|  |  |  |
| --- | --- | --- |
| ZnakUniverziteta | Univerzitet u Novom Sadu  Tehnički fakultet »MihajloPupin«  Zrenjanin | LOGOpupin |

**Školska 2025/26 godina**

**Nastavni predmet:** Arhitektura informacionih sistema

**SEMINARSKI RAD**

Tema: Fudbalski Turnir

|  |  |
| --- | --- |
| Predmetni nastavnik: | Student: |
| Prof. Dr. Biljana Radulović  Predmetni asistent:  MSc Velibor Premčevski | Dušan Slankamenac IT 26/22 |

Zrenjanin, 2026. godina

Sadržaj

[1. Uvod 2](#_Toc218792979)

[2. Korišćene tehnologije 3](#_Toc218792980)

[3. Korisničko uputstvo 4](#_Toc218792981)

[4. Ključni delovi koda 11](#_Toc218792982)

[4.1. Poslovna pravila i ograničenja 11](#_Toc218792983)

[4.2. Ograničenja u aplikaciji 13](#_Toc218792984)

[Literatura 16](#_Toc218792985)

# 1. Uvod

Ovaj seminarski rad opisuje razvoj veb aplikacije **"Fudbalski turnir"**, namenjene digitalizaciji procesa organizacije sportskih takmičenja. Glavni cilj aplikacije je da omogući efikasno upravljanje podacima o igračima, klubovima, korisnicima, menadžerima, sponzorima, turnirima i utakmicama, uz automatsko generisanje tabela grupne faze i vizuelni prikaz eliminacione faze (knockout system).

Sistem je dizajniran tako da jasno razgraniči uloge administratora, koji vrši unos i validaciju podataka, i običnih korisnika koji mogu da pregledaju turnir, rezultate...

# 2. Korišćene tehnologije

Aplikacija je zasnovana na modernom **Microsoft .NET** ekosistemu:

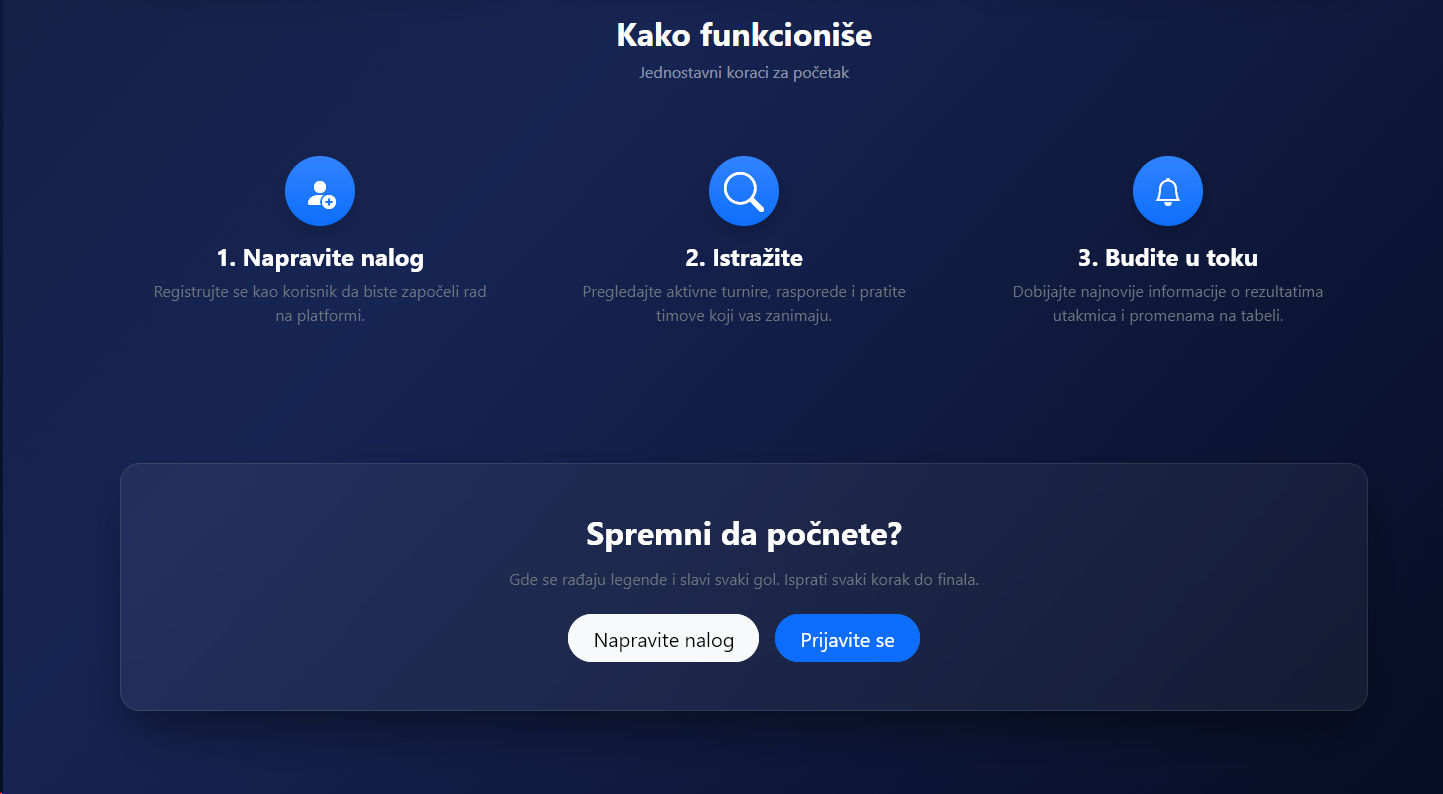
* **C# i .NET 8:** Osnovni programski jezik i radni okvir za razvoj **backend** logike.
* **ASP.NET Core MVC:** Korišćen za implementaciju **prezentacionog sloja** kroz **Model-View-Controller** obrazac.
* **Entity Framework Core (EF Core):** ORM alat za **mapiranje objektnog modela** u **relacionu bazu podataka** (Code First pristup).
* **Microsoft SQL Server:** Relaciona **baza podataka** za trajno skladištenje informacija.
* **ASP.NET Core Identity:** Sistem za **autentifikaciju** i **autorizaciju** korisnika na osnovu uloga (Roles).
* **Bootstrap i Razor:** Za **responsivni dizajn** i **dinamičko renderovanje** HTML sadržaja na strani servera.
* **Git** – Sistem za **kontrolu verzija** koji omogućava precizno praćenje izmena i upravljanje razvojnim granama (branches)
* **GitHub –** Servis koji služi kao cloud platforma za udaljeno skladištenje (remote repository) projekta. Uz pomoć **GitHub-a**, omogućava se siguran **backup** celog izvornog koda u oblaku i olakšava sinhronizacija između timova ljudi koji rade na 1 projektu. Platforma služi kao mesto za pregled **dokumentacije** i **upravljanje projektom**. **Issues** se koriste za praćenje zadataka i grešaka u programu (bagova), a **Pull Requests** za integraciju novih funkcionalnosti i popravljanje grešaka u programu.

# 3. Korisničko uputstvo

Početna stranica predstavlja centralni navigacioni panel aplikacije. Osmišljena je da korisniku pruži uvid u ključne funkcionalnosti sistema i olakša pristup najbitnijim sekcijama veb aplikacije.

