Web-platforma za organizaciju rekreativnih sportskih aktivnosti Sportscape

Vrančić, Karlo

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:188110

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-23**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1572

WEB-PLATFORMA ZA ORGANIZACIJU REKREATIVNIH SPORTSKIH AKTIVNOSTI SPORTSCAPE

Karlo Vrančić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1572

WEB-PLATFORMA ZA ORGANIZACIJU REKREATIVNIH SPORTSKIH AKTIVNOSTI SPORTSCAPE

Karlo Vrančić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1572

Pristupnik:	**Karlo Vrančić (0036541994)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Ivica Botički

Zadatak: **Web-platforma za organizaciju rekreativnih sportskih aktivnosti Sportscape**

Opis zadatka:

U okviru završnog rada potrebno je izraditi web-platformu za organizaciju sportskih aktivnosti. Web-platforma treba sadržavati četiri osnovne komponente: pregled dostupnih termina sportskih prostora, online rezervaciju termina, modul za priključenje na jedan od već rezerviranih termina unutar kojeg je moguća komunikacija u stvarnom vremenu te modul za pregled korisničkih profila. Registracija na platformu omogućit će korisnicima kreiranje profila te rezervaciju i priključivanje na već rezervirane sportske termine, dok će vlasnici objekata moći objavljivati dostupne termine. Korisnici se međusobno mogu recenzirati, a mogu ocijeniti i same sportske prostore.

Rok za predaju rada: 14. lipnja 2024.

ZAHVALA

Zahvaljujem se izv. prof. dr. sc. Ivici Botičkom na mentoriranju ovog završnog rada. Također, iskreno zahvaljujem svakom pojedincu koji je na bilo koji način bio dio mog dosadašnjeg studentskog života.

Sadržaj

Uvod *4*

1. Korištene tehnologije *6*

1.1. Supabase **6**

1.2. Next.js **8**

1.3. JavaScript **9**

1.4. Mantine **10**

1.5. Tailwind CSS **11**

2. Arhitektura sustava *12*

2.1. Funkcionalni zahtjevi **12**

2.2. Nefunkcionalni zahtjevi **13**

2.3. Osnovna arhitektura **15**

2.4. Model baze podataka **15**

2.5. Temeljni obrazac uporabe **20**

2.6. Model dijaloga ekrana **20**

2.7. Implementirani mehanizmi **24**

2.7.1. *Row Level Security* (RLS) 24

2.7.2. Okidači u bazi podataka 26

2.7.3. Dopisivanje u stvarnom vremenu 27

2.7.4. Autentifikacija 28

2.7.5. Međuslojni kod (*middleware*) 29

2.7.6. Učitavanja 30

2.7.7. Responzivni dizajn 31

2.7.8. Tamni način 33

2.7.9. Pretraživanje teksta 34

2.7.10. Geolokacijske usluge 35

2.7.11. Postavljanje platforme na Internet 37

2.7.12. Kontinuirana integracija i kontinuirana isporuka 37

3. Korisničke upute *38*

3.1. Naslovna stranica za neregistrirane korisnike **38**

3.2. Registracija i prijava **40**

3.3. Stvaranje novog oglasa za iznajmljivanje termina **43**

3.4. Kupnja termina **45**

3.5. Priključenje na kupljeni termin **50**

3.6. Dopisivanje u stvarnom vremenu među članovima **54**

3.7. Recenzije **55**

3.7.1. Recenziranje termina 55

3.7.2. Recenziranje profila 56

Zaključak *57*

Literatura *58*

Sažetak *60*

Summary *61*

Uvod

Tijekom povijesti, ljudi su morali biti izuzetno fizički aktivni kako bi preživjeli.

Lov, sakupljanje hrane i rad na poljima zahtijevali su stalno kretanje i napor. Neizbježna fizička aktivnost bila je sastavni dio svakodnevnog života.

S razvojem tehnologije, strojevi su preuzeli mnoge fizički zahtjevne zadatke. Ipak, ljudi su i dalje imali potrebu za aktivnošću radi održavanja hormonalne ravnoteže i mentalnog zdravlja. Dakle, tjelesna aktivnost je postala manje nužna za preživljavanje, ali je imala sekundarne koristi uključujući lučenje hormona poput endorfina i dopamina koji su izazivali osjećaj zadovoljstva i blagostanja. Upravo su te potrebe ljudi ispunjavali kroz boravak na otvorenom i rekreativno bavljenje sportom.

Međutim, daljnji napredak tehnologije doveo je do pojave videoigara, društvenih mreža i drugih oblika digitalne zabave koji su postali novi izvori hormonalnih skokova. Takve digitalne aktivnosti više ne zahtijevaju fizički napor. Ovo je dovelo do ekstremno sjedilačkog načina života, koji negativno utječe na tjelesno i mentalno zdravlje.

Dok je u prošlosti tehnološki napredak uvelike utjecao na produljenje životnog vijeka i rast kvalitete života, danas smo suočeni s paradoksom u kojem tehnologija pridonosi sve većem broju zdravstvenih problema. Sedentarni način života povezan je s povećanim rizikom od pretilosti, dijabetesa tipa 2, hipertenzije i muskuloskeletalnih problema. Također, nedostatak fizičke aktivnosti negativno utječe na mentalno zdravlje, povećavajući rizik od anksioznosti, depresije i smanjenja kognitivnih funkcija [1].

Dodatnu prepreku stvorila je pandemija koronavirusa, zbog koje se velik dio populacije još više izolirao od fizičkog svijeta čija je interakcija preduvjet za bilo kakvu sportsku aktivnost [2]. Jedini fizički aktivni pojedinci su profesionalni sportaši koji pak izlažu tijelo ekstremnim naporima, što opet ima štetan utjecaj na zdravlje.

Kako bi se suprotstavili ovim negativnim posljedicama, nužno je pronaći načine za integraciju tehnologije u promicanje uravnotežene fizičke aktivnosti. U tom kontekstu, razvijena je web-platforma za organizaciju rekreativnih sportskih aktivnosti.

Sportscape je osmišljen kao centralizirano mjesto na kojem sportaši-rekreativci mogu pronaći druge entuzijaste, formirati timove i započeti nova sportska prijateljstva. Vlasnici sportskih objekata mogu putem ove platforme oglašavati dostupne termine, čime se olakšava proces rezervacije i korištenja sportskih prostora.

Pristup sportskim aktivnostima često je ograničen nedostatkom informacija i osobnih poznanstava, što rezultira neiskorištenim terminima i smanjenom participacijom u sportskim aktivnostima. Sportscape rješava ove probleme omogućujući jednostavan pristup informacijama o dostupnim sportskim aktivnostima i prostorima te olakšava formiranje sportskih timova u kontekstu ljudskih zajednica.

1. Korištene tehnologije

Početna ideja bila je primijeniti dobro poznati tehnološki *stack* – PostgreSQL, Node.js, Express i React zbog njihove pouzdanosti i široke primjene u razvoju web aplikacija.

PostgreSQL napredni je objektnorelacijski sustav za upravljanje bazama podataka, poznat po stabilnosti i bogatim značajkama. Node.js omogućuje izgradnju skalabilnih mrežnih aplikacija koristeći JavaScript na strani poslužitelja, dok Express pruža minimalistički okvir za razvoj API-ja. React, popularna biblioteka za izgradnju korisničkih sučelja, poznata je po svojoj učinkovitosti i fleksibilnosti.

Ipak, s obzirom na prirodu završnog rada odlučeno je posvetiti značajno vrijeme pronalasku potencijalno optimalnijih softverskih alata za izgradnju moderne web platforme s funkcionalnostima sličnim onima koje nudi Sportscape.

Nakon razmatranja različitih mogućnosti, odabrane su tehnologije koje su temeljene na početnom planu, ali pružaju naprednije mogućnosti i bolje odgovaraju specifičnim zahtjevima projekta.

Odlučeno je koristiti Supabase, Next.js, Tailwind CSS i Mantine. Ove tehnologije nude modernija rješenja koja omogućuju brži razvoj i veću fleksibilnost.

1.1. Supabase

Kao platforma bazirana na PostgreSQL objektno-relacijskom sustavu za upravljanje bazama podataka, Supabase nudi izvanredne mogućnosti u pogledu SQL upita, transakcija i integritetskih ograničenja. Ove značajke ne samo da osiguravaju stabilnost i skalabilnost, već i visoku razinu sigurnosti podataka čime se omogućava efikasno upravljanje velikim i složenim skupovima podataka te postizanje visokih performansi web i mobilnih aplikacija.

Jedinstvenost Supabasea proizlazi iz njegove *open-source* osnove, što osigurava visoku razinu transparentnosti i fleksibilnosti u razvoju. U usporedbi s konkurentskim platformama kao što je Firebase koji koristi NoSQL pristup, Supabase se oslanja na bogate SQL kapacitete PostgreSQL-a i esencijalno je relacijska baza podataka što omogućava realizaciju složenijih upita i funkcija. Osim temeljnih SQL mogućnosti, Supabase posjeduje mrežno upravljačko sučelje za upravljanje bazom podataka zbog čega nije potrebno instalirati najčešće glomazne

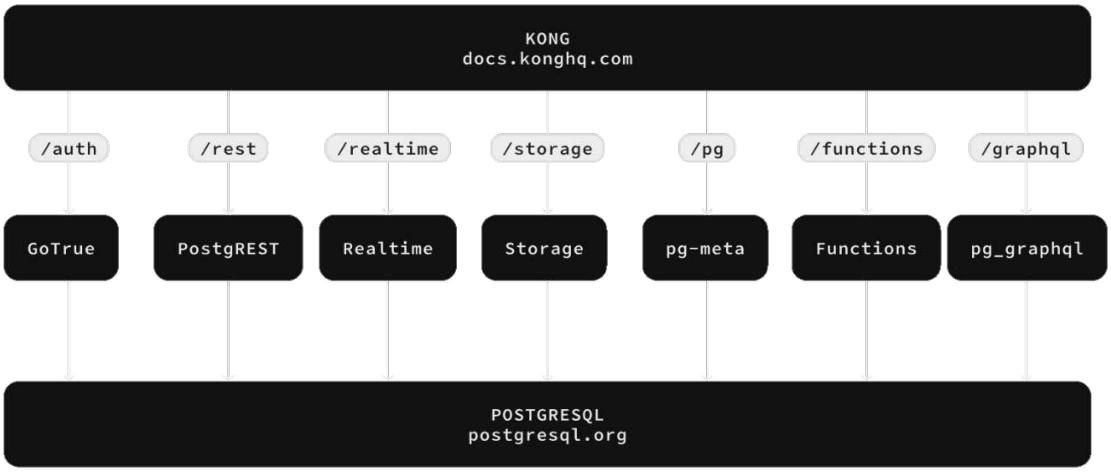
aplikacije na administratorsko računalo, ali i niz funkcionalnosti koji se tradicionalno razvijaju na poslužiteljskoj strani.

U području autentifikacije[1], Supabase implementira sigurnosne protokole kao što su JWT (JSON Web Tokens) i OAuth standarde. Ova integrirana rješenja omogućuju jednostavno i sigurno upravljanje korisničkim sesijama, ali i RLS-om (Row Level Security) za autorizaciju[2], smanjujući kompleksnost i vrijeme razvoja aplikacija [3].

Autentifikacija će detaljnije biti opisana u poglavlju 2.7.4.

Supabase također pruža učinkovite mogućnosti upravljanja pohranom, omogućujući aplikacijama da pohranjuju i dohvaćaju datoteke kao što su slike, videozapisi i dokumenti.

Dodatno, Supabase nudi brojne druge alate (Slika 1.1) kao što su auto-generirani REST i GraphQL API-ji, tzv. *webhooks,* rubne funkcije (engl. *edge functions*), samostalni hosting te osluškivanje baze podataka u stalnom vremenu koje je detaljnije opisan u poglavlju 2.7.3 [4].



Slika 1.1 Osnovna arhitektura alata integriranih u Supabase [5]

Međutim, kao i većina BaaS (*Backend as a Service*) aplikacija, Supabase je limitiran u smislu da koliko god olakša jednostavne situacije, toliko i dodatno komplicira složenije zadatke za koje nije dizajniran. Često se dogodi situacija u kojoj se više vremena izgubi na traženje

[1] Provjera da je korisnik onaj za koga se predstavlja

[2] Provjera da je korisniku ovlašten pristup nekom resursu

nekog tzv. *workaround*-a nego što se vremena uštedi na unaprijed implementiranim funkcionalnostima. Stoga je Supabase izuzetno koristan za manje projekte, no za veće i složenije projekte, poput Sportscapea, potrebno je imati dodatnu tehnologiju za poslužiteljsku stranu u rezervi. Upravo tu u priču dolazi Next.js.

1.2. Next.js

Next.js razvijen je od strane Vercela (prije poznatog kao Zeit) 2016. godine kao framework temeljen na Facebookovom okviru React s novim mogućnostima u vidu jednostavne implementacije server-side renderinga (u daljnjem tekstu SSR) i generiranja statičkih stranica (SSG) .

Next.js nasljeđuje mnoge osnovne koncepte Reacta, uključujući kompozitni pristup, virtualni DOM i deklarativno programiranje (Pseudokod 1.1)

```
Odi u kuhinju
Otvori hladnjak
Ukloni jaja iz hladnjaka
Donesi hranu za stol
```

Pseudokod 1.1. Primjer imperativnog stila

```
Želim večeru s jajima.
```

Pseudokod 1.2. Primjer deklarativnog stila

Kao i React, Next.js je dizajniran za izgradnju dinamičkih i interaktivnih korisničkih sučelja kroz upotrebu komponenti[3], malih i ponovno upotrebljivih dijelova koda koji predstavljaju dijelove korisničkog sučelja. Za potrebe Sportscapea, izrađeno je preko 60 komponenti, od kojih je u pravilu gotovo svaka korištena više puta.

Ove komponente koriste virtualni Document Object Model (DOM), što omogućava efikasno ažuriranje i renderiranje korisničkog sučelja uspoređujući prethodno i trenutno stanje DOM-a te ažurirajući samo one dijelove koji su se promijenili.

React je često osuđivan zbog nemogućnosti generiranja statičkih stranica (SSG) i izostanka rendera na poslužiteljskoj strani (SSR). Next.js rješava sve navedene nedostatke

[3] Česta (i jako dobra) analogija za izradu aplikacije pomoću komponenti jest gradnja kipa LEGO kockicama koji je tradicionalno građen iz jednog komada kamena

implementirajući istovremeno renderiranje i na serverskoj i na klijentskoj strani, a dodatno može i poslužiti kao razvojni okvir za serversku stranu što ga čini *full-stack* razvojnim okvirom i kao takvim idealnom kombinacijom sa Supabaseom.

Zadani pristup u Next.js-u je upravo SSR, a za renderiranje na klijentskoj strani koja omogućuje postojanje stranica s interaktivnim elementima dovoljno je na početak datoteke upisati ključnu riječ `'use client'`.

SSR poboljšava sigurnost i performanse aplikacija jer klijenti dobivaju potpuno renderiranu stranicu čime se smanjuje vrijeme potrebno za učitavanje na klijentskoj strani. Dodatno, SSR značajno unapređuje optimizaciju za tražilice (engl. *search engine optimization* – SEO) jer se sadržaj već renderiranih stranica može unaprijed indeksirati, što poboljšava vidljivost u rezultatima pretraživanja. Ovo je ključno za bilo koju aplikaciju koja želi postići visoku dostupnost na internetu.

Ipak, jedna od najznačajnijih prednosti Next.js-a koja ga čini full-stack radnim okvirom jest sposobnost integriranja API ruta unutar same aplikacije (Kôd 1.3).

```
export default function handler(req, res) {
  res.status(200).json({ message: 'Hello from Next.js API
Route' });
}
```

Kôd 1.3 API rute unutar radnog okvira Next.js

Upravo je ključna prednost okvira Next.js pri odabiru tehnologija za razvoj ove aplikacije bila njegova sposobnost da nadomjesti nedostatke platforme Supabase pružajući potpunu poslužiteljsku funkcionalnost kroz svoje API rutine i SSR.

1.3. JavaScript

JavaScript je temeljni jezik za izradu dinamičkih web aplikacija koji omogućuje izravno manipuliranje strukturalnim modelom dokumenata (DOM) te podržava programiranje vođeno događajima (engl. *event-driven programming*).

Njegova sposobnost upravljanja asinkronim operacijama ostvarena je kroz obećanja i asinkrone funkcije te je ključna za učinkovito izvođenje koda koji komunicira s mrežnim uslugama.

Međutim, dinamička priroda JavaScripta može otežati održavanje koda u većim projektima, što je potaknulo razvoj TypeScripta od strane Microsofta. TypeScript, kao nadskup JavaScripta, uvodi strogu tipizaciju čime se poboljšava održivost koda i omogućava rano otkrivanje pogrešaka.

Ova značajka posebno dolazi do izražaja u složenim projektima gdje TypeScript dodaje dodatni sloj sigurnosti i strukturiranosti, olakšavajući upravljanje projektima, pronalazak pogrešaka i podizanje kvalitete koda. Integracija TypeScripta u JavaScript projekte primjer je dobre prakse koja rezultira aplikacijama koje su pouzdane, efikasne i lako održive, što je od presudne važnosti za uspješan razvoj suvremenih web aplikacija.

Međutim, TypeScript se pri izvođenju prevodi u JavaScript pa je u projektu zbog jednostavnosti u nekim dijelovima korišten Javascript, dok se u kompleksnijim komponentama prešlo na TypeScript zbog veće strukturne čistoće i pouzdanosti.

1.4. Mantine

Mantine se ističe kao alat za izradu korisničkih sučelja unutar Reactovog ekosustava koji je primarno fokusiran na efikasnost i minimalno opterećenje sustava.

Razvijen je da bude iznimno fleksibilan, ovaj okvir smanjuje vrijeme potrebno za implementaciju složenih korisničkih sučelja koristeći modularne komponente iz svojih biblioteka kao što su `@mantine/core` i `@mantine/dates`.

Mantine pridonosi bržem razvoju aplikacija nudeći gotove komponente, a podržava i responzivnost te automatsku prilagodbu teme (više o istima u poglavlju 2.7.7 i 2.7.8) bez previše dodatnog koda.

Međutim, i Mantine dolazi uz svoje kompromise. Iako modularan, često je puno teže izmijeniti neku unaprijed pripremljenu komponentu na određeni način nego iz temelja stvoriti vlastitu komponentu. Upravo zato je Mantine korišten samo u ciljanim slučajevima, dok je veći dio ostalih komponenti napravljen bez korištenja knjižnica komponenti.

1.5. Tailwind CSS

Tailwind CSS (engl. *Cascading Style Sheets*) suvremeni je alat za oblikovanje web stranica koji se ističe svojim utilitarnim pristupom. Umjesto pisanja prilagođenih pravila za stiliziranje elemenata, Tailwind koristi unaprijed definirane klase koje se primjenjuju direktno na HTML (engl. *HyperText Markup Language*) elemente. Ovaj pristup omogućuje brz i učinkovit razvoj, smanjujući potrebu za prebacivanjem između datoteka sa stilovima i strukture.

