

Web aplikacija za vođenje navijačkih skupina u nogometu

Jančec, Nikola

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:620379>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-28**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Nikola Jančec 0313023045

Web aplikacija za vođenje navijačkih skupina u nogometu
Završni rad

Čakovec, veljača 2024.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Nikola Jančec 0313023045

Web aplikacija za vođenje navijačkih skupina u nogometu
Web application for leading football fan groups

Završni rad

Mentorica:

dr. sc. Sanja Brekalo, prof. struč. stud.

Čakovec, veljača 2024.



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

PRIJAVA TEME I OBRANE ZAVRŠNOG/DIPLOMSKOG RADA

Stručni prijediplomski studij:

Računarstvo Održivi razvoj Menadžment turizma i sporta

Stručni diplomski studij Menadžment turizma i sporta:

Pristupnik: Nikola Janžec, JMBAG: 0313023045
(ime i prezime)

Kolegij: izrada web sadržaja
(na kojem se piše rad)

Mentor: Sanja Brekalo, prof. struč. stud.
(ime i prezime, zvanje)

Naslov rada: Web aplikacija za vođenje navijačkih skupina u nogometu

Naslov rada na engleskom jeziku: Web application for the management of football fan groups

- Članovi povjerenstva: 1. Bruno Trstenjak, v. pred., predsjednik
(ime i prezime, zvanje)
2. Jurica Trstenjak, v. pred., član
(ime i prezime, zvanje)
3. Sanja Brekalo, prof. struč. stud., mentor
(ime i prezime, zvanje)
4. Marija Mižančuk, v. pred., zamjenski član
(ime i prezime, zvanje)

Broj zadatka: 2021-RAC-R-71

Kratki opis zadatka: _____

Tema rada je izrada u JavaScriptu (React library) web aplikaciju za upravljanje navijačkim skupinama. Aplikacija

poim dioničkih dječaka upravljanja kolicima i njihovim pravima počinje do za upis i vođenje skupina. Imajući sustav je CMS sustav koji olakšava

pratiće i rad navijačkih skupina

Datum: 28.02.2024.

Potpis mentora: Sanja Brekalo 

SAŽETAK

Ovaj završni rad ima za cilj istražiti i razviti aplikaciju za vođenje navijačkih skupina koja je kreirana u React biblioteci i koja predstavlja jedan od suvremenih razvojnih okvira za kreiranje korisničkih sučelja, odnosno aplikacija. Aplikacija za vođenje navijačkih skupina usredotočuje se na olakšavanje procesa organizacije, komunikacije i praćenja relevantnih informacija unutar navijačkih skupina. Metodologija rada kombinira teorijski i praktični pristup kako bi se dublje istražila problematika. Prikupljeno je teorijsko znanje o programiranju, programskim jezicima, te o biblioteci React putem izvora dostupnih na internetu. Na temelju informacija prikupljenih istraživanjem, identificirani su izazovi vezani uz vođenje navijačkih skupina, postavljena je svrha rada koja uključuje razvoj aplikacije što olakšava organizaciju događanja i unaprjeđuje komunikaciju unutar članova pojedine navijačke skupine. Ciljevi rada obuhvaćaju stjecanje praktičnoga znanja u primjeni React tehnologije, odnosno primjenu teorijskoga znanja u praktičnom okruženju te izradu aplikacije. U praktičnom dijelu rada, izrađena je aplikacija koja primjenjuje koncepte naučene tijekom teorijskoga istraživanja. Aplikacija sadrži organizaciju događanja, omogućava interakciju među članovima te se koristi kao primjer primjene React tehnologije. U praktičnom zadatku razvija se aplikacija za vođenje navijačkih skupina i istražuju se potencijalna rješenja za optimizaciju komunikacije i organizacije unutar tih skupina. Fokus rada također leži na izradi dokumentacije aplikacije, koja će služiti kao osnova za razumijevanje arhitekture, funkcionalnosti i tehničkih aspekata aplikacije. Dokumentacija je korisna za budući razvoj, održavanje i unaprjeđenje aplikacije te pruža smjernice za buduće izmjene u vođenju navijačkih skupina.

Ključne riječi: *React, web aplikacija, baza podataka, vođenje navijačkih skupina*

ABSTRACT

The aim of this final paper is to research and develop an application for managing fan groups, which was created in the React library, and which represents one of the modern development frameworks for creating user interfaces, or applications. Fan group management application focuses on facilitating the process of organization, communication and tracking of relevant information within fan groups. The methodology of the work combines a theoretical and practical approach in order to explore the issue more deeply. Theoretical knowledge about programming, programming languages, and the React library was collected through sources available on the Internet. Based on the information collected through research, challenges related to the management of fan groups were identified, the purpose of the work was set, which includes the development of an application that facilitates the organization of events and improves communication within the members of individual fan groups. The goals of the work include the acquisition of practical knowledge in the application of React technology, that is, the application of theoretical knowledge in a practical environment and the creation of an application. Through the practical part of the work, an application was created that applies the concepts learned during the theoretical research. The application contains the organization of events, enables interaction between members and is used as an example of the application of React technology. In addition, the paper combines theory and practice to provide a deeper understanding of the development of applications based on React technology. Through a practical task, an application for managing fan groups is developed and potential solutions for optimizing communication and organization within these groups are explored. The focus of the work also lies in the creation of application documentation, which will serve as a basis for understanding the architecture, functionality and technical aspects of the application. The documentation is useful for the future development, maintenance and improvement of the application and provides guidance for future changes in the management of fan groups.

Keywords: React, web application, database, fan group manag, final work

Sadržaj

SAŽETAK	3
ABSTRACT	4
1. UVOD.....	1
1. PROGRAMSKI ALATI.....	2
3.1. JavaScript.....	3
3.2. React	4
3.2.1. Značajke Reacta.....	6
3.2.2. Komponente u Reactu	7
3.2.3. Prednosti Reacta	8
3.2.4. Nedostatci React-a.....	9
3.3. Pokretanje novog React projekta	10
4. OPIS APLIKACIJE.....	11
4.1. Primjena aplikacije.....	18
4.2. Testiranje aplikacije	22
4.3. Planirani izvještaji.....	23
4.4. Puštanje aplikacije u rad	23
5. ZAKLJUČAK.....	25
LITERATURA.....	27
POPIS SLIKA I GRAFOVA.....	29

1. UVOD

Suvremeno digitalno okruženje pruža brojne mogućnosti za stvaranje inovativnih rješenja koja olakšavaju organizaciju i koordinaciju različitih aktivnosti. Navijačke skupine su dio sportskoga iskustva, a na sportskim događanjima pridonose istom. Ovaj završni rad ima za cilj razvoj aplikacije za navijače kako bi se unaprijedilo navijanje i okupljanje navijača u virtualnom svijetu. Aplikacija je posebno osmišljena za vođenje navijačkih skupina, a za razvoj aplikacije se koristi React biblioteka. Kroz razvoj aplikacije, istraživale su se mogućnosti korištenja suvremenih tehnoloških alata za poboljšanje angažmana i interakcije unutar navijačkih skupina.

Rad kombinira teorijski i praktični pristup kako bi istražio problematiku razvoja aplikacije za vođenje navijačkih skupina korištenjem biblioteke React. Metodologija rada obuhvaća istraživanje te praktičnu izradu aplikacije. Dostupna literatura i resursi s raznih stručnih web stranica analizirani su kako bi se stekao dublji uvid u tehnologije koje će biti korištene u izradi aplikacije.

U izradi aplikacije, primjenjuje se React tehnologija, organizacija komponenata, upravljanje stanjem i interakcijom unutar aplikacije. Rad na aplikaciji obuhvaća više koraka, uključujući kreiranje komponenata, implementaciju funkcionalnosti za vođenje skupina, organizaciju događanja te komunikaciju među članovima. Na kraju rada, zaključuje se usklađenost između teorijskoga istraživanja i praktične primjene.

1. PROGRAMSKI ALATI

Alat za programiranje može se odnositi na bilo koju softversku aplikaciju ili uslužni program dizajniran za pomoć razvojnim programerima softvera ili programerima u različitim zadacima koji se odnose na stvaranje, uređivanje, otklanjanje pogrešaka, održavanje ili izvođenje specifičnih zadataka unutar područja programiranja i razvoja. Alternativno, poznat je i kao alat za razvoj softvera [1]. Softverski alat funkcionira kao sistemski program koji uspostavlja vezu između programa i izvora koji generira njegove ulazne podatke, ili alternativno, povezuje ishode programa s primateljem koji ih koristi. Sastoji se od zbirke računalnih programa koje su programeri koristili u svrhu generiranja, upravljanja, otklanjanja pogrešaka ili pružanja podrške drugim aplikacijama i programima. Takvi alati obuhvaćaju niz elemenata uključujući uređivače koda, jezične biblioteke, programe za ispravljanje pogrešaka, alate za analizu performansi, GUI dizajnere i još mnogo toga [1].

Softverski alati igraju ključnu ulogu u svim fazama životnoga ciklusa softvera. Olakšavaju izvođenje i analizu poslovnih procesa, dokumentiraju napredak razvoja i optimiziraju razne procese. Ovi alati značajno pomažu programerima u učinkovitom upravljanju tijekovima rada projekta [2]. Postoje dvije primarne aktivnosti unutar područja računalstva [2]:

1. razvoj programa i
2. korištenje aplikacijskoga softvera.

