

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Hrvoje Dumančić

**Baza podataka za Američku nacionalnu
nogometnu ligu**

SEMINARSKI RAD IZ KOLEGIJA BAZE PODATAKA 2

Varaždin, 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Hrvoje Dumančić

Matični broj: 43981/15–R

Studij: Informacijski sustavi

**Baza podataka za Američku nacionalnu
nogometnu ligu**

SEMINARSKI RAD IZ KOLEGIJA BAZE PODATAKA 2

Mentor:

Martina Šestak, mag inf.

Varaždin, prosinac 2017.

Hrvoje Dumančić

Izjava o izvornosti

Izjavljujem da je moj projektni rad izvorni rezultat mog rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Baza podataka za Američku nacionalnu nogometnu ligu je projekt stvoren u svrhu kolegija Baza podataka 2. Projekt se sastoji od backend i frontend. Backend dio je izrađen u programskom jeziku MySQL, koristeći alat Workbench za kreiranje same baze podataka, dok je frontend dio projekta izrađen u programskim jezicima PHP, HTML i CSS. Za pisanje samih kodova u jezicima za frontend korištena je softverska platforma NetBeans.

Sadržaj

1. Uvod	1
2. Korišteni alati	2
2.1. MySQL Workbench	2
2.2. XAMPP	3
2.3. NetBeans.....	3
3. Kreiranje baze podataka	4
3.1. Tablice	4
3.2. ERA model	9
4. Okidači.....	11
5. Web aplikacija	14
5.1. Navigacijska traka	14
5.2. Početna stranica / upis novog igrača	15
5.3. Upis susreta	18
5.4. Tablica rezultata	21
5.5. Upiti.....	22
6. Zaključak	24
7. Literatura	25

1. Uvod

U sažetku su površno opisani alati u kojima je izrađena baza podataka, no nije objašnjeno koji je motiv za izradu baze podataka američke nogometne lige. Osobno imam veliko zanimanje za američki nogomet koji pratim već duže vrijeme, zbog čega sam se odlučio na izradu baze podataka upravo za to. Veliko rivalstvo između nekih timova i igrača, preokreti u utakmicama u zadnjim minutama te nepredvidivost svake utakmice su upravo stvari zbog kojih mi je Nacionalna nogometna liga u segmentima bolja od europskog nogometa. Samim radom na bazi podataka ću raditi na stvari koja me interesira i usput učiti nove pojmove koji mi prije nisu bili poznati, u svrhu što bolje izrade aplikacije i prezentiranja projekta.

Htio sam napraviti kompaktnu bazu podataka samo s najosnovnijim elementima zbog čega imam svega jedanaest tablica u bazi koje su sve međusobno povezane na određene načine radi optimalnog korištenja te koje će kasnije biti detaljnije objašnjene.

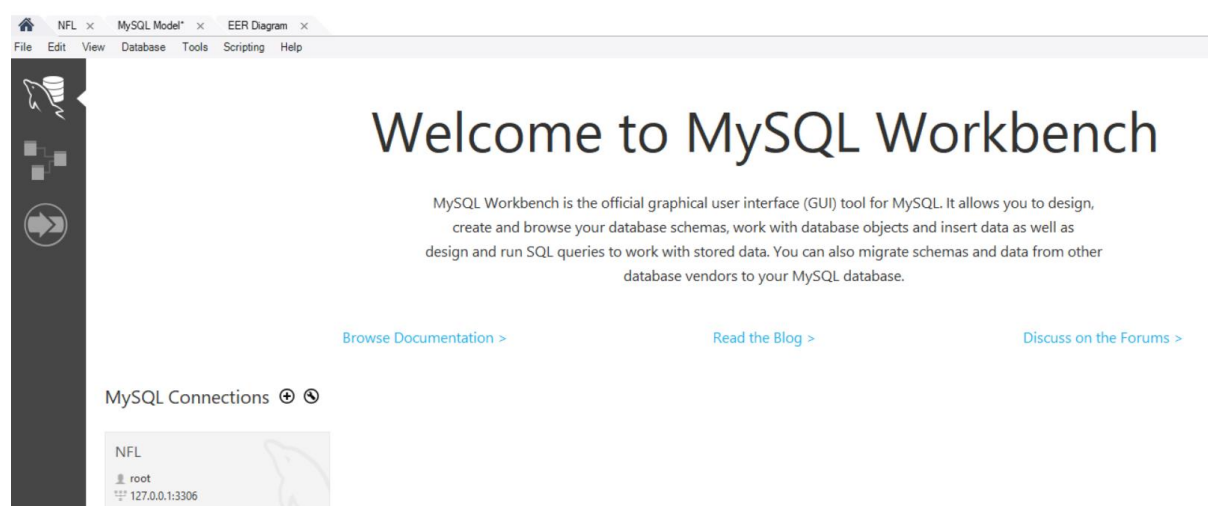
2. Korišteni alati

Kao što je već prethodno objašnjeno, za backend se koristio programski jezik MySQL pomoću alata Workbench, dok se za frontend koristio HTML/CSS, PHP te su se kodovi za te programske jezike pisali pomoću alata NetBeans. Virtualni web poslužitelj napravljen je koristeći XAMPP, besplatan paket koji obuhvaća razne samostalne elemente, Apache Servers, MySQL Server, PHP te olakšava postupak spajanja baze podataka na lokalni server te na poslužitelj.

2.1. MySQL Workbench

Alat koji se koristi za modeliranje baze podataka, MySQL Workbench je grafički alat za rad s MySQL serverima i bazama podataka koji nudi razne mogućnosti kao što su SQL razvoj, administracija, dizajn baze podataka te druge koje nisu fokus ovog projekta.

Kreiranje i razvoj baza podataka možemo realizirati pomoću pisanja samog MySQL koda u SQL Editoru, a možemo i preko izgrađenog sučelja za izradu i popunjavanje tablica. U Workbenchu nema ograničenja na broj primarnih ključeva, kao što je to slučaj kod drugih alata (npr. Oracle Apex), zbog čega se tablice u potpunosti mogu kreirati s primarnih i vanjskim ključevima, kao što je u projektu i rađeno. S druge strane, lakši način za kreiranje okidača je preko SQL Editora, gdje se pišu u cijelosti, te se kasnije mogu uređivati u sučelju za izradu tablica pod kategorijom Triggers. Također, jedna od prednosti Workbencha je mogućnost izrade ERA dijagrama već postojeće baze podataka temeljem Reverse Engineeringa. [1]