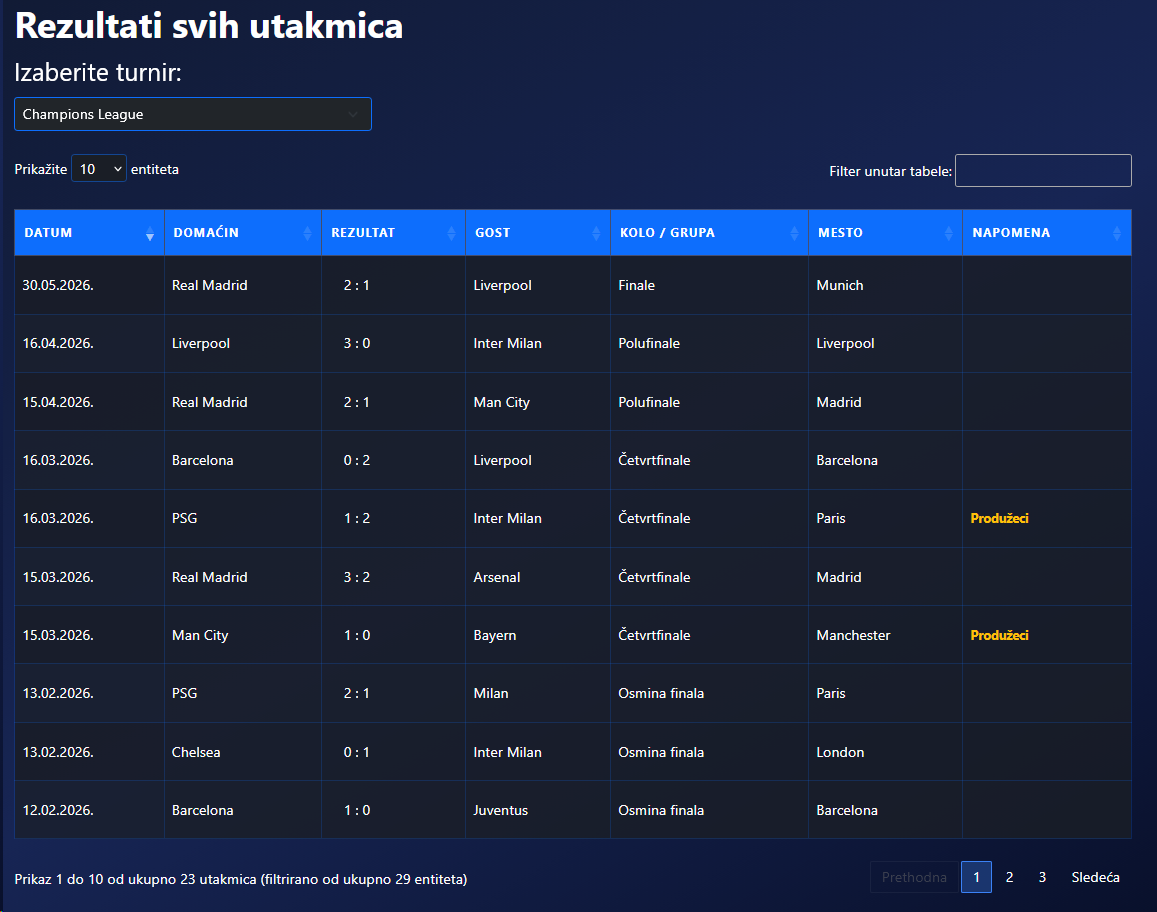


[Slika 1. Prikaz početne stranice]



[Slika 2. Prikaz početne stranice]

Stranica sa rezultatima nudi pregled svih odigranih mečeva. Implementiran je sistem filtriranja koji omogućava korisniku da brzo izoluje rezultate specifičnog turnira i svi podaci se automatski sortiraju po najnovijem igranom meču.

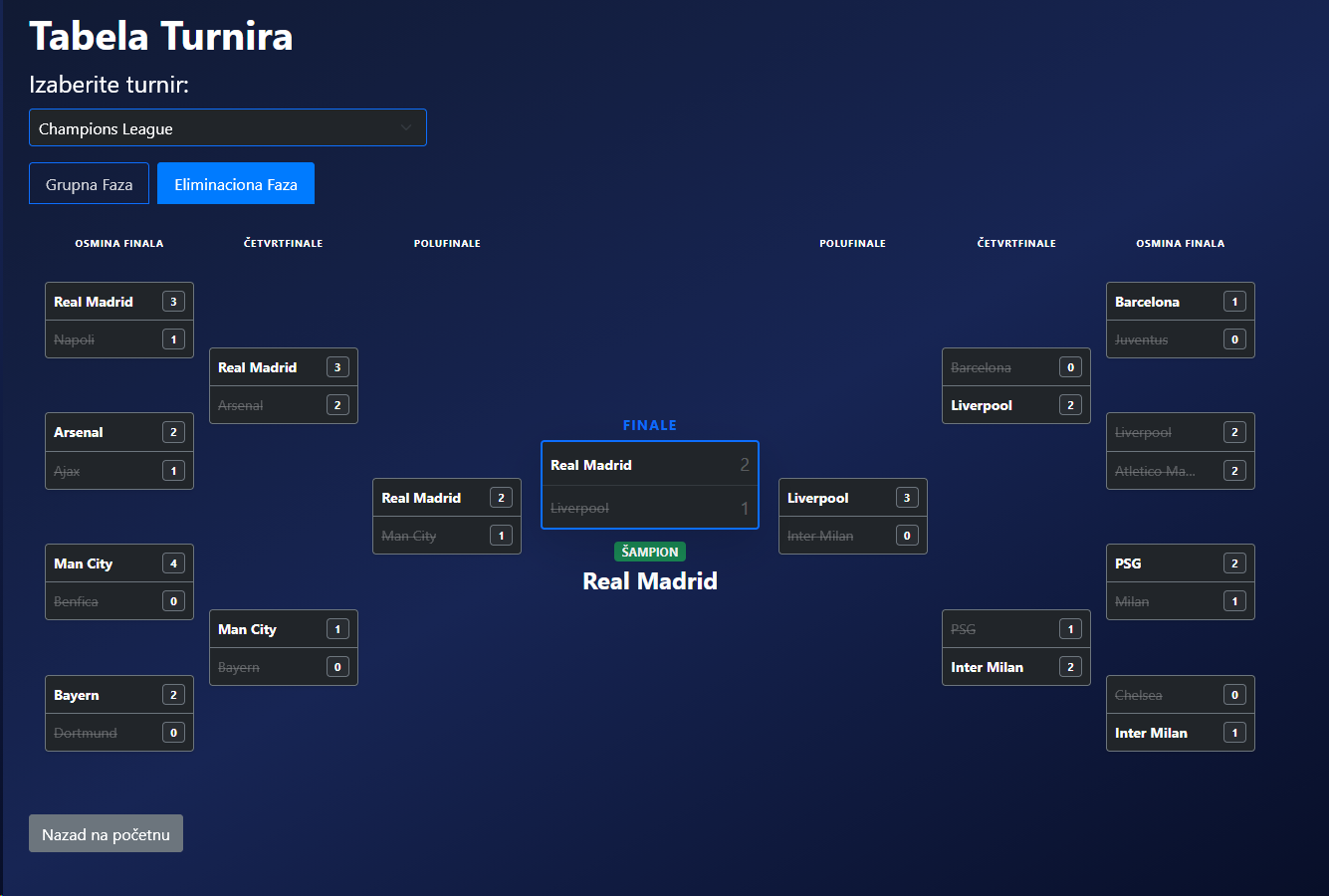


[Slika 3. Prikaz “Rezultati” stranica]

Ova sekcija namenjena je vizuelnom praćenju toka takmičenja. Korisnik može birati između **Grupne faze**, sa tabelarnim prikazom poena tokom grupne faze, i **Eliminacione faze**, prikazane u formi "bracket" dijagrama za praćenje knockout završnice.