Dok jezični procesori i operativni sustavi imaju značajnu ulogu u ovim aktivnostima, postoji još jedna, ključna uloga koju ispunjavaju programi osmišljeni pomoći u razvoju ili korištenju programa. Ovi programi obrađuju niz administrativnih zadataka koji se odnose na stvaranje programa i korištenje aplikacije. Njihova integracija nudi praktičnost i povećava učinkovitost i aktivnost korisnika [2].

Softverski alat odnosi se na sistemski program koji učinkovito uspostavlja vezu između programa i drugih programa ili korisnika unutar njegova okruženja. Zadatci koje izvršavaju softverski alati obuhvaćaju širok spektar, u rasponu od rutinskih zadataka koji uključuju sučelja do složenijih zadataka koji povećavaju učinkovitost angažmana korisnika [2].

Softverski alati odnose se na programe ili aplikacije dizajnirane za pomoć u stvaranju, testiranju i održavanju softvera [2]. Za potrebe izrade aplikacije za ovaj rad korištena su Integrirana razvojna okruženja (eng. *Integrated Development Environment - IDE*) i Visual studio code (VS Code).

IDE služi kao specijalizirani softverski alat koji nudi sveobuhvatno okruženje za razvoj softvera, integrirajući značajke kao što su uređivač teksta, kompajler, program za ispravljanje pogrešaka i drugo. Visual Studio Code je uređivač teksta koji se preporučuje i iskusnim profesionalcima i početnicima, a riječ je o proizvodu tvrtke Microsoft. Ističe se svojom dostupnošću, budući da je besplatan alat, otvorenoga koda i kompatibilan sa sustavima Windows, Linux i macOS. Popularnost Visual Studio Codea može se pripisati jednostavnom postupku instalacije i minimalnim zahtjevima za prostorom na disku.

Pružna podršku za širok raspon programskih jezika, uključujući C++, Java, Python i druge. Ima mogućnost proširenja uređivača kao što su programi za ispravljanje pogrešaka, alati za razvoj weba i drugo. Sučelje je prilagođeno korisniku. Jedna od istaknutih značajki Visual Studio Code-a je njegova funkcionalnost i stalna ažuriranja. Proširenja, koja su značajke trećih strana, omogućuju programerima da poboljšaju kvalitetu koda, funkcionalnost i izgled. Ova proširenja mogu uvesti podršku za dodatne jezike, programe za ispravljanje pogrešaka i razne alate, a programeri imaju pristup širokom spektru značajki i ažuriranja [3].

3.1. JavaScript

JavaScript (JS) je skriptni jezik koji se pretežno koristi na internetu. Služi za poboljšanje HTML stranica i često je integriran unutar HTML koda. JavaScript radi kao interpretirani jezik čime se negira potreba za kompilacijom. Omogućuje web stranicama postati interaktivnima i dinamičnima olakšavajući njihovu reakciju na događaje, daju prikaz specijalnih efekata, prihvaćanje varijabilnoga teksta, provjeru valjanosti podataka, omogućuju stvaranje kolačića, i otkrivanje preglednika [4]. Za izvršavanje, JavaScriptu nije potreban server, već se može izvršiti izravno u web pregledniku. Skriptu je moguće umetnuti bilo gdje na stranici, iako je optimalno umetnuti ju na dno dokumenta, netom prije zatvaranja *body taga* [5].

HTML stranice su prikladne za predstavljanje statičnoga sadržaja poput jednostavnih slika ili teksta. Međutim, moderne web stranice često su daleko od statičnih. Mnoge suvremene stranice sadrže dinamičke elemente kao što su izbornici, obrasci, dijaprojekcije i interaktivne slike.

JavaScript se pojavljuje kao jezik koji su web programeri odabrali kako bi olakšali ovu vrstu interakcije [4]. Budući da JavaScript surađuje s HTML stranicama, programer mora posjedovati HTML znanje kako bi u potpunosti iskoristio potencijal ovog skriptnoga jezika. U praktičnom smislu, JavaScript je najdominantniji [4].

Aplikacijska programska sučelja (API) pružaju dodatne mogućnosti za korištenje u JavaScript kodu. Naime, API su gotovi skupovi gradivnih blokova koda koji razvojnim programerima omogućuju implementaciju programa koje bi inače bilo teško ili nemoguće implementirati. API su ugrađeni u web preglednik i mogu izložiti podatke iz okolnoga računalnog okruženja ili raditi korisne složene stvari od kojih se neke navode u nastavku [6].

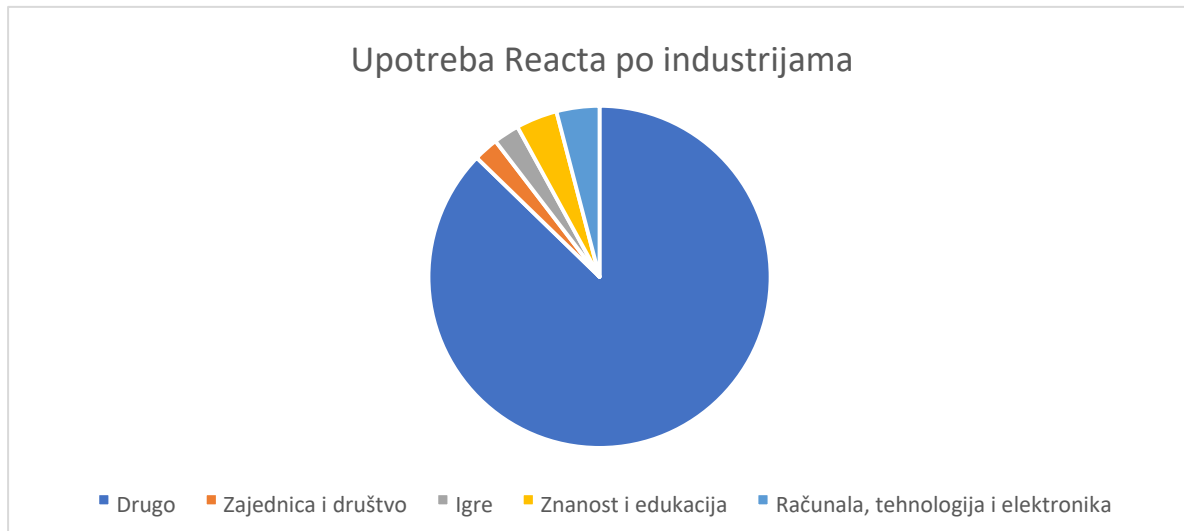
Postoje dva pristupa za integraciju JavaScripta u HTML datoteku [4]. Prva metoda uključuje ugradnju cijeloga JavaScript koda unutar samog HTML koda. Drugi pristup uključuje korištenje zasebne JavaScript datoteke, koja se poziva korištenjem Script elementa okruženoga Script oznakama. JavaScript datoteke obično se identificiraju ekstenzijom `.js.`, dok se JavaScript uglavnom koristi za interakcije s HTML objektima, također se može proširiti za interakciju s entitetima koji nisu HTML kao što su dodatci preglednika, svojstva CSS-a (Cascading Style Sheets), trenutačni datum ili čak sam preglednik [4].

3.2. React

React je biblioteka za razvoj korisničkoga sučelja temeljena na JavaScriptu. Razvio ga je Facebook, a koristi se kako bi se izradile web aplikacije. React predstavlja programski okvir koji omogućuje izgradnju pojedinih komponenti, a potom i njihovo sastavljanje u aplikaciju. React omogućuje izradu web i mobilnih aplikacija u *front-end* dijelu [7].

React se koristi u brojnim raznim primjenama, od web stranica zasnovanih na sadržaju poput web stranice The New York Timesa do aplikacija poput Twittera i Facebooka, koji se uvelike oslanja na interakciju korisnika s elementima web stranice. Dakle, gotovo svi dinamički elementi Facebook web aplikacije izgrađeni su s React.js [7]. Industrije u kojima se najčešće koristi React, prikazane su grafom u nastavku.

Slika 1: Primjena Reacta po industrijama



Izvor: izrada autora prema [9]

Iako je React biblioteka, a ne jezik, naširoko se koristi u web razvoju. Ova biblioteka se prvi put pojavila u svibnju 2013. godine i sada je jedna od najčešće korištenih frontend biblioteka za web razvoj. Važnost Reacta je u tome što se može koristiti za razvoj web i mobilnih aplikacija. Postoji okvir pod nazivom *React Native*, izveden iz samoga Reacta, koji je iznimno popularan i koristi se za stvaranje mobilnih aplikacija. React nudi različita proširenja za cjelokupnu arhitektonsku podršku aplikacije, kao što su *Flux* i *React Native*, izvan korisničkoga sučelja [7].

Kao jedna od najčešće prihvaćenih JavaScript biblioteka, React nudi svestranost u izgradnji različitih web elemenata, te udovoljava različitim zahtjevima u rasponu od jednostavnih, ali zanimljivih aplikacija poput Instagrama, do kompleksnijih platformi poput Netflix, prilagođavajući se širokim bazama korisnika i višejezičnoj podršci [7].