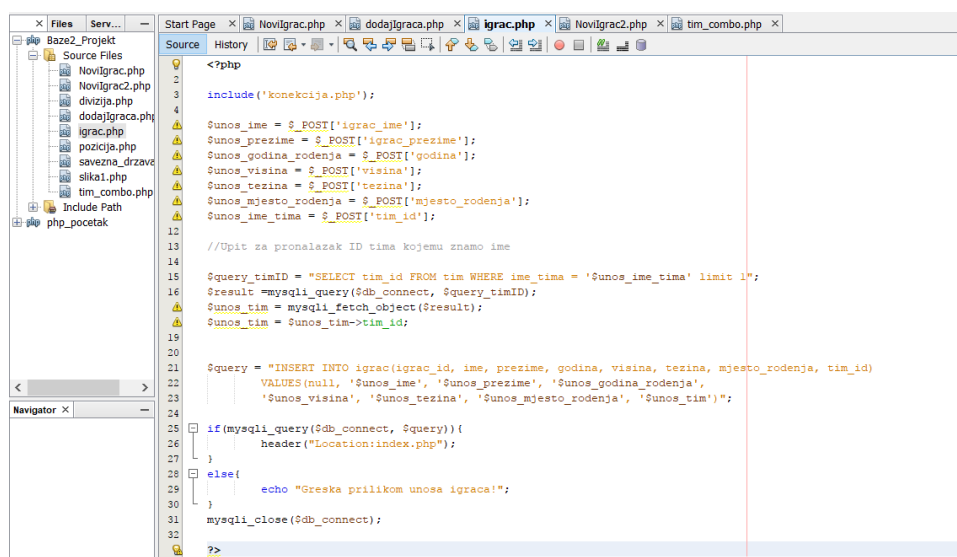
Slika 1. MySQL početni zaslon

2.2. XAMPP

Cross-platform skup alata za kreiranje jednostavnog web servera je najprepoznatljivija karakteristika XAMPP-a. Sve potrebno se već nalazi unutar skupa kojeg je potrebno samo instalirati nakon čega se može započeti rad na frontendu za bazu podataka. Sadržava podršku za PHP, HTML/CSS zbog čega je jedan od najpopularnijih alata za razvoj web aplikacija. Samo ime XAMPP je zapravo skraćenica u kojoj svako slovo predstavlja jedan od alata koji ga tvore ili kojima je namijenjen. X označava mogućnost korištenja na bilo kojoj platformi, Linuxu, Windowsima ili Apple operacijskim sustavima, A označava Apache HTTP Server, M za MariaDB, odnosno prijašnji MySQL, prvo P za PHP te drugo P za Perl programske jezike. [2]

2.3. NetBeans

Preferirana platforma za izradu web stranica i aplikacija je NetBeans, u kojoj je i rađen ovaj projekt. Moguće je raditi datoteke u PHPu, HTMLu i CSSu bez ikakvih problema s ugrađenim potporama za pojedine Internet preglednike kako bi svaka promjena bili vidljiva odmah na uvid. Osim tih programskih jezika, moguće je raditi i u drugima kao što je Java ili JavaScript te izravno povezivanje na GitHub kako bi se svi podaci distribuirali svim potrebnim ljudima unutar napravljenih skupina, koje također rade na istom projektu. Kao i XAMPP, također je multiplatformskog tipa te je moguće koristiti Netbeans na Microsoft Windowsima, Apple operacijskim sustavima te Linuxu. [3]



Slika 2. Prikaz koda u NetBeansu

3. Kreiranje baze podataka

3.1. Tablice

Nakon prethodno obavljenih potrebnih spajanja pomoću XAMPPa, moguće je početi raditi bazu podataka u MySQL Workbenchu. Baza podataka o američkoj nogometnoj ligi sastoji se od deset međusobno povezanih tablica koje govore o timovima, igračima i njihovim statistikama, divizijama i odigranim susretima.

Prva tablica baze podataka je tablica igrač koja sad atribute `igrac_id`, koji je ujedno i primarni ključ tablice, ime, prezime, visina, težina, godina koji označava godinu rođenja pojedinog igrača, mjesto_rođenja i `tim_id`, vanjski ključ na tablicu `tim` u kojoj se nalaze momčadi lige. Iako se u bazu podataka upisuje vanjski ključ pojedinog tima u polje `tim_id` koji je definiran kao serial, u frontendu je radi pojednostavljivanja korisničkog unosa napravljeno da se tim odabire iz padajuće liste koja se povlači direktno iz baze podataka te tako daje izbor svih timova prema njihovim punim imenima, kako bi korisnik koji unosi igrače mogao unijeti samo tim, bez potrebno prethodnog poznavanja samog ID-a tima.

igrač	Igrac_id (int 10)	Primary Key, Unsigned, Auto Increment
	Ime varchar(45)	Not Null
	Prezime varchar(45)	Not Null
	Godina (int 10)	Not Null, Unsigned
	Visina (int 10)	Not Null, Unsigned
	Težina (int 10)	Not Null, Unsigned
	Mjesto_rođenja varchar(45)	Not Null
	Tim_id (int 10)	Foreign Key(Tim), Not Null, Unsigned

`Igrac_id` je vanjski ključ u dvjema tablicama, pozicija igrača te statistike igrača. Pozicija igrača je određena među tablica, nastala zbog potrebe spajanja dvije tablice između kojih je veza više naprema više, zbog čega smo morali stvoriti određenu tablicu koja sadržava atribute `igrac_id`, `pozicija_id` te postava. `Igrac_id` i `pozicija_id` djeluju kao zajednički primarni

ključevi, ali su ujedno reference na tablice susreti i igrač, dok postava govori o tome označava li pojedina pozicija da je momčad u napadu, obrani ili je specijalni tim na terenu. [4]

pozicija_igrača	Igrac_id (int 10)	Primary Key, Foreign Key (Igrač), Unsigned, Auto Increment
	Pozicija_id (char 2)	Primary Key, Foreign Key (Pozicija)
	Postava (varchar 10)	Not Null

Tablica na koju se tablica pozicija_igrača isto nadovezuje je pozicija, tablica koja ima zapisane sve moguće pozicije igrača na terenu. Sastoji se od tri atributa, pozicija_id koji je glavni ključ, te se u njega upisuje skraćenica od dva slova za svaku poziciju na terenu, ime_pozicije u kojem se nalazi puno ime svake pozicije te postava, govori li se o poziciji u napadu, obrani ili o specijalnom timu.

pozicija	Pozicija_id (char 2)	Primary Key
	Ime_pozicije (varchar 45)	Not Null
	Postava (varchar 10)	Not Null

Kao što je prije rečeno, dvije tablice se referenciraju na tablicu igrač, jedna je opisana te je ostala jedna, statistike_igrača. Ta tablica govori o pojedinim uspjesima igrača u utakmicama, koje se upisuju radi poznavanja njihove statistike i mjere uspješnosti u igri. Tako se tablica sastoji, osim od igrač_id, vanjskog ključa na tablicu igrač te primarnog ključa same tablice, još od atributa rushing_yards, odnosno broja prijeđenih jardi, touchdowns, broja postignutih polaganja kao igrača ili kao quaterbacka, bacača lopti, interceptions, broja ukradenih lopti za vrijeme igre od druge momčadi te sacksa, broja uspješnog savladavanja protivničkih igrača za vrijeme igre. Potrebno je naglasiti kako svi atributi osim primarnog ključa imaju postavljene vrijednosti na 0 po defaultu, te se one mijenjaju prilikom unosa, nakon čega više ne smiju biti prazne. Također su svi atributi osim rushing_yards postavljeni s ograničenjem unsigned, jer ne smiju poprimiti negativne vrijednosti, dok atribut

rushing_yards smije, jer postoji mogućnost da osoba prilikom pokušaja osvajanja zapravo izgubi jarde, što je nepovoljno za njegov tim.

Statistike_igrača	Igrač_id (int 10)	Primary Key, Foreign Key (Igrač), Unsigned, Auto Increment
	Rushing_yards (int 10)	
	Touchdowns (int 10)	Unsigned
	Interceptions (int 10)	Unsigned
	Sacks (int 10)	Unsigned

Tablica igrač ima vanjski ključ na tablicu tim, koju možemo smatrati okosnicom cijele baze podataka jer povezuje najviše tablica i jedna je od najbitnijih. Sastoji se od atributa tim_id, ime_tima, stadion, naziv_divizija te drzava_id. Tim_id je primarni ključ tablice definiran kao integer koji se auto inkrementira, odnosno kao serial. Sljedeći atribut je ime_tima u koji zapisujemo potpuno ime tima, zbog čega je stavljen varchar 255, kao i u atribut stadion. Naziv_divizija te drzava_id su vanjski ključevi od tablica divizija te savezna_drzava koji daju informacije sukladno imenima, u kojoj diviziji se nalazi momčad te iz koje savezne države dolazi.

