje malonogometnih turnira. Organizatori odnosno administratori stranice imaju mogućnost kreiranja vlastitih turnira, a korisnici imaju mogućnost prijave na otvorene turnire dok neregistrirani korisnici imaju uvid u otvorene turnire, ali nemaju mogućnost prijave na iste. Za izradu aplikacije korišten je HTML, CSS, PHP, MySQL, JavaScript te za responzivan dizajn front-end razvojni okvir Bootstrap. Za pisanje izvornog kôda korišten je Visual Studio Code radi lakšeg pisanja i pregleda kôda.

**Ključne riječi:** Bootstrap, malonogometni turniri, MySQL, PHP, responzivni dizajn

## ABSTRACT

**Title:** Web application for managing futsal tournament

The task of this paper is to create a web application for management futsal tournaments. The organizers who are administrators have the ability to create their own tournaments while users are allowed to sign up for an tournament and unregistered users have only insight into the tournament. Technologies used to create application are HTML, CSS, PHP, JavaScript and a responsive front-end framework Bootstrap. Visual Studio Code is used for easier writing and reviewing of the code.



**Keywords:** Bootstrap, Futsal tournaments, MySQL, PHP, Responsive design

# ŽIVOTOPIS

Krunoslav Kovač rođen 19.8.1997. godine u Đakovu. Završio je osnovnu školu Vladimira Nazora u Vinkovcima 2012. godine nakon čega iste godine upisuje Tehničku školu Nikole Tesle u Vukovaru, smjer Tehničar za računalstvo koju je završio 2016. godine. Iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija. Stručnu praksu je odradio u firmi HGSPOT informatika u Vinkovcima gdje je stekao iskustvo u održavanju i sastavljanju računala. U sklopu kolegija Multimedijska tehnika napravio je osobnu web stranicu. Koristi se engleskim jezikom te ima vrlo dobro znanje na računalu (korisnik Microsoft Office paketa) te poznaje programske jezike kao što su C, Java i Python. Iskustvo u izradi web stranica s primjenom HTML-a za prikaz i CSS-a za stiliziranje elemenata, JavaScript za dinamičko upravljanje i PHP/MySQL za upravljanje bazom podataka. Poznaje Microsoft .NET platforme i rad u Microsoft Visual Studio-u te ima iskustvo rada u grafičkim programima PhotoShop, Inkscape.

# PRILOZI

1. Projektna mapa s izvornim kôdom nalazi se na optičkom disku koji je priložen uz printanu verziju rada zajedno s word i pdf dokumentom.
2. Kôd se može pronaći na Git repozitoriju na linku: https://gitlab.com/kruuno/zavrsni-rad-futsal