3.2.1. Značajke Reacta

React.js je popularna JavaScript biblioteka koja se koristi za izradu sučelja (eng. *Front End*) web aplikacija. Njime aktivno upravlja Meta, u suradnji sa stručnom zajednicom programera i tvrtki. Poznat po brzini, skalabilnosti i jednostavnosti, a posebno mogućnosti ažuriranja podataka na web stranicama bez potrebe za punim ponovnim učitavanjem, React.js nalazi široku primjenu u izradi web aplikacija [10].

Kako bi se definirao izgled korisničkoga sučelja u React-u, koristi se JSX, što je sintaktičko proširenje JavaScripta. JSX omogućuje pisanje HTML struktura unutar iste datoteke kao i JavaScript kod.

Virtualni DOM predstavlja Reactovu varijantu Stvarnoga DOM-a i znatno je brži. Kada dođe do promjene u stanju objekta, virtualni DOM ažurira samo taj objekt u stvarnom DOM-u, čime poboljšava učinkovitost i smanjuje potrebno vrijeme za osvježavanje prikaza. Ovaj pristup olakšava razvoj korisničkih sučelja u React-u, pridonoseći bržem i efikasnijem iskustvu korisnika [10].

Jednosmjerni protok podataka znači da je uobičajeno ugnijezditi podređene komponente unutar roditeljskih komponenti prilikom razvoja React projekta. Govoreći o ekstenzijama, React nadilazi samo UI (eng. *User interface*) okvir, te sadrži mnoga proširenja koja pokrivaju cjelokupnu arhitekturu aplikacije. Pomaže u izgradnji mobilnih aplikacija i pruža renderiranje na strani poslužitelja [10]. *Flux i Redux*, između ostalog, mogu proširiti React. *Flux i Redux* su dvije arhitekture za upravljanje stanjem u razvoju React aplikacija. *Flux* je arhitekturni obrazac koji je razvio Facebook za upravljanje stanjem u jednostranskoj (one-way) React aplikaciji.

Karakterizira ga Jednosmjerni tok podataka, što znači da se podatci kreću kroz aplikaciju u jednom smjeru, što pojednostavljuje praćenje i razumijevanje tijeka podataka. Sustav se sastoji od akcija (*Action*), dispečera (*Dispatcher*) i trgovina (*Store*). Akcije predstavljaju događaje, dispečer upravlja njihovom distribucijom, a trgovine čuvaju stanje aplikacije, a komponente ne komuniciraju izravno međusobno, već putem dispečera.

Redux je biblioteka za upravljanje stanjem, slična *Flux* konceptima, ali pojednostavljena i usmjerena na pojednostavljenje upravljanja stanjem u React aplikacijama. Sve informacije o stanju aplikacije čuvaju se u jednom glavnom objektu, što pojednostavljuje praćenje i

održavanje stanja. Akcije predstavljaju događaje, reduktori (*Reducers*) su čiste funkcije koje opisuju kako akcije mijenjaju stanje, a trgovina (*Store*) čuva stanje aplikacije. Stanje se mijenja putem nepromijenjivih (*immutable*) objekata kako bi se olakšalo praćenje promjena.

React aplikacije su jednostavne za upotrebu i jednostavne za testiranje. Facebook primjerice, nudi proširenje preglednika koje pojednostavljuje i ubrzava otklanjanje pogrešaka u React aplikacijama, a naziva se "*React Developer Tools*". Ovo proširenje omogućava programerima lakšu inspekciju i analizu komponenti Reacta izravno iz preglednika. Nudi mnoge korisne značajke, uključujući pregled hijerarhije komponenti, praćenje promjena stanja i props-a, te duboko inspekciju i manipulaciju s virtualnim DOM-om [10].

MVC (eng. Model View Controller) je arhitektonski uzorak koji dijeli sloj aplikacije na model, prikaz i kontroler. Model se odnosi na svu logiku koja se odnosi na podatke, pogled se koristi za UI logiku aplikacije, a kontroler je sučelje između modela i pogleda. Govoreći o arhitekturi Model View Controller (MVC), React je 'View' odgovoran za to kako aplikacija izgleda i kakav doživljaj zapravo pruža korisniku [10].

3.2.2. Komponente u Reactu

Komponenta je autonomni dio koda koji se može ponovno koristiti i koji fragmentira korisničko sučelje u manje segmente. Na primjer, kada se izrađuje korisničko sučelje za React web mjesto pomoću Reactjs-a, moguće je podijeliti korisničko sučelje u odjeljke kojima je lakše upravljati. Umjesto sastavljanja cijeloga korisničkoga sučelja unutar jedne datoteke poput HTML-a, različiti odjeljci mogu se odvojiti u različite autonomne fragmente, koji su poznati kao komponente. Svaka od komponenti se pohranjuje u vlastitu JavaScript datoteku [11].

U Reactu postoje dvije primarne vrste komponenti: funkcionalne i klasne komponente, te se svaki element smatra komponentom. Iako svaka komponenta generira neki HTML izlaz, može se vratiti samo pojedinačni HTML element. Unutar ovoga elementa mogu se ugnijezditi brojni podređeni elementi [11]. Umjesto sastavljanja cijeloga koda unutar jedne datoteke, aplikacija se može podijeliti na više komponenti. Posljedično, moguće je izdvojiti kod koji služi za posebne svrhe u zasebne komponente. Na taj način se ekstrahirana komponenta može koristiti u cijeloj aplikaciji bez dupliciranja ili ponovnoga pisanja koda. Ova praksa naziva se ekstrakcija i ponovna uporaba komponenti [11].

Primarna, najznačajnija i preporučena vrsta komponente u Reactu je funkcionalna komponenta.

Funkcionalna komponenta je u biti JavaScript funkcija koja proizvodi React element (JSX). Funkcionalne komponente se deklariraju pomoću ključne riječi "*function*". U biti, funkcionalna komponenta je *JavaScript* funkcija ili ES6 funkcija koja prihvata svojstva kao argumente i generira JSX sadržaj. Konvencionalno, nazivi komponenti uvijek bi trebali počinjati velikim slovima [11].

Druga kategorija komponenti je komponenta klase. Analogno funkcijama JavaScripta, klase su također prisutne u jeziku. Komponente klase izgrađene su kao ES6 klase i generiraju JSX izlaz. Na primjer, funkcionalna komponenta "*HelloWorld*" može se transformirati u komponentu klase. Komponente klase započinju ključnom riječi "*class*" i nasljeđuju od konstruktora React "*Component*". Ove komponente uključuju metodu "renderiranja" koja daje JSX izlaz. Koriste se ne samo za renderiranje JSX-a, već i za upravljanje dinamičkim podacima unutar komponenti, poboljšavajući odziv aplikacije i jednostavnost upotrebe, ovi se podaci nazivaju „stanje“, a mogu ga imati obje vrste komponenti, i klasne i funkcionalne [11].

Rekviziti (engl. *props*) služe kao komunikacijski mehanizam između komponenti. React koristi rekvizite za prijenos podataka s jedne komponente na drugu, pretežno na jednosmjernan način (od nadređene do podređene komponente). Važno je napomenuti da rekviziti ne olakšavaju prijenos podataka između komponenti na istoj razini hijerarhije [11]. Rekviziti su prilagođene vrijednosti koje pridonose dinamičkoj prirodi komponenti.

3.2.3. Prednosti Reacta

React je popularan *front-end* razvojni okvir. Uz pomoć React-a, moguće je kreiranje dinamičkih aplikacija. Za razliku od JavaScript-a, React olakšava kreiranje dinamičkih i interaktivnih web aplikacija, zahtijeva manje kodiranja, te istovremeno nudi znatno više funkcionalnosti. Prednost Reacta je i poboljšanje performansi. Naime, React koristi Virtual DOM s kojim je moguće znatno brže kreirati web aplikacije. Virtualni DOM ima funkciju uspoređivanja prijašnjih stanja komponenti, te ažurira samo one stavke koje su u stvarnom DOM-u i koje su promijenjene, a za razliku od konvencionalnih web aplikacija, ne ažurira ih ponovo sve.

React slijedi jednosmjerni tijek podataka. To podrazumijeva da kada programeri dizajniraju aplikaciju u Reactu, ugrađuju podređene komponente unutar nadređenih komponenti. Budući da podatci teku u jednom smjeru, postaje znatno lakše otkloniti pogreške i otkriti gdje se u

danom trenutku pojavljuje problem u aplikaciji. Komponente su građevni blokovi svake React aplikacije, a jedna aplikacija obično se sastoji od više komponenti. Ove komponente imaju svoju logiku i kontrole i mogu se ponovno koristiti u cijeloj aplikaciji, što znatno pridonose racionalizaciji vremena potrebnoga za razvoj aplikacije.

React najčešće kombinira osnovne HTML i JavaScript koncepte s nekim korisnim dodatcima. Ipak, kao što je slučaj i s drugim alatima i okvirima, potrebno je utrošiti određeno vrijeme kako bi se na ispravan način shvatila i razumjela Reactova biblioteka.