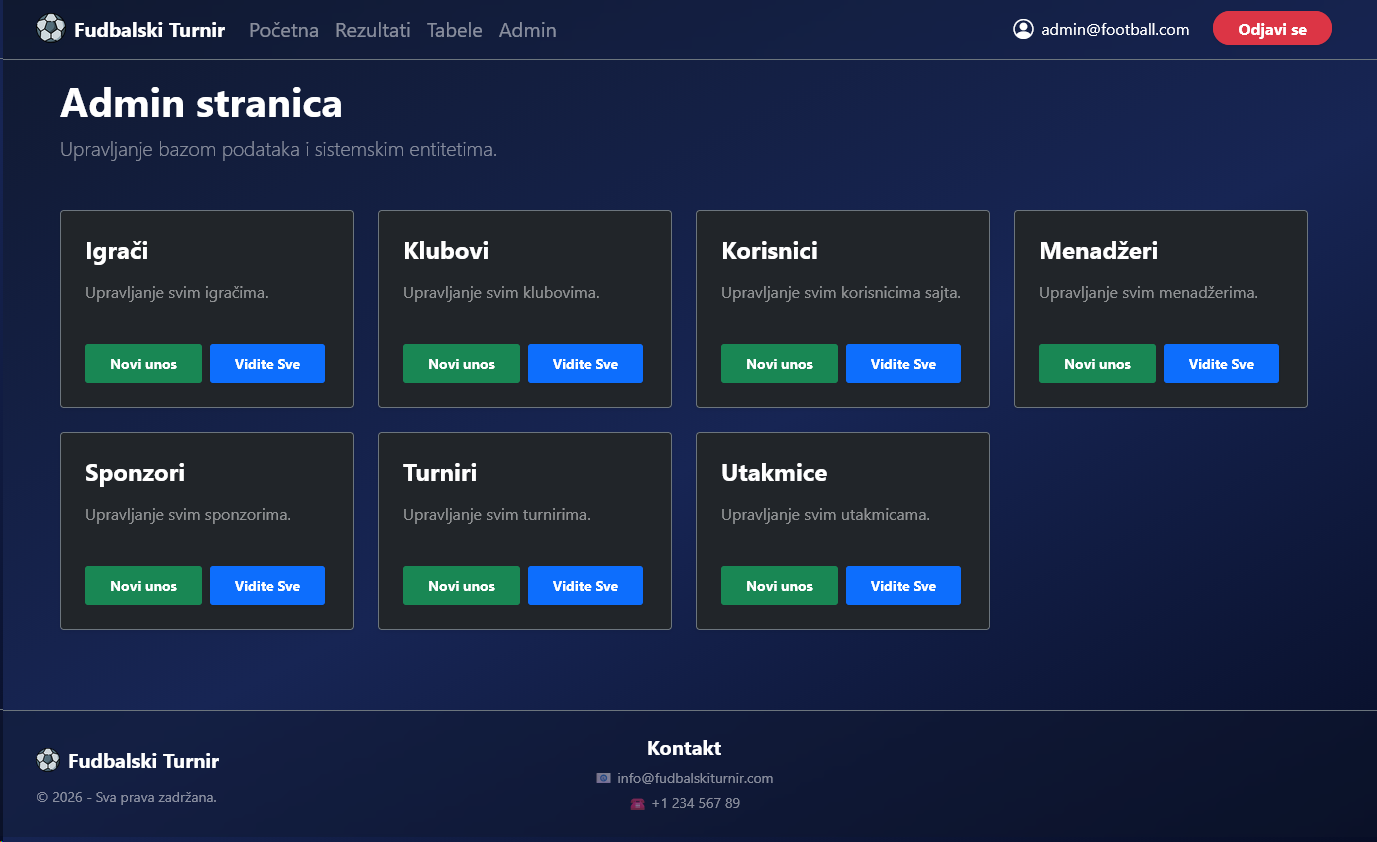


[Slika 4. Prikaz “Tabele” stranica, grupna faza deo]



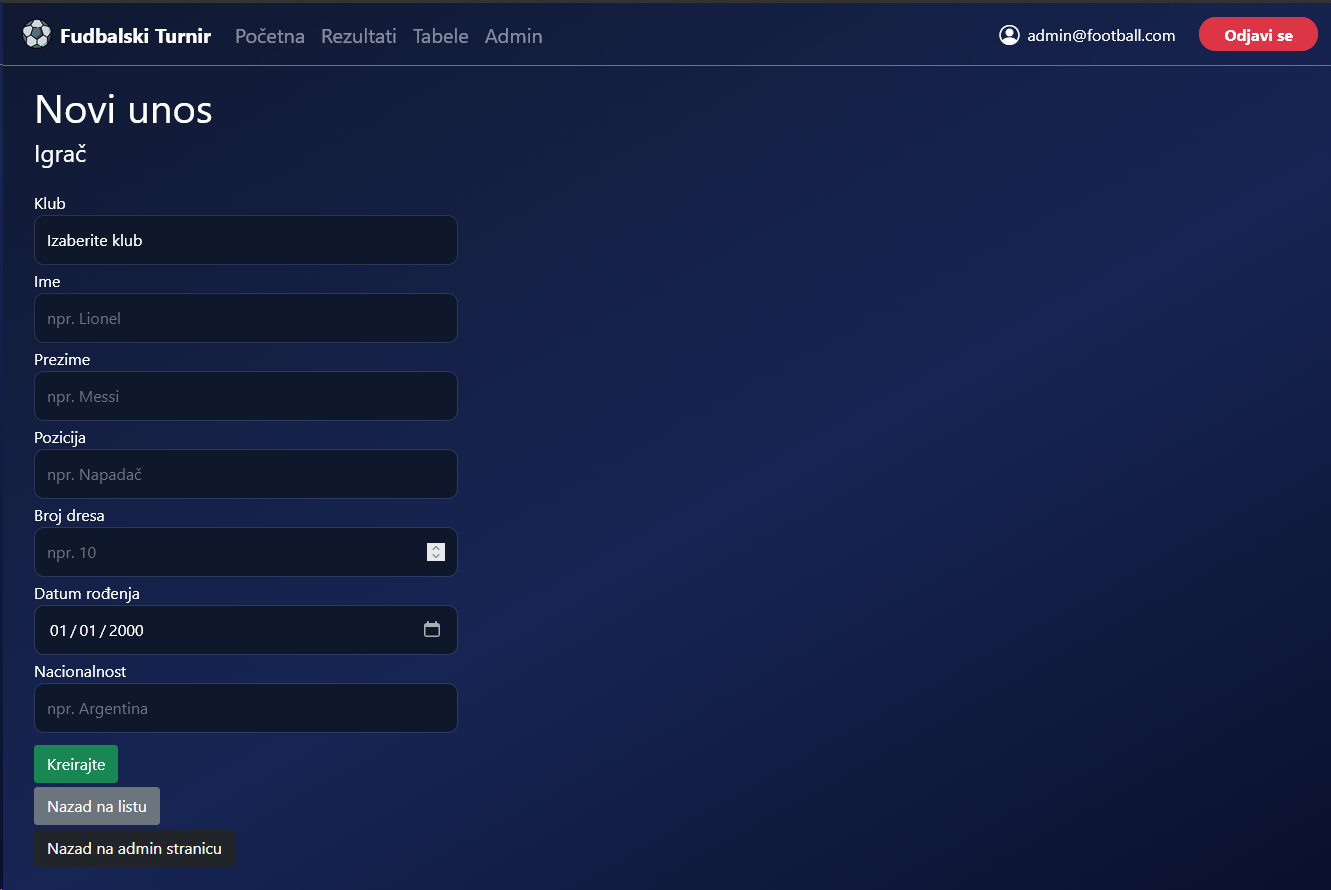
[Slika 5. Prikaz “Tabele” stranica, eliminaciona faza deo]