Tim	Tim_id (int 10)	Primary Key, Unsigned, Auto Increment
	Ime_tima (varchar 255)	Not Null, Unique
	Stadion (varchar 255)	Not Null, Unique
	Naziv_divizija (varchar 45)	Not Null
	Drzava_id (char 2)	Not Null

Sljedeća tablica koju je potrebno opisati je savezna_država, jednostavna tablica koja sadrži popis svih saveznih država Sjedinjenih Američkih Država, kako bi mogli za svaki tim odrediti državu iz koje dolazi. Iako postoji pedeset saveznih država, a svega trideset i dva tima se nalazi u ligi, ne nalazi se svaki tim u jednoj državi, zbog čega je potrebno napraviti vezu jedan naprema više, gdje vanjski ključ uvijek ide na stranu više, odnosno na tablicu tim. Tako država California ima čak 4 tima, Oakland Raiderse, San Francisco 49erse Los Angeles Ramse te Los Angelese Chargerse koji su prošle godine prešli iz San Diega u Los Angeles, dok velik broj država sjevera i sjeverozapada nema niti jedan tim, kao što su Nevada, Utah, Idaho, Oregon, Montana, Wyoming te mnoge druge. [4]

Savezna_država	Država_id (char 2)	Primary Key
	Naziv_drzave (varchar 45)	Not Null, Unique

Divizija se također referencira u tablici tim, te nam je potrebna kako bismo svakom timu pridružili diviziju u kojoj se nalazi i u kojoj igra, te kako bismo u aplikaciji za bazu podataka mogli napraviti ispis rezultata po diviziji za svaki tim, odnosno tablicu. Tako se ta tablica sastoji od atributa naziv te broj_timova. Naziv je glavni ključ divizije te je tipa varchar veličine četrdeset pet gdje upisujemo naziv divizije. Broj_timova je drugi atribut koji govori koliko svaka divizija broj timova, te se u trenutku stvaranja baze podataka u svakoj diviziji nalazi četiri tima, ali se taj broj može mijenjati s obzirom na rast i širenje lige.

Divizija	Naziv (varchar 45)	Primary Key
	Broj_timova (int 10)	Not Null, Unsigned

Za svaku utakmicu potrebni su suci, kojih na terenu treba biti više od jednog, točnije njih sedmero u svakom trenutku. Pozicionirani su oko svih igrača kako bi pružili što bolji uvid u igru i kako bi pokrili svaki dio terena i vidjeli sve prekršaje koji se događaju u trenucima igre. Tablica suci sastoji se od atributa suci_id, integera koji se automatski inkrementira, imena, prezimena i pozicije na kojoj se nalaze prilikom suđenja na terenu. Pozicije se označavaju kraticama s jednim ili dva znaka, stoga je taj atribut definiran kao varchar 2.

Suci	Suci_id (int 10)	Primary Key, Auto Increment
	Ime (varchar 45)	Not Null
	Prezime (varchar 45)	Not Null
	Pozicija (varchar 2)	Not Null

Sljedeća tablica koju je potrebno opisati je susreti. U tu tablicu se upisuju susreti za svaku odigranu utakmicu u određenoj sezoni. Za svaku utakmicu potrebno je da postoji jedinstvena identifikacija tog susreta, stoga trebamo imati atribut `susreti_id`, koji ujedno služi kao primarni ključ te tablice, zatim `domaci_tim` te `gostujuci_tim`, `bodovi_domacina`, `bodovi_gosta` te `suci_id`. `Domaci_tim` i `gostujuci_tim` su atributi koji se referenciraju na istu tablicu, tim, iz kojih se izvlači `tim_id` koji se upisuje u svako polje. Za jednostavnost upisa tima u aplikaciji za bazu podataka napravljeno je da korisnik bira u padajućem izborniku odgovarajući tim iz liste timova te se to u pozadini frontenda pretvara u `tim_id` kako bi se mogao upisati u polja `domaci_tim` po odgovarajućem timu i u polje `gostujuci_tim`. `Bodovi_domacina` i `bodovi_gosta` su polja tipa integer u koja se upisuju bodovi svakog tima odigrane utakmice, ta polja moraju biti popunjena i ne smije vrijednost u njima biti negativna. `Suci_id` je referenca na tablicu `suci` u kojoj se nalaze suci koji su na utakmici. Iako za svaku utakmicu upisujemo samo jednog suca, u pravilu ih treba biti upisano sedmero po susretu, ali zbog pojednostavljivanja same baze podataka upisujemo samo jednog po susretu i to u tablicu `susreti`, jer je veza jedan naprema više pa upisujemo na stranu više.

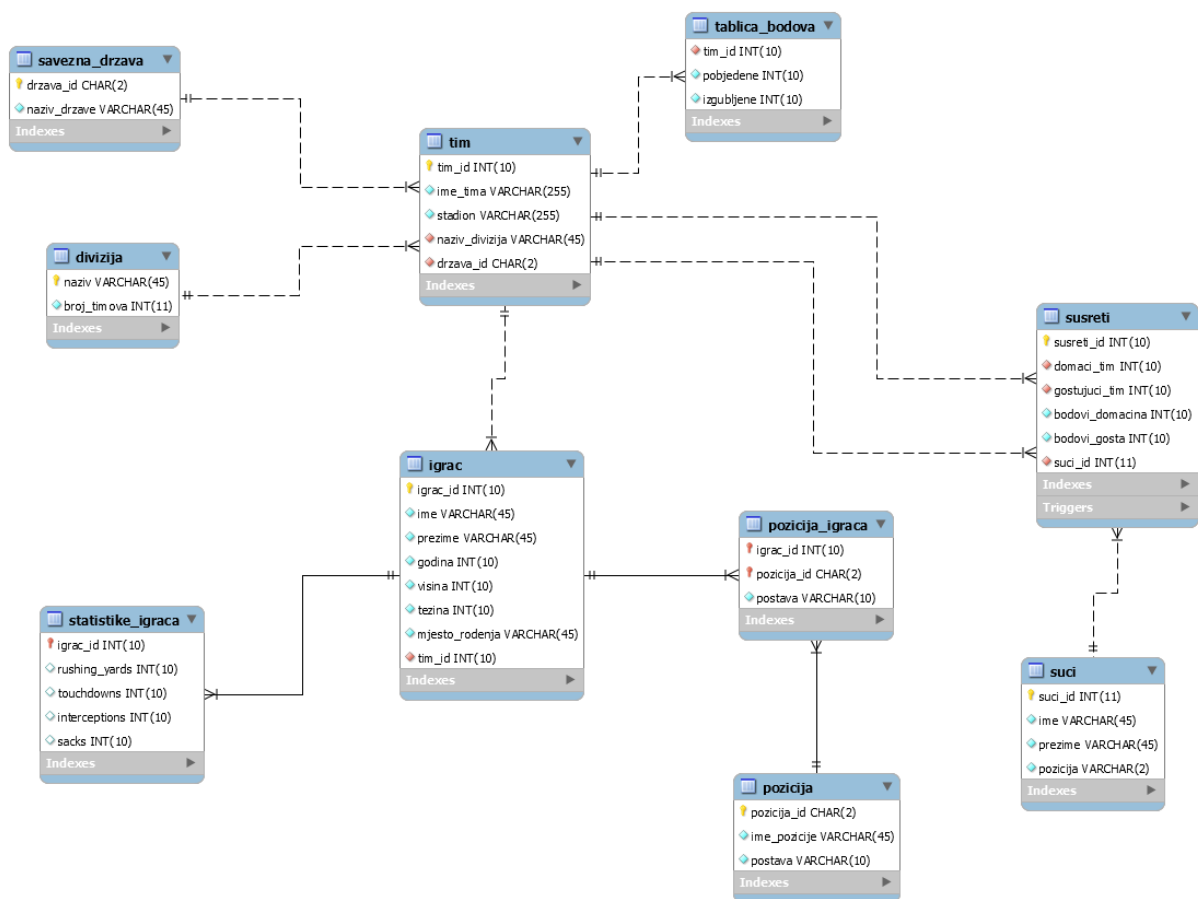
Susreti	Susreti_id (int 10)	Primary Key, Unsigned, Auto Increment
	Domaci_tim (int 10)	Foreign Key (Tim), Unsigned
	Gostujuci_tim (int 10)	Foreign Key (Tim), Unsigned
	Bodovi_domacina (int 10)	Not Null, Unsigned
	Bodovi_gosta (int 10)	Not Null, Unsigned
	Suci_id (int 10)	Not Null, Unsigned

Tablica bodova je zadnja, deseta tablica u bazi podataka, u koju se automatski upisuju podaci na temelju odigranih susreta. Prethodno u tablici se nalaze svi timovi na temelju tim_id, zbog čega je tablica povezana s tablicom tim, te atributi pobjedene i izgubljene, koji označavaju koliko utakmica je svaki tim pobijedio, odnosno izgubio. Ova tablica je zanimljiva po tome što se svi podaci upisuju na temelju napravljenih okidača, nakon svakog unosa u tablicu susreti. Svi okidači biti će objašnjeni u sljedećem poglavlju.

Tablica_bodova	Tim_id (int 10)	Foreign Key (Tim), Not Null, Unsigned
	Pobjedene (int 10)	Not Null
	Izgubljene (int 10)	Not Null

3.2. ERA model

Nakon kreiranja baze podataka, moguće je generirati ERA model, u kojemu je potrebno samo premjestiti određene tablice radi bolje čitljivosti. ERA model sadržava jedan broj tablica kao i baza podataka, njih deset, te prikazuje kardinalosti veza između pojedinih tablica. Tako na slici ispod možemo vidjeti da je tablica tim povezana s najviše drugih tablica, zbog čega je i smatramo okosnicom baze podataka. Sa svim tablicama je povezana vezama jedan naprema više, gdje se za tablicu savezna_država te divizija strane više nalaze na tablici tim, a kod tablica igrač, susreti i tablica bodova se vanjski ključ nalazi u tim tablicama. Iako ima šest veza, povezana je s pet tablica. Razlog tomu je što se dvaput povezujemo na tablicu susreti s istim vezama zbog toga što trebamo imati vanjski ključ za gostujući i za domaći tim. Povezivanjem tablice tim s tablicom igrač dobivamo u tablici igrač tim_id svakog igrača. što u aplikaciji je realizirano na način da se odmah bira ime tima, radi jednostavnosti za korisnike. Tablica igrač povezana je s tablicom statistike_igrača vezom jedan naprema više gdje je vanjski ključ u tablici statistike_igrača. Također, tablica igrač povezana je s tablicom pozicija_igrača, što je tablica nastala kao potreba zbog postojanja tablice pozicije. Naime, svaki igrač može igrati više pozicija, a na određenim pozicijama, kao što su obrambene pozicije, mora se na utakmici nalaziti više igrača, zbog čega je to veza više naprema više. Na tablici susreti povezana je tablica suci, u kojoj se nalaze svi suci te za svaku utakmicu je potrebno izabrati jednog suca kako bi bila regularna, zbog čega je i tu veza jedan naprema više gdje se vanjski ključ nalazi na tablici susreti.



Slika 3. ERA model

4. Okidači

Kao što je rečeno u prošlom poglavlju, na tablici susreti su definirani okidači, i to njih 5, kako bi se omogućio pravilan upis u tablicu bodova, gdje se nalaze sve pobjeđene i izgubljene utakmice svih upisanih momčadi. Tako postoji okidač koji osigurava da se prilikom svakog upisa u tablicu susreti jasno definira pobjednik, odnosno netko ima više bodova od dva tima. Ukoliko timovi imaju isti broj bodova, tada se javlja poruka upozorenja, kojom se prikazuje kako se susret ne može unijeti ako su bodovi_domacina i bodovi_gosta jednaki. Naravno, taj okidač se događa prije svakog upisa (before insert) kako se u tablicu ne bi unijeli neželjeni podaci.

```
1 CREATE DEFINER='root'@'localhost' TRIGGER provjeriRezultateTrigger
2 BEFORE INSERT ON susreti
3 FOR EACH ROW
4 BEGIN
5     IF NEW.bodovi_domacina = NEW.bodovi_gosta THEN
6         SIGNAL SQLSTATE '45000'
7         SET MESSAGE_TEXT = 'Timovi ne mogu igrati nerijeseno!';
8     END IF;
9 END
```

Slika 4. provjeriRezultateTrigger

Sljedeći okidači možemo reći da dolaze u paru i događaju se poslije svakog unosa susreta (after insert), te se zovu domacinTrig i gostTrig. Okidači djeluju tako da prilikom unosa u tablicu susreti gdje se za svaku pobjeđenu ili izgubljenu utakmicu tablica tablica_bodova se osvježava s novim rezultatima u prikladnim poljima, odnosno povećava se za jedan polje izgubljene ili polje pobjeđene za dobivenu, odnosno izgubljenu utakmicu.

```
CREATE DEFINER='root'@'localhost' TRIGGER domacinTrig
AFTER INSERT ON susreti
FOR EACH ROW
BEGIN
    IF NEW.bodovi_domacina > NEW.bodovi_gosta
    THEN UPDATE tablica_bodova SET pobjedene = pobjedene+1 WHERE NEW.domaci_tim = tablica_bodova.tim_id;
    ELSE
    UPDATE tablica_bodova SET izgubljene = izgubljene+1 WHERE NEW.domaci_tim = tablica_bodova.tim_id;
    END IF;
END
```

Slika 5. domacinTrigger upis susreta


```

CREATE DEFINER='root'@'localhost' TRIGGER gostTrig
AFTER INSERT ON susreti
FOR EACH ROW
BEGIN
    IF NEW.bodovi_domacina > NEW.bodovi_gosta
    THEN UPDATE tablica_bodova SET izgubljene = izgubljene+1 WHERE NEW.gostujuci_tim = tablica_bodova.tim_id;
    ELSE
    UPDATE tablica_bodova SET pobjedene = pobjedene+1 WHERE NEW.gostujuci_tim = tablica_bodova.tim_id;
    END IF;
END

```

Slika 6. gostTrigger upis susreta

Sljedeći okidač potreban je ukoliko smo upisali krivi rezultat za susret te ne želimo brisati cijeli susret nego samo urediti rezultate, odnosno izmijeniti bodove_domacina i bodove_gosta. U tom slučaju potreban je jedan kompliciraniji okidač koji prilikom svakog ažuriranja podataka ažurira i podatke u tablici tablica_bodova. Taj okidač djeluje na način da se prvotno u jednom grananju provjeravaju stari podaci, odnosno već uneseni bodovi_domacina i bodovi_gosta. Nakon što je program odabrao točnu vrijednost, u njoj se provjerava jesu li novo unesene vrijednosti obrnute, odnosno ako su prije domaćini imali više bodova imaju li ih sada više gosti i obrnuto. Ukoliko je to slučaj, tada se događa inkrementiranje polja pobjeđenih te smanjivanje polja izgubljenih utakmica za jedan te obrnuto kod drugog tima. Ovim okidačem dobivamo stanje u kojemu ukoliko smo obrnuli slučajno bodove prilikom upisa u te tablice da ispravkom njihovih vrijednosti tablica tablica_bodova i dalje ostaje relevantna i prikazuje točne podatke o pobjeđenim i izgubljenim utakmicama pojedinih momčadi. Veliki broj grananja u ovom okidaču je zbog želje stvaranje što manjeg broja okidača, ali i zbog problema nastalih prilikom stvaranja dva okidača koji djeluju prije upisivanja u tablicu, što u ovom projektu nije bilo moguće implementirati.

```

CREATE DEFINER='root'@'localhost' TRIGGER Trig_update_susreta
BEFORE UPDATE ON susreti
FOR EACH ROW
BEGIN
    IF OLD.bodovi_gosta > OLD.bodovi_domacina
    THEN
        IF NEW.bodovi_domacina > NEW.bodovi_gosta
        THEN UPDATE tablica_bodova SET izgubljene = izgubljene+1, pobjedene = pobjedene-1 WHERE NEW.gostujuci_tim = tablica_bodova.tim_id;
        END IF;
    ELSE IF OLD.bodovi_gosta > OLD.bodovi_domacina
    THEN
        IF NEW.bodovi_domacina > NEW.bodovi_gosta
        THEN UPDATE tablica_bodova SET izgubljene = izgubljene-1, pobjedene = pobjedene+1 WHERE NEW.domaci_tim = tablica_bodova.tim_id;
        END IF;
    ELSE IF OLD.bodovi_domacina > OLD.bodovi_gosta
    THEN
        IF NEW.bodovi_gosta > NEW.bodovi_domacina
        THEN UPDATE tablica_bodova SET izgubljene = izgubljene-1, pobjedene = pobjedene+1 WHERE NEW.gostujuci_tim = tablica_bodova.tim_id;
        END IF;
    ELSE IF OLD.bodovi_domacina > OLD.bodovi_gosta
    THEN
        IF NEW.bodovi_gosta > NEW.bodovi_domacina
        THEN UPDATE tablica_bodova SET izgubljene = izgubljene+1, pobjedene = pobjedene-1 WHERE NEW.domaci_tim = tablica_bodova.tim_id;
        END IF;
    END IF;
END

```

Slika 7. updateSusreta trigger

Peti i zadnji okidač u cijelom projektu, ali i na tablici susreti je okidač brisanja susreta iz tablice, koji se točno mora na odgovarajuće načine prikazati i u tablici tablica_bodova. Taj okidač djeluje u skladu sa svojim imenom, prije samog brisanja (before delete) podataka iz tablice susreti, prvo se obrišu odgovarajući podaci u tablici tablica_susreta na odgovarajućim mjestima, odnosno na polju pobjeđene se broj pobjeda smanji za jedan kod prikladnog tima, te se na polju izgubljene broj također smanji za jedan kod prikladnog tima. Svi ovi opisani okidači zajedno uvjetuju savršeno funkcioniranje tablice tablica_bodova u koju se podaci upisuju automatski.

```
CREATE DEFINER='root'@'localhost' TRIGGER Trig_brisanje_susreta
BEFORE DELETE ON susreti
FOR EACH ROW
BEGIN
    IF OLD.bodovi_domacina > OLD.bodovi_gosta
    THEN UPDATE tablica_bodova SET pobjedene = pobjedene-1 WHERE OLD.domaci_tim = tablica_bodova.tim_id;
    ELSE
    UPDATE tablica_bodova SET izgubljene = izgubljene-1 WHERE OLD.domaci_tim = tablica_bodova.tim_id;
    END IF;

    IF OLD.bodovi_domacina < OLD.bodovi_gosta
    THEN UPDATE tablica_bodova SET izgubljene = izgubljene-1 WHERE OLD.gostujuci_tim = tablica_bodova.tim_id;
    ELSE
    UPDATE tablica_bodova SET pobjedene = pobjedene-1 WHERE OLD.gostujuci_tim = tablica_bodova.tim_id;
    END IF;
END
```

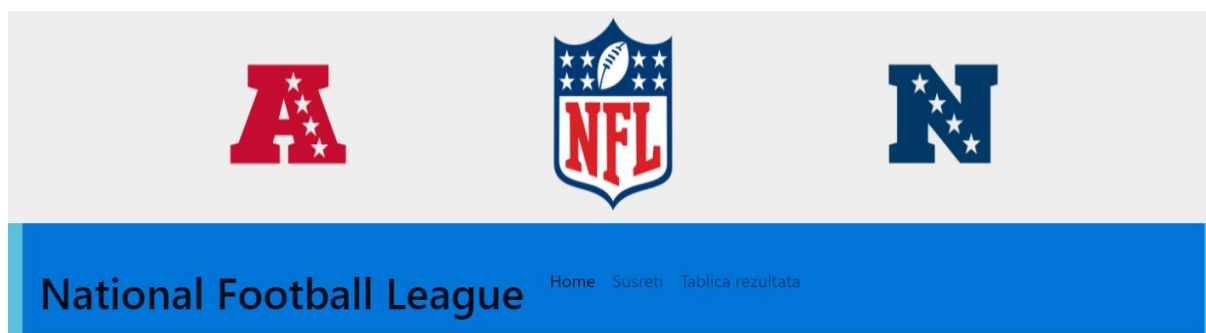
Slika 8. brisanjeSusreta trigger

5. Web aplikacija

Aplikacija za napravljenu bazu podataka izrađivana je u programski jezicima HTML/CSS i PHP, koristeći za pisanje u njima Notepad++ te NetBeans. U web aplikaciji se ne nalaze sve tablice baze podataka, samo određene najvećeg značaja, stoga su za CRUD odabrane tablice igrač i susreti. Tablica igrač je odabrana iz razloga zbog toga što je jedna od bitnijih tablica u bazi podataka i omogućuje implementiranje određenih pogodnosti za korisnike kao što je combo-box za odabir tima kako bi krajnjem korisniku unos igrača bio što jednostavniji.

5.1. Navigacijska traka

Prva što se uoči prilikom otvaranja web aplikacije je jednostavna navigacijska traka koja na vrhu ima sliku glavnih divizija i logo Nacionalne nogometne lige. Ispod se nalazi naziv „National Football League“, odnosno skraćeno NFL, engleskog naziva za ligu. Desno od naziva nalaze se hyperlinkovi Home, Susreti te Tablica rezultata. Hyperlink Home je stranica koja se otvori prilikom otvaranja same web aplikacije, odnosno početna stranica, za koju sam odabrao unesi igrača te ispis svih igrača, što će kasnije biti objašnjeno. Na slikama iznad nalazi se izgled navigacijske trake te kod kojim se ona dobiva.



Slika 9. Izgled navigacijske trake

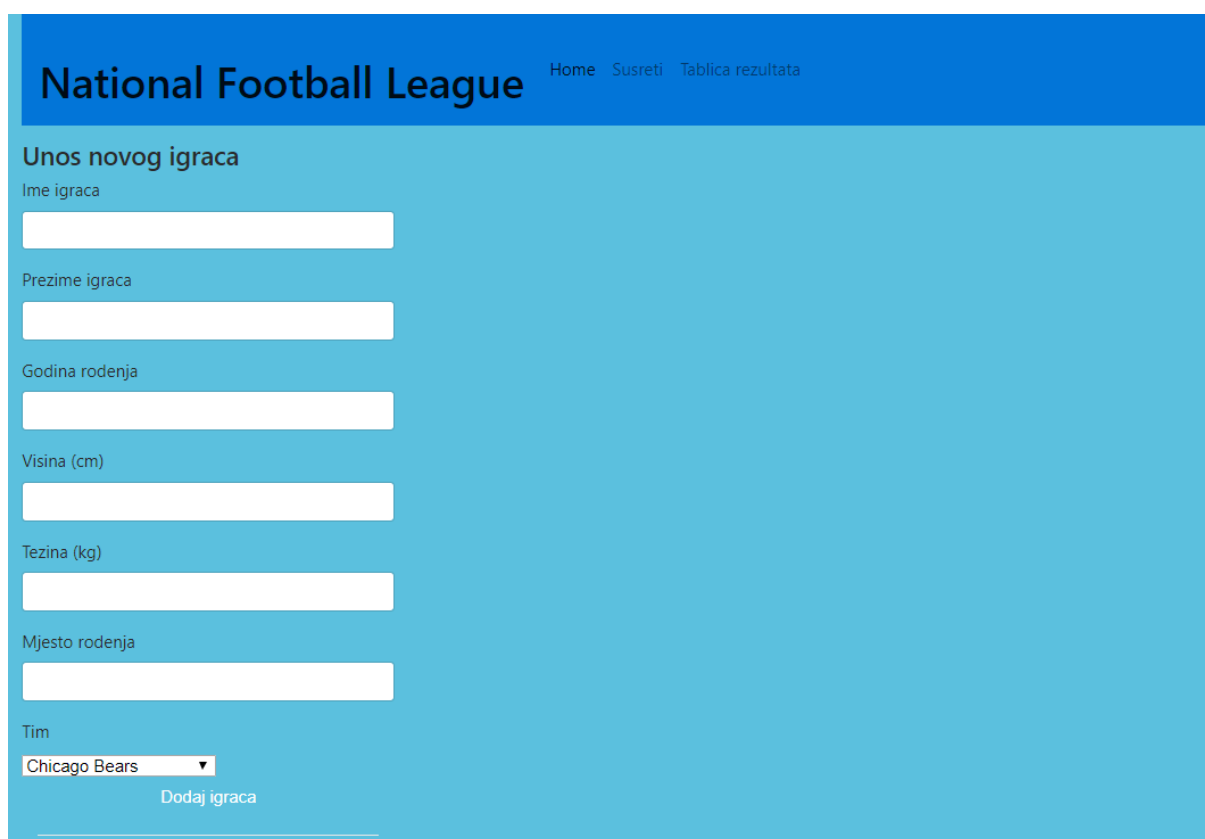
```
<div class="container bg-info">
<nav class="navbar navbar-toggleable-md navbar-light bg-primary">
<button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<a class="navbar-brand" href="#">

<h1>National Football League</h1>
</a>
<div class="collapse navbar-collapse" id="navbarNavAltMarkup">
<div class="navbar-nav">
<a class="nav-item nav-link active" href="index.php">Home <span class="sr-only">(current)</span></a>
<a class="nav-item nav-link" href="susreti.php">Susreti</a>
<a class="nav-item nav-link" href="tablica_bodova.php">Tablica rezultata</a>
</div>
</div>
</nav>
```

Slika 10. Kod navigacijske trake

5.2. Početna stranica / upis novog igrača

Prvo stranica kad se otvori web aplikacija je igrač, u kojemu imamo opciju unes novog igrača (Create komponenta skraćeno CRUD), čitanje svih upisanih igrača (Read komponenta), brisanje i uređivanje pojedinog igrača (UD komponenta). Kao što je prikazano na slici, potrebno je upisati u sva predviđena polja potrebne stvari kako bi se igrač mogao unijeti u bazu podataka. Sva polja odgovaraju poljima definiranim u bazi podataka, osim polja `tim_id`, koje je ovdje zamijenjeno s combo-boxom iz kojega korisnik koji unosi pojedine podatke izabire tim. Poslije ispunjavanja svih polja te odabirom tima, pritiskom na tipku dodaj igrača ti podaci prosljeđuju se u pozadinu, u drugu formu, gdje se odabrani tim na temelju jednostavnog upita poistovjećuje s `tim_id` koji nam je potreban za unos u bazu podataka. Kada dobijemo traženi `tim_id`, tada sve dobivene i izvučene podatke upisujemo u redak u tablicu igrač u bazu podataka, ili u suprotnom dobijemo grešku kako unos nije ispravan iz određenih razloga kao što su, netočni tip podataka ili ne uneseni podaci.



National Football League Home Susreti Tablica rezultata

Unos novog igrača

Ime igrača

Prezime igrača

Godina rođenja

Visina (cm)

Tezina (kg)

Mjesto rođenja

Tim

Slika 11. Upis novog igrača

```

1  <?php
2
3  include('konekcija.php');
4
5  $unos_ime = $_POST['igrac_ime'];
6  $unos_prezime = $_POST['igrac_prezime'];
7  $unos_godina_rodenja = $_POST['godina'];
8  $unos_visina = $_POST['visina'];
9  $unos_tezina = $_POST['tezina'];
10 $unos_mjesto_rodenja = $_POST['mjesto_rodenja'];
11 $unos_ime_tima = $_POST['tim_id'];
12
13 //Upit za pronalazak ID tima kojemu znamo ime
14
15 $query_timID = "SELECT tim_id FROM tim WHERE ime_tima = '$unos_ime_tima' limit 1";
16 $result =mysql_query($db_connect, $query_timID);
17 $unos_tim = mysql_fetch_object($result);
18 $unos_tim = $unos_tim->tim_id;
19
20
21 $query = "INSERT INTO igrac(igrac_id, ime, prezime, godina, visina, tezina, mjesto_rodenja, tim_id)
22 VALUES (null, '$unos_ime', '$unos_prezime', '$unos_godina_rodenja',
23 '$unos_visina', '$unos_tezina', '$unos_mjesto_rodenja', '$unos_tim')";
24
25 if(mysql_query($db_connect, $query)){
26     header("Location:index.php");
27 }
28 else{
29     echo "Greska prilikom unosa igraca!";
30 }
31 mysql_close($db_connect);
32
33 >?

```

Slika 12. Kod za unos novoga igrača

U nastavku stranice nalazi se ispis svih igrača, koji se direktno povlači iz baze podataka, te nakon što upišemo novoga igrača, odmah je vidljiv na dnu ispisa svih igrača. Za svakog igrača pored svih njegovih unesenih karakteristika postoje dva gumba, uredi igrača i obriši igrača. Pritiskom na gumb uredi igrača prelazimo u novi obrazac u kojemu možemo urediti sve upisane podatke o igraču, kao i prilikom upisa. Tako prilikom ispravka neke karakteristike, svi podaci se ponovno šalju u bazu podataka, bili oni ispravljeni ili ne, te se naredbom Update upisuju u bazu podataka na mjesto odgovarajuće identifikacijske oznake `igrac_id`, koje smo proslijedili još iz glavne forme preko URL-a. Potrebno je naglasiti kako se već svi upisani podaci nalaze u odgovarajućim poljima, radi lakše preglednosti i pronalaženja pogreške. Pritiskom na gumb obriši igrača briše se taj igrač iz baze podataka, što je vidljivo odmah nakon pritiska na taj gumb jer se stranica automatski ažurira te više taj igrač se ne nalazi na listi svih igrača koja se vuče iz baze podataka.