Neke od dodatnih prednosti Reacta su [10]:

- modularnost: React omogućuje rastavljanje koda na manje, ponovno upotrebljive komponente. Ovo poboljšava održavanje i fleksibilnost aplikacija.
- deklarativna priroda: React koristi deklarativno programiranje, olakšavajući opisivanje željenih rezultata. To uključuje rad Document Object Model (DOM) na deklarativan način, smanjujući potrebu za izravnom manipulacijom DOM-a.
- dosljednost na više platformi: Pisanje koda za React omogućuje izvođenje na različitim platformama.
- jednostavnost: React se ističe jednostavnošću dizajna čime se povećava produktivnost programera.
- brzina i učinkovitost: React nudi visoke performanse i učinkovito prikazivanje korisničkih sučelja zahvaljujući virtualnom DOM-u.
- stabilnost koda: React potiče stabilnost koda kroz uporabu komponenti za višekratnu upotrebu i jasnu strukturu smanjujući šanse za uvođenje pogrešaka.
- čvrsta podrška zajednice: React se može pohvaliti snažnom i aktivnom zajednicom programera koja neprestano pridonosi njegovom rastu, pruža podršku i dijeli znanje, čime se osiguravaju stalna poboljšanja i rješenja za izazove.

3.2.4. Nedostatci React-a

Pored mnogih prednosti i popularnosti, React ima određene nedostatke i izazove, te je važno razumjeti i analizirati potencijalne nedostatke kako bi se omogućilo temeljito i uravnoteženo razmatranje prilikom odabira tehnologije za razvoj projekata. U nastavku su istraženi ključni

nedostatci Reacta, te se pruža uvid u to kako se ovi izazovi mogu svladati ili ublažiti putem različitih pristupa i rješenja.

- brzi tempo razvoja: Brzi rast Reacta kao biblioteke i česte promjene mogu biti izazov za programere koji moraju kontinuirano prilagođavati svoje prakse kodiranja.
- fleksibilnost i dvosmislenost: Reactova fleksibilnost može predstavljati izazov jer različiti timovi koriste različite prakse i konvencije. To može otežati novim programerima prilagodbu različitim standardima, posebno ako su navikli na strukturirane okvire s jasno definiranim konvencijama.
- djelomična priroda okvira: Unatoč bogatstvu značajki, React se usredotočuje samo na prikaz u MVC arhitekturi. To može dovesti do neorganiziranih struktura koda i obrazaca, dok puni okviri poput Angulara nude cjelovito MVC okruženje s poboljšanom strukturom.
- neodgovarajuća dokumentacija: Brza evolucija Reacta i stalni priljev novih alata i obrazaca često rezultiraju nedostatkom detaljne dokumentacije. To može otežati novim programerima ulazak u React projekte i usporiti razvoj timova manje iskusnih programera.

JSX složenost: Iako JSX olakšava razvoj, njegova mješavina JavaScripta i HTML sintakse može izazvati poteškoće, posebno zbog novih atributa i sintakse. Nedostatak sveobuhvatne dokumentacije dodatno komplicira rad s JSX-om [8].

3.3. Pokretanje novoga React projekta

Pokretanje novoga React projekta uključuje nekoliko koraka kako bi se olakšao proces razvoja. Pri izradi nove aplikacije ili web stranice pomoću Reacta, preporuča se odlučiti za jedan od okvira koji pokreće React. Ovi okviri nude niz značajki koje su bitne za većinu aplikacija i web stranica, kao što su usmjeravanje, dohvaćanje podataka i generiranje HTML-a. Prije početka projekta, potrebno je imati instaliran Node.js za potrebe lokalnoga razvoja. Iako se Node.js također može koristiti u proizvodnji, to nije obvezno. Mnogi React okviri nude funkcionalnost izvoza projekata u statičnu HTML/CSS/JS mapu. Među izborom dostupnih okvira React proizvodne razine ističe se nekoliko istaknutih opcija [12]:

1. Next.js - olakšava kreiranje React aplikacija različite složenosti, bilo da se radi o pretežno statičkom blogu ili sofisticiranoj dinamičkoj aplikaciji. Next.js projekti mogu se

implementirati na velik raspon hosting rješenja. Štoviše, potpuno statične Next.js aplikacije mogu se postaviti na bilo koju statičnu uslugu hostinga.

2. Remix - alternativni *full-stack React framework*, uvodi mogućnosti ugniježdenoga usmjeravanja, što omogućuje segmentiranje aplikacije u ugniježđene komponente, omogućujući paralelno učitavanje podataka i besprijekorno osvježavanje kao odgovor na interakcije korisnika.

3. Gatsby - još jedan React framework, specijaliziran je za izradu brzih web stranica podržanih CMS-om. Može se pohvaliti opsežnim ekosustavom dodataka i GraphQL podatkovnim slojem, pojednostavljajući integraciju sadržaja, API-ja i usluga u objedinjenu web stranicu.

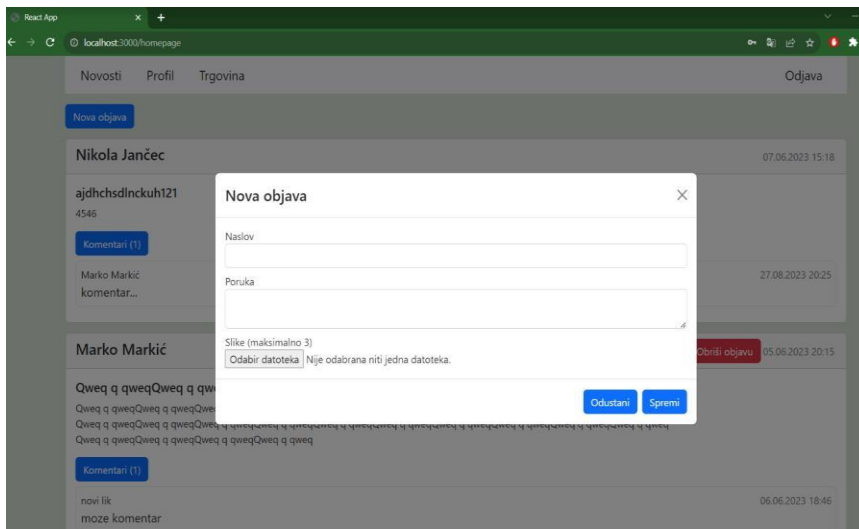
4. Expo (za izvorne aplikacije) – koristi se u područje nativnih aplikacija, Expo nudi okvir React koji podržava razvoj univerzalnih Android, iOS i web aplikacija, od kojih se svaka može pohvaliti nativnim korisničkim sučeljima. Expo isporučuje SDK za React Native, pojednostavljujući korištenje izvornih komponenti.

4. OPIS APLIKACIJE

Aplikacija je osmišljena za organiziranje navijača u navijačke skupine i njihovu međusobnu interakciju tako da svaki navijač odabere navijačku skupinu kojoj pripada ili klub za koji navija. Korištenje aplikacije namijenjeno je postojećim navijačkim skupinama i klubovima u tuzemstvu i inozemstvu. Također, aplikacija je namijenjena regionalnim i lokalnim klubovima.

Aplikacija je namijenjena desktop i mobilnoj verziji (Ios i Android). Osim okupljanja navijača i drugih sportskih sudionika, te razmjene informacija i objavljivanja sadržaja, aplikacija služi i kao platforma koja korisnicima može ponuditi odgovore na postavljena pitanja i pružiti prijedloge ili organizirane navijačke radnje. Izgled korisničkoga sučelja pri upisivanju javnih poruka u aplikaciji, prikazan je slikom 2.

Slika 2: Korisničko sučelje aplikacije



Izvor: Izrada autora

Korisnici imaju mogućnost komentiranja, postavljanja multimedijalnoga sadržaja i slično. Sadržajem upravljaju administratori. Planirana su redovita ažuriranja i nadogradnja aplikacije. Planirana je i mogućnost kreiranja individualnih grupa u *chatu* koje će biti zaštićene lozinkama, dok će odgovorne osobe biti verificirane osobnim identifikacijskim brojem i *emailom*. Verifikacija računa u aplikaciji igra ključnu ulogu u osiguravanju sigurnosti, povjerenja i ispravnoga funkcioniranja. Prvi korak u verifikaciji računa je registracija korisničkoga računa, a obrazac je prikazan slikom 3. Korisnici trebaju pružiti osnovne informacije kao što su e-mail adresa, korisničko ime i lozinka.

Slika 3: Registracija korisnika

Registracija

Datum rođenja ne može biti u budućnosti

Ime*
1222

Prezime*
qwe

E-pošta*
asdasd@gmail.com

Lozinka*
.....

Ponovite lozinku*
.....