Administrativni interfejs predstavlja kontrolu same aplikacije i njihovih podataka. Dizajniran je za administratore, kako bi se omogućilo efikasno upravljanje bazom podataka i rad nad svim sistemskim entitetima.



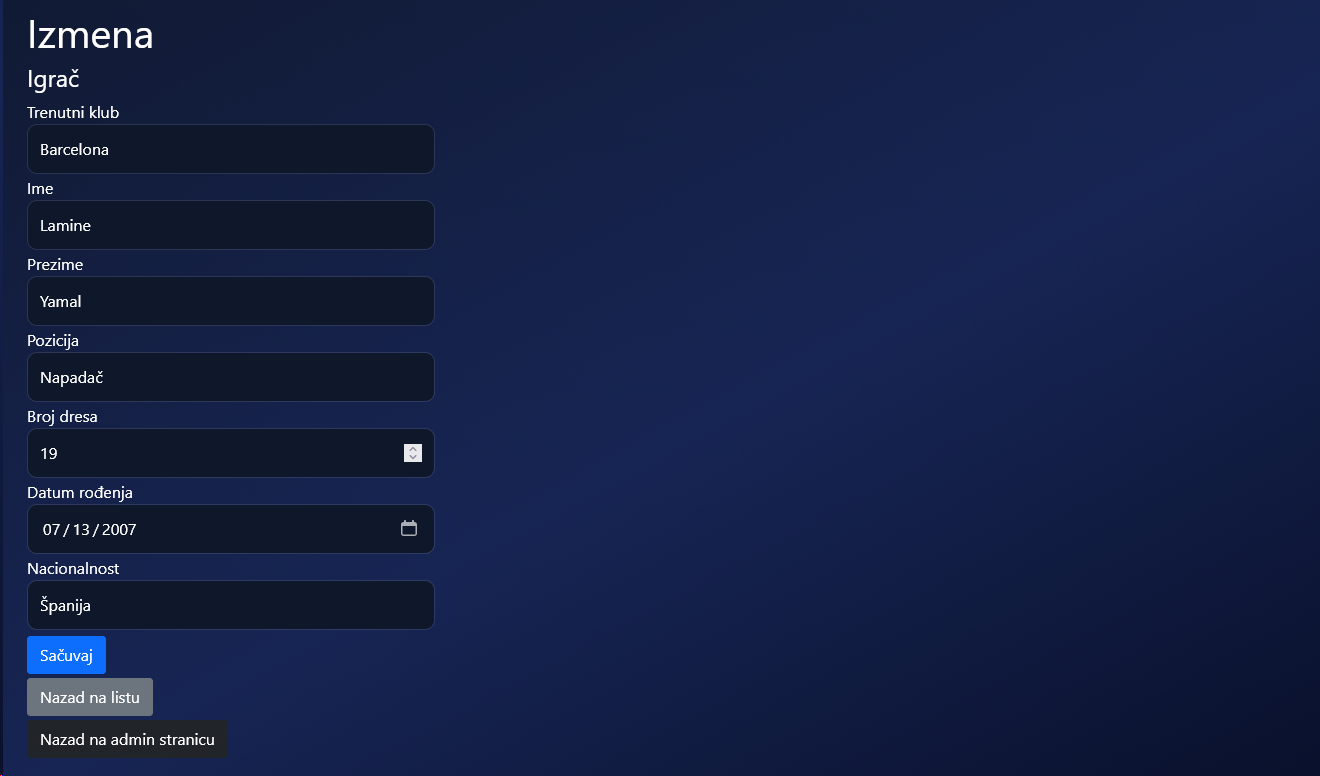
[Slika 6. Prikaz “Admin” stranice]

Nov unos stranica služi za ubacivanje novih podataka u bazu podataka.



[Slika 7. Prikaz stranice “Nov unos” za igrača]

Izmena je stranica koja omogućava ažuriranje postojećih informacija.



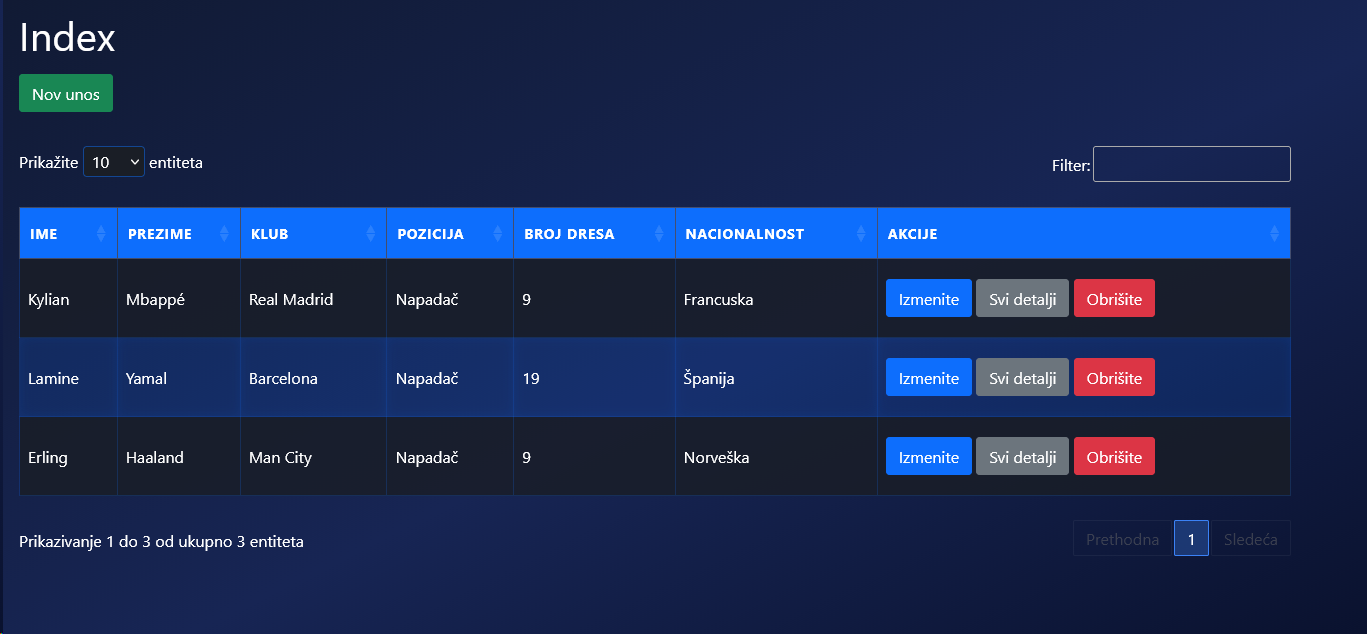
[Slika 8. Prikaz stranice “Izmenite” za igrača]

Svi detalji stranica pokazuje detaljan prikaz svih atributa specifičnog entiteta na jednom mestu.