Na primjer, umjesto da se piše prilagođeni CSS za boju pozadine, *padding*, stil teksta i promjenu pri tzv. *hoveru,* koristi se niz klasa (Kôd 1.4).

```
<button class="bg-blue-500 hover:bg-blue-700 text-white bold
p-4">
      Klikni me
</button>
```

Kôd 1.4 Tailwind CSS klase

2. Arhitektura sustava

U prethodnom je poglavlju već detaljno opisana arhitekturalna pozadina odabira svake od korištenih tehnologija, a u ovom će poglavlju fokus biti na samoj ostvarenoj arhitekturi.

2.1. Funkcionalni zahtjevi

Sportscape je zamišljen kao platforma na koja olakšava iznajmljivanje sportskih terena, ali i okupljanje rekreativnih sportaša u timove. Najlakša analogija ciljane funkcionalnosti bio bi hibrid oglasnika i sportske društvene mreže. Dakle, vlasnici prostora prvo iznajmljuju svoj prostor jednom sportašu, a zatim taj sportaš okuplja tim. Sve to će biti detaljnije opisano u ovom potpoglavlju.

Dionici:

a) Vlasnik sportskog objekta (u nastavku teksta: prodavač)

b) Sportaš u ulozi unajmljivača prostora

c) Sportaši u ulozi člana ekipe

Prodavač može:

1. Napraviti račun i prijaviti se u sustav
2. Kreirati oglas za iznajmljivanje sportskog terena
3. Pratiti status i statistike svojih oglasa s kontrolne ploče
4. Pregledavati zahtjeve za kupnju
5. Izbrisati oglas

Sportaš može:

1. Napraviti račun s dodatnim detaljima i prijaviti se u sustav
2. Koristiti kontrolnu ploču za pregled svojih termina
3. Kupiti termin

   3.1. Pretraživati i filtrirati oglase za kupnju

   3.2. Pregledati informacije o odabranom terminu

3.3. Poslati zahtjev za kupnju prodavaču

3.4. Uređivati kupljeni oglas

3.5. Kontrolirati zahtjeve za učlanjenje na vlastiti termin

4. Pridružiti se terminu

4.1. Pretraživati i filtrirati oglase za pridruživanje

4.2. Pregledati informacije, recenzije i članove odabranog termina

4.3. Poslati zahtjev za pridruživanje sportašu-najmoprimcu

5. Sudjelovati u grupnom razgovoru među članovima svakog termina
6. Recenzirati termina kojih je član
7. Pregledavati profile drugih sportaša i recenzirati ih

Detaljan popis svih obrazaca uporabe je izostavljen iz konačnog rada jer je ocijenjeno da su gornji zahtjevi u kombinaciji s detaljnim opisom čitavog tijeka korištenja platforme koji je dan u četvrtom poglavlju dovoljno deskriptivni.

2.2. Nefunkcionalni zahtjevi

Nefunkcionalni zahtjevi igraju ključnu ulogu u osiguravanju sveobuhvatne kvalitete softverskog sustava. Za platformu koja je hibrid oglasnika i društvene mreže za sportaše, nefunkcionalni zahtjevi definiraju standarde za performanse, skalabilnost, sigurnost, pouzdanost, održavanje, kompatibilnost, upotrebljivost, efikasnost, prenosivost i usklađenost s relevantnim standardima. Ovi zahtjevi osiguravaju da sustav ne samo da zadovoljava funkcionalne potrebe korisnika, već i pruža robusno, sigurno i visokokvalitetno iskustvo.

Performanse

- Platforma mora odgovarati na korisničke zahtjeve unutar 1.5 sekunde za 95% operacija.
- Sve stranice moraju se učitavati unutar 3 sekunde.

Skalabilnost

- Sustav mora biti sposoban horizontalno skalirati kako bi podržao povećanje broja korisnika bez smanjenja performansi.

Dostupnost

- Sustav mora biti dostupan 99% vremena tijekom godine.
- Mora biti uspostavljena kontinuirana integracija i isporuka (CI/CD), a u slučaju njenog neuspjeha treba biti implementiran tzv. *fallback* na prošlu isporučenu verziju.

Sigurnost

- Korištenje dvofaktorske autentifikacije (2FA) i granularne kontrole pristupa podacima.
- Svi osjetljivi podaci moraju biti šifrirani u prijenosu (TLS) i mirovanju.

Kompatibilnost

- Potrebno je implementirati podršku za različite uređaje i operativne sustave (desktop, mobilni uređaji, tableti).
- Platforma treba biti dovoljno modularna kako bi se u bilo kojem trenutku mogle uspostaviti integracije s drugim aplikacijama i servisima putem API-ja.

Upotrebljivost

- Pristupačnost mora biti u skladu s Web Content Accessibility Guidelines *(WCAG)* standardima.

Efikasnost

- Server i baza podataka moraju biti optimizirani u svrhu minimiziranja troškova.

Prenosivost

- Migracija Podataka: Jednostavna migracija podataka u slučaju promjene infrastrukture ili dobavljača usluga (Supabase je esencijalno BaaS i može se očekivati da će uvelike smanjiti funkcionalnosti za besplatno korištenje).

Pridržavanje Standarda

- Svi spremljeni podaci moraju biti u skladu s lokalnim i međunarodnim zakonima o zaštiti podataka (GDPR).

2.3. Osnovna arhitektura

Sportscape se temelji na Supabase infrastrukturi kojom je izvedena poslužiteljska strana koristeći niz alata za povezivanje integrirane PostgreSQL baze podataka s Next.js aplikacijom (Slika 2.1)

Slika 2.1. Arhitektura sustava temeljenog na Supabase platformi [7]

Kao posrednik između klijentske i poslužiteljske strane, korištene su API rute implementirane u Next.js-u (više o *middleware*-u u poglavlju 2.7.5) dok je i sama klijentska strana razvijena upravo u tom razvojnom okviru.

2.4. Model baze podataka

Arhitektura baze podataka ključna je komponenta svakog sustava koji se oslanja na učinkovito i pouzdano upravljanje podacima.

Dizajn baze podataka zahtijeva pažljivo planiranje kako bi se osiguralo da podaci budu pravilno strukturirani, dostupni i dosljedni.

U znanstvenoj literaturi, normalizacija podataka prepoznata je kao standardna praksa za postizanje optimalne organizacije podataka. Proces normalizacije uključuje razbijanje složenih struktura podataka u manje, povezane tablice koje zadovoljavaju određene normalne forme.

Postizanje viših normalnih formi, kao što su treća normalna forma (3NF) i Boyce-Codd normalna forma (BCNF), ključno je za minimiziranje redundancije i osiguranje integriteta podataka. Međutim, postoji inherentna tenzija između želje za visokom razinom normalizacije i potrebe za praktičnom izvedivošću i performansama sustava. Pretjerana normalizacija može dovesti do kompleksnih struktura koje otežavaju razumijevanje i upravljanje bazom podataka, osobito u slučaju kada je potrebno upravljati velikim brojem stranih ključeva.

Upravo zbog ove tenzije, arhitekti baza podataka često moraju pronaći balans između rigorozne normalizacije i praktične denormalizacije. Težina takvog zadatka bila je evidentna i tijekom razvoja platforme Sportscape.

Platforma koristi dvije glavne sheme: `public` i `auth`. Shema `public` sadrži sve osnovne tablice, dok shema `auth` upravlja autentifikacijom korisnika. U nastavku potpoglavlja ćemo se fokusirati na shemu `public` koja je podijeljena u deset tablica (Tablica 2.1)

Naziv	Opis
`profile`	Pohranjuje osnovne podatke o korisnicima, kao što su URL slike profila (`picture_url`), tip korisnika (`type`) i ime (`name`). Povezana je s autentifikacijskom shemom putem `id` polja.
`athlete`	Pohranjuje specifične informacije o sportašima, uključujući dob, visinu, težinu te razine vještina za košarku, nogomet i odbojku. Svaki sportaš je identificiran jedinstvenim `athlete_id`, koji je također povezan s profilom u tablici `profile`.
`vendor`	Pohranjuje informacije o prodavačima koji iznajmljuju sportske termine. Svaki prodavač ima jedinstveni `vendor_id`, koji je povezan s korisničkim profilom u tablici `profile`.

offer	Sadrži podatke o ponudama između sportaša i vlasnika dvorana. Svaka ponuda ima jedinstveni `offer_id` i povezana je s prodavačem (`vendor_id`), sportašem (`athlete_id`), terminom (`slot_id`) te statusom (`status`)
slot	Sadrži informacije o terminima za sportaše. Svaki termin ima jedinstveni `slot_id`, te sadrži detalje kao što su vrijeme početka i završetka (`start_time, end_time`), dostupnost (`is_available`), fotografija termina (`slot_photo`) i adresa (`address`).
slot_request	Pohranjuje zahtjeve za pridruživanje terminima. Svaki zahtjev sadrži `slot_id, athlete_id`, status zahtjeva (`status`) te opcionalnu poruku (`message`)
slot_member	Služi za povezivanje sportaša s terminima u kojima sudjeluju. Svaki zapis sadrži `slot_id`, `athlete_id`, informacije o tome je li sportaš vlasnik termina (`is_owner`) te datum pridruživanja (`date_joined`)
slot_review	Sadrži recenzije termina koje su napisali sportaši. Svaka recenzija sadrži `athlete_id, slot_id`, ocjenu (`rating`), tekst recenzije (`text`) i informaciju o tome je li recenzija anonimna (`is_anonymous`)
user_review	Pohranjuje recenzije koje korisnici pišu jedni o drugima. Svaka recenzija sadrži `from_user_id, to_user_id`, tekst recenzije (`text`) i ocjenu (`rating`)
chat_message	Pohranjuje poruke koje se razmjenjuju između korisnika unutar određenih termina (`slot`). Svaka poruka ima jedinstveni `message_id`, a također pohranjuje informacije o vremenu slanja (`timestamp`), pošiljatelju (`sender_id`) i tekstu poruke (`message_text`).

Tablica 2.1 Struktura sheme public u bazi podataka

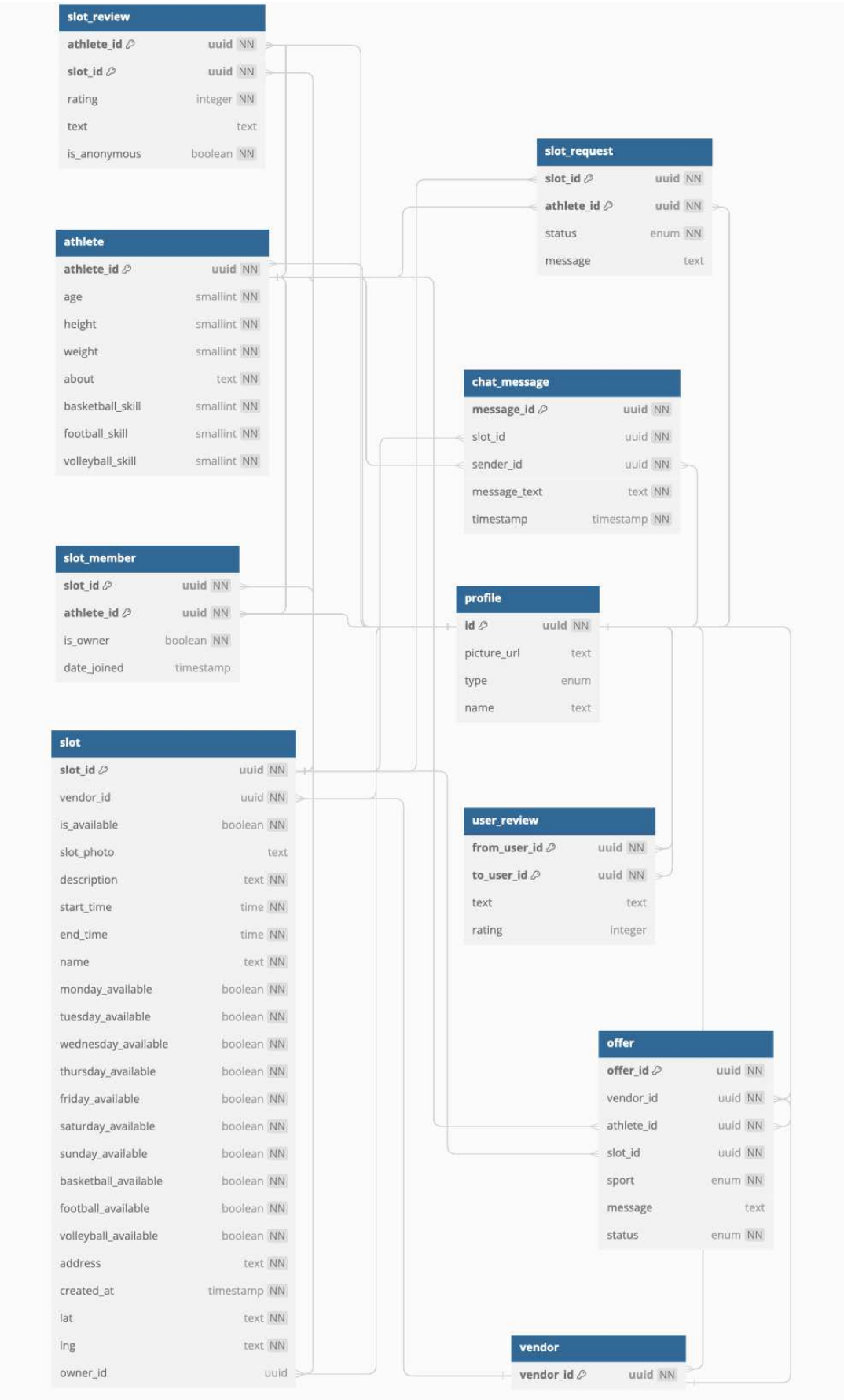
Relacijsku shemu baze podataka prikazuje Slika 2.2.

Shema pruža vizualni pregled različitih tablica, njihovih atributa, primarnih i stranih ključeva te relacija među tablicama.

Primijetite kako se u tablici `slot` umjesto enumeracije `sport_type` koriste tri zasebna atributa (`basketball_available`, `football_available`, `volleyball_available`). Ovo rješenje omogućava terminima da budu dostupni za više sportova istovremeno dok paralelno ne krše prvu normalnu formu po kojoj svaki atribut ima samo jednu vrijednost iz domene. U ostatku modela se također nastojalo održati normalizaciju, ali je vidljivo koliko se posljedično stvorilo relacija i stranih ključeva.

Upravo zato je vođena posebna briga da implementacija budućih nadogradnji bude što jednostavnija zbog čega su unaprijed definirane tablice kao npr. `vendor` koja pohranjuje specifične informacije o prodavaču. Za razliku od tablice `athlete`, ova tablica trenutno ne sadrži nikakve dodatne informacije o korisniku iz tablice `profile` pa je kao takva u ovom trenutku redundantna. Međutim, u budućim verzijama planirano je proširiti funkcionalnosti uvođenjem plaćanja u aplikaciji i sl. (više o tome u zaključku) te ova tablica služi kao tzv. *placeholder* kako ne bi trebalo naknadno mijenjati veze i strane ključeve među tablicama.

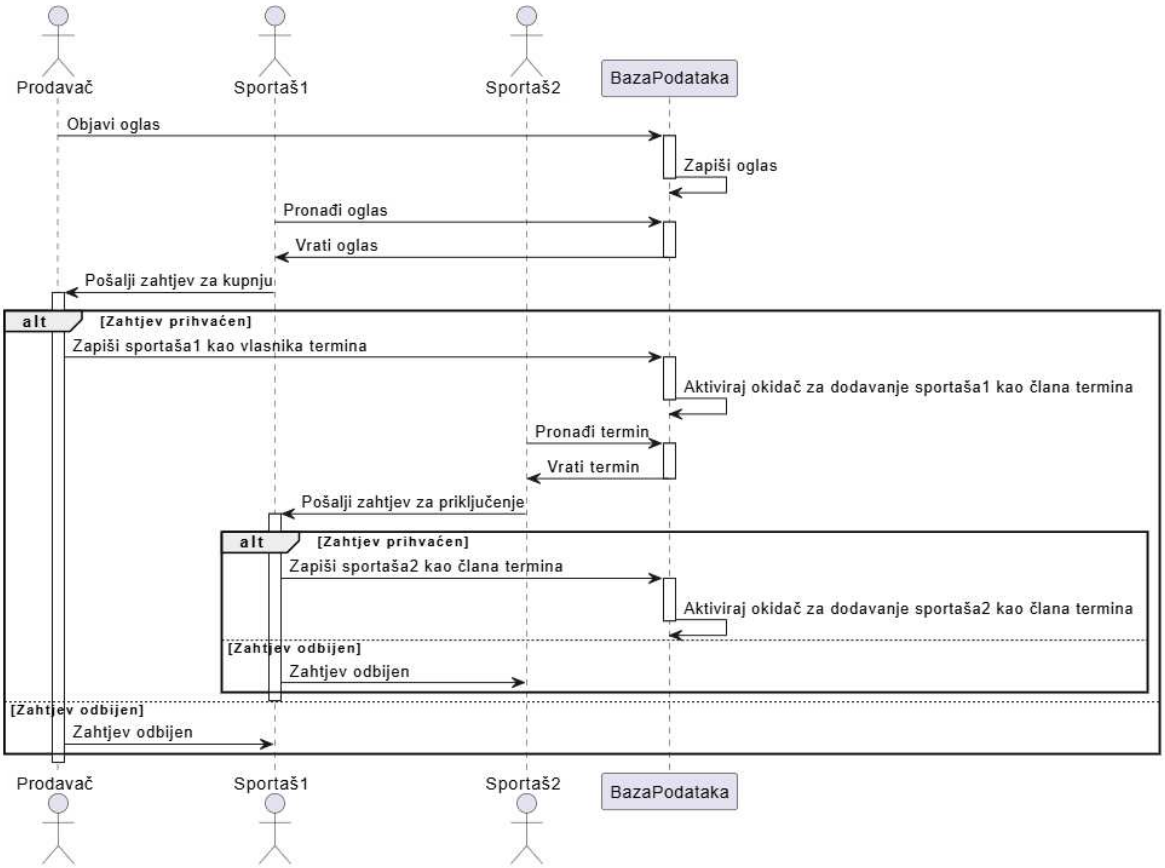
Nastavno, funkcije i okidači unutar baze podataka automatiziraju (i time pojednostavljuju) ključne procese. Na primjer, funkcija `handle_accept_join_request` automatski dodaje sportaša u tablicu `slot_member` kada je zahtjev za pridruživanje prihvaćen. Slično tome, funkcija `handle_accept_offer` ažurira informacije o vlasništvu termina i dostupnosti kada je ponuda prihvaćena. Više o funkcijama i okidačima se može pročitati u poglavlju 2.7.2.



Slika 2.2 Vizualizacija sheme baze podataka

2.5. Temeljni obrazac uporabe

Temelj funkcionalnosti čitave platforme može se dati kroz jedan jedini visokorazinski obrazac uporabe prikazan sekvencijskim dijagramom (Slika 2.3).