Ispis svih igrača								
ID	Ime	Prezime	Godina rođenja	Visina(cm)	Tezina(kg)	Mjesto rođenja	Tim	Opcije
1	Aaron	Rodgers	1982	185	90	Chico, California	Green Bay Packers	Uredi igrača Obrisi igrača
2	Tom	Brady	1977	193	87	San Mateo, California	New England Patriots	Uredi igrača Obrisi igrača
3	Odell	Beckham	1993	180	90	Baton Rouge, Louisiana	New York Giants	Uredi igrača Obrisi igrača
4	Cam	Newton	1989	195	97	Atlanta, Georgia	Miami Dolphins	Uredi igrača Obrisi igrača
5	J.J.	Watt	1989	196	98	Waukesha, Wisconsin	Houston Texans	Uredi igrača Obrisi igrača
6	Antonio	Brown	1988	185	90	Miami, Florida	Pittsburgh Steelers	Uredi igrača Obrisi igrača
7	Adrian	Peterson	1985	185	90	Palestine, Minnesota	New York Giants	Uredi igrača Obrisi igrača
8	Vonnie	Miller	1989	191	93	Dallas, Texas	Green Bay Packers	Uredi igrača Obrisi igrača

Slika 13. Ispis svih igrača

Uredi igrača

Ime igrača

Prezime igrača

Godina rođenja

Visina(cm)

Tezina(kg)

Mjesto rođenja

Tim

Green Bay Packers ▾

[Uredi igrača](#)

Slika 14. Uredi igrača

```

1  <?php
2
3  $obrisi_igrac_id = $_GET['igrac_id'];
4
5  include('konekcija.php');
6
7
8  $upit_brisanje = "DELETE FROM igrac WHERE igrac_id='$obrisi_igrac_id'";
9
10 if(mysqli_query($db_connect, $upit_brisanje))
11 {
12     header("Location:index.php");
13 }
14 else
15 {
16     echo "Neuspjelo brisanje";
17 }
18
19 mysqli_close($db_connect);
20
21 ?>

```

Slika 15. Kod za brisanje igrača

5.3. Upis susreta

Druga tablica odabrana za CRUD operacije je susreti, te je napravljena u potpunosti na web aplikaciji. U duhu prošlog combo-boxa za odabir tima, i ovdje se putem istoga načina odabiru domaćin i gost povučeni iz baze podataka iz tablice tim te čak i sudac povučen iz tablice suci, radi pojednostavljivanja korištenja web aplikacije i same baze podataka za korisnike. Samo upisivanje koje ostaje za korisnika je bodovi_domaćina i bodovi_gosta, koji ne smiju biti isti, odnosno ne smiju se upisati isti iznosi u oba polja što bi označavalo neriješen rezultat, te bi se prikazalo kao neuspješno upisivanje u bazu podataka. Kod napisan u PHP, HTML/CSS programskim jezicima za unos susreta je kompliciraniji od napravljenog koda za unos igrača jer zahtjeva više upita u bazu podataka za pronalazak potrebnih identifikacijskih oznaka za polja odabrana u combo-boxovima. Tako je potrebno napraviti tri upita za bazu podataka u kojima dobivamo dvaput tim_id, jednom za domaćina, jednom za gosta te suci_id za glavnog suca kojeg odaberemo. Tek nakon dobivenih svih potrebnih podataka možemo upisati u redak susreti sve vrijednosti.

```

1 <?php
2
3 include('konekcija.php');
4
5
6 $odabrani_domaci_tim = $_POST['ime_domaceg_tima'];
7 $bodovi_domaceg_tima = $_POST['bodovi_domacina'];
8 $odabrani_gostujuci_tim = $_POST['ime_gostujuceg_tima'];
9 $bodovi_gostujuceg_tima = $_POST['bodovi_gosta'];
10 $prezime_glavni_sudac = $_POST['odabir_suca'];
11
12
13 //Upit za pronalazak ID tima kojemu znamo ime (domacin)
14
15 $query_domaci_timID = "SELECT tim_id FROM tim WHERE ime_tima = '$odabrani_domaci_tim' ";
16 $rezultat_domacin = mysqli_query($db_connect, $query_domaci_timID);
17 $unos_domaci_tim = mysqli_fetch_object($rezultat_domacin);
18 $unos_domaci_tim = $unos_domaci_tim->tim_id;
19
20 //Upit za pronalazak ID tima kojemu znamo ime (gost)
21
22 $query_gostujuci_timID = "SELECT tim_id FROM tim WHERE ime_tima = '$odabrani_gostujuci_tim' ";
23 $rezultat_gost = mysqli_query($db_connect, $query_gostujuci_timID);
24 $unos_gostujuci_tim = mysqli_fetch_object($rezultat_gost);
25 $unos_gostujuci_tim = $unos_gostujuci_tim->tim_id;
26
27
28 //Upit za pronalazak ID suca kojemu znamo ime
29
30 $query_glavni_sudac = "SELECT suci_id FROM suci WHERE prezime= '$prezime_glavni_sudac' limit 1";
31 $rezultat_sudac = mysqli_query($db_connect, $query_glavni_sudac);
32 $unos_glavni_sudac = mysqli_fetch_object($rezultat_sudac);
33 $unos_glavni_sudac = $unos_glavni_sudac->suci_id;
34
35
36 $query = "INSERT INTO susreti(susreti_id, domaci_tim, gostujuci_tim, bodovi_domacina, bodovi_gosta, suci_id) values(null, '$unos_domaci_tim', '$unos_gostujuci_tim',
37     '$bodovi_domaceg_tima', '$bodovi_gostujuceg_tima', '$unos_glavni_sudac')";
38
39 if(mysqli_query($db_connect, $query)){
40     header("Location:susreti.php");
41 }
42 else{
43     echo "Greska prilikom unosa susreta!";
44 }
45 mysqli_close($db_connect);
46
47 >

```

Slika 16. Kod za unos susreta

The screenshot shows a web interface for the National Football League. At the top, there's a navigation bar with 'Home', 'Susreti', and 'Tablica rezultata'. The main content area has a light blue background. It contains three sections: 'Domaci tim' (Home team) with a dropdown menu showing 'Chicago Bears' and a 'Bodovi domaceg tima' (Home team points) input field; 'Gostujuci tim' (Away team) with a dropdown menu showing 'Chicago Bears' and a 'Bodovi gostujuceg tima' (Away team points) input field; and 'Glavni sudac' (Referee) with a dropdown menu showing 'Allen'. At the bottom, there's a 'Unesite susret' (Add match) button.

Slika 17. Unos susreta

Ispod toga, u nastavku te stranice imamo ispis svih susreta, pored kojih e, kao i kod igrača, nalaze dva gumba, jedan za uređivanje susreta, jedan za brisanje susreta po potrebi. Forma za uređivanje susreta je drukčija od forme za uređivanje igrača jer određena polja nisu moguća za izmjenu, nego je potrebno obrisati susret te upisati novi. Polja koja se mogu mijenjati su bodovi_domaćina te bodovi_gosta, dok su ostala osjenčana te nisu u mogućnosti

za izmjenu. Nakon uređenih bodova potrebno je stisnuti na gumb uredi susret, nakon čega se u pozadini događa opet update susreta, ali samo u poljima bodovi_domaćina i bodovi_gosta, što je drukčije nego u prvom slučaju kada su se sve stavke mogle urediti kod tablice igrač, nakon čega se događa aktiviranje okidača koji su definirani prethodno u tekstu.

Ispis svih susreta						
ID susreta	Domaci tim	Bodovi domacina	Bodovi gosta	Gostujući tim	Glavni sudac	
9	Miami Dolphins	30	35	New York Giants	Barry Poe	<div>Uredi susret</div> <div>Obrisi susret</div>
13	Oakland Raiders	21	17	New York Giants	Brad Allen	<div>Uredi susret</div> <div>Obrisi susret</div>

Slika 18. Ispis susreta

Uredi susret

Domaca momcad

New England Patriots

Gostujuca momcad

Chicago Bears

Domaci tim

13

Gostujući tim

17

Suci id

Allen

Uredi susret

Slika 19. Uredi susret

Prilikom klika na gumb obriši susret, u pozadini se obriše susret na temelju prosljeđene identifikacijske oznake susreta, slično kao i kod uređivanja susreta, samo što se tamo šalje i identifikacijska oznaka domaćina, gosta te suca kako bi se polja mogla ispuniti te kako bi korisnik web aplikacije mogao vidjeti koje se vrijednost nalaze u poljima.

```

1  <?php
2
3  $obrisi_susreti_id = $_GET['susreti_id'];
4
5  include('konekcija.php');
6
7
8  $upit_brisanje_susreta = "DELETE FROM susreti WHERE susreti_id='$obrisi_susreti_id'";
9
10 if(mysqli_query($db_connect, $upit_brisanje_susreta))
11 {
12     header("Location:susreti.php");
13 }
14 else
15 {
16     echo "Neuspjelo brisanje";
17 }
18
19 mysqli_close($db_connect);
20
21 ?>

```

Slika 20. Kod za brisanje susreta

5.4. Tablica rezultata

Osim dvaju CRUD tablica, igrač te susret, u web aplikaciji postoji tablica rezultata, koja se generira iz podataka unesenih u bazu podataka te prikazuje tablicu rezultata ukupne lige, ili pritiskom na dugme divizija, koje se nalazi pored svakog tima, može se prikazati tablica divizije za svaki tim.

National Football League			
		Home	Susreti
		Tablica rezultata	
Ispis tablice rezultata			
Ime tima	Pobjedene	Izgubljene	
Philadelphia Eagles	2	0	Divizija
Chicago Bears	1	0	Divizija
New York Giants	1	1	Divizija
Seattle Seahawks	0	0	Divizija
Green Bay Packers	0	0	Divizija
Houston Texans	0	0	Divizija

Slika 21. Tablica rezultata

National Football League		
Home Susreti Tablica rezultata		
NFC East		
Ime tima	Pobjedene	Izgubljene
Philadelphia Eagles	2	0
New York Giants	1	1
Dallas Cowboys	0	1

Slika 22. Tablica divizije

5.5. Upiti

U tablici `tablica_bodova` definiranoj u bazi podataka nalazi se koliko je svaki tim pobijedio i izgubio utakmica upisanih u tablici `susreta`. Zbog toga što se u bazi podataka nalaze samo brojevi koji označavaju identifikacijske oznake, potrebno ih je prilikom prikaza povezati na određeni način, upitima, kako bi se krajnjem korisniku prikazale potpune informacije, kao što su imena sudaca, imena momčadi i slično. Zbog toga je u web aplikaciji napravljeno nekoliko složenih upita koji će biti objašnjeni u sljedećem odlomku.

```

14 <?php
15 include('konekcija.php');
16
17 //stvaranje upita
18
19 $upit_ispis_tablice = "SELECT t.tim_id, t.ime_tima, b.pobjedene, b.izgubljene from tim as t join tablica_bodova as b on t.tim_id=b.tim_id order by b.pobjedene desc, b.izgubljene asc";
20
21 $result = mysqli_query($db_connect, $upit_ispis_tablice);
22
23
24
25

```

Slika 23. Upit tablice rezultata

Ovaj upit nalazimo u HTML kodu kojim definiramo što želimo izvlačiti iz baze podataka. Tako ovaj upit govori kako želimo dobiti identifikacijsku oznaku tima, puno ime tima te pobijedene i izgubljene utakmice, i to sve da se prezentira na način tako da se prvo na vrhu prikazuju sve momčadi koje su što više utakmica pobijedile, a na dnu sve momčadi koje su utakmice izgubile. Zbog toga što uzimamo podatke iz dviju tablica, potrebno ih je na određeni način preko zajedničkog ključa povezati, što radimo putem spajanja preko identifikacijske oznake tima.

Sljedeći upit je potreban prilikom ispisa tablice divizije. Svaki tim pripada jednoj diviziji u ligi zbog čega je moguće ispisati i tablicu timova za pojedine divizije što je i implementirano u web aplikaciji. Zbog toga je potrebno ovaj prvotni upit malo proširiti, s još jednim poljem, a to je divizija_id. Prije svega je potrebno na temelju tim_id identifikacijske oznake izvući ime divizije iz tablice tima, nakon čega možemo ju proslijediti u upitu bazi podataka.

```
14 <?php
15
16 $trazeni_tim_id = $_GET['tim_id'];
17
18 include('konekcija.php');
19
20 //stvaranje upita za pronalazak trazene divizije
21
22 $result_divizija = mysqli_query($db_connect, "SELECT naziv_divizija FROM tim WHERE tim_id='$trazeni_tim_id'");
23 $result_trazena_divizija = mysqli_fetch_object($result_divizija);
24 $result_trazena_divizija = $result_trazena_divizija->naziv_divizija;
25
26 //stvaranje upita za ispis svih iz trazene divizije
27
28 $upit_ispis_divizije = "SELECT t.tim_id, t.ime_tima, b.pobjedene, b.izgubljene from tim as t join tablica_bodova as b on t.tim_id=b.tim_id where naziv_divizija='$result_trazena_divizija' order by b.pobjedene desc, b.izgubljene asc";
29
30 $result = mysqli_query($db_connect, $upit_ispis_divizije);
31
32 ->
```

Slika 24. Upit divizije

S ta dva složena upita dobivamo ispis tablica rezultata i tablica divizije za pojedine timove, koje su implementirane u web aplikaciji.

6. Zaključak

Radom na ovom projektu upoznao sam se s mnogim elementima koji će mi biti korisni u daljnjim projektima u privatne svrhe ili obrazovanju. MySQL Workbench je odličan alat za stvaranje baza podataka, koji zbog svoje jednostavnosti korištenja odlično služi početnicima u savladavanju početnih problema kod stvaranja baze podataka. XAMPP je odličan alat koji objedinjuje sve stvari potrebne za stvaranje virtualne konekcije i rada na web aplikacijama za pojedine baze podataka.

Nacionalna nogometna liga je izrazito zanimljiva tema na kojoj se može provoditi puno više vremena i koja se može raditi daleko više u dubinu, no za potrebe ovog projekta napravljene su osnovne tablice i veze između njih koje daju dovoljno jasnu sliku nogometne lige spremne za korištenje u web aplikaciji.

7. Literatura

- [1] MySQL Workbench. (2017). *MySQL.com*. Preuzeto 28. prosinca 2017. s <https://www.mysql.com/products/workbench>
- [2] XAMPP. (2017.) *Wikipedia.com* Preuzeto 28. prosinca 2017. s <https://en.wikipedia.org/wiki/XAMPP>
- [3] NetBeans (2017.) *Wikipedia.com* Preuzeto 28. prosinca 2017. s <https://en.wikipedia.org/wiki/NetBeans>
- [4] M.Maleković i K.Rabuzin (2016.) *Uvod u baze podataka*, (udžbenik Sveučilišta u Zagrebu). Varaždin: TIVA-FOI.