[Slika 9. Prikaz stranice “Svi detalji” za igrača]

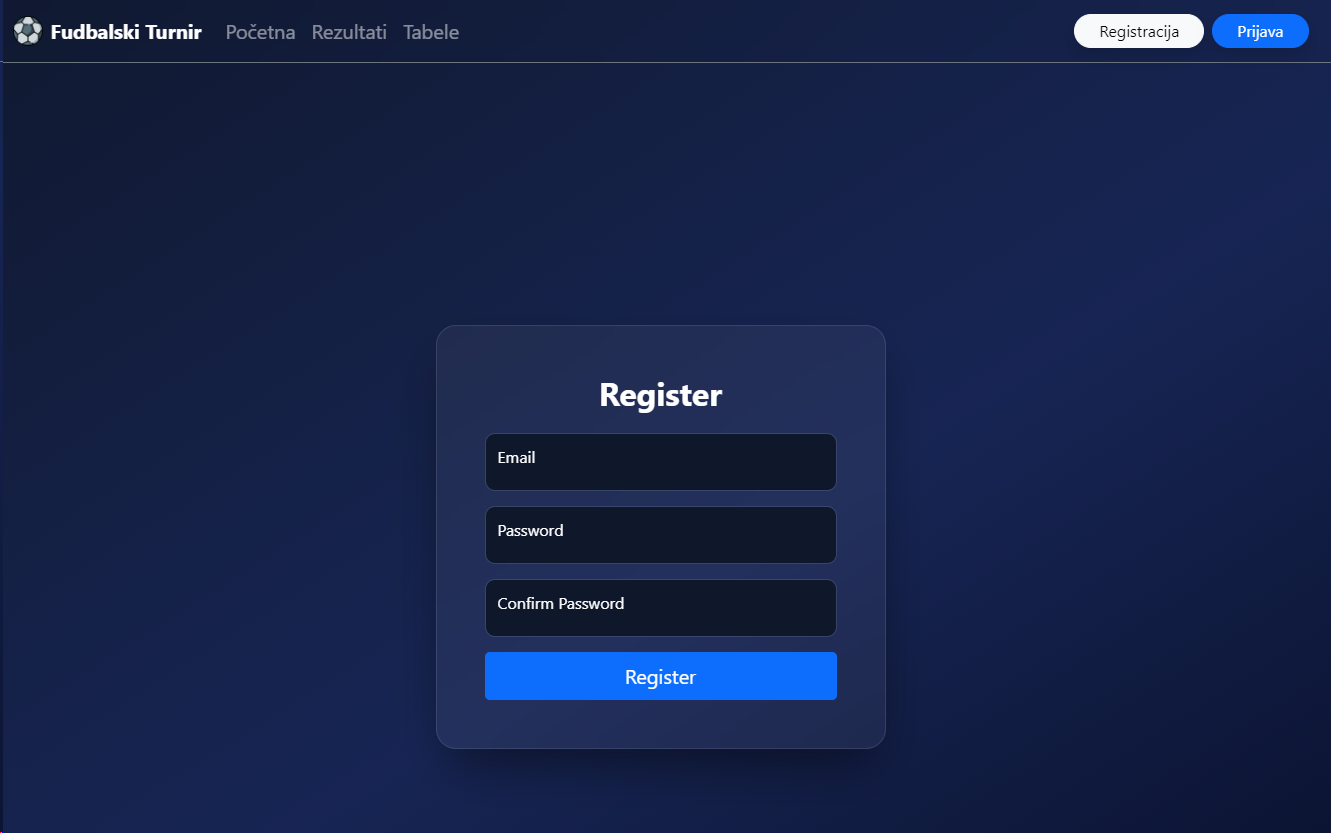
Centralna tabela koja prikazuje najvažnije informacije o svim stavkama određenog entiteta, uz mogućnost pretrage i brze navigacije.



[Slika 10. Prikaz stranice “Index” za igrača]

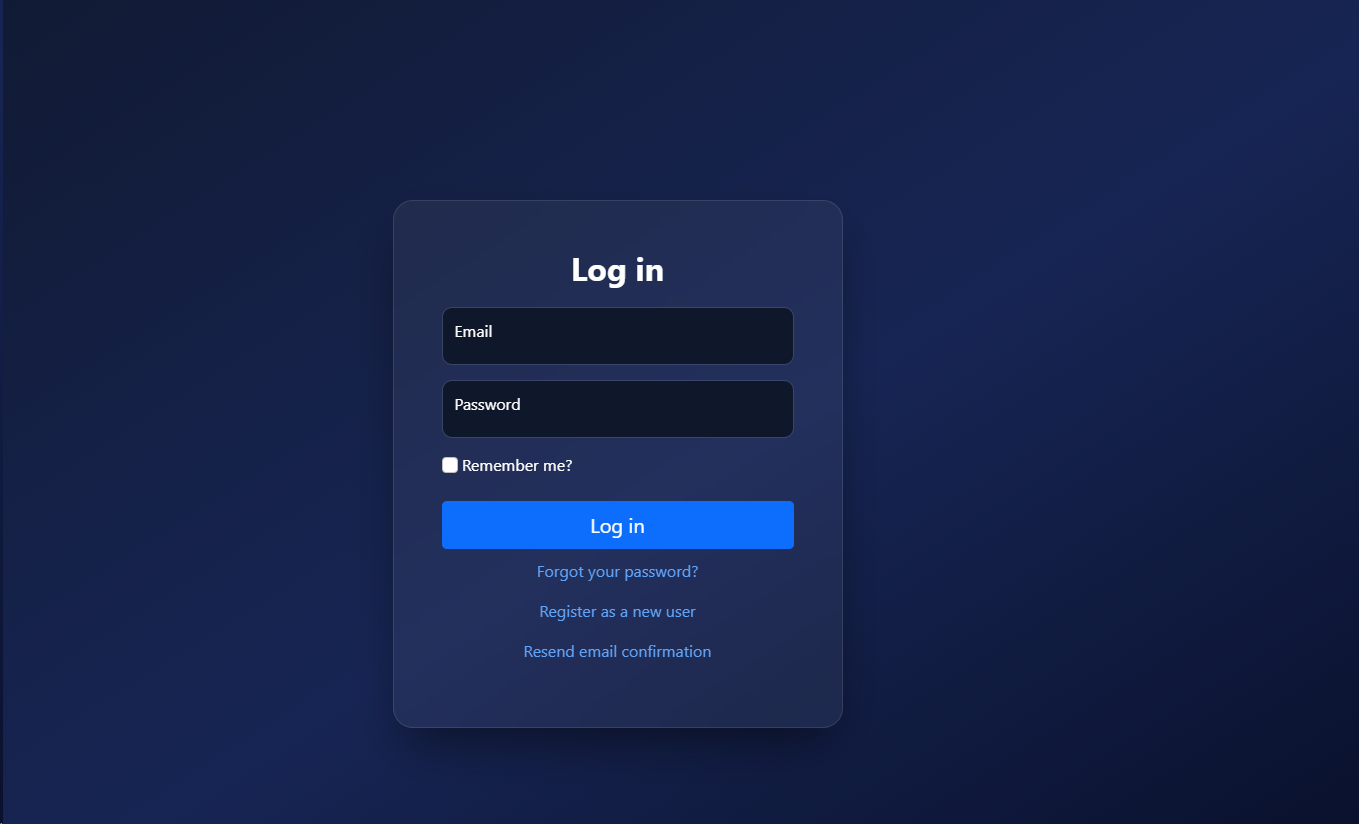
**NAPOMENA: Sve tabele** (Igrač, Klubovi, Korisnici, Menadžeri, Sponzori, Turniri, Utakmice) **koriste identične obrasce za interakciju** (Nov unos, Izmena, Svi Detalji, Index)

Registracija stranica omogućava novim korisnicima registraciju naloga na web platformi.



[Slika 11. Prikaz stranice “Registracija” za korisnika]

Prijava stranica služi za prijave i autentifikaciju postojećih korisnika na web aplikaciji.



[Slika 12. Prikaz stranice “Registracija” za korisnika]

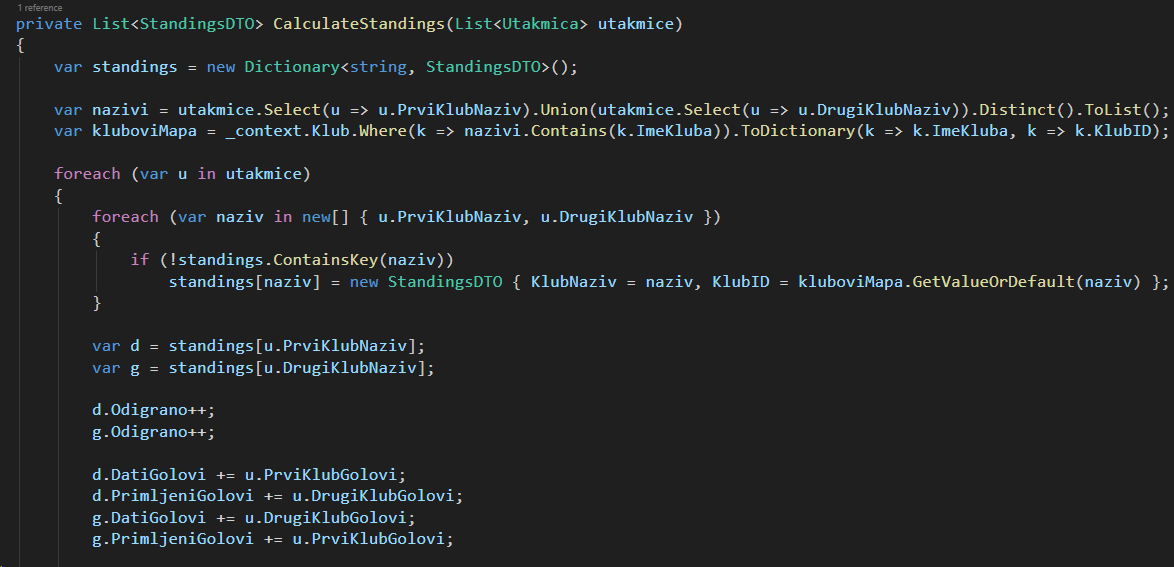
# 4. Ključni delovi koda

## 4.1. Poslovna pravila

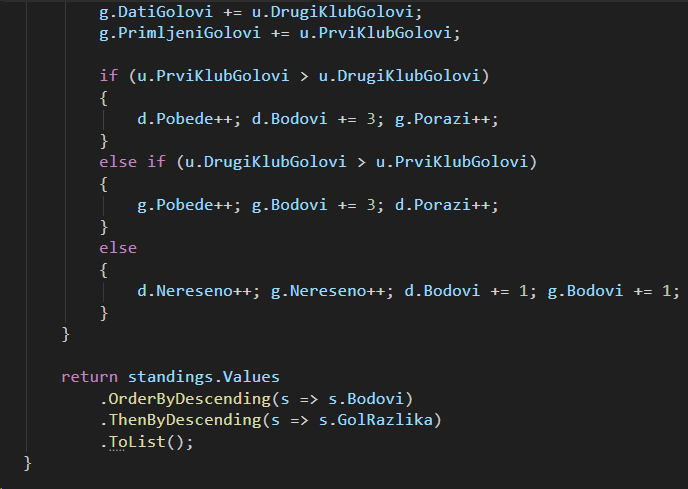
Poslovna pravila predstavljaju srž aplikacije i definišu način na koji sistem radi sa sportskim rezultatima i generišu tabelarni prikaz. Ova logika je implementirana u servisnom sloju i automatski se aktivira prilikom svake promene rezultata.

**Ažuriranje tabele u grupi**

* **Sistem bodovanja i automatizacija:** Aplikacija implementira zvanični sistem bodovanja gde pobeda donosi 3 boda, nerešen ishod 1 bod, dok se poraz ne boduje. Nakon svake registrovane utakmice, ponovo se vrši kalkulacija statistike za oba kluba (broj pobeda, poraza, izjednačenih , kao i datih i primljenih golova).
* **Kriterijumi rangiranja (Tie-breaking):** U slučaju da dva ili više klubova imaju isti broj bodova, poslovna logika primenjuje automatsko sortiranje prema gol-razlici. Ovim se osigurava objektivan i fer prikaz plasmana u realnom vremenu.
* **Integritet takmičarskih faza:** Sistem osigurava da se rezultati grupne faze ispravno akumuliraju, formirajući finalnu tabelu koja je osnov za dalji tok turnira.



[Slika 13. Prikaz prvog dela koda za kalkulisanje grupne faze]



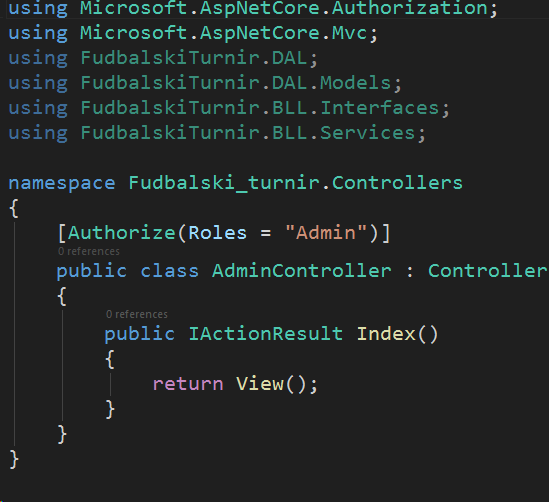
[Slika 14. Prikaz drugog dela koda za kalkulisanje grupne faze]

## 4.2. Ograničenja u aplikaciji

Kako bi se osigurao integritet podataka i regularnost takmičenja, implementiran je **sistem ograničenja**. Navedene stavke predstavljaju samo **karakteristične primere** primenjenih pravila koja se prožimaju kroz sve module aplikacije:

**Pristup podacima (Autorizacija):**

* **Admin:** Ima apsolutna prava (unos, izmena, brisanje svih podataka).
* **Korisnik:** Autorizovani korisnik koji može samo da pregleda rezultate, tabele i detalje, bez mogućnosti menjanja sadržaja.
* **Neulogovan korisnik:** Ima pristup isključivo početnoj stranici, kao i stranicama za registraciju i prijavu.



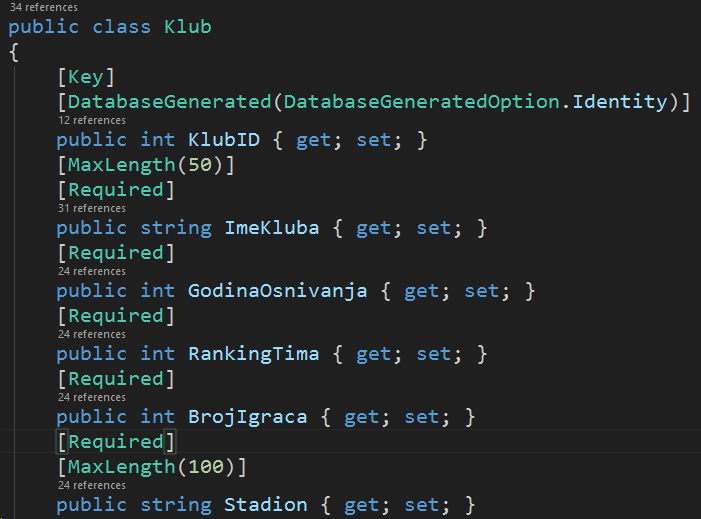
[Slika 15. Primer autorizacije admin stranice]

**Tehnička ograničenja polja (Database Constraints):** Aplikacija striktno kontroliše svako ulazno polje. Dole su navedeni reprezentativni primeri ovih ograničenja

* **Maksimalna dužina karaktera:** Sva tekstualna polja su definisana pomoću nvarchar tipa kako bi se optimizovao prostor. (Npr. Ime kluba do 50, Stadion do 100, Pozicija do 30 karaktera).
* **Obavezni unosi (Required):** Sva ključna polja (nazivi, datumi, brojevi dresova) su definisana kao NOT NULL, čime se sprečava kreiranje nepotpunih entiteta.
* **Numerički opsezi (Range):** Vrednosti poput godine osnivanja ili broja igrača su ograničene na realne logičke okvire.

**Bezbednosna polisa:**

* **Lozinka:** Implementirana Identity polisa koja zahteva minimalnu dužinu (6+ karaktera) i kompleksnost (kombinacija velikog i malog slova i specijalnog karaktera).

**Napomena:** Navedena ograničenja su deo šire logike validacije. Svaki entitet u sistemu (sponzori, turniri, itd.) imaju slična striktna pravila koja garantuju da baza podataka uvek ostaje u dobrom stanju. 

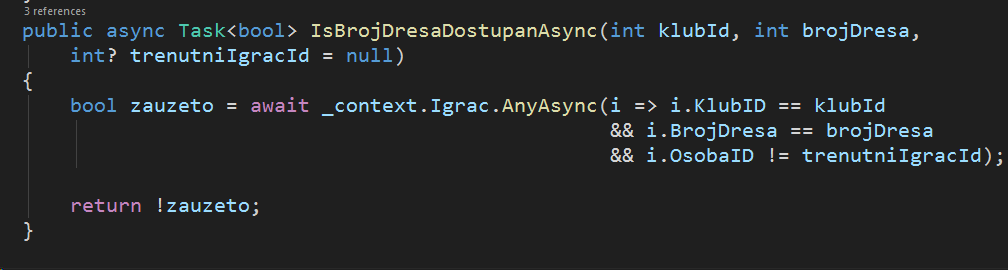
[Slika 16. Primer validacije podataka unutar modela “Klub”]

## 4.3. Specifične validacije

Validacije u aplikaciji nisu ograničene samo na osnovne provere, već čine **sistem kontrola** koji prati svaki unos podataka. Primarni cilj je prevencija logičkih nedoslednosti i zaštita baze podataka od nevalidnih informacija.

**Provera jedinstvenosti broja dresa u klubu**

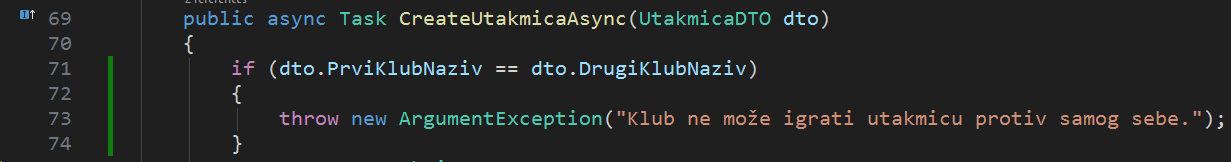
Aplikacija sadrži validacionu logiku na nivou servisa i modela koja osigurava da svaki igrač unutar istog kluba nosi jedinstven broj dresa, ako u bazi već postoji igrač dodeljen tom klubu sa istim brojem dresa, aplikacija blokira upis i vraća poruku o grešci.



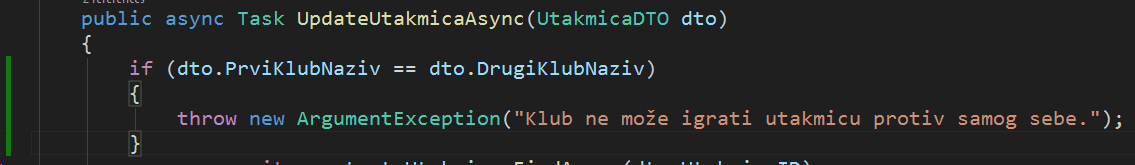
[Slika 17. Prikaz koda za proveru broj dresa]

**Validacija dupliranja klubova na utakmici**

Aplikacija sprečava unos/izmenu utakmice u kojoj jedan klub igra sam protiv sebe. Sistem upoređuje nazive izabranog domaćina i gosta pre čuvanja u bazu. Ukoliko je izabran klub sa istim imenom dvaput aplikacija blokira unos/izmenu.



[Slika 18. Prikaz prvog dela koda za validaciju klubova u utakmici]



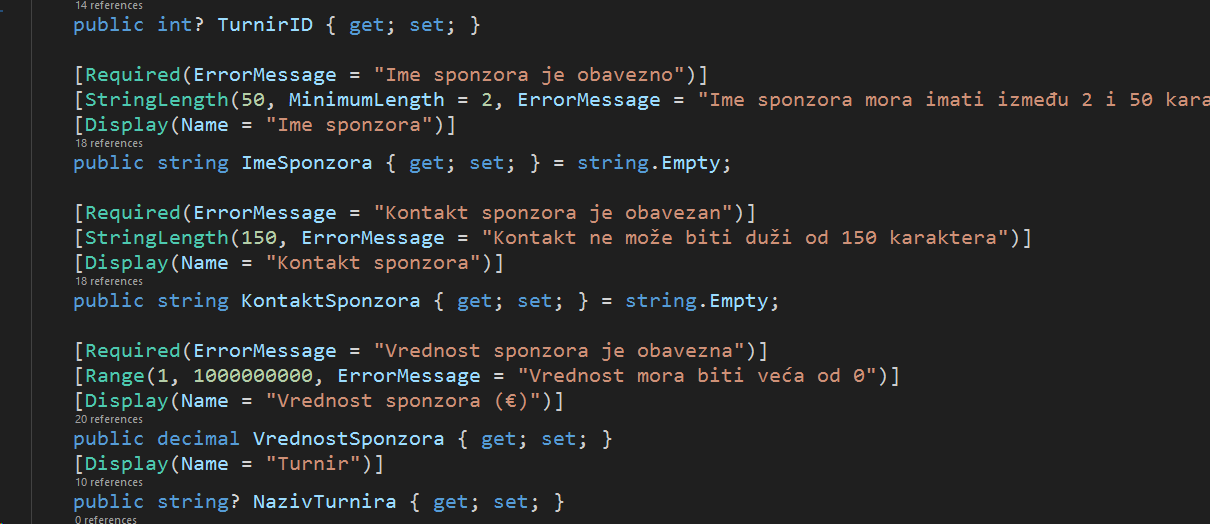
[Slika 19. Prikaz drugog dela koda za validaciju klubova u utakmici]

**Kategorizacija ostalih validacionih pravila**

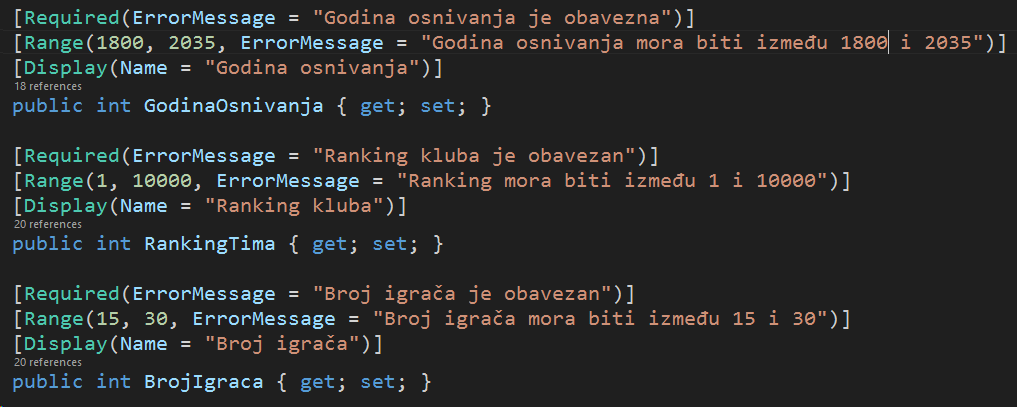
Iako sistem sadrži različite validacije za svaki entitet (igrači, klubovi, korisnici, menadžeri, sponzori, turniri i utakmice) one se mogu grupisati u sledeće reprezentativne kategorije:

* **Validacija formata i dužine:** Sva tekstualna polja prolaze proveru dužine kako bi se očuvao integritet dizajna i baze. Na primer, ime sponzora je ograničeno na 2 do 50 karaktera, dok su kontakt informacije limitirane na 150 karaktera radi preciznosti.
* **Logički numerički opsezi:** Aplikacija striktno kontroliše realnost unetih brojeva. To uključuje provere poput godine osnivanja kluba (opseg 1500–2035) ili broja igrača u timu (15–30), čime se sprečavaju faktografske greške.
* **Finansijska i statistička ispravnost:** Vrednosti poput sponzorskih ugovora moraju biti strogo pozitivne (veće od 0), dok se za golove i titule primenjuje logika koja onemogućava unos negativnih vrednosti.
* **Relaciona i poslovna pravila:** Pored navedenih primera, sistem sprovodi i kompleksnije provere poput sprečavanja dupliranja brojeva dresova unutar istog kluba ili onemogućavanja da klub odigra utakmicu sam protiv sebe.

**Napomena:** Navedeni primeri predstavljaju samo deo implementiranog sistema validacije. Svako polje u aplikaciji poseduje specifične atribute i serversku logiku koja garantuje da sistem ostaje stabilan i zaštićen od pogrešnog unosa.



[Slika 20. Primer validacija podataka za sponzore]



[Slika 21. Primer validacija podataka za klubove]

# 5. Zaključak

Razvijena veb aplikacija „Fudbalski turnir” uspešno digitalizuje ključne aspekte organizacije sportskih takmičenja. Korišćenjem savremenih tehnologija poput **ASP.NET Core MVC** i **Entity Framework Core**, kreiran je sistem koji osigurava integritet podataka kroz stroge validacije i poslovna pravila. Implementirani algoritmi za automatsko ažuriranje tabela i vizuelni prikaz eliminacione faze značajno olakšavaju rad administratorima, dok korisnicima pružaju transparentan i intuitivan uvid u tok turnira.

**Prostor za unapređenja**

**Napredna statistika:** Uvođenje detaljnije analitike za igrače (broj postignutih golova, asistencije, kartoni).

**Sistem za generisanje izveštaja:** Izvoz tabela i statistika u PDF ili Excel format.

**Algoritam za automatsko generisanje knockout faze:** Implementacija logike koja bi, nakon zvaničnog završetka grupne faze, automatski uparivala klubove za osminu finala (npr. Prvi iz Grupe A protiv Drugog iz Grupe B). Ovo bi eliminisalo ručni unos parova i smanjilo mogućnost ljudske greške u organizaciji završnice.

**Implementacija dvomečnog eliminacionog sistema:** Proširenje knockout faze opcijom za igranje revanš mečeva (kod kuće i u gostima). Ovo bi zahtevalo nadogradnju poslovne logike za obračun ukupnog rezultata (aggregate score) i automatsko prepoznavanje potrebe za penalima nakon druge utakmice.

**Github repozitorijum link**

<https://github.com/DusanSl/Fudbalski-Turnir-ASP.NET>

# Literatura

1. Bootstrap documenation - <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
2. Microsoft Documentation Views - <https://learn.microsoft.com/en-us/aspnet/core/mvc/views/overview?view=aspnetcore-10.0&WT.mc_id=dotnet-35129-website>
3. Microsoft Documentation Controllers - <https://learn.microsoft.com/en-us/aspnet/core/mvc/controllers/actions?view=aspnetcore-10.0&WT.mc_id=dotnet-35129-website>
4. Microsoft Documentation ASP.NET Core MVC - <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-10.0&WT.mc_id=dotnet-35129-website>
5. Microsoft Documentation code first approach - <https://learn.microsoft.com/en-us/ef/ef6/modeling/code-first/workflows/new-database>
6. GitHub documentation - <https://docs.github.com/en>
7. Jquery documentation - <https://datatables.net/reference/option/>
8. GitHub repo - <https://github.com/VeliborP/ais-2025>