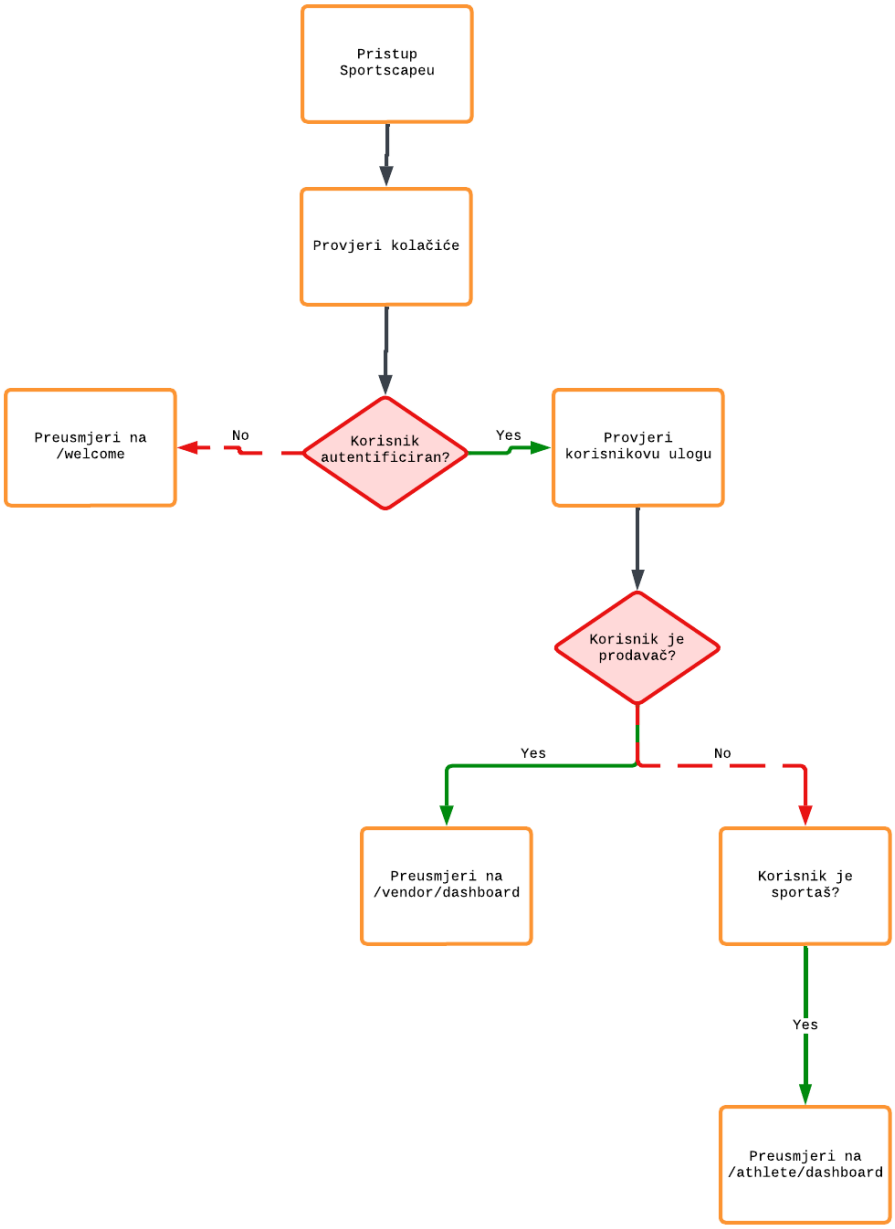
Slika 2.3 Osnovni tijek korištenja

U platformu su dodane i ostale mogućnosti poput filtriranja, recenziranja i dopisivanja, a detaljan pregled kroz ovaj čitav tijek obogaćen dodatnim funkcionalnostima i popraćen snimkama zaslona dan je u četvrtom poglavlju.

2.6. Model dijaloga ekrana

Odlaskom na adresu platforme Sportscape izvršava se provjera kolačića te ako je korisnik neautentificiran, preusmjerava ga se na `/welcome` stranicu, a ako se radi o autentificiranom korisniku provjerava se njegova uloga (Slika 2.4):

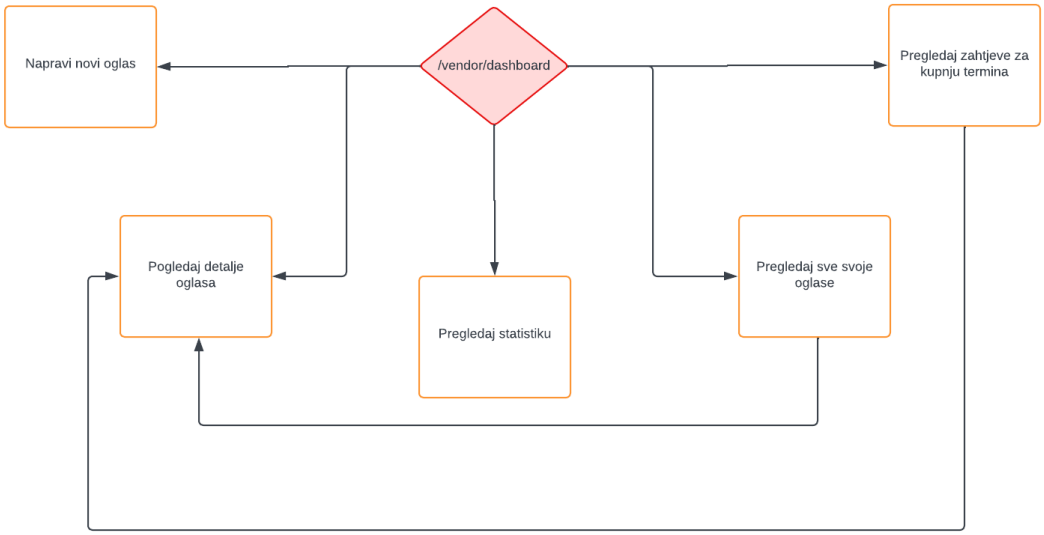
- ako je korisnik prodavač, preusmjerava se na `/vendor/dashboard`
- ako je korisnik sportaš, preusmjerava se na `/athlete/dashboard`



Slika 2.4 Dijagram logike preusmjeravanja na početnoj stranici

Nadalje, Slika 2.5 prikazuje na koje stranice prodavač može otići počevši od svoje kontrolne ploče:

- stranica za pregled statistike
- stranica pregleda detalja nekog od prethodna 3 oglasa
- stranica za pregled svih svojih oglasa
- stranica za pregled svih zahtjeva za kupnju njegovih termina

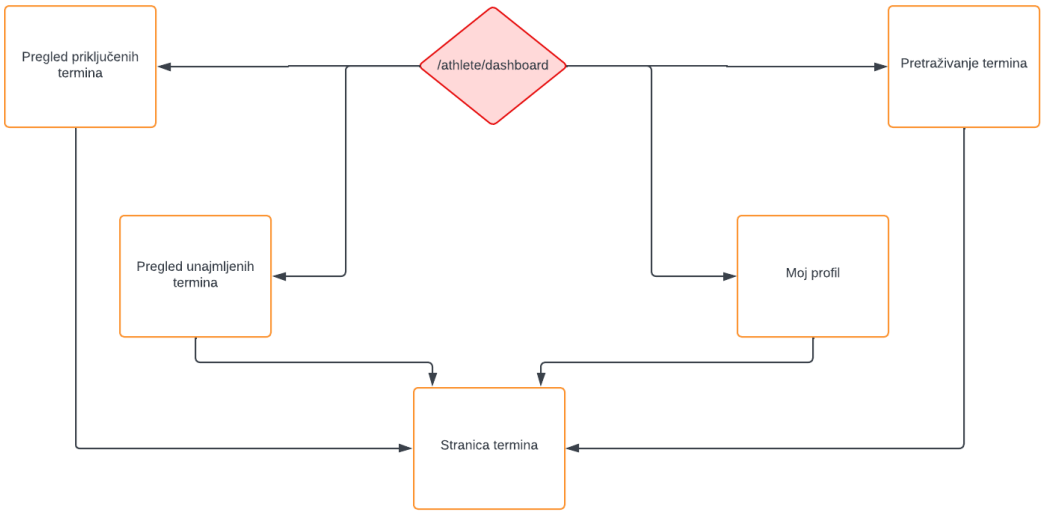


Slika 2.5 Navigacija s prodavačeve korisničke ploče

S druge strane, sportaš sa svoje kontrolne ploče može:

- pregledati popis termina na kojima je član
- pregledati popis termina koje unajmljuje
- otići na pretragu novih termina za najam ili pridruživanje
- otići na svoj profil

Sa svake od tih stranica korisnik može između ostaloga otići na stranicu nekog od termina (Slika 2.6).



Slika 2.6 Pojednostavljeni prikaz navigacije sa sportaševe kontrolne ploče

Upravo stranica termina predstavlja centralnu komponentu platforme Sportscape, koja omogućuje različite interakcije korisnika s terminima ovisno o njihovoj ulozi. Ova stranica

je polimorfna, što znači da se njezin izgled i funkcionalnost dinamički mijenjaju ovisno o tome je li korisnik potencijalni kupac, vlasnik termina, član termina ili potencijalni član. Korištenje Next.js-a i Reactovog komponentnog pristupa omogućava modularno i fleksibilno upravljanje ovim prikazima. Svaka komponenta je dizajnirana da bude samostalna jedinica koja se može lako integrirati i prilagoditi u različitim kontekstima. Upravo je to bilo ključno jer unatoč tome što na stranici za termine postoje različiti prikazi, oni najčešće dijele iste komponente.

Na stranici se uvijek prikazuju osnovne informacije o terminu (`SlotBasicInfo` komponenta) koja uključuje podatke kao što su naziv termina, vrijeme, lokacija na karti, dostupni sportovi i opis dok ostali dijelovi variraju ovisno o prikazu:

- Ako je sportaš potencijalni kupac termina, bit će prikazana terminalna (engl. *leaf*) komponenta `BuyRequestForm`
- Ako je sportaš vlasnik termina, bit će prikazana kompozitna `OwnedSlot` komponenta sa svojom djecom (engl. *children components*)
- Ako je sportaš član termina, bit će prikazana kompozitna `JoinedSlot` komponenta sa svojom djecom
- Ako je sportaš potencijalni član termina, bit će prikazana kompozitna `ToJoinSlot` komponenta sa svojom djecom

S obzirom da se stranica sastoji od preko 20 različitih komponenti, niti jedan UML dijagram ili dijagram toka nije pogodan za prikaz hijerarhije pa je ista pojednostavljena i prikazana jednostavnim ASCII art-om (Slika 2.7).

```
PAGE
  SlotBasicInfo
  ├── OwnedSlot
  │   └── SegmentedControl
  │       ├── EditSlot
  │       ├── Chat
  │       ├── BoughtSlotInfo
  │       │   ├── Reviews (noedit)
  │       │   └── People
  │       └── Requests
  ├── ToJoinSlot
  │   └── SegmentedControl
```

```
│       ├── JoinRequestForm
│       ├── BoughtSlotInfo
│       │   ├── Reviews (noedit)
│       │   └── People
├── JoinedSlot
│   └── SegmentedControl
│       ├── Chat
│       ├── BoughtSlotInfo
│       │   ├── Reviews (edit)
│       │   └── People
└── BuyRequestForm
```

Slika 2.7 Dijagram hijerarhije komponenti na stranici za prikaz termina

2.7. Implementirani mehanizmi

U ovom potpoglavlju su objašnjeni neki od glavnih sigurnosnih i funkcionalnih mehanizama koji čine jezgru platforme Sportscape, krenuvši od baze podataka, preko poslužitelja i poslovne logike do klijentske strane.

2.7.1. *Row Level Security* (RLS)

Redna sigurnost (engl. *Row level security* – RLS) omogućava provođenje sigurnosti na razini baze podataka umjesto na razini aplikacije. Na primjer, recimo da imamo model podataka gdje su korisnici članovi timova (korisnik ima strani ključ koji referencira tim, omogućujući korisnicima da budu članovi samo jednog tima) i želimo osigurati da samo korisnik koji je član tima može pristupiti ili mijenjati podatke tog tima.

U tradicionalnom pristupu, API rukovatelj ili neki drugi kod na razini aplikacije eksplicitno bi pokrenuo SQL upite kako bi provjerio je li korisnik član tima i dopustio operaciju samo ako aplikacija odluči da je to u redu. Ako postoji greška u ovom kodu ili ako ovaj kod na razini aplikacije uopće ne postoji, sigurnosni zahtjevi na modelu podataka i njihovo provođenje prestali bi postojati.

S RLS-om, baza podataka može sama izvršiti ovu provjeru kao dio definiranog sustava integriteta relacijskih podataka. Ovo dodatno poboljšava pouzdanost naših podataka čineći ove sigurnosne zahtjeve ključnim dijelom modela [8].

Supabase koristi Postgresov mehanizam za RLS, omogućujući implementaciju složenih SQL pravila koja se mogu kombinirati s Supabase Auth za sveobuhvatnu sigurnost korisnika od preglednika do baze podataka. RLS pruža višeslojnu obranu (engl. *defense in depth*) kako bi zaštitio podatke čak i kada im se pristupa putem alata s treće strane (engl. *third party*).

RLS je zasnovan na pravilima (engl. *policy*). Svako pravilo je vezano za tablicu te je provedeno svaki put kada se pristupa tablici.

Ovaj mehanizam se može percipirati kao da dodajemo `WHERE` svakom upitu.

Naprimjer, pravilo po kojem samo autentificirani vlasnik profila može uređivati svoj profil izgledalo bi kao Kôd 2.1.

```
CREATE POLICY "Users can update their own profiles."
ON public.profile
FOR UPDATE
TO authenticated
USING ( (select auth.uid()) = id )
WITH CHECK ( (select auth.uid()) = id );
```

Kôd 2.1 Primjer pravila

Uz ovo pravilo će se upit `UPDATE public.profile SET name = 'New Name' WHERE id = 'some-uuid';` automatski pretvoriti u Kôd 2.2.

```
UPDATE public.profile
SET name = 'New Name'
WHERE id = 'some-uuid' AND auth.uid() = profile.id;
```

Kôd 2.2 Primjer upita s automatski primijenjenim pravilom

Pravila slično funkcioniraju i za SELECT, INSERT i DELETE naredbe u bazi podataka.

Dodatno, Supabase nudi nekoliko pomoćnih funkcija koje dodatno olakšavaju RLS.

Funkcija `auth.uid()` koju smo već demonstrirali u primjerima gore vraća jedinstveni identifikator korisnika koji izvršava zahtjev.

Nadalje, funkcija `auth.jwt()` pruža dodatnu razinu fleksibilnosti vraćanjem JSON Web Tokena (JWT) korisnika. Ovaj token sadrži različite metapodatke, uključujući podatke pohranjene u stupcima `raw_app_meta_data` i `raw_user_meta_data`.
Dok `raw_user_meta_data` može ažurirati sam korisnik, `raw_app_meta_data` ostaje nepromijenjeno od strane korisnika, čime postaje idealno mjesto za pohranu autorizacijskih podataka. Na taj način, `auth.jwt()` omogućava složene provjere i uvjete unutar sigurnosnih politika, poput određivanja pripadnosti korisnika određenom timu ili grupi.

Međutim, ključno je naglasiti važnost svježine JWT tokena. JWT nije uvijek trenutno ažuriran, što može dovesti do situacija gdje promjene u korisničkim ovlastima nisu odmah reflektirane u sigurnosnim politikama. Primjerice, ako korisnika uklonimo iz tima i ažuriramo polje `app_metadata`, te promjene neće biti vidljive kroz `auth.jwt()` dok se JWT ne osvježi. Ovo naglašava potrebu za pravilnim upravljanjem i osvježavanjem JWT tokena kako bi se osigurala dosljednost i točnost primjene sigurnosnih politika.

Također, sve stupce koji se koriste u pravilima, a nisu primarni ključevi bi bilo dobro indeksirati. Službeni *benchmark* u Supabase dokumentaciji ukazuje kako je indeksirani upit izvršen za manje od 0.1 milisekunde dok je istom upitu bez indeksa oko 170 milisekundi [9].

2.7.2. Okidači u bazi podataka

Okidači omogućuju dinamičku reakciju na promjene u podacima, osiguravajući konzistentnost i integritet baze podataka.

Jedan od najvažnijih okidača u sustavu za platformu Sportscape je onaj koji se aktivira prilikom prihvaćanja ponude. Ovaj okidač osigurava da se odgovarajuće promjene automatski izvrše u drugim relevantnim tablicama kada se ponuda prihvati. Funkcija `handle_accept_offer` je definirana kako bi upravljala ovim procesom (Kôd 2.3).

```
CREATE OR REPLACE FUNCTION handle_accept_offer()
RETURNS TRIGGER AS $$
BEGIN
```

```
  IF NEW.status = 'accepted' THEN
    -- Insert into slot_member table
    INSERT INTO slot_member (slot_id, athlete_id, is_owner)
    VALUES (NEW.slot_id, NEW.athlete_id, TRUE);

    -- Update the slot table
    UPDATE slot
    SET owner_id = NEW.athlete_id, is_available = FALSE
    WHERE slot_id = NEW.slot_id;
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_accept_offer
AFTER UPDATE OF status
ON offer
FOR EACH ROW
WHEN (NEW.status = 'accepted')
EXECUTE FUNCTION handle_accept_offer();
```

Kôd 2.3 Funkcija `handle_accept_offer` i okidač koji ju aktivira

Osim navedenog okidača i pozivajuće funkcije, implementirani su okidači za rukovanje dodavanja novog korisnika i učlanjenja sportaša na postojeći termin.

2.7.3. Dopisivanje u stvarnom vremenu

Platforma Sportscape koristi moć Supabase Realtime servera kako bi omogućila slanje i primanje poruka u stvarnom vremenu između povezanih klijenata. Ova funkcionalnost je implementirana unutar chat komponente, osiguravajući trenutnu komunikaciju među korisnicima. Supabase Realtime koristi distribuirani klaster servera za pružanje niskolatentnih poruka, praćenje prisutnosti korisnika i slušanje promjena u Postgres bazi podataka.

Da bismo omogućili Realtime na tablici `chat_message` postavljamo slušanje promjena na klijentskoj strani (Kôd 2.4):

```
const channel = supabase.channel('custom-filter-channel')
```

```
      .on(
        'postgres_changes',
        { event: 'INSERT', schema: 'public', table:
'chat_message', filter: `slot_id=eq.${slot_id}` },
        async (payload) => {
          console.log('Change received!', payload);
          const newMessage = payload.new;

          // Fetch the sender's profile information
          const { data: profile, error: profileError } =
await supabase
            .from('profile')
            .select('id, name, picture_url')
            .eq('id', newMessage.sender_id)
            .single();

          if (profileError) {
            console.error('Error fetching profile for new
message:', profileError);
          } else {
            newMessage.profile = profile;
            setMessages((prevMessages) => [...prevMessages,
newMessage]);
          }
        }
      )
      .subscribe();
```

Kôd 2.4 Implementacija slušanja u stvarnom vremenu u komponenti za dopisivanje

Supabase Realtime dolazi s još nekoliko funkcionalnosti, a kao najzanimljivije ističu se *broadcast* za npr. praćenje pozicije kursora u online igrama i *presence* za praćenje stanja korisnika.