Datum rođenja
12-Jan-2025

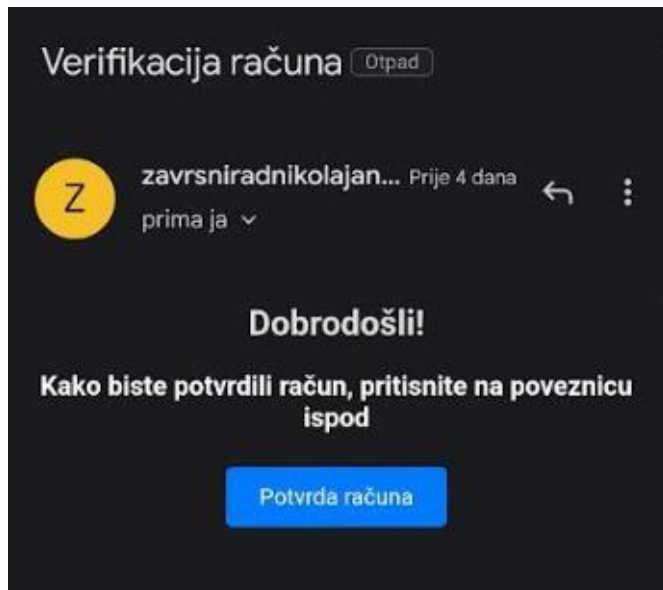
Registriraj se

Već imate korisnički račun? [Prijava](#)

Izvor: izrada autora

Nakon registracije, korisniku se šalje verifikacijski e-mail (prikazan na slici u nastavku teksta), s jedinstvenim linkom ili kodom koji mora unijeti kako bi potvrdio svoju elektroničku adresu što osigurava da korisnik pruža valjanu adresu i povećava sigurnost komunikacije. Svi korisnici koji su verificirani imaju pravo na upis poruka na za to predviđeno mjesto u aplikaciji.

Slika 4: Verifikacijska poruka putem e-maila korisnika



Izvor: izrada autora

Nakon uspješne prijave, novi korisnici imaju pristup svojoj personaliziranoj stranici na kojoj će moći istraživati sadržaj vezan za njihov klub i pridružiti se raznim aktivnostima i raspravama. Pri registraciji, potrebno je upisivti podatke kako je određeno u aplikaciji, a što podrazumijeva da ime i prezime korisnika mora imati više slova, te nije dopušteno upisivanje jednoga ili dva slova. Ujedno, pri registraciji se vrši i validacija datuma rođenja što je prikazano slikom u nastavku.

Slika 5: Validacija pri registraciji novih korisnika

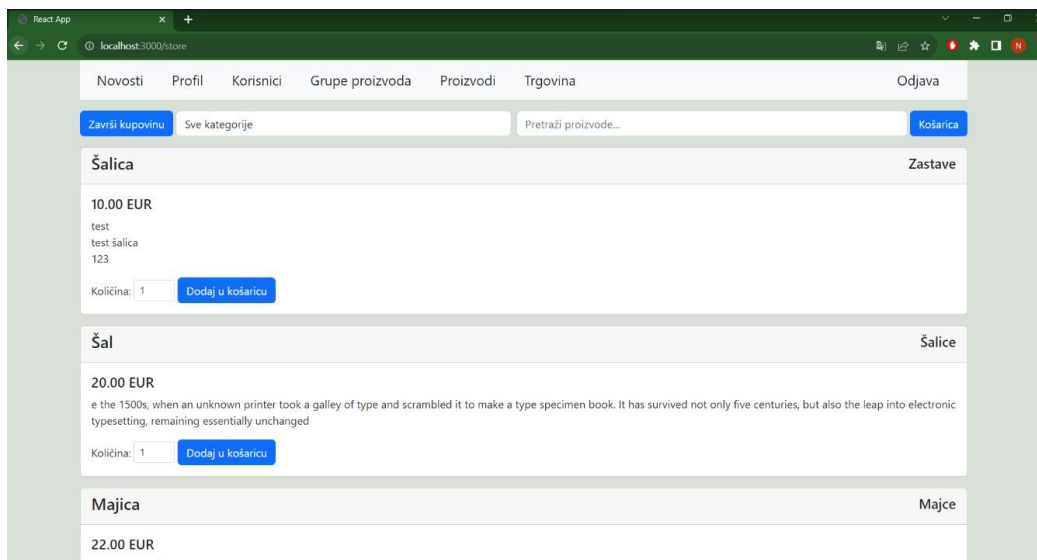
Datum rođenja ne može biti u budućnosti		Ime je prekratko		Datum rođenja ne može biti u budućnosti	
Ime*	ana	Ime*	a	Ime*	ana
Prezime*	balog	Prezime*	b	Prezime*	balog
E-pošta*	hgbinj@ghih.vom	E-pošta*	hgbinj@ghih.vom	E-pošta*	hgbinj@ghih.vom
Lozinka*	*****	Lozinka*	***	Lozinka*	*****
Ponovite lozinku*	*****	Ponovite lozinku*	****	Ponovite lozinku*	*****
Skupina	Nedefinirano	Skupina	Nedefinirano	Skupina	Nedefinirano
Datum rođenja	30.11.2039.	Datum rođenja	30.11.2039.	Datum rođenja	30.11.2039.

Izvor: izrada autora

Osim temeljnih komunikacijskih alata, u aplikaciju je integriran sustav Internet trgovine koji će omogućiti korisnicima kupovinu navijačkih rekvizita i suvenira. Ova mogućnost pruža korisnicima dodatnu vrijednost unutar aplikacije. Slika 6 prikazuje korisničko sučelje Internet trgovine. Važno je napomenuti kako će korisnici u svakom trenutku imati mogućnost vidjeti koji su proizvodi dodani u košaricu, ujedno, moći će iz nje oduzimati ili dodavati nove proizvode, te mijenjati količine pojedinih proizvoda u košarici prije izvršenja kupovine, a biti će im prikazan i iznos košarice.

Korisnicima će biti omogućeno automatsko popunjavanje podataka dok su prijavljeni u aplikaciju, što će poboljšati njihovo korisničko iskustvo. Kada korisnici budu prijavljeni u svoje račune, njihove osnovne informacije biti će pohranjene kako bi se koristile za automatsko popunjavanje obrazaca i polja unutar aplikacije, čime se smanjuje potreba za ponovnim ručnim unosom tih informacija.

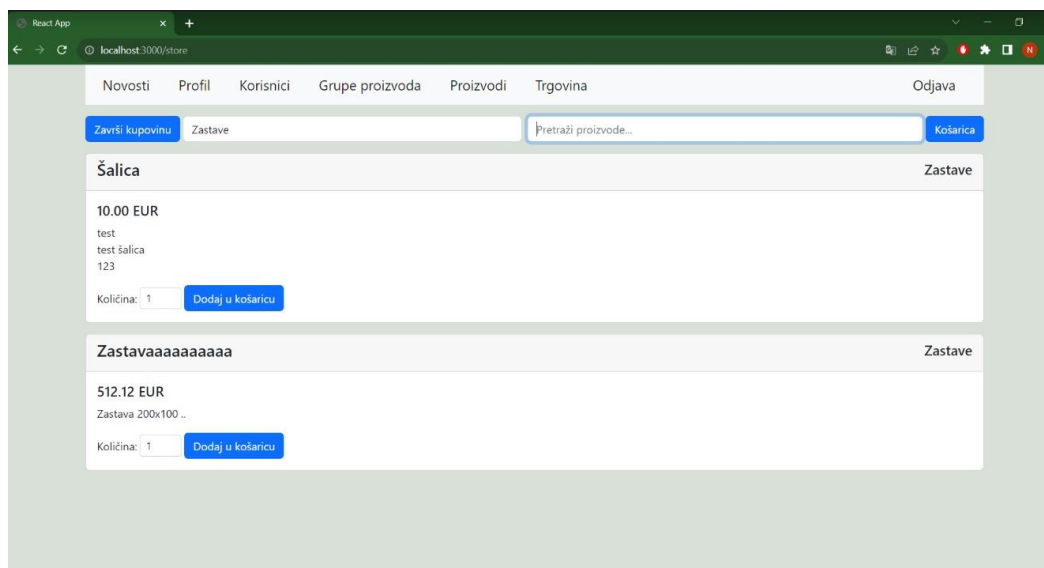
Slika 6: Korisničko sučelje Internet trgovine



Izvor: Izrada autora

Webshop je organiziran prema kategorijama proizvoda, što korisnicima omogućuje lakšu kupovinu, odnosno pregled izbora proizvoda. Dostupne kategorije proizvoda su majice, kape, zastave i kategorija suveniri kao što su šalice.

Slika 7: Kupovina u webshopu



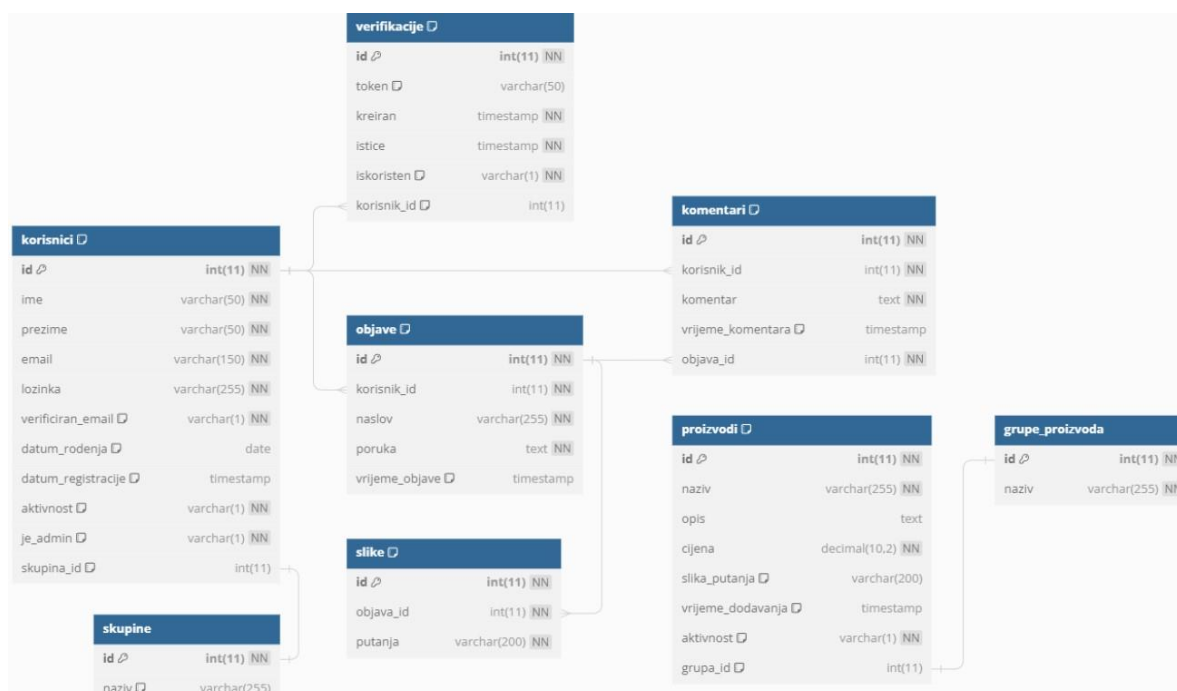
Izvor: Izrada autora

Korisnici imaju opciju pregledavanja i, po potrebi, promjene automatski popunjenih podataka. To će im omogućiti da prilagode informacije prema potrebama, na primjer ako žele koristiti različitu adresu dostave ili ažurirati svoje kontakt informacije. Važno je napomenuti da će se podaci korisnika štititi kako bi se osigurala privatnost i sigurnost informacija.

Bazama podataka koji se unose u aplikaciju upravlja MySQL baza. Definirana je struktura baze podataka pri čemu su određene tablice koje sadrže podatke, određene su vrste podataka koje će se čuvati u svakom stupcu tablice, postavljeni ključevi za povezivanje tablica i drugih elemenata. Podatci se organiziraju u tablice, s definiranim stupcima (atributima) i redovima (unosima podataka). Na primjer, može postojati tablica "Korisnici" s atributima kao što su "ID", "Ime", "Prezime", "Email" i "Lozinka". Za komunikaciju s MySQL bazom koristi se SQL, specifičan jezik za upravljanje i manipulaciju podacima u relacijskim bazama podataka. SQL se koristi za kreiranje, ažuriranje, brisanje i čitanje podataka iz baze.

Korišteni su SQL upiti kako bi se izvršile različite operacije nad podacima. Na primjer, *SELECT* upit se koristi za dohvaćanje podataka iz baze, *INSERT* za dodavanje novih podataka, *UPDATE* za ažuriranje postojećih podataka i *DELETE* za brisanje podataka. Kada korisnici popunjavaju registracijski obrazac, dodaju primjerice proizvode u košaricu, aplikacija koristi SQL upite kako bi komunicirala s MySQL bazom. Podatci koje korisnici unesu putem sučelja aplikacije obrađuju se i pohranjuju u MySQL bazi. Shema baze i tablice koje se koriste u aplikaciji, prikazane su slikom 8. u nastavku. Prije nego što se podatci upišu u bazu, obično se vrši validacija kako bi se osiguralo da su podatci ispravni. Također, primjenjuju se sigurnosne prakse kako bi se spriječilo neovlašteno pristupanje ili zlouporaba podataka.

Slika 8: Shema cijele baze



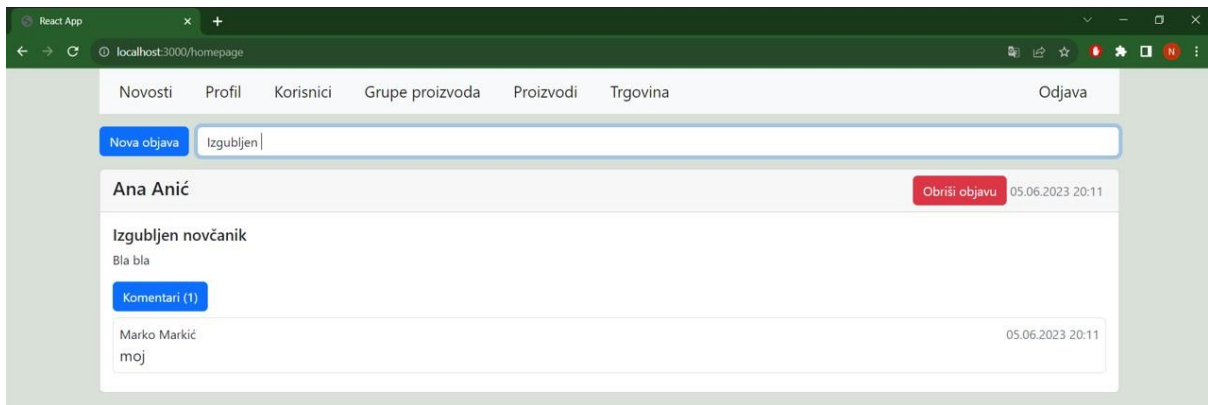
Izvor: Izrada autora

Tražilica je jedna od ključnih komponenti za pružanje korisničkoga iskustva u aplikaciji. Ključne karakteristike tražilice su da pretražuje prema pojmovima (uključujući naslove tema), imenima i datumima, odnosno moguće je pretraživati korisnike, razne pojmove, ili datume. Nakon što korisnici izvrše pretragu, tražilica treba prikazati relevantne rezultate na temelju ključnih riječi. Osim toga, korisnicima omogućuje filtrirati rezultate pretrage prema različitim kriterijima, kao što su nadnevak objave ili vrsta proizvoda. Planirano je i praćenje korištenja tražilice budući da ono može pružiti korisne uvide u to kako korisnici traže informacije unutar aplikacije, odnosno koji su traženi pojmovi, što može pomoći u optimizaciji algoritama pretrage i poboljšanju korisničkoga iskustva.

1. Pretraga po pojmovima:

Korisnici mogu pretraživati različite pojmove, riječi ili fraze unutar foruma i web shopa. To uključuje pretragu po naslovima, opisima proizvoda, korisničkim komentarima i postovima na forumu. Napredna tražilica trebala bi omogućiti brzu i preciznu pretragu neovisno o tome koliko velik je sadržaj, a primjer je prikazan slikom u nastavku.

Slika 9: Primjer rezultata u tražilici pri pretraživanju pojmova

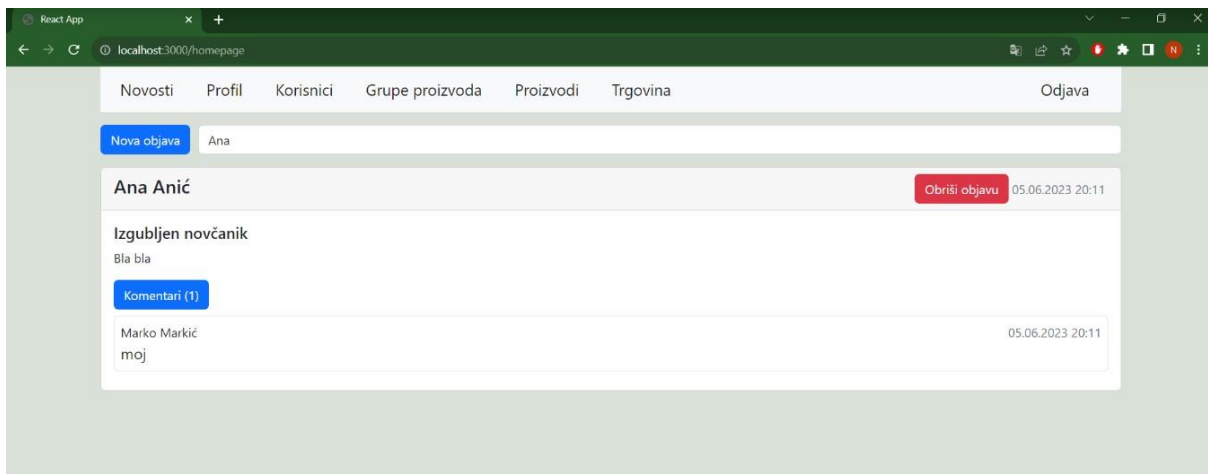


Izrada autora

2. Pretraga korisnika (autora upisa):

Osim pretrage sadržaja, korisnici mogu pretraživati i korisničke profile, odnosno upise po autoru upisa. To može biti korisno za povezivanje s određenim članovima foruma i brže traženje postova nekoga korisnika, odnosno informacije. Tražilica treba omogućiti pretragu po korisničkim imenima, a primjer je prikazan slikom u nastavku.

Slika 10: Primjer rezultata u tražilici pri pretraživanju korisnika

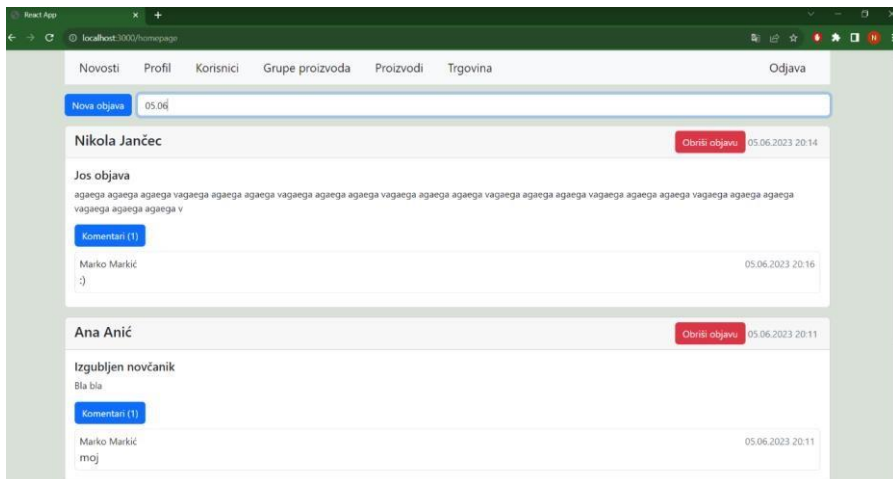


Izvor: izrada autora

3. Pretraga nadnevk:

Ukoliko određeni upisi u aplikaciji uključuju podatke koji sadrže nadnevak, kao što je nadnevak upisa određenoga korisnika, tražilica treba podržavati pretragu ovih brojki. Korisnicima treba omogućiti pretraživanje upisa korisnika na određeni nadnevak. Primjer pretraživanja po nadnevku prikazan je slikom u nastavku.

Slika 11: Primjer rezultata u tražilici pri pretraživanju po nadnevku:



Izvor: Izrada autora

4.1. Primjena aplikacije

Predviđena su dva slučaja primjene aplikacije:

1. slučaj primjene 1 za postojeće navijačke skupine, te

2. slučaj primjene 2 za nove navijačke skupine

Slučaj primjene 1

Odnosi se na već postojeće organizirane skupine navijača koje su administratori unijeli u aplikaciju. Putem linka za prijavu, korisnika se izravno upućuje na stranicu s prijavom koji potom unosi svoje podatke za registraciju, ukoliko još nije registriran u aplikaciju. Po registraciji, odnosno logiraju, korisnik ulazi u aplikaciju koja pokazuje korisnikovu grupu i klub za koji navija. Registracijom se ujedno vrši i verifikacija korisnika, koji na svoju e-adresu dobiva potvrdu o istoj što je prikazano ranije u radu. Kako bi pristupio aplikaciji, korisnik za prijavu u sustav mora imati valjano korisničko ime i e-adresu te jedinstvenu lozinku, a prilikom prijave navodi puno ime prezime, datum rođenja, adresu elektroničke pošte, te kreira lozinku za pristup svom računu. Osobni podatci korisnika neće biti javno dostupni drugim korisnicima, već će se korisnici razlikovati samo izabranim nadimcima.

Slučaj primjene 2

Aplikacija se koristi za povezivanje novih navijačkih skupina, a programiranje sučelja i procesa pristupa novim klubovima za navijanje zahtijeva određene korake. Administratori aplikacije programiraju sučelja nove navijačke skupine. Korisnici prolaze proces registracije kako bi postali dijelom aplikacije prema istim koracima kako je opisano u slučaju primjene 1. Prilikom registracije, korisnik unosi svoje osobne podatke kao što su puno ime, prezime i adresa elektroničke pošte, te kreiraju lozinku. Korisnik odabire korisničko ime.

Kreiranje novih navijačkih skupina u aplikaciji odvija se putem posebne funkcionalnosti koja administratorima aplikacije omogućuje kreiranje novih navijačkih skupina za klubove koji dosad nisu bili prisutni u aplikaciji. Administrator će unijeti osnovne informacije o novoj skupini, kao što su naziv kluba, kratak opis, simboli ili boje. Nakon stvaranja skupine, korisnici će moći pristupiti toj skupini i klubu putem svojih korisničkih računa. Popis svih korisnika vidljiv je administratoru, kao što je vidljiv i podatak u koju navijačku skupinu pripada pojedini korisnik. Pri otvaranju pojedinoga profila, administrator ima uvid u navijačku pripadnost te ju može mijenjati, odnosno ukoliko korisnik želi promijeniti navijačku skupinu, mora kontaktirati administratora.

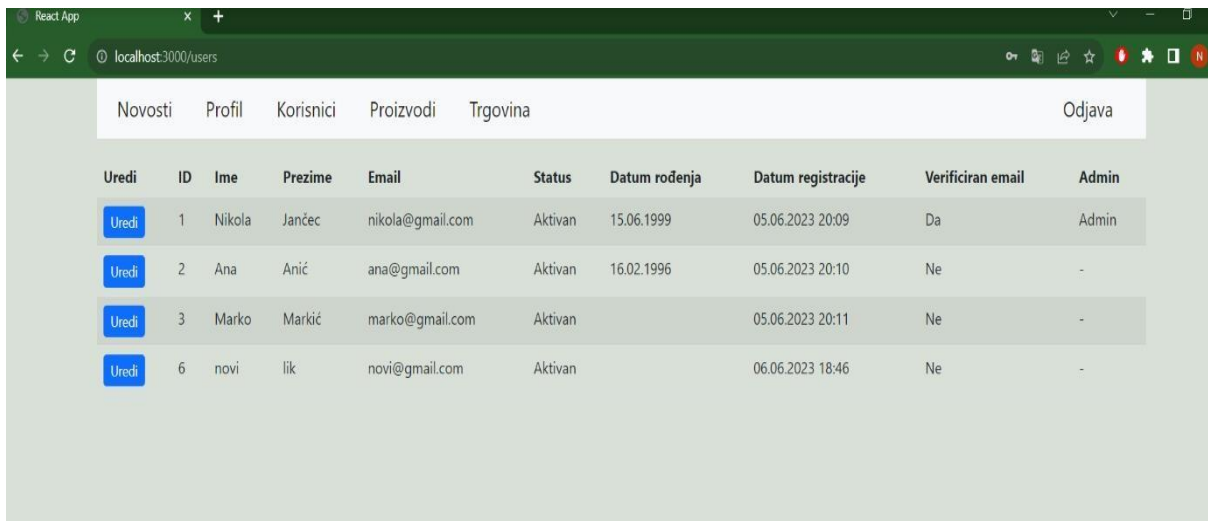
Nakon što su novi korisnici uspješno registrirani, mogu pretraživati i pristupiti različitim navijačkim skupinama unutar aplikacije. Korisnici mogu pretraživati navijačke skupine ili

klub, pregledati relevantne informacije, pridružiti se diskusijama, događanjima i aktivnostima koje su vezane za taj klub. Govoreći o funkcionalnostima koje će biti dostupne korisnicima unutar aplikacije, naglasak je na opcijama koje omogućuju raznoliku interakciju.

U cilju zaštite zloupotrebe korisničkih računa, ukoliko korisnik pogrešno unese lozinku i nakon nekoliko pokušaja, biti će blokiran na 3 minute, nakon toga bit će blokirani na 10 minuta, nakon čega na svoju elektroničku adresu dobiva obavijest o zaštiti, te će provjerom identiteta moći ponovo pristupiti aplikaciji. Kao i u slučaju primjene 1, korisnici će komunicirati međusobno koristeći nadimke umjesto svojih stvarnih imena.

Administratori aplikacije su odgovorni za stvaranje novih klubova i navijačkih skupina, a adminovo sučelje prikazano je slikom 4.

Slika 12: Sučelje administratora



The screenshot shows a web browser window with the URL localhost:3000/users. The page has a navigation bar with links: Novosti, Profil, Korisnici, Proizvodi, Trgovina, and Odjava. Below the navigation bar is a table with columns: Uredi, ID, Ime, Prezime, Email, Status, Datum rođenja, Datum registracije, Verificiran email, and Admin. The table contains four rows of user data.

Uredi	ID	Ime	Prezime	Email	Status	Datum rođenja	Datum registracije	Verificiran email	Admin
Uredi	1	Nikola	Jančec	nikola@gmail.com	Aktivan	15.06.1999	05.06.2023 20:09	Da	Admin
Uredi	2	Ana	Anić	ana@gmail.com	Aktivan	16.02.1996	05.06.2023 20:10	Ne	-
Uredi	3	Marko	Markić	marko@gmail.com	Aktivan		05.06.2023 20:11	Ne	-
Uredi	6	novi	lik	novi@gmail.com	Aktivan		06.06.2023 18:46	Ne	-

Izvor: Izrada autora

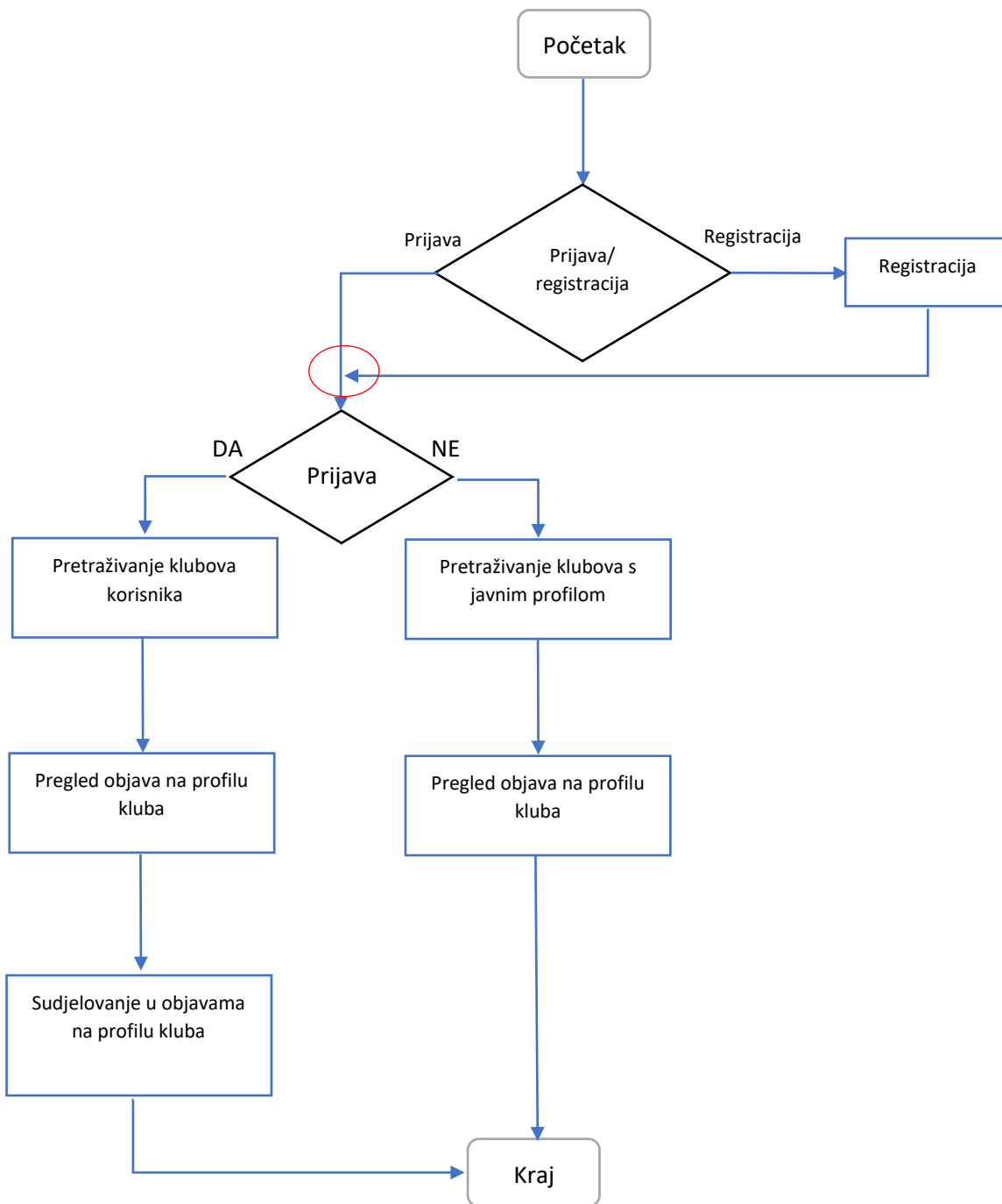
Administratori imaju mogućnost moderiranja sadržaja, upravljanja korisnicima i osiguravanja da pravila i smjernice aplikacije budu poštovane. Korisnici će potom moći pretraživati i pristupiti novim skupinama navijača, te aktivno sudjelovati u komunikaciji i aktivnostima koje su povezane s njihovim interesom za taj klub. U nastavku je prikazan dijagram kojim se prikazuju koraci procesa pristupa i korištenja aplikacije.

Slika 13: Dijagram procesa



-  Početak/Kraj -Odluka

Proces



4.2. Testiranje aplikacije

Testiranje aplikacije odnosi se na oblik testiranja softvera koji koristi skripte za otkrivanje softverskih pogrešaka, s ciljem osiguravanja efektivnoga funkcioniranja cijele aplikacije.

Unutar područja softverskoga inženjeringa, testiranje aplikacija obuhvaća više kategorija kao što su grafičko korisničko sučelje, funkcionalnost, pozadinska baza podataka, testiranje opterećenja i drugo [13]. Odabir specifičnih testova koji će se provesti ovisi o prirodi aplikacije koja se ocjenjuje. Testiranje aplikacije sustavno je podijeljeno u tri različita segmenta [13]:

- testiranje web aplikacija
- testiranje aplikacije za stolna računala
- testiranje mobilne aplikacije

Kada se radi o aplikacijama za stolna računala, proces testiranja obuhvaća više aspekata uključujući korisničko iskustvo, odnosno korisničko sučelje, poslovnu logiku, operacije baze podataka, generiranje izvješća, korisničke uloge i dopuštenja, integritet podataka, upotrebljivost, funkcionalnost, izvedbu, sigurnost, hardver i kompatibilnost softvera, te protok podataka.

Plan testiranja uključuje korištenje aplikacije prijavljenoga i anonimnoga korisnika. Anonimni ili neprijavljeni korisnik ne može imati pristup podacima u aplikaciji kao što ih ima prijavljeni korisnik, te će mu korištenje aplikacije biti ograničeno na pretrage javnih profila. Stoga je u procesu provjere, odnosno testa provjere važno utvrditi pristupe i mogućnosti anonimnih korisnika. Nasuprot njima su prijavljeni korisnici za koje je također potrebno uraditi testove provjere ima li prijavljeni korisnik pristup svim funkcijama na koje kao prijavljeni korisnik ostvaruje prava.

Kod prijave, odnosno registracije, provjerava se ispravnost unesenih podataka. Potrebno je također provjeriti i spremaju li se podaci registriranih korisnika u bazu podataka, dok je u slučaju prijavljenih korisnika potrebno osigurati dohvaćanje istih. Testiranje se odnosi i u slučaju opcije odabira željene navijačke grupe tako da se testira tražilica, i način funkcioniranja. Govoreći o odabiru kluba, potrebno je testirati vodi li odabir na ispravan način, odnosno vodi li aplikacija na odabrani ili neki treći klub. Sigurnost u virtualnom prostoru danas predstavlja vrlo važan čimbenik, stoga će se posebna pažnja u izradi aplikacije posvetiti sigurnom korištenju aplikacije u svakom segmentu.

U prvoj fazi zasebno je testirana svaka jedinica, nakon čega sljedi provjera funkcionalnosti cjelokupnog sustava. Pri tome je ispitano je li sustav prikladan za namijenjenu svrhu. Također, u ovoj fazi je izvršena provjera postoje li određene neočekivane interakcije između pojedinih funkcionalnosti sustava.

Posljednja faza testiranja odnosi se na ponašanje sustava u odnosu na zahtjeve korisnika. Stoga je pažnju potrebno posvetiti izradi korisničkoga sučelja, kako bi se korisnik lako, brzo i jednostavno kretao aplikacijom. Zato je ovaj dio testiranja od iznimne važnosti za rad aplikacije. Aplikacija mora biti pregledna i jednostavna za korištenje svakom korisniku. Testiranje ovoga segmenta razvoja aplikacije provoditi će se u periodu od 30 dana među 20 dragovoljaca. Na temelju njihove povratne informacije, aplikacija će se dodatno usavršavati i unaprjeđivati.

4.3. Planirani izvještaji

Planirani izvještaji važni su za administratore i analize u okviru aplikacije. Ovi izvještaji se dijele na administratorske i korisničke izvještaje, te pružaju specifične informacije koje su važne za praćenje performansi, optimizaciju, te unaprjeđenje korisničkoga iskustva. Ovaj izvještaj osigurava jasnu sliku o korisničkim interakcijama dajući administratorima uvid u obrasce ponašanja korisnika. Kroz analizu ovoga izvještaja, administratori mogu sagledati popularnost aplikacije u odnosu na broj korisnika koji su pristupili, a također dobivaju i kvantitativni uvid u razinu posjećenosti. Praćenje kretanja korisnika unutar aplikacije omogućava dublju analizu njihovih preferencija, što se kasnije koristi za poboljšanje funkcionalnosti i korisničkoga angažmana.

Važno je napomenuti da, uz sve prednosti, planirani izvještaji također služe kao alat za prepoznavanje i rješavanje problema. U slučaju nenadanih prekida rada aplikacije, neplanirani izvještaj se generira i šalje administratorima kako bi identificirali uzrok problema i poduzeli potrebne korake za održavanje stabilnosti i performansi aplikacije.

4.4. Puštanje aplikacije u rad

Kako bi se testiranje smatralo uspješnim, aplikacija mora raditi bez greške. Nakon plasiranja aplikacije, ista će biti dostupna putem App trgovina u aplikacijama Google Play za Android

sustave, te u App Store za iPhone korisnike. Osim na navedene načine, aplikacija će biti dostupna i putem web sjedišta.

Važno je napomenuti kako su korisnici samim pristupanjem aplikaciji obvezni poštivati uvjete korištenja aplikacije, a samom registracijom ih prihvaćaju. Pri registraciji korisnik jamči da ima najmanje šesnaