

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 920

**Web-aplikacija za praćenje
podataka o nogometnim
utakmicama.**

Marko Jerkić

Zagreb, svibanj 2023.

SADRŽAJ

1. Uvod	1
2. Analiza postojećih rješenja na internetu	2
2.1. Prikaz liste utakmica	2
2.2. Prikaz detalja utakmice	3
2.3. Prikaz detalja igrača, izbornika, ekipa i natjecanja	3
3. Relacijski model baze podataka	5
4. Odabrane tehnologije i arhitektura	7
4.1. Backend i frontend tehnologija	7
4.2. Poslužitelj	8
5. Izrada aplikacije	9
6. Zaključak	10
Literatura	11

1. Uvod

Uvod rada. Nakon uvoda dolaze poglavlja u kojima se obrađuje tema. Test uvod.

2. Analiza postojećih rješenja na internetu

Za analizu postojećih rješenja na internetu su korištene aplikacije Sofascore¹, OneFootball² i Rezultati.com³.

2.1. Prikaz liste utakmica

Jedan od glavnih dijelova aplikacija koje se koriste za praćenje nogometnih rezultata je stranica pregleda popisa dostupnih utakmica. Većina postojećih rješenja na internetu koriste ovu stranicu kao središnju lokaciju s kojih se pregleda većina sadržaja. To je istina za aplikacije Sofascore i Rezultati.com.

U slučaju te dvije aplikacije, prilikom pristizanja na samu web-lokaciju, prikazana nam je lista postojećih utakmica koje se igraju na dan posjećivanja aplikacije.

Središnja komponenta početne stranice Sofascore-a je lista utakmica grupirana po natjecanjima i državama. Prvih nekoliko prikazanih natjecanja su istaknuta ili spremljena natjecanja. U slučaju kada korisnik prvi put posjećuje stranicu ili nema spremljenih podataka u pregledniku, istaknuta natjecanja uključuju velika svjetska natjecanja, kao engleska Premier liga, njemačka Bundesliga i slično te velika natjecanja u državi iz koje korisnik posjećuje stranicu. Za korisnika koji posjećuje stranicu iz Hrvatske, među istaknutim natjecanjima se nalazi Hrvatska nogometna liga. Nakon istaknutih natjecanja, prikazuju se natjecanja iz cijelog svijeta koja sadrže utakmice za odabrani dan. Natjecanja su grupirana po državama u kojima se održava to natjecanje.

Osim popisa utakmica, koji se nalazi u sredini aplikacije, Sofascore sadrži kalendar s lijeve strane popisa utakmica. Kalendar se koristi za odabir datuma prema kojemu se prikazuju dostupne utakmice. Ispod kalendara se nalazi popis istaknutih natjecanja

¹<https://sofascore.com>

²<https://onefootball.com>

³<https://rezultati.com>

i popis svih dostupnih natjecanja s poljem za unos teksta koji se koristi za sužavanje izbora.

S desne strane popisa utakmica se nalazi pretpregled istaknute utakmice. Pretpregled sadrži ime i grb ekipa koje su sudjelovale u utakmici te rezultat utakmice. Ispod toga se nalazi popis istaknutih igrača te reklame za sportsko klađenje.

Aplikacija Rezultati.com ima dosta sličan raspored na početnoj stranici, ali ne prikazuje istaknuta natjecanja, utakmice niti igrače. Aplikacija OneFootball ne koristi popis dostupnih utakmica na početnoj stranici. OneFootball je aplikacija koja se fokusira prvenstveno na nogometne novosti. Do stranice na kojoj se prikazuje popis dostupnih utakmica se dolazi preko poveznice koja se nalazi u navigacijskoj traci.

2.2. Prikaz detalja utakmice

Sve tri aplikacije koje su korištene za analizu postojećih rješenja imaju dosta različito ponašanje prilikom odabira utakmice s popisa. Prilikom odabira utakmice na aplikaciji Sofascore otvori se pretpregled desno od popisa utakmica. Aplikacija Rezultati.com otvori detalje utakmice u skočnom prozoru, dok OneFootball otvori novu stranicu.

Zaglavlje stranice detalja utakmice prikazuje ime i grb ekipa koje su sudjelovale u utakmici, datum utakmice i trenutni rezultat. Detaljni podatci su raspodijeljeni u više kategorija koje se odabiru preko kartica. Zadana kartica sadrži događaje utakmice, kao što su pogodci, kartoni i zamjene, kronološki poredane po minuti.

Pored kartice događaja po minuti, bitne su još kartica početnih postava dviju ekipa te kartica statistike utakmice. Kartica početnih postava utakmice grafički prikazuje prvih jedanaest igrača obje ekipe, formaciju, boju dresova te broj na dresovima. Kartica statistike prikazuje podatke o posjedu lopte, broju udaraca, broju udaraca u vrata, broju udaraca iz kuta i sl., za obje ekipe.

2.3. Prikaz detalja igrača, izbornika, ekipa i natjecanja

Nakon stranica prikaza popisa utakmica i detalja pojedinačne utakmice, sporednu ulogu u odabranim aplikacijama zauzimaju stranice pregleda detalja igrača, izbornika, ekipa i natjecanja.

Stranice pregleda detalja igrača i izbornika su dosta slične. Obje prikazuju ime osobe, sliku, datum rođenja i nacionalnost. Ispod toga se prikazuju kartice s popisom najnovijih utakmica u kojima je sudjelovala ta osoba, osobna statistika te popis ekipa

za koje je ta osoba igrala ili koje je ta osoba trenirala.

Stranica prikaza detalja o ekipi prikazuje ime ekipe, grb i državu iz koje dolazi. Nakon toga se prikazuju kartice s podacima o nedavnim utakmicama koje je igrala ekipa, trenutnim igračima koji igraju za tu ekipu, te tablica poretka natjecanja u kojemu ta ekipa sudjeluje.

Kod stranice pregleda natjecanja, najbitnije su kartice s tablicom poretka te popisom utakmica po sezoni za odabrano natjecanje.

3. Relacijski model baze podataka

Dizajnirani relacijski model podataka sadrži 19 tablica te 6 enumeracijski vrsti.

Središnji entitet relacijskog modela je tablica "Game," koja sadrži osnovne podatke o utakmici. Ovaj entitet sadrži veze na natjecanje kojemu pripada utakmica, domaću i gostujuću ekipu, izbornika domaće i gostujuće ekipe, sezonu u kojoj se igra utakmica te na tablicu koja sadrži detaljniju statistiku utakmice. Tablica "Game" sadrži foreign key na tablicu "GameStatistic," ali taj podatak nije obavezan jer svaka utakmica ne mora imati podatke o statistici. U slučaju kada utakmica još nije počela, entitet "Game" ne sadrži podatke o statistici, pa to polje sadrži vrijednost "null."

Entitet "Game" također sadržava informaciju o vremenu početka utakmice, vremenu nadoknade u prvom i drugom poluvremenu, boju dresova domaće i gostujuće ekipe, kao i boje dresova vratara, status utakmice, te početne postavbe dviju ekipa. Status utakmice se sprema kao enumerirana vrijednost koja može poprimiti vrijednosti "NOT_STARTED," "STARTED," "HALFTIME" te "OVER".

Početne postavbe prvih jedanaest se spremaju kao "jsonb"¹ (PostgreSQL, 2021) vrsta podataka. Podatci se spremaju kao JSON polje od jedanaest vrijednosti koje predstavljaju jedanaest igrača početne postavbe. U tom JSON podatku je spremljen identifikator igrača, broj dresa koji igrač nosi na toj utakmici, broj reda u početnoj formaciji te redni broj igrača u tome redu gledajući s lijeva na desno. Odlučeno je spremati početnu postavbu kao JSON vrijednost umjesto da se stvori novi entitet koji bi imao vezu na entitet "Game" jer bi to stvorilo previše unosa u bazi podataka, točnije 22 po svakoj spremljenoj utakmici. Pored toga, spremanje tog podatka kao JSON je olakšalo način izrade administracije početne postavbe kroz klijetsku aplikaciju.

Entitet "GameStatistics" sadrži podatke kao ukupni udarci, udarci u vrata, ubačaji iz kuta, prekršaji, stvorene velike prilike, broj dodavanja, broj ubacivanja lopte u šesnaesterac, broj driblinga i broj uklizavanja za obje ekipe.

Pored nabrojanih podataka iz entiteta "GameStatistics," izdvojeni su podatci o zabilježenim pogotcima, podijeljenim kartonima te zamijenama kao posebni entiteti. Razlog

¹<https://www.postgresql.org/docs/9.5/functions-json.html>

tomu je taj što svi ovi podatci sadrže bar jednu vezu na nekog igrača. Entitet "Goal" sadrži podatak o igraču koji je zabio pogodak, o igraču koji je asistirao te vrijeme u minutama kada je zabijen pogodak. Entitet "CardAwarded" sadrži igrača koji je dobio karton, minutu u kojoj je igrač dobio karton, te vrstu kartona koji je dodijeljen. A entitet "Substitution" sadrži igrača koji je izašao s terena, igrača koji ulazi u teren te minutu u kojoj je zamjena izvršena.

Svaka utakmica pripada jednom natjecanju u određenoj sezoni. Entitet "Competition" sadrži ime, podatak o državi u kojoj se igra te podatak o tome jeli to natjecanje označeno kao istaknuto. Država u kojoj se igra natjecanje je spremljena kao poseban entitet koji sadrži naziv države te poveznicu na sličicu zastave te države. Svako natjecanje može imati više sezona, te za jednu sezonu može postojati više natjecanja. Sezona je spremljena kao vlastiti entitet i jedini podatak koji sadrži, pored identifikatora, je naziv te sezone.

Entitet "Team" sadrži podatke o nazivu ekipe, državi iz koje potječe klub te poveznicu na sličicu grba ekipe. Entitet "Team" sadrži više-na-više vezu s poveznom tablicom "CompetitionInSeason" koja sadrži vezu na jedno natjecanje i sezonu.

Entiteti "Manager" i "Player" su dosta slični. Oba sadrže ime, prezime, datum rođenja, trenutnu ekipu (koja nije obavezna) te nacionalnost. Oba imaju više-na-više vezu s poveznom tablicom "TeamInCompetition" koja sadrži podatak o ekipi koja igra u određenom natjecanju u određenoj sezoni. To znači da jedan izbornik ili igrač može pripadati više ekipa u jednoj sezoni, ali također ne mora pripadati niti jednoj ekipi za određenu sezonu.

Pored nabrojanih entiteta postoje još entiteti potrebni za autentifikaciju te spremanje najdražih igrača i ekipa korisnika.

4. Odabrane tehnologije i arhitektura

Arhitektura aplikacije se sastoji od NodeJS¹ koja obrađuje HTTP zahtjeve te poslužuje statični sadržaj (kao HTML, JavaScript i CSS datoteke) i dinamične podatke u obliku JSON-a te PostgreSQL² baze podataka.

4.1. Backend i frontend tehnologija

NodeJS aplikacija je napisana koristeći meta-framework SolidStart³. Meta-framework (D'Avanzo, 2022) je software koji je napisan da upotpuni neki postojeći frontend framework. Većim meta-framework propisuje načine na koji se radi dohvat i pisanje podataka te donosi vlastiti usmjerivač⁴ (eng. router).

SolidStart je meta-framework oko popularnog frontend framework-a SolidJS⁵. SolidJS je postao popularan u posljednjih nekoliko godina zbog svog modela reaktivnosti klijentskih JavaScript aplikacija. SolidJS je obnovio ideju signala stanja. Ryan Cariansi (Cariansi, 2023) opisuje signal kao reaktivno stanje aplikacije kojemu se pristupa preko observer obrazca⁶. Stanje se drži u jednom objektu i može se dijeliti na više mjesta. Na svakom mjestu gdje se želi pristupiti tome stanju potrebno je se pretplatiti na izvor signala. To se učini tako da se pozove funkcija signala. Svagdje gdje se pozove funkcija signala automatski se pretplati na sami izvor. Najveća prednost signala je ta što je moguće pretplatiti JavaScript DOM⁷ na signal. To znači da svaki put kada se u signal upiše nova vrijednost, izvornik pošalje obavijest svim pretplatnicima te ako je DOM među pretplatnicima, SolidJS runtime preuzme najnoviju vrijednost tog signala te izmjeni samo one čvorove HTML stabla koji su pretplaćeni na njega.

¹<https://nodejs.org/en>

²<https://www.postgresql.org/>

³<https://start.solidjs.com>

⁴<https://hr.wikipedia.org/wiki/Usmjerivač>

⁵<https://solidjs.com>

⁶https://en.wikipedia.org/wiki/Observer_pattern

⁷https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

Ovaj način ažuriranja HTML stabla je jako efikasan. U prosjeku mu treba manje vremena da ažurira HTML stablo od frameworka kao React⁸ koji koriste virtualni DOM⁹ na koji se pišu sve promjene te usporede virtualni i stvarni DOM pa se ažuriraju samo promijenjeni čvorovi. SolidJS time također rezultira manjim prevedenim JavaScript datotekama jer ne mora s njima slati cijeli virtualni DOM. Ovaj način ažuriranja HTML stabla je poznat kao "fine-grained reactivity"(Carniato, 2021).

4.2. Poslužitelj

Aplikacija se poslužuje na poslužitelju Vercel¹⁰. Aplikacije se poslužuje koristeći tvz. "bezposlužiteljsku" (eng. serverless). To znači da postoji jedan poslužitelj koji konstantno vrti aplikaciju i čeka zahtjeve. Svaki put kada dođe novi zahtjev, provjeri se postoji li uključena Lambda¹¹ instanca, a ako ne postoji onda se pokrene nova, i onda ta instanca obradi zahtjev. Ako nakon nekoliko minuta ta lambda instanca ne dobije novi zahtjev, onda se ona ugasi (McCumskey, 2022).

Baza podataka koristi poslužitelj Supabase¹² gdje je pokrenuta instanca PostgreSQL servera. Supabase se također koristi za obavijesti o promjenama podataka određenih entiteta u bazi na osnovu kojih se na klijentu ažuriraju podatci u stvarnom vremenu.

Aplikacija je poslužena pod domenom *<https://football.jerkic.dev>*.

⁸<https://react.dev>

⁹<https://legacy.reactjs.org/docs/faq-internals.html>

¹⁰<https://vercel.com>

¹¹<https://aws.amazon.com/lambda/>

¹²<https://supabase.com/>

5. Izrada aplikacije

6. Zaključak

Zaključak.

LITERATURA

Ryan Cariato. The evolution of signals in javascript, 2023. URL <https://dev.to/this-is-learning/the-evolution-of-signals-in-javascript-8ob>.

Ryan Carniato. A hands-on introduction to fine-grained reactivity, 2021. URL <https://dev.to/ryansolid/a-hands-on-introduction-to-fine-grained-reactivity-3ndf>.

Lewis D'Avanzo. What is a javascript meta-framework?, 2022. URL <https://www.ombulabs.com/blog/javascript/what-is-a-javascript-meta-framework.html>.

Gareth McCumskey. A guide to serverless architecture, 2022. URL <https://www.serverless.com/blog/serverless-architecture>.

PostgreSQL. Json functions and operators, Feb 2021. URL <https://www.postgresql.org/docs/9.5/functions-json.html>.

Web-aplikacija za praćenje podataka o nogometnim utakmicama.

Sažetak

Ukratko opisati najznačajnije mogućnosti odabranih postojećih aplikacija namijenjenih prikazu statističkih podataka o nogometnim utakmicama. Oblikovati relacijski model baze podataka za praćenje relevantnih podataka o klubovima, igračima, sudjelovanju igrača u utakmicama, formacijama, rangju natjecanja, zamjenama, trenerima ili izbornicima, posjedu lopte, asistencijama, udarcima u vrata i slično. Na osnovi izgrađenog modela implementirati bazu podataka i web-aplikaciju za prikupljanje podataka i prikaz obrađenih podataka o utakmicama, klubovima i igračima u prikladnom korisničkom sučelju. Za odabrani skup prikazanih podataka osigurati osvježavanje podataka u korisničkom sučelju u stvarnom vremenu.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.