2.7.4. Autentifikacija

Autentifikacija i autorizacija ključne su komponente platforme Sportscape. One osiguravaju da su korisnici oni za koje se predstavljaju i da imaju odgovarajući pristup resursima. Sportscape koristi Supabase Auth za implementaciju ovih funkcionalnosti.

Supabase Auth se temelji na JSON Web Tokenima (JWT) za autentifikaciju. JWT omogućuje kontrolu pristupa bazi podataka na razini pojedinačnih redaka u kombinaciji s RLS politikama. Integracija s PostgreSQL bazom podataka omogućuje povezivanje korisničkih podataka s vlastitim tablicama koristeći okidače i vanjske ključeve, čime se povećava sigurnost i konzistentnost podataka.

Platforma Sportscape koristi PKCE tijek (engl. *Proof Key for Code Exchange flow*) za dodatnu sigurnost. PKCE tijek omogućuje poslužiteljsko autentificiranje, osiguravajući dodatni korak razmjene tokena prije nego što aplikacija dobije pristupni token. Time se smanjuje mogućnost presretanja tokena i povećava sigurnost autentifikacijskog procesa.

Uz PKCE tijek, Supabase Auth omogućuje autentifikaciju putem čarobnih poveznica i jedinstvenu prijavu (SSO). Ove metode mogu se poslati iz administratorskog sučelja baze podataka, ali nisu implementirane u klijentsku aplikaciju platforme Sportscape zbog redundancije.

Supabase pruža pomoćne funkcije koje olakšavaju implementaciju sigurnosnih pravila u Sportscapeu. Funkcija `auth.uid()` vraća ID korisnika koji izvršava zahtjev, omogućujući preciznu kontrolu pristupa. Funkcija `auth.jwt()` vraća JWT korisnika, omogućujući pristup metapodacima pohranjenim u tokenu, što je korisno za složene provjere autorizacije.

2.7.5. Međuslojni kod (*middleware*)

Middleware u platformi Sportscape igra ključnu ulogu u upravljanju pristupom, autorizaciji i sigurnosti korisnika. Ovaj posrednički sloj osigurava ažuriranje korisničkih sesija, provjeru autentifikacije, dohvaćanje korisničkih profila i preusmjeravanje korisnika na odgovarajuće *dashboard* stranice na temelju njihovih uloga.

Nakon ažuriranja sesije, *middleware* provjerava je li korisnik autentificiran. Ako korisnik nije prijavljen, preusmjerava ga na stranicu dobrodošlice, čime se osigurava da samo ovlašteni korisnici imaju pristup zaštićenim dijelovima aplikacije. Dohvaćanje korisničkog profila omogućuje određivanje uloge korisnika (`vendor` ili `athlete`), što je ključno za preusmjeravanje na odgovarajuće *dashboard* stranice.

Middleware također štiti specifične rute unutar aplikacije.

Prema OWASP-u, upravo je loše implementirana kontrola pristupa 2021. godine odabrana kao najveći sigurnosni rizik web-aplikacija [11]. Primjer loše implementacije predstavlja očekivanje da će korisnik pretraživati samo one putanje kojima može pristupiti preko klijentske aplikacije. To zlonamjerni napadači iskorištavaju tako da direktno u tražilice ručno upišu putanje. Takvo ponašanje u platformi Sportscape sprječava međuslojni kod.

Na primjer, korisnici s ulogom `athlete` neće moći pristupiti rutama namijenjenim `vendor` korisnicima i obrnuto. Ovaj mehanizam osigurava da korisnici mogu pristupiti samo relevantnim dijelovima aplikacije, čime se povećava sigurnost i pravilno funkcioniranje sustava.

2.7.6. Učitavanja

U platformi Sportscape, datoteka `loading.js` se koristi za prikaz stanja učitavanja koristeći React Suspense. Ova datoteka omogućuje prikaz trenutnog stanja učitavanja odmah pri navigaciji, pružajući bolje korisničko iskustvo. Time se osigurava da korisnici vide indikatore učitavanja poput kostura ili *spinnersa* čim započne učitavanje nove stranice.

Implementacija `loading.js` je jednostavna, što omogućuje brzo postavljanje i prilagodbu. U slučaju Sportscapea, koristi se Mantine komponenta za prikaz *loadera*, što dodatno poboljšava vizualnu privlačnost sučelja (Slika 2.8).

Slika 2.8 Različiti prikazi učitavanja ovisno o tipu komponente

Dodatno, implementirani su i kosturi koji se prikazuju umjesto sporijih komponenti dok se iste ne učitaju što čini iskustvo pregledavanja stranice još dinamičnijim.

Jedna od ključnih prednosti korištenja `loading.js` u Next.js je trenutna navigacija, čak i uz server-centrično usmjeravanje. Navigacija je prekidljiva, što znači da promjena ruta ne mora čekati da se sadržaj potpuno učita prije nego što se pređe na drugu rutu. Zajednički izgledi

(*shared layouts*) ostaju interaktivni dok se novi segmenti ruta učitavaju, čime se dodatno poboljšava korisničko iskustvo.

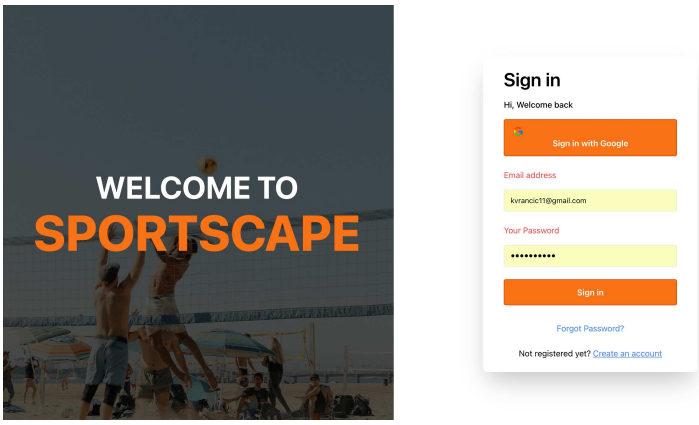
2.7.7. Responzivni dizajn

S obzirom na prirodu aplikacije, logično je očekivati kako će korisnici potencijalno češće pristupati platformi Sportscape preko mobilnih uređaja nego preko stolnih računala. Zato je tijekom razvoja vođena posebna briga o responzivnom dizajnu, odnosnu rasporedu komponenti koji se mijenja ovisno o dimenzijama korisnikovog ekrana.

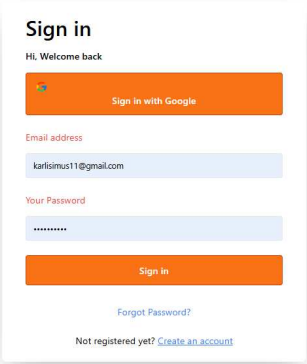
Takva je funkcionalnost uz detaljno planiranje prije implementacije s lakoćom ostvarena koristeći Tailwind CSS i neke Mantine komponente s unaprijed implementiranom responzivnošću.

Tailwind CSS omogućava stvaranje responzivnog dizajna putem svojih klasa koje su specifično dizajnirane za različite veličine ekrana. U Tailwindu se koriste tzv. *breakpoints* kako bi se definirale različite stilizacije za različite dimenzije ekrana. Primjerice, klase poput `sm:`, `md:`, `lg:`, `xl:` i `2xl:` koriste se za primjenu stilova na malim, srednjim, velikim i jako velikim ekranima.

Međutim, u samoj implementaciji različitih prikaza su korištene i komponente koje su vidljive samo u jednoj vrsti prikaza, dok u drugom nisu. Najjednostavniji primjer takve situacije jest ponovno stranica za prijavu (Slika 2.9 i Slika 2.10)

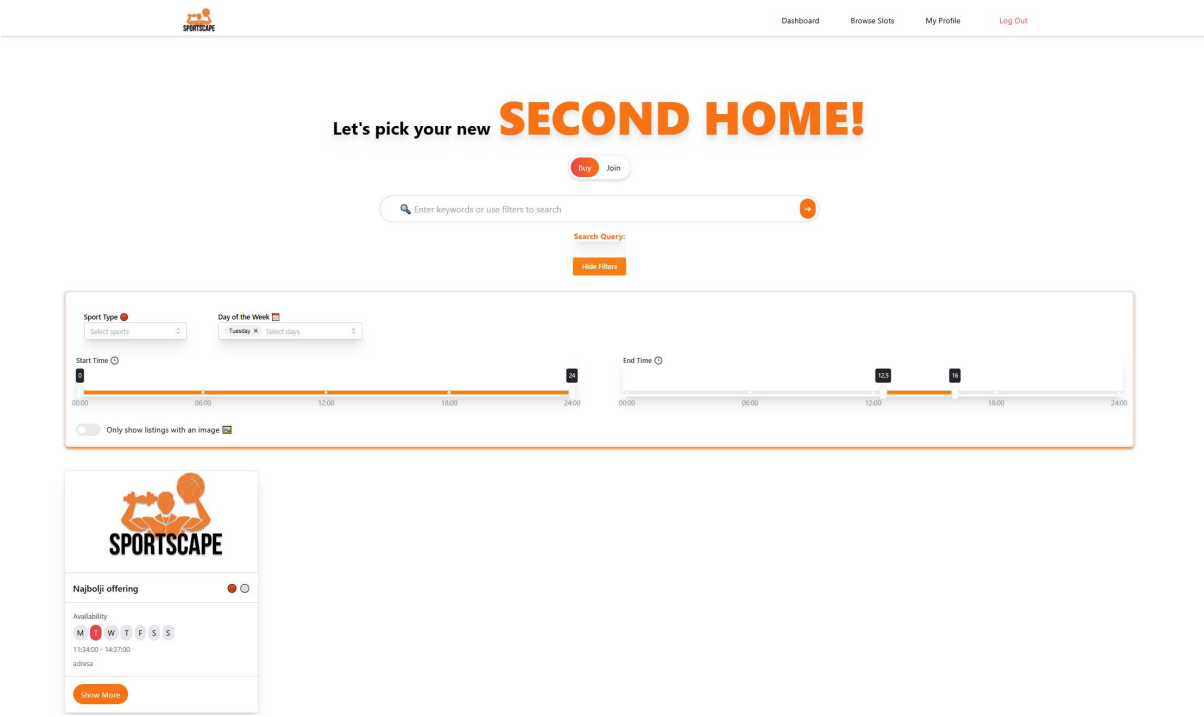


Slika 2.9 Prikaz prijave na računalnom monitoru

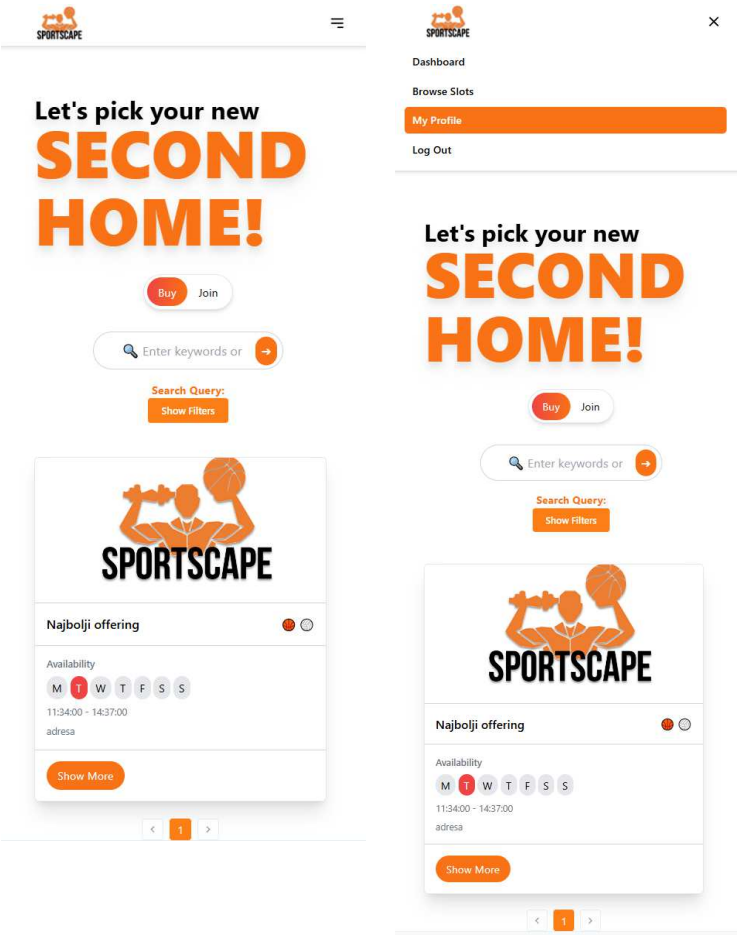


Slika 2.10 Prikaz prijave na mobitelu

Još jedan primjer takve situacije je navigacijska traka za uloge prodavača i sportaša. Na širokim ekranima se ista nalazi na vrhu stranice (sportaši – Slika 2.11) ili u jednom elementu *grid* komponente (prodavač), dok se kod obojice na mobilnim prikazima navigacija odvija klikom na tzv. *burger* u gornjem kutu stranice nakon kojeg se prikazuje navigacijski izbornik (Slika 2.12).



Slika 2.11 Široki prikaz stranice za pretraživanje termina u ulozi sportaša



Slika 2.12 Mobilni prikaz stranice za pretraživanje termina u ulozi sportaša

2.7.8. Tamni način

Iako iznimno popularan posljednjih nekoliko godina, tzv. tamni način u aplikacijama nije ništa novo. Još su sedamdesetih godina prošlog stoljeća računala koristila tamnu temu kao zadanu, ali pojavom LCD tehnologije u devedesetima prikaz svijetlih slova na tamnoj podlozi postao je neprecizan zbog niske rezolucije takvih ekrana [12].

U međuvremenu, tehnologija ekranskih panela je napredovala zbog čega su navedeni problem postali stvar prošlosti pa je 2016. godine Microsoft uveo mogućnost tamnog načina [13], što je 2018. napravio i Apple u macOS Mojaveu [14]. Uskoro su na trend uskočili i ostali tehnološki lideri jer su korisnici dočekali ove promjene širom otvorenih ruku.

Istraživanje provedeno 2022. godine pokazalo je kako mladi pokazuju mnogo više pozitivnih emocija pri korištenju tamnog načina i da ge preferiraju iz estetskih razloga [15]. Dodatno, Američka oftalmološka akademija (AAO) na svojim stranicama preporučuje korištenje

digitalnih uređaja u tamnom načinu [16]. Isto tako, na Sveučilištu Centralna Florida provedeno je istraživanje koje je pokazalo znatno povećanje u vizualnoj aktivnosti i smanjenje vizualnog umora kod subjekata koji su koristili tamni način u usporedbi sa svijetlim načinom. Međutim, ispitanici nisu koristili obične uređaje već naglavne ekrane (engl. *see-through head-mounted displays* – OST-HMD) [17]. Što se računalnih i mobilnih ekrana tiče, još nisu pronađeni čvrsti dokazi koji bi ukazali na korelaciju između tamnog načina i smanjenja štetnosti ekrana za ljudske oči.

Ipak, činjenica je da mnogi ljudi svejedno preferiraju isti te je zato implementiranje tamnog načina u platformu Sportscape bilo potrebno.

Dodatno, posljednjih nekoliko godina su na tržištu sve prisutniji ekrani s OLED panelom koji pri prikazu tamnih dijelova štede energiju.

Tamni način u platformi Sportscape implementiran je pomoću Mantine biblioteke. Mantine komponente nativno podržavaju tamni način rada, što omogućava jednostavnu i efikasnu implementaciju bez dodatnih koraka. Sve što je potrebno jest omotati aplikaciju unutar `MantineProvider` komponente i specificirati `colorScheme` kao '*dark*'.

Međutim, pošto je većina komponenti stvoreno koristeći i vlastito izrađene elemente izvan Mantine knjižnice, bilo je potrebno uskladiti njihovo ponašanje s onima koje kontrolira `MantineProvider` što je uspješno napravljeno.

Zbog izbjegavanja redundancije, snimke zaslona u tamnom načinu neće biti prikazane u ovom poglavlju, već u poglavlju s korisničkim uputama u kojem različiti načini prikaza služe za diferencijaciju između sportaša – člana termina i sportaša – vlasnika termina.

2.7.9. Pretraživanje teksta

Pretraživanje teksta unutar baze podataka ključno je za poboljšanje funkcionalnosti platforme Sportscape jer omogućuje korisnicima lako pretraživanje termina pomoću ključnih riječi. PostgreSQL nudi napredne ugrađene funkcije za *full-text search*, koje se koriste za implementaciju ove značajke. Postupak započinje funkcijom `to_tsvector()`, koja tekstualne podatke pretvara u niz pretražljivih tokena. Na primjer, upit `SELECT to_tsvector('basketball training session');` pretvara tekst u vektor koji sadrži riječi '*basketball*', '*training*' i '*session*' zajedno s njihovim pozicijama. Ti tokeni služe kao

osnovni entiteti za pretraživanje unutar baze podataka. Sljedeća ključna funkcija, `to_tsquery()`, pretvara korisnički upit u pretražive tokene. Na primjer, `SELECT to_tsquery('basketball & training');` konvertira upit u format koji omogućava *fuzzy matching*, pronalaženje podudaranja sličnih pojmova.

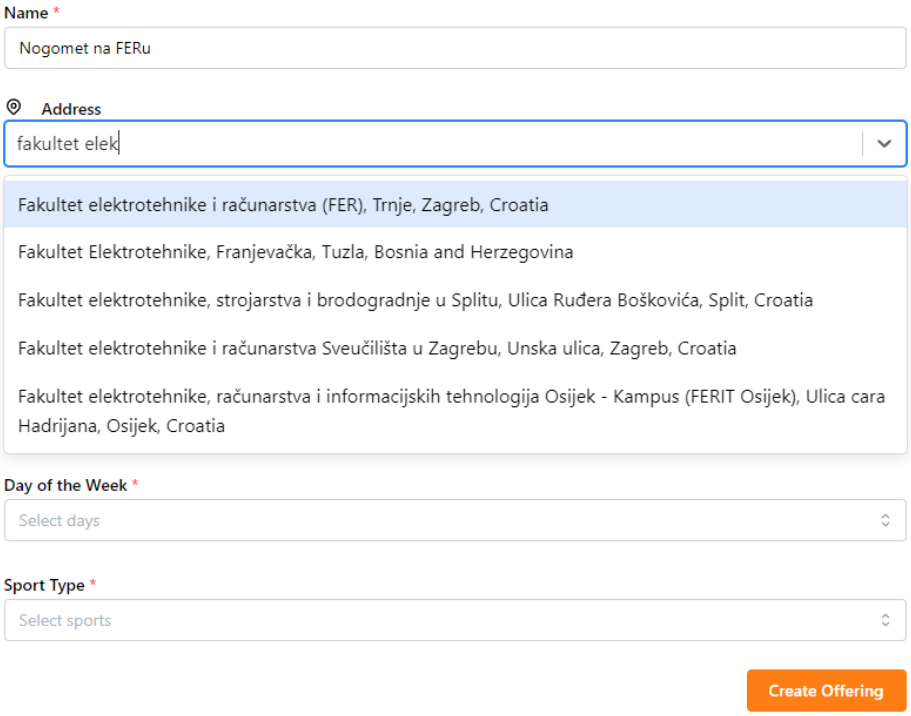
Operator `@@` koristi se za usporedbu rezultata funkcija `to_tsvector` i `to_tsquery`, omogućujući pretraživanje svih zapisa koji se podudaraju s korisničkim upitom. U Sportscapeu, pretraživanje termina može obuhvatiti više stupaca, spajajući ih u jedan pretraživi vektor. Na primjer, `SELECT * FROM slots WHERE to_tsvector(description || ' ' || title) @@ to_tsquery('basketball');` pretražuje tekstualne podatke u stupcima `description` i `title`, osiguravajući precizne rezultate pretraživanja.

Kako bi se osigurala efikasnost pretraživanja, koristi se indeksiranje. Indeksi omogućuju PostgreSQL-u da unaprijed pripremi dokumente za pretraživanje, smanjujući vrijeme potrebno za izvršenje upita. Na primjer, `ALTER TABLE slots ADD COLUMN fts tsvector GENERATED ALWAYS AS (to_tsvector('english', description || ' ' || title)) STORED;` i `CREATE INDEX slots_fts ON slots USING gin(fts);` omogućuju brzo pretraživanje svih relevantnih podataka. Ovaj pristup osigurava da su podaci uvijek spremni za brzo pretraživanje, poboljšavajući korisničko iskustvo i omogućavajući sportašima da lako pronađu i rezerviraju sportske termine koji najbolje odgovaraju njihovim potrebama.

2.7.10. Geolokacijske usluge

U platformu Sportscape integrirane su lokacijske usluge koje omogućuju automatsko popunjavanje adrese prilikom kreiranja novog termina pomoću Google Maps API-ja. Prilikom unosa nove lokacije, korisnici koriste komponentu `GooglePlacesAutocomplete` iz knjižnice `react-google-places-autocomplete` koja omogućava predlaganje adresa u stvarnom vremenu na temelju unosa korisnika.

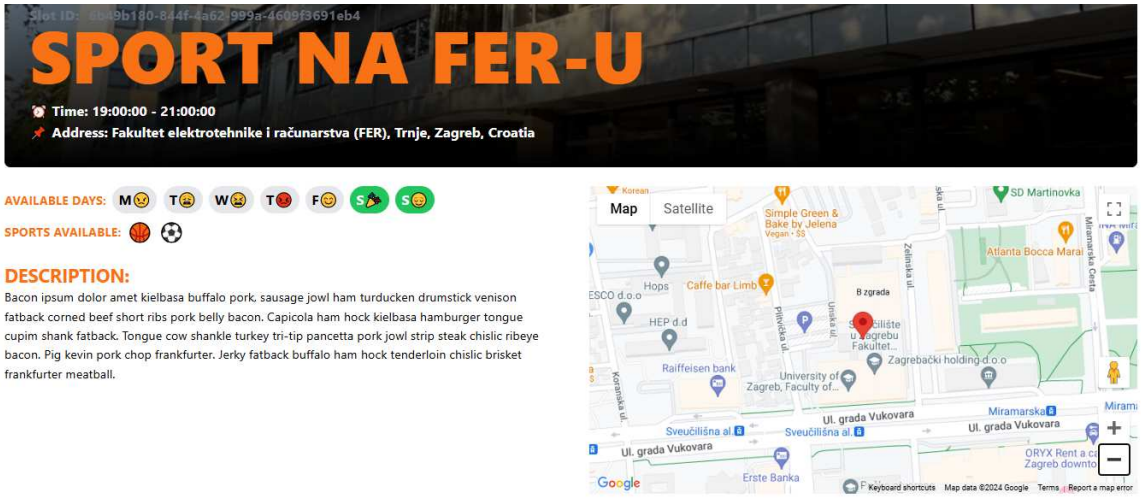
Komponenta radi tako da korisnik unosi dio adrese, a knjižnica zatim koristi Google Maps API za predlaganje mogućih adresa (Slika 2.13).



Slika 2.13 Predlaganje adrese pomoću Google Maps API-ja

Nakon odabira adrese, automatski se dohvaćaju geografske koordinate (širina i dužina) te se te informacije zapisuju u tablicu `slots` unutar baze podataka.

Te koordinate se zatim koriste u drugoj komponenti za prikaz lokacije na Google karti. Komponenta `GoogleMapComponent` koristi `useJsApiLoader` iz knjižnice `@react-google-maps/api` kako bi učitala Google Maps JavaScript API, a zatim prikazuje marker na karti na temelju unesenih koordinata (Slika 2.14).



Slika 2.14 Prikaz adrese na karti

Ovaj pristup pojednostavljuje proces unosa podataka za korisnike, ali i osigurava preciznost i relevantnost podataka o lokaciji, čime se značajno poboljšava korisničko iskustvo na platformi Sportscape.

2.7.11. Postavljanje platforme na Internet

S obzirom da je razvojni okvir Next.js razvijen od strane Vercel-a koji je stvoren kao platforma za postavljanje klijentske strane aplikacija na Internet, logičan je korak bilo upravo tamo postaviti aplikaciju zbog besprijekorne i maksimalno jednostavne integracije.

Platforma je dostupna na adresi https://sportscape.vercel.app.

S druge strane, Supabase je *self-hosted* baza podataka koja se *hosta* na AWS serveru u Njemačkoj. Ova konfiguracija osigurava visoku dostupnost i performanse baze podataka, uz zadržavanje potpune kontrole nad podacima.

Poslužitelj i klijent su ponovno povezani unaprijed napravljenom integracijom Supabase-a u Vercelu, tako da se unaprijedno planiranje u vidu uzimanja u obzir kompleksnosti postavljanja platforme na Internet pri odabiru tehnologija pokazao kao pravi izbor koji je uštedio mnogo vremena.

2.7.12. Kontinuirana integracija i kontinuirana isporuka

Kontinuirana integracija (CI) i kontinuirana isporuka (CD) ključni su procesi u modernom razvoju softvera koji omogućuju automatsko testiranje i implementaciju koda. U platformi Sportscape, CI/CD proces osigurava da se svaka promjena koda automatski testira i, ako je uspješna, odmah implementira na produkcijski server.

Svaki put kada se kod promijeni i pošalje na GitHub (*push*), automatski se pokreće niz skripti za testiranje kako bi se osiguralo da nove promjene ne uvode greške. Iako kod ponekad radi lokalno na računalu, često može pasti na testovima za implementaciju zbog različitih okruženja ili nedosljednosti.

Nakon uspješnog prolaska svih testova, Vercel omogućuje da su promjene odmah vidljive na adresi platforme. Integracija s GitHubom i Vercelom je vrlo jednostavna, omogućujući besprijekoran i automatiziran proces implementacije.

3. Korisničke upute

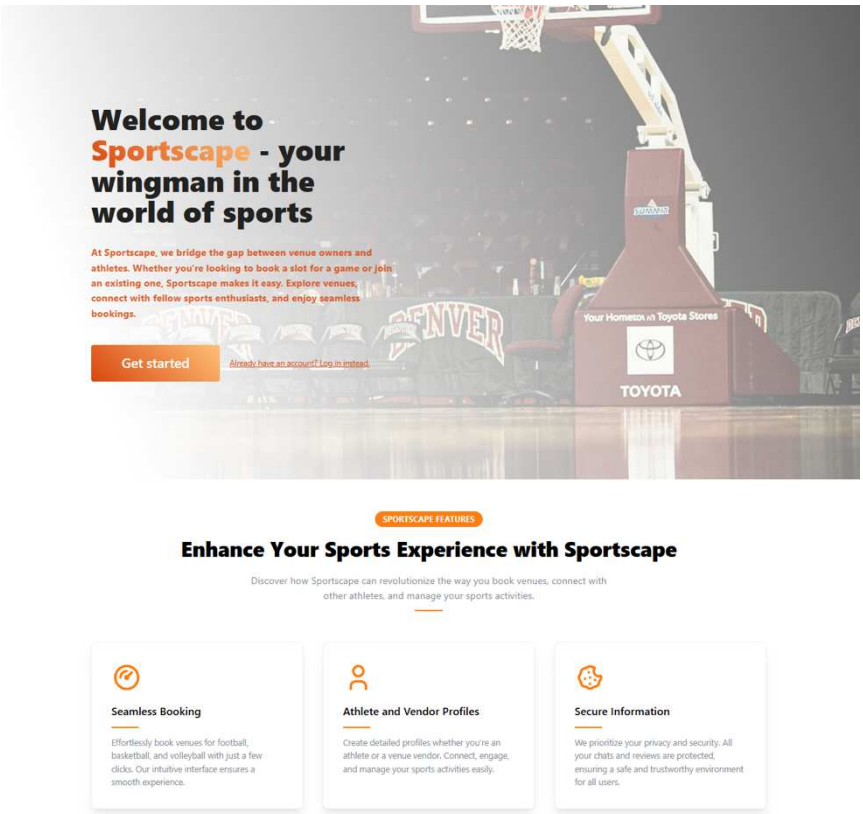
U ovom poglavlju je opisan potpun ciklus korištenja platforme Sportscape, od inicijalne registracije preko objave oglasa o prodaji termina, kupnje termina, priključivanja sportaša, dopisivanja i dogovora sve do konačne igre i kasnijeg recenziranja.

3.1. Naslovna stranica za neregistrirane korisnike

Naslovna stranica napravljena je kako bi neregistriranim korisnicima dala učinkovit uvid u funkcionalnosti platforme Sportscape. To je jedina stranica koju neregistriran korisnik može vidjeti te kao što je spomenuto u poglavlju 2.6., svaki pokušaj učitavanja neke druge stranice bit će preusmjeren na ovu stranicu.

S druge strane, korisnici koji su trenutno prijavljeni u sustav s ove će stranice biti automatski preusmjereni na korisničku ploču njihove uloge (sportaš ili vlasnik prostora).

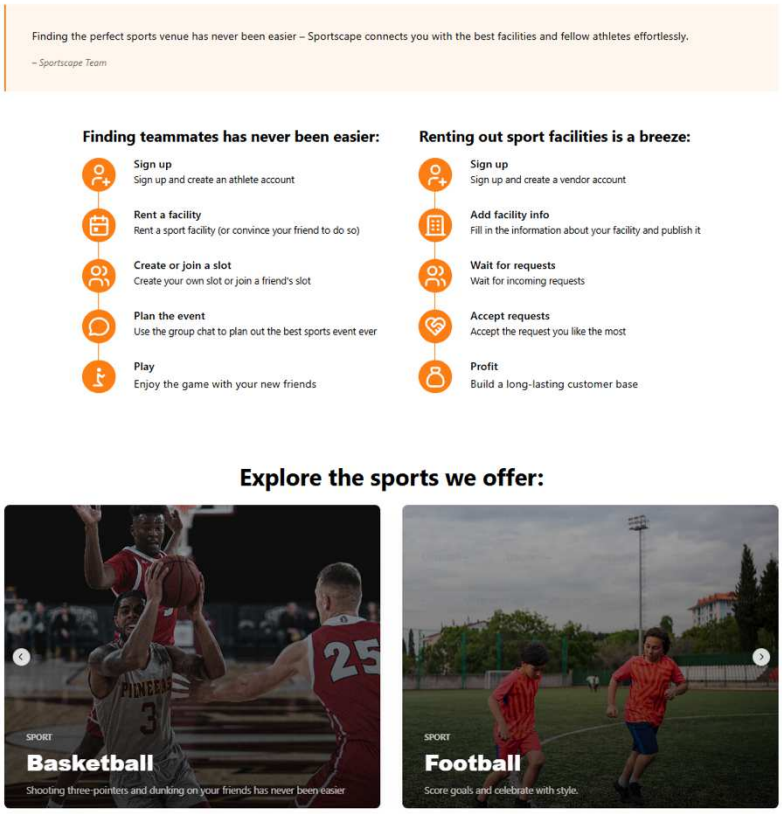
Nakon učitavanja stranice, vidljiva je tzv. *hero* sekcija na kojoj se ističe CTA (engl. *call-to-action*) gumb za registraciju (Slika 3.1).



Slika 3.1. Hero sekcija i kartice s funkcionalnostima

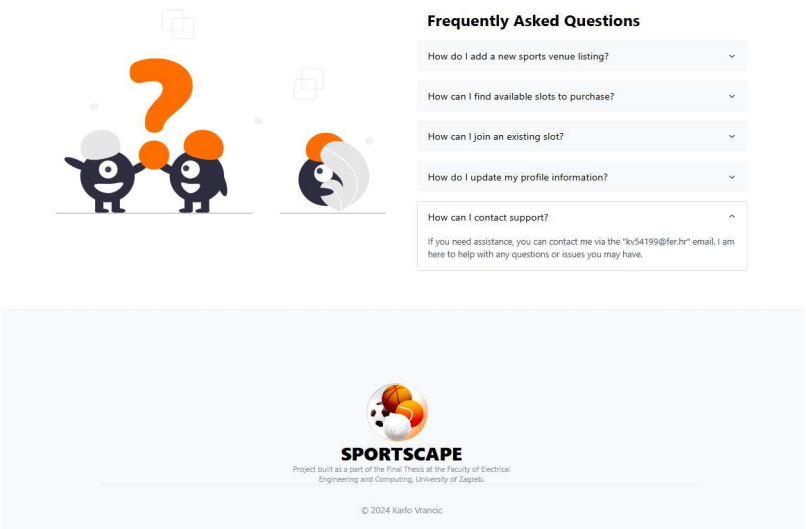
Pored njega je manje upadljiv link za prijavu u platformu, a ispod se nalaze kartice s popisom nekih osnovnih funkcionalnosti koje Sportscape nudi.

Nakon toga napisan je „citat“ koji objašnjava poantu platforme, a zatim je nabrojen slijed radnji za ulogu sportaša i vlasnika prostora koji će biti detaljno objašnjeni upravo u ovom vodiču. Osim toga, napravljene su i kartice s podržanim sportovima na platformi. (Slika 3.2).



Slika 3.2 Citat, osnovni obrazac upotrebe i dostupni sportovi

Na samom dnu naslovne stranice nalazi se odjeljak s često postavljenim pitanjima i podnožje u kojem je rečeno kako je projekt prije svega stvoren u svrhe pisanja ovog završnog rada (Slika 3.3).

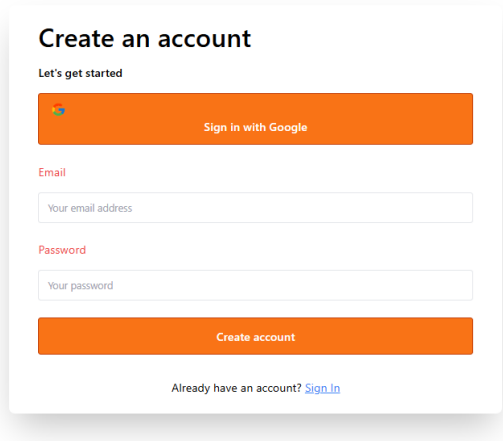


Slika 3.3 Često postavljana pitanja i podnožje

3.2. Registracija i prijava[4]

Nakon pritiska na gumb za registraciju s naslovne strane, otvara se stranica za registraciju koja je već opisana u poglavljima 2.7.7 i 0 u kontekstu mobilnog prikaza i tamne teme.

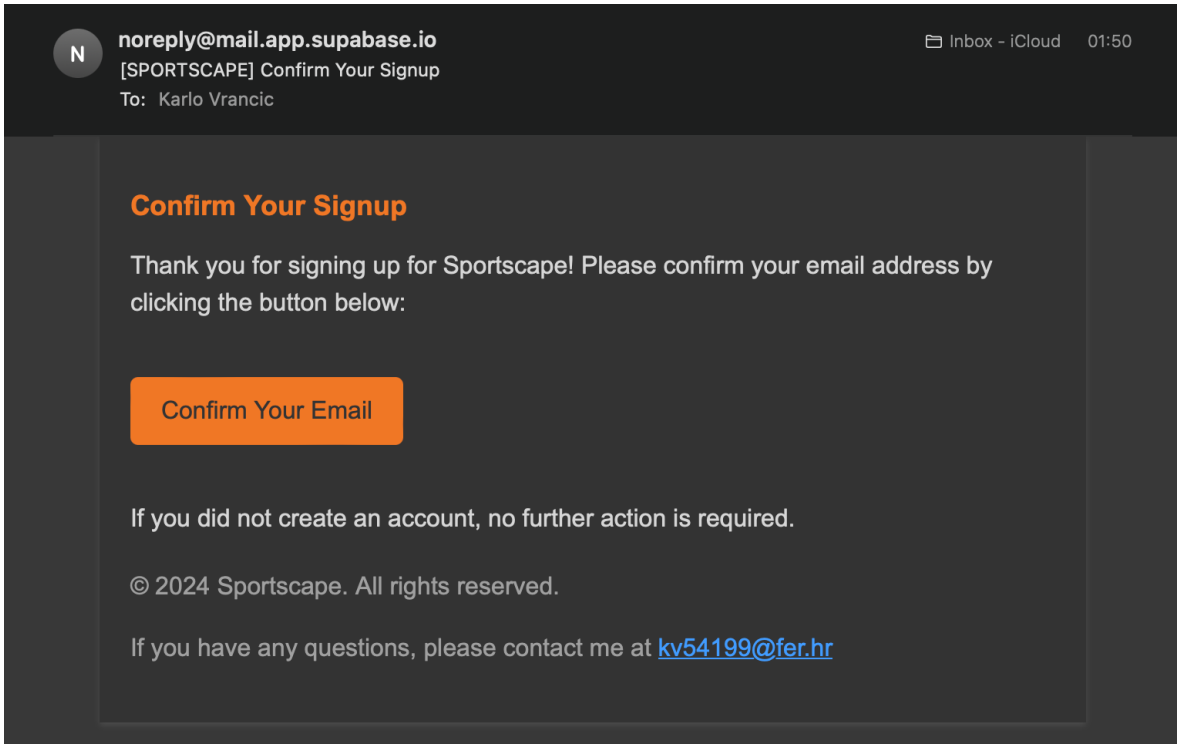
Novi se korisnik može registrirati preko Google računa ili stvoriti novi Sportscape račun (Slika 3.4).



Slika 3.4 Prozor za registraciju

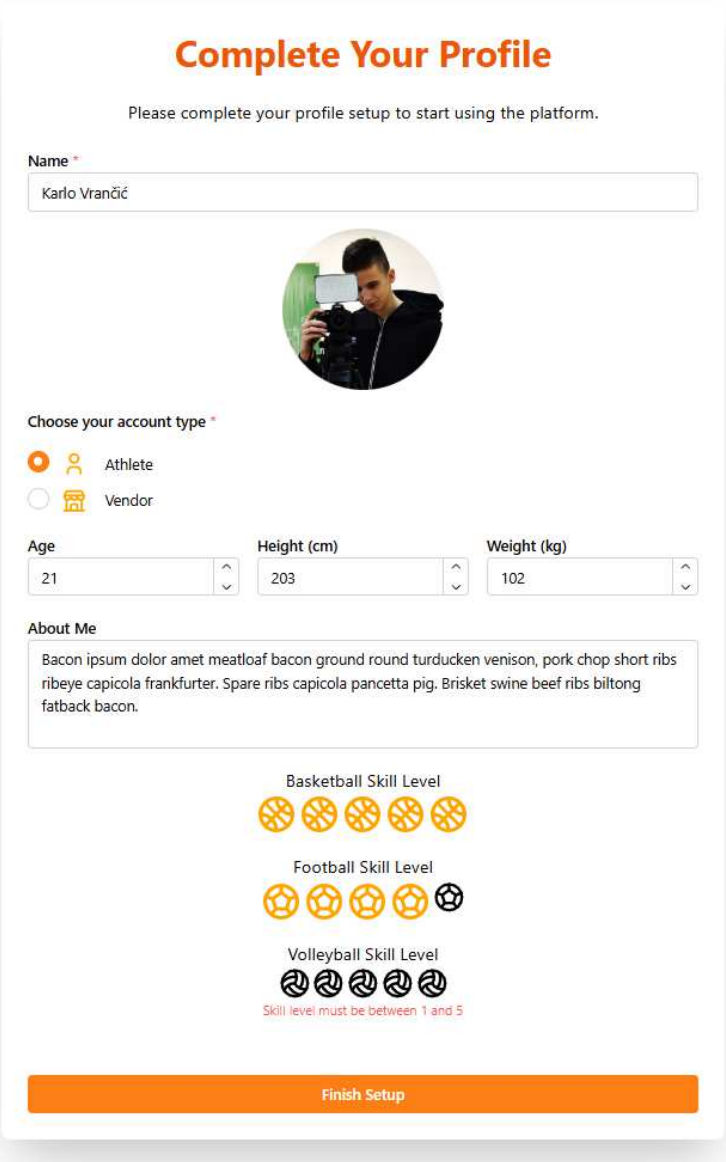
[4] S obzirom da registracija, prijava i obnavljanje lozinke funkcioniraju na sličan način, u nastavku će biti opisana samo registracija.

Nakon upisa adrese elektroničke pošte i zaporke, korisnik svoj račun aktivira klikom na aktivacijsku poruku primljenu na upisanu adresu e-pošte. (Slika 3.5)



Slika 3.5 Aktivacijska poruka

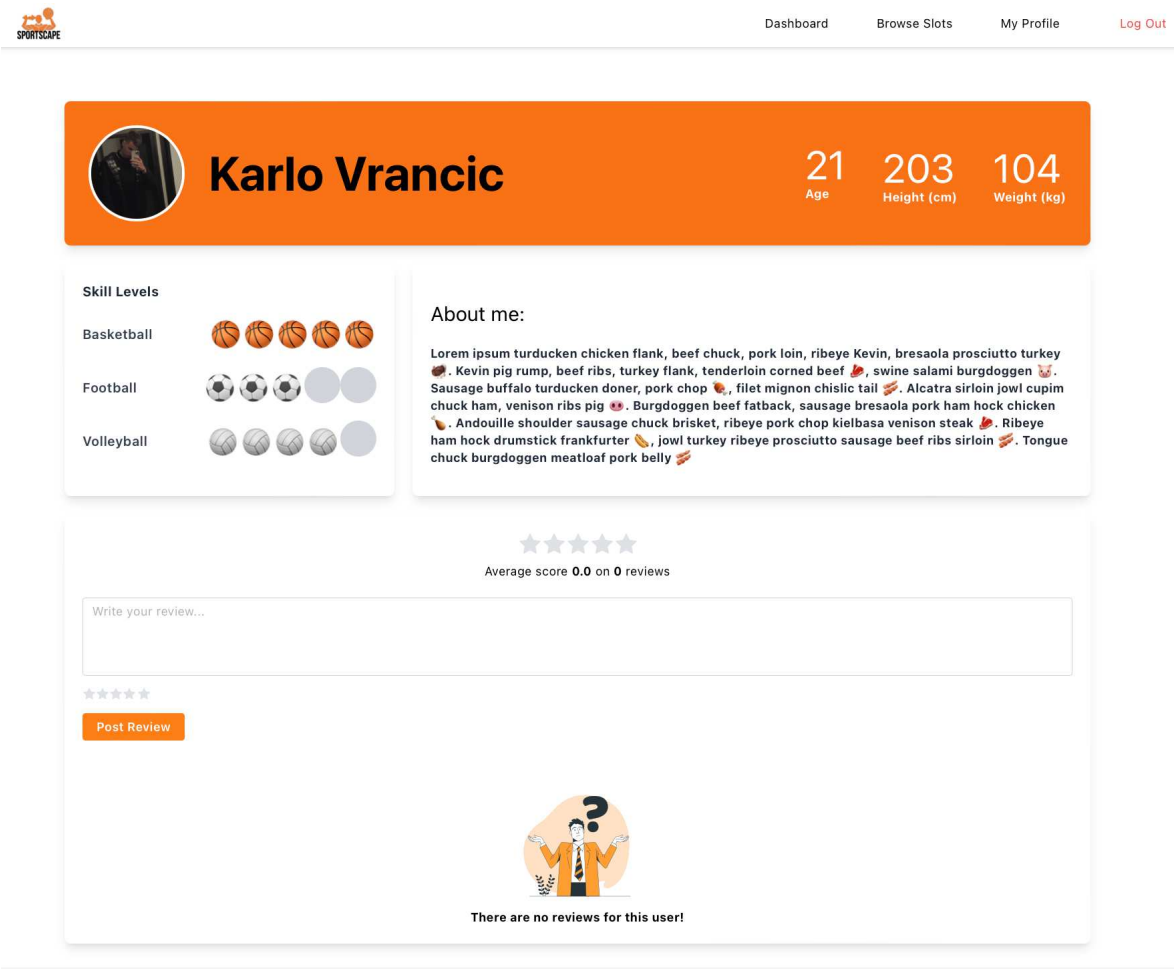
Korisnik je nakon toga doveden na obrazac za dovršavanje profila u kojoj popunjava detaljnije informacije i odabire tip korisničkog računa. Svaki korisnik mora unijeti ime i profilnu fotografiju, a ako se prijavljuje za profil sportaša i dob, visinu, težinu, opis te vještinu u košarci, nogometu i odbojci na skali od 1 do 5 klikom na određeni broj lopti (Slika 3.6).



Slika 3.6. Dovršavanje registracije

Sva polja se validiraju te ukoliko je sve ispravno upisano korisnički je profil spremljen u bazu podataka.

Profili i podaci o istima su vidljivi svim korisnicima kako bi mogli procijeniti žele li prihvatiti uključenje pojedine osobe u njihov termin (Slika 3.7). Dodatno, korisnici se mogu i međusobno recenzirati što je također vidljivo na profilu, ali o tome će biti više riječi u nadolazećim poglavljima.

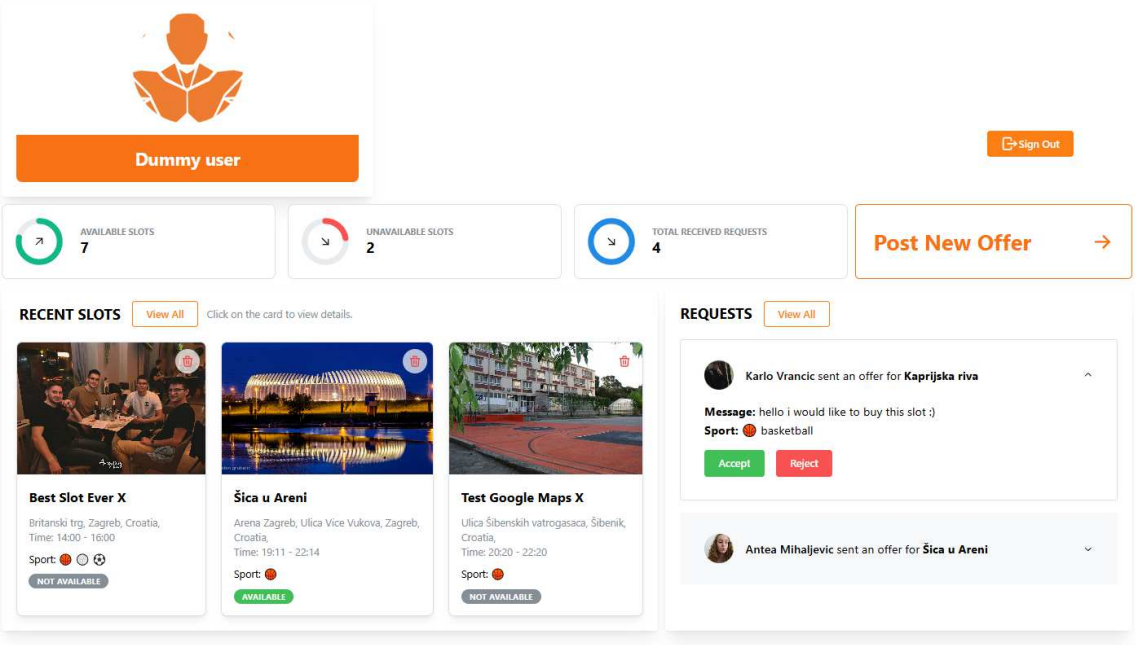


Slika 3.7 Prikaz korisničkog profila sportaša

3.3. Stvaranje novog oglasa za iznajmljivanje termina

Za stvaranje novog oglasa za iznajmljivanje terena potrebno je biti prijavljen u sustav kao vlasnik prostora (u nastavku teksta prodavač).

Prodavač objavljuje novi oglas za iznajmljivanje prostora postavlja preko svoje kontrolne ploče (Slika 3.8). Na nadzornoj ploči prodavač vidi statistike svojih oglasa, posljednje postavljene oglase te zahtjeve za unajmljivanje termina.

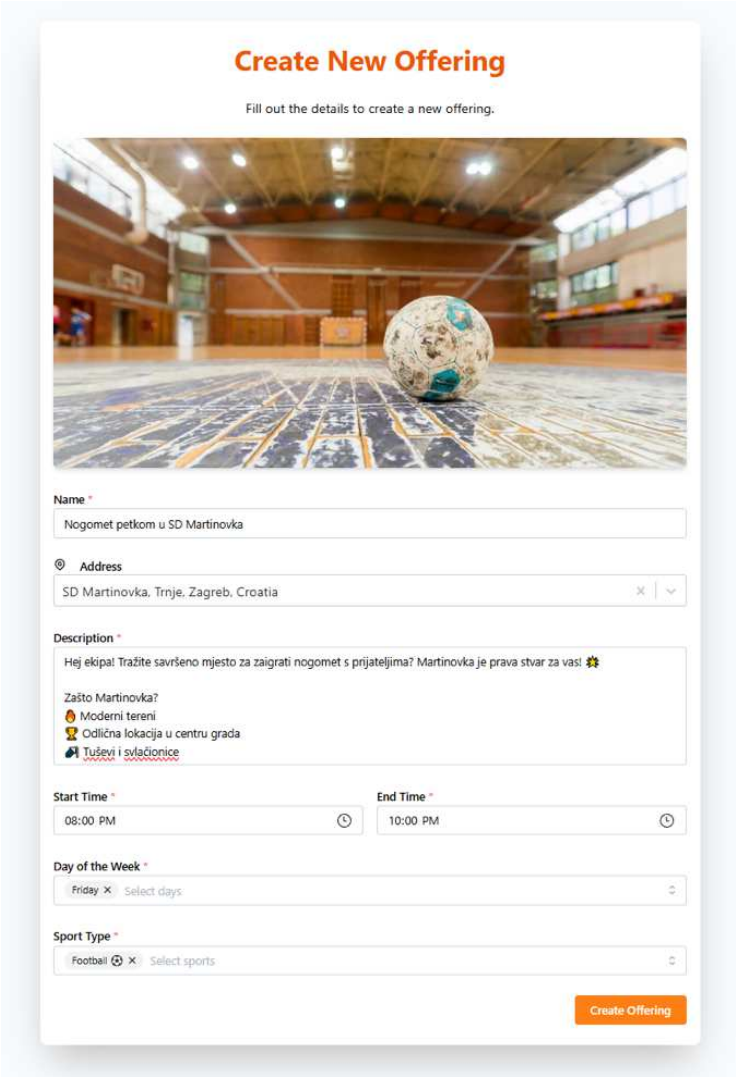


Slika 3.8 Kontrolna ploča prodavača

Klikom na gumb „*Post New Offer*“ otvara se obrazac za stvaranje novog oglasa (Slika 3.9).

Za svaki je termin moguće postaviti sliku, dodati ime, opis, dostupne dane i sportove te adresu. Za polje adrese je implementirano automatsko popunjavanje te je moguće unijeti samo stvarnu adresu kako bi se kasnije koordinate te adrese mogli prikazati na karti.

Nakon što je oglas stvoren, prodavač prima obavijest te je vraćen na svoju kontrolnu ploču. Na njoj se sada nalazi kartica (Slika 3.10) s novostvorenim oglasom koju je klikom moguće proširiti na čitavu stranicu s detaljnim informacijama i popisom svih zahtjeva za kupnju određenog termina. U kutu kartice se nalazi gumb za brisanje koji direktno briše stvoreni termin.

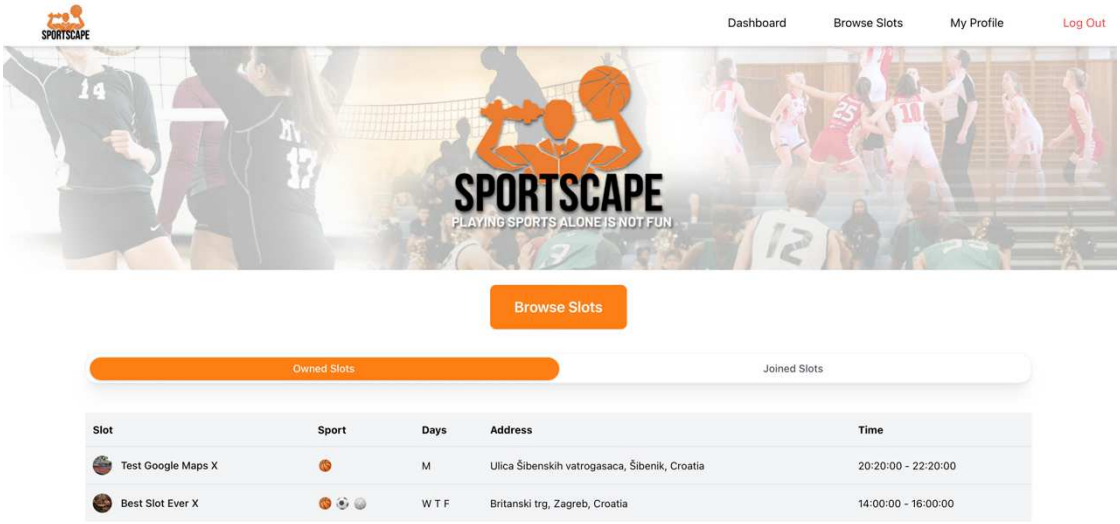


Slika 3.9 Stvaranje novog oglasa za termin

Slika 3.10. Kartica objavljenog termina

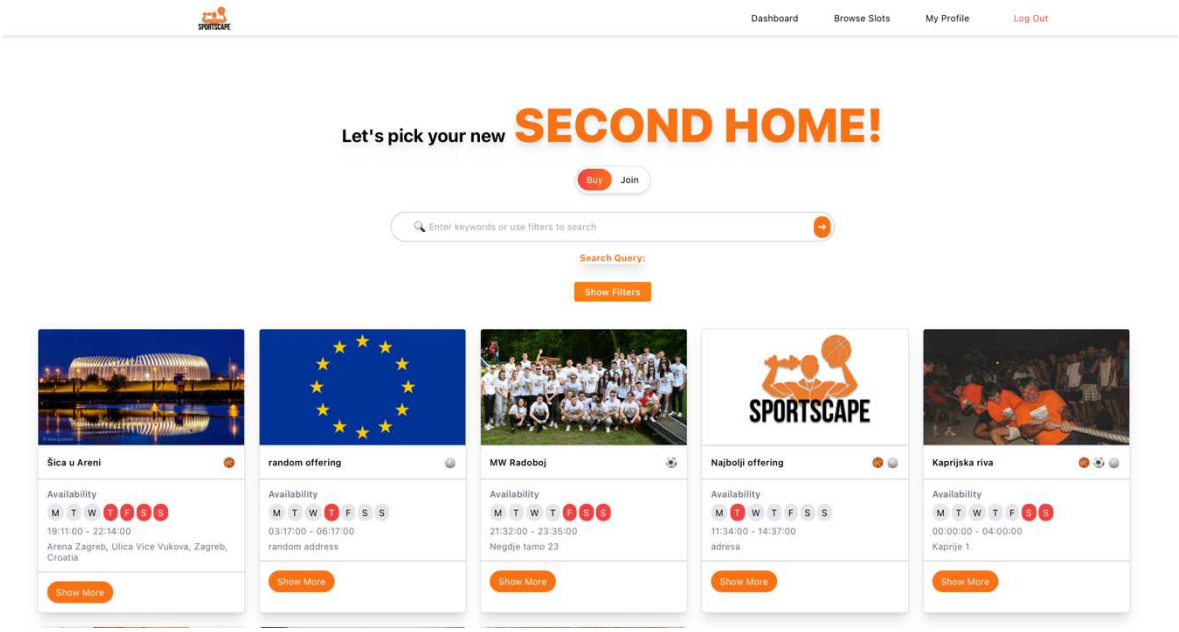
3.4. Kupnja termina

Kupnju termina izvršava sportaš. Nakon prijave u sustav, sportaš je preusmjeren na svoju korisničku ploču (Slika 3.11) s koje može pristupiti terminima koje on unajmljuje, kao i terminima na kojima sudjeluje, a unajmio ih je drugi sportaš.



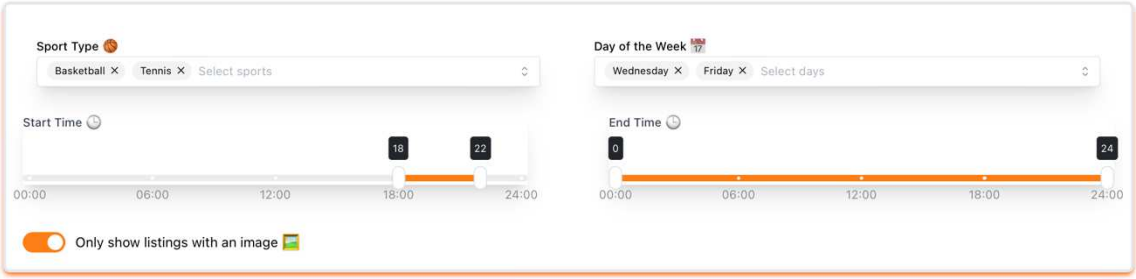
Slika 3.11 Kontrolna ploča sportaša

On s kontrolne ploče ili navigacijske trake može pritisnuti gumb „*Browse Slots*“ koji ga vodi na stranicu za pretraživanje termina (Slika 3.12).



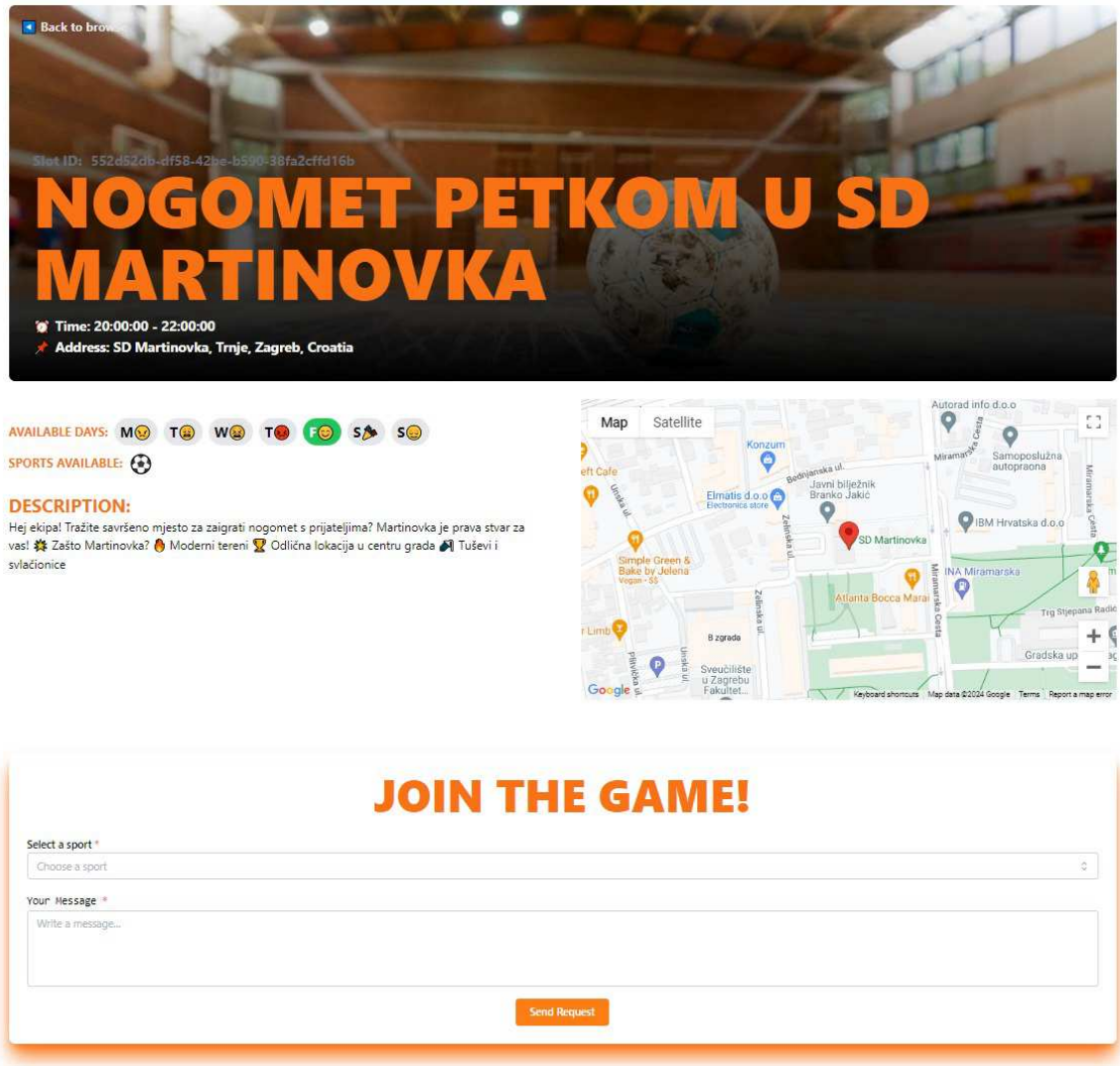
Slika 3.12 Pretraživanje termina

Na stranici za pretraživanje moguće je odabrati traži li sportaš termin koji želi sam unajmiti ili se želi pridružiti već unajmljenom terminu. Dodatno, omogućeno mu je pretraživanje po ključnim riječima (spominjano u poglavlju 2.7.9) ili po brojnim filtrima (Slika 3.13)



Slika 3.13 Filtracija termina

Filtracija se odvija u stvarnom vremenu što korisniku omogućuje efikasno sužavanje izbora pri odabiru idealnog sportskog termina. Nakon odabira sportaš pritišće gumb „*Show more*“ te mu se otvara stranica s detaljnijim podacima o terminu (Slika 3.14).



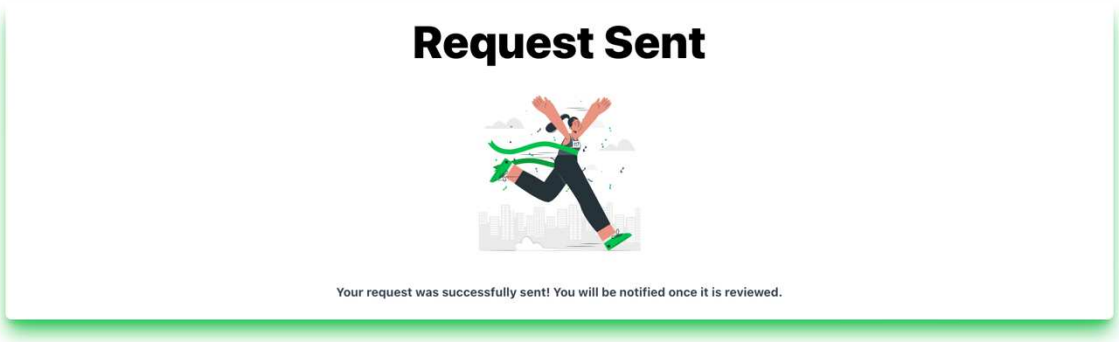
Slika 3.14 Pregled ponude termina za kupnju

Korisnik zatim ispunjava zahtjev za kupnju koji se šalje prodavaču (Slika 3.15)

Slika 3.15 Zahtjev za kupnju termina

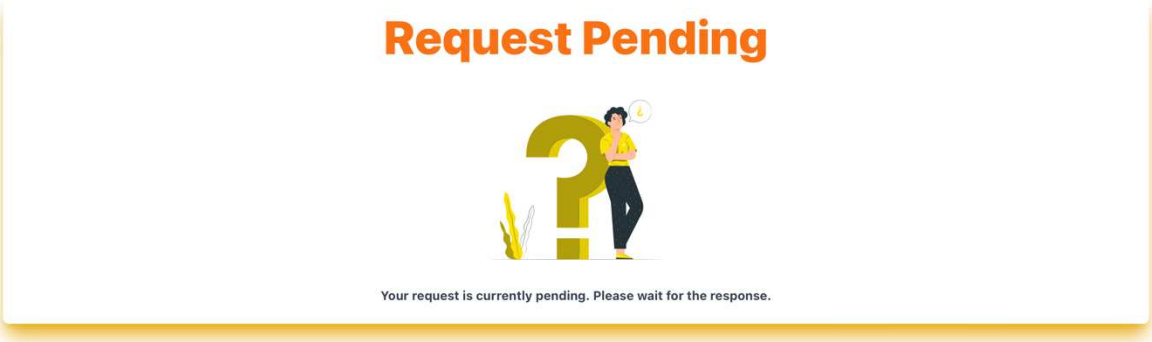
Za to vrijeme, kako korisnik ne bi mogao pretrpavati prodavača s ponudama, umjesto obrasca prikazan je status kupnje koji može biti:

- Uspješno poslana ponuda (prikazano samo nakon inicijalnog slanja) (Slika 3.16)



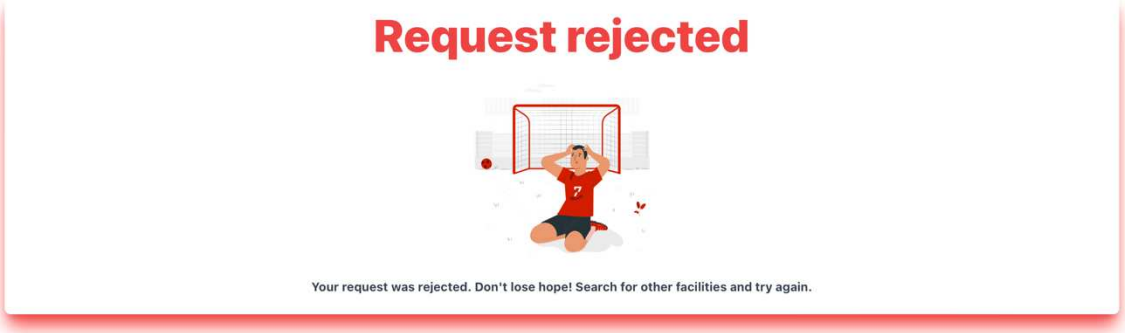
Slika 3.16

- Na čekanju (prodavač još nije niti prihvatio niti odbio ponudu) (Slika 3.17)



Slika 3.17

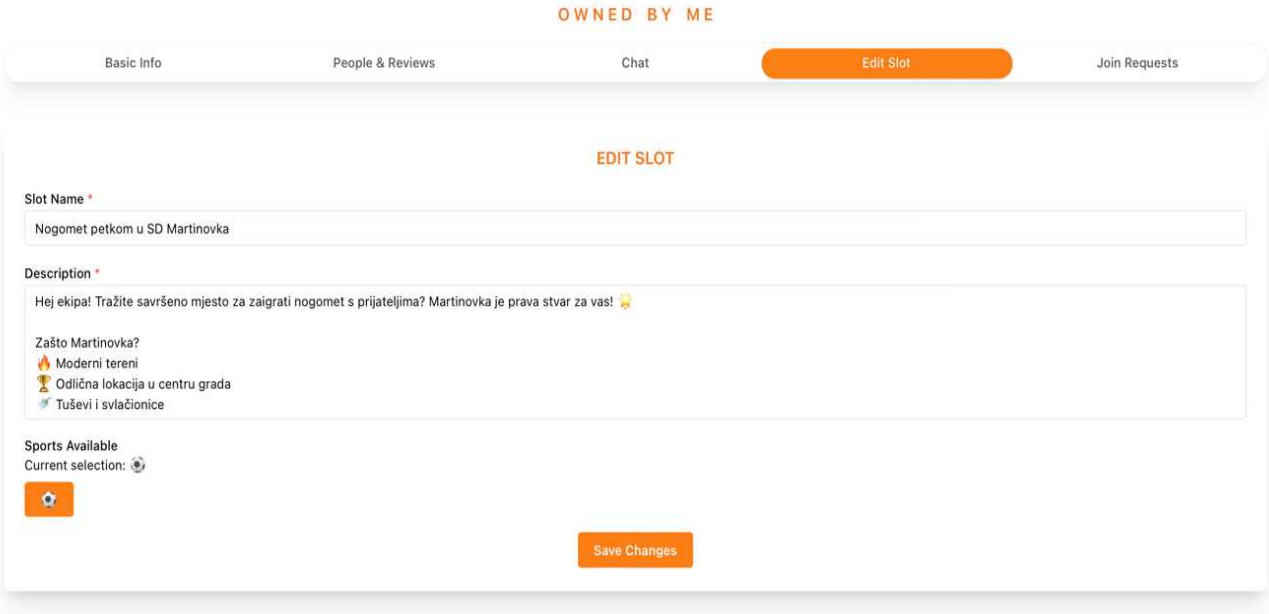
- Ponuda odbijena (Slika 3.18)



Slika 3.18

Konačno, ako je ponuda prihvaćena, aktivira se okidač u bazi podataka (detaljnije objašnjen u poglavlju 2.7.2) koji upisuje sportaša kao novog vlasnika termina i termin iz faze prodaje prelazi u fazu u kojoj mu se moguće priključiti.

Svim sportašima se sada prikazuje i odjeljak s dodatnim informacijama (recenzije[5] i popis članova), a vlasnik i članovi vide prostor za međusobno dopisivanje (engl. *chat* koji će biti opisan u poglavlju 3.6). Dodatno, sportaši koji nisu članovi vide obrazac sa zahtjevom za priključenje, a vlasnik vidi poglavlja za prihvaćanje zahtjeva i uređenje informacija o terminu kako bi mogao dodati vlastiti opis, promijeniti ime i odabrati jedan sport između sportova koji su prigodni za termin (Slika 3.19)



Slika 3.19 Uređivanje termina

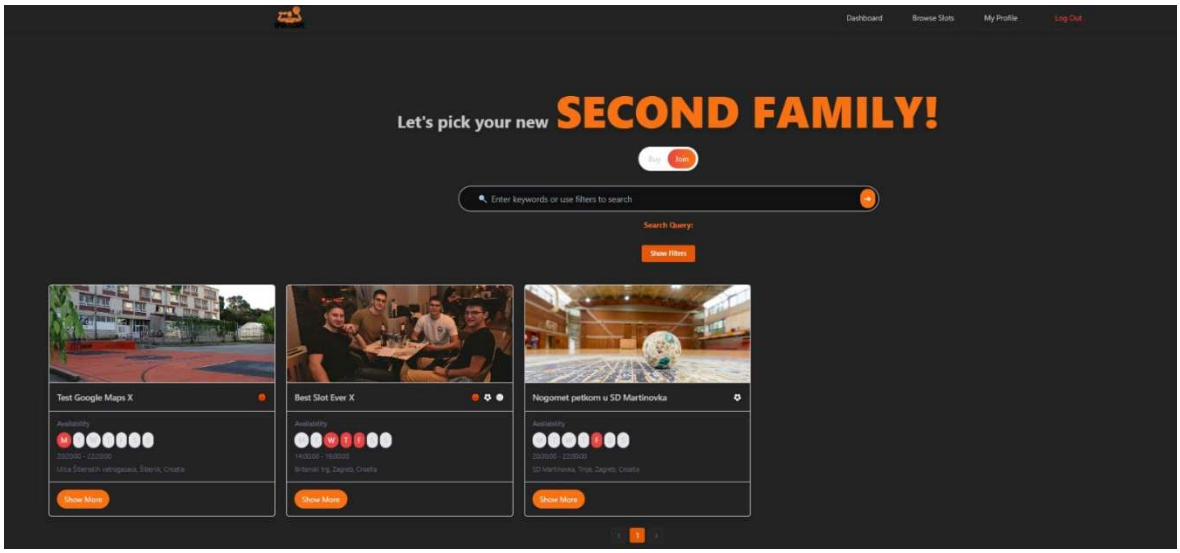
3.5. Priključenje na kupljeni termin

Nakon što je jedan sportaš kupio termin, ostali sportaši se priključuju istome.

Kako bi nadolazeći tijek događaja bio pregledniji, sportaš koji je od prodavača unajmio termin i dalje koristi svijetlu temu, dok će sportaš koji se priključuje terminu koristiti tamni način.

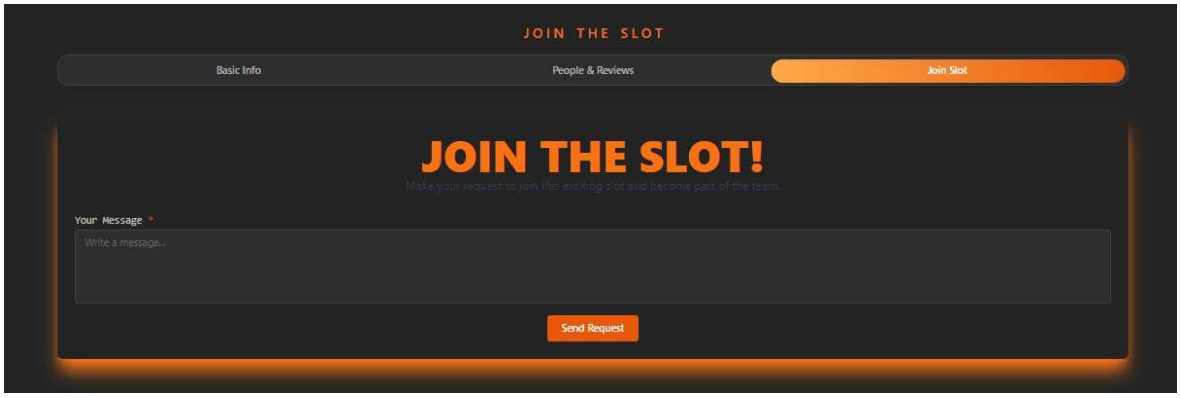
[5] Vlasnik i korisnici koji nisu priključeni terminu mogu samo pregledavati recenzije, dok članovi termina mogu i pisati iste

Taj novi sportaš ulazi u prozor za pretraživanje termina i na gumbu za segmentiranu kontrolu odabire „Join“ kojim mu se mijenja naslovni tekst (usporediti sa Slika 3.12) i filtri, a prikazuju se samo one kartice kod kojih je termin već kupljen i čeka druge sportaše na priključenje (Slika 3.20).



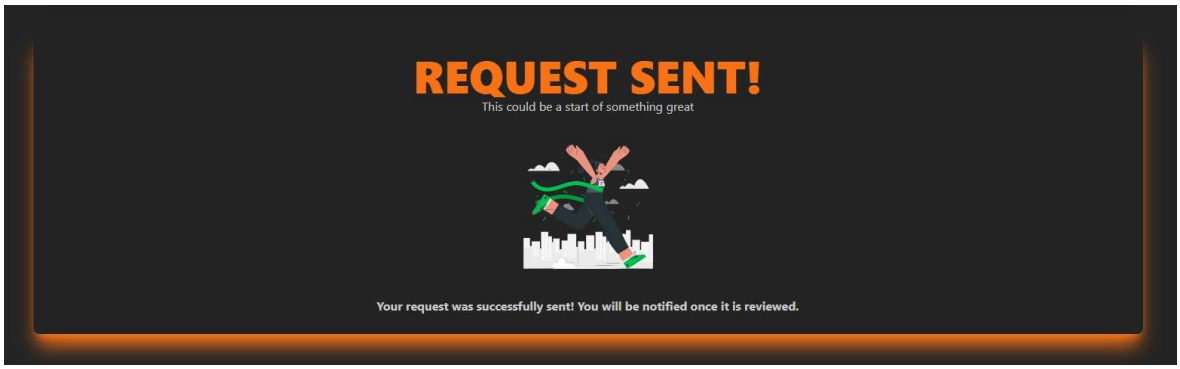
Slika 3.20 Prikaz pretraživanja termina za priključenje

Klikom na odabrani termin, otvara se prikaz za nepriključene sportaše koji između ostalog mogu odabrati opciju ispunjavanja obrasca za priključenje terminu (Slika 3.21).



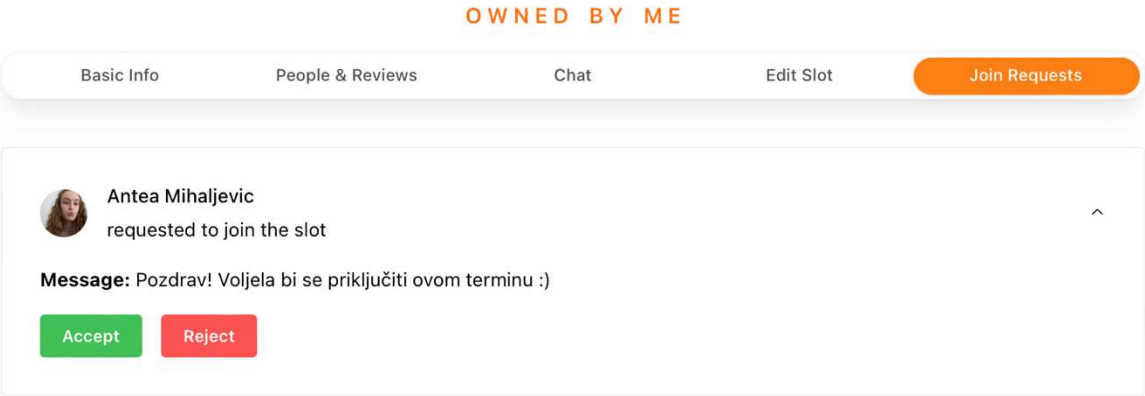
Slika 3.21 Obrazac za priključenje

Ponašanje obrasca je slično prethodno opisanom obrascu za kupnju termina, pa potvrda ispunjavanja (Slika 3.22) funkcionira analogno kao Slika 3.16, a isto vrijedi i za čekanje te odbijanje zahtjeva.



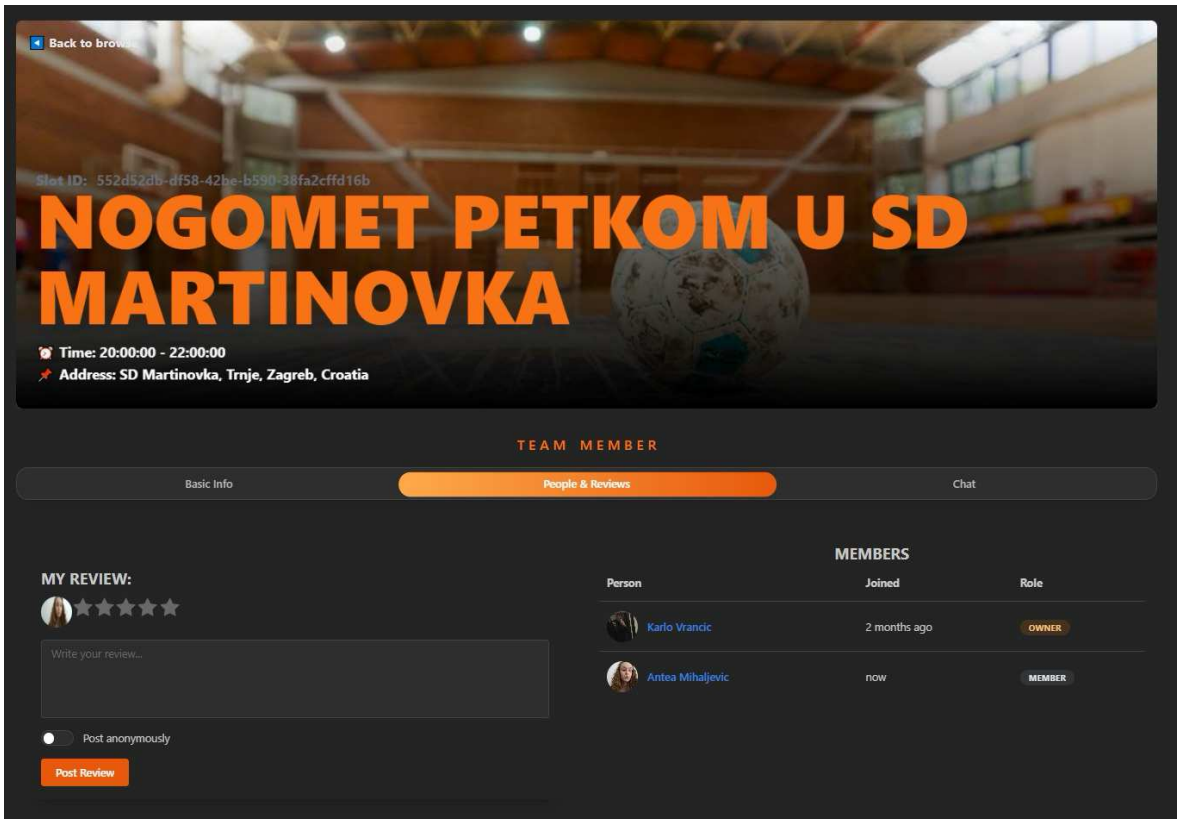
Slika 3.22 Potvrda o slanju zahtjeva za priključenje

Vlasnik termina prima zahtjev za priključenje te odabire hoće li ga prihvatiti (Slika 3.23)



Slika 3.23 Zahtjev za priključenje

Nakon prihvaćanja termina aktivira se okidač u bazi podataka kojim sportaš postaje član termina te mu se sada otkrivaju dijelovi za dopisivanje i pregled članova. Također, omogućeno mu je pisanje recenzija koje je do sada mogao samo čitati (Slika 3.24).

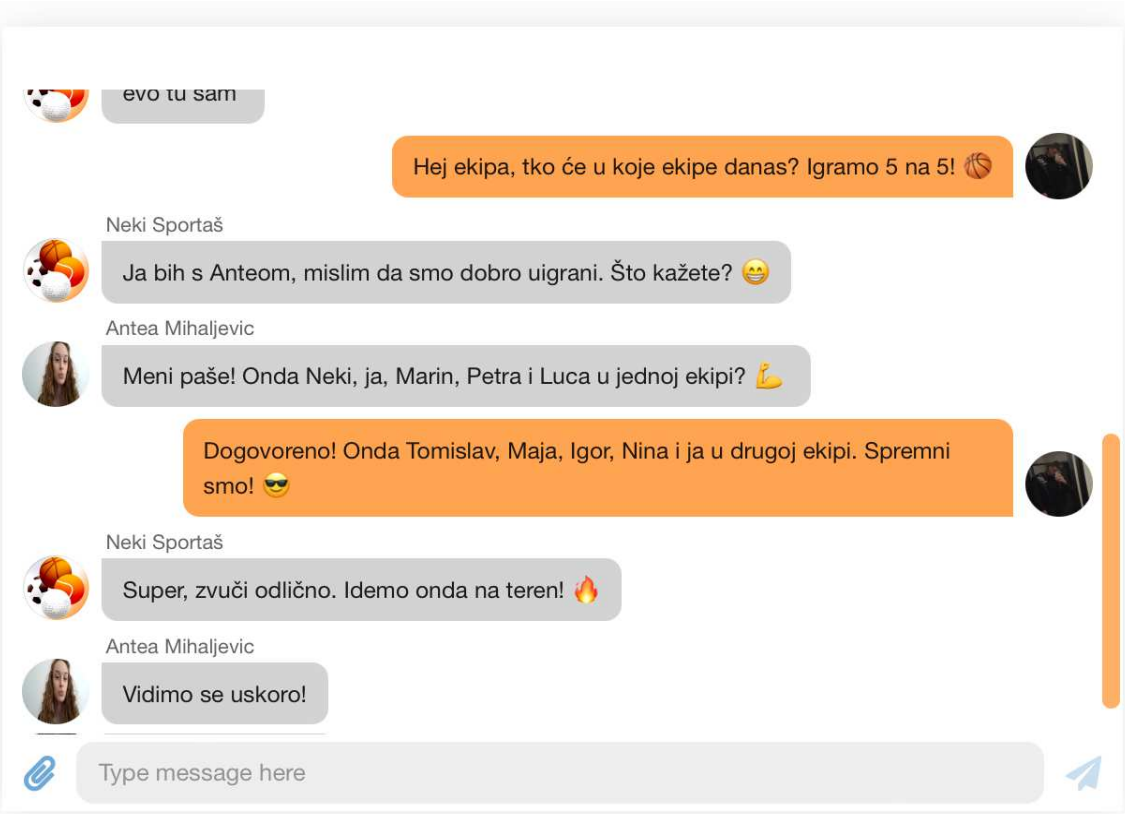


Slika 3.24 Prikaz termina iz uloge člana termina

3.6. Dopisivanje u stvarnom vremenu među članovima

Kada se okupi dovoljno ljudi za održavanje termina, vrijeme je za konkretne dogovore. Članovi to rade odabirom na „*Chat*“ u segmentiranoj kontroli stranice za termin.

Time se prikazuje odjeljak u kojem sudionici vide međusobne poruke što im omogućuje detaljne dogovore za svaki pojedinačni termin (Slika 3.25).



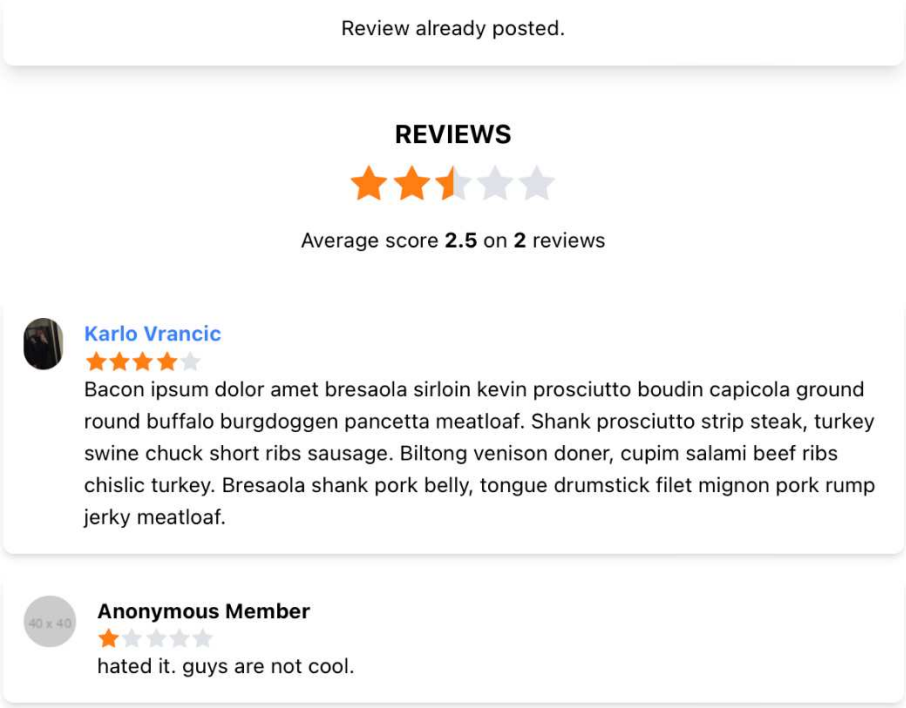
Slika 3.25 Prikaz prozora za dopisivanje

3.7. Recenzije

3.7.1. Recenziranje termina

Nakon odigranih termina, svaki član ima pravo napisati recenziju za termin. Zbog privatnosti, implementirana je i funkcionalnost anonimnih recenzija.

Sportaš piše recenziju u prozor za pisanje, a nakon ispunjavanja recenzije mu je zabranjeno dodavanje nove recenzije (Slika 3.26).



Slika 3.26 Prikaz odjeljka s recenzijama. Primijetiti gornji dio slike.

3.7.2. Recenziranje profila

Do korisničkog profila osim pretragom moguće je doći i iz tablice članova termina (Slika 3.24) ili klikom na ime u njihovoj recenziji nekog termina.

Osim osnovnih informacija o korisniku, na profilu su vidljive recenzije o tom korisniku te svaki sportaš s kojim dijeli termin može napisati svoju recenziju (Slika 3.27).

Slika 3.27 Recenzije na korisničkom profilu

Zaključak

Tehnologija je stvorena kako bi čovjeku uljepšala život. Međutim, zbog svih stvari koje je omogućila pojedincu, on više nema potrebu za aktivnim životom što dugoročno djeluje detrimentalno na fizičko i mentalno zdravlje.

Ideja iza Sportscapea kao platforme jest iskoristiti tehnologiju u svrhu povratka običnih ljudi ka fizičkoj aktivnosti i sportu na moderan način. Zamišljen je kao hibrid oglasnika za sportske terene koji ne postoji na domaćem tržištu kao specijalizirana aplikacija i društvene mreže za sportaše rekreativce, koja također ne postoji.

Dakle, ovim je radom implementiran uspješan sustav za oglašavanje sportskih termina i njihovo unajmljivanje, ali i još bitnije – sustav koji omogućuje običnim ljudima da na transparentan, brz i učinkovit način pronađu sportske kolege i vrate se davno zaboravljenim sportskim radostima. Sportscape i u trenutnoj fazi efektivno zamjenjuje niz drugih platformi i servisa koji su bili potrebni da bi se izveo osnovni obrazac korištenja opisan u ovom radu.

To znači da platformu već u ovom obliku možemo promatrati kao zaokruženu cjelinu s funkcionalnošću koja nema ekvivalent u vidu svjetski poznate javno dostupne platforme. Međutim, evidentno je da i dalje postoji prostor za daljnji razvoj. U slučaju korištenja aplikacije u stvarnome životu, jasno je da su cijene i plaćanja komponente osnovnog obrasca uporabe koje u ovome radu kao akademskom projektu nisu implementirane. Dodatno, same funkcionalnosti društvene mreže bi se mogle dodatno proširiti uvođenjem mehanika poput praćenja, objavljivanja sportskih postignuća i sl.

Dodatno, platforma bi mogla sama organizirati razne sportske izazove, natjecanja i turnire koji bi se odvijali kako virtualno, tako i fizički. Na taj bi način Sportscape postao temeljni poligon za razvoj rekreativnog sporta, ali postoji i potencijal za širenje na profesionalno područje. Kao konačni cilj Sportscape možemo opisati kao LinkedIn za sportske aktivnosti.

Literatura

[1] Woessner Mary N. , Tacey Alexander , Levinger-Limor Ariella , Parker Alexandra G. , Levinger Pazit , Levinger Itamar, *The Evolution of Technology and Physical Inactivity: The Good, the Bad, and the Way Forward,* Frontiers in Public Health, 9, 2011.
Poveznica: www.frontiersin.org/journals/public-health/articles/10.3389

[2] Hanna Fahad , You Emily , El-Sherif Mohamed *Editorial: The impact of sedentary behavior and virtual lifestyle on physical and mental wellbeing: social distancing from healthy living,* Frontiers in Public Health, 11, 2023.
Poveznica: /www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2023.1265814/full

[3] Auth | Supabase Docs; pristupljeno 1. lipnja 2024.

[4] Features | Supabase Docs; pristupljeno 1. lipnja 2024.

[5] Architecture | Supabase Docs; pristupljeno 1. lipnja 2024.

[6] Getting started with Next.js | Supabase Docs; pristupljeno 3. lipnja 2024.

[7] Architecture and Technology Stack of Supabase (workingsoftware.dev); pristupljeno 1. lipnja 2024.

[8] *Tips for Row Level Security in Postgres and Supabase | Poveznica:*

https://maxlynch.com/2023/11/04/tips-for-row-level-security-rls-in-postgres-and-supabase/; pristupljeno 3. lipnja 2024.

[9] Row level Security | Supabase Docs; pristupljeno 3. lipnja 2024.

[10] Password-based Auth | Supabase Docs; pristupljeno 2. lipnja 2024.

[11] OWASP Top10 | https://owasp.org/www-project-top-ten; pristupljeno 4. lipnja 2024.

[12] The Dark Mode Trend Explained: History, Benefits, and Evolution | Axigen; pristupljeno 2. lipnja 2024.

[13] The Anniversary Update's most exciting features: Windows 10 users weigh in | PCWorld; pristupljeno 6. lipnja 2024.

[14] macOS Mojave: Dark Mode, Stacks, & More (macrumors.com); pristupljeno 6. lipnja 2024.

[15] Sethi T., Ziat M. *Dark mode vogue: Do light-on-dark displays have measurable benefits to users?*, Ergonomics Volume 66, 2023
Poveznica: www.tandfonline.com/doi/full/10.1080/00140139.2022.2160879

[16] Digital Devices and Your Eyes - American Academy of Ophthalmology (aao.org); pristupljeno 8. lipnja 2024.

[17] Kangsoo K., Erickson A., Lambert A., Bruder G., Welch G.F., *Effects of Dark Mode on Visual Fatigue and Acuity in Optical See-Through Head-Mounted Displays*, SUI '19: Symposium on Spatial User Interaction, 2019.
dl.acm.org/doi/10.1145/3357251.3357584

Sažetak

Završni rad prati razvoj web aplikacije Sportscape koja kroz hibrid oglasnika i društvene mreže za sportaše ima cilj stvoriti univerzalnu platformu za popularizaciju rekreativnog sporta i tako mogućnosti moderne tehnologije okrenuti u korist malog čovjeka i njegova zdravlja.

Sportscape je stvoren za dvije vrste korisnika: vlasnike sportskih prostora koji žele iznajmljivati terene i sportaše koji iste terene unajmljuju te se zatim povezuju s ostalim sportašima sličnih profila.

Dakle, vlasnik može objaviti oglas, pratiti statistike svojih oglasa i odgovarati na ponude za iste. S druge strane, sportaši mogu pretraživati oglase, slati zahtjeve za iznajmljivanje termina ili pak za pridruživanje već unajmljenim terminima. Svaki član termina istome može napisati recenziju, a sportaši mogu i međusobno gledati svoje profile i recenzirati se. Dodatno, svaki termin ima vlastiti razgovor u stvarnom vremenu (engl. *chat*) preko kojeg sportaši mogu dogovarati detalje svoje buduće sportske aktivnosti.

Platforma je izrađena koristeći PostgreSQL, Supabase, Next.js, Tailwind CSS i Mantine.

Ključne riječi: PostgreSQL, Supabase, Next.js, TailwindCSS, web-aplikacija, platforma, oglasnik, društvena mreža, sport, recenzije, dopisivanje.

Summary

The final thesis follows the development of the web application Sportscape, which aims to create a universal platform for popularising recreational sports through a hybrid of a classifieds site and a social network for athletes. The goal is to leverage modern technology to benefit the average person and their health.

Sportscape is designed for two types of users: sports facility owners who want to rent out their spaces, and athletes who rent these spaces and then connect with other athletes with similar interests.

Thus, an owner can post an advertisement, track the statistics of their ads, and respond to offers. On the other hand, athletes can browse ads, send requests to rent slots or join already booked slots. Each slot member can write a review for that specific slot, and athletes can also view each other's profiles and review each other. Additionally, each slot has its real-time chat through which athletes can arrange the details of their future sports activities.

The platform is built using PostgreSQL, Supabase, Next.js, Tailwind CSS, and Mantine.

Keywords: PostgreSQL, Supabase, Next.js, Tailwind CSS, web application, platform, classifieds, social network, sports, reviews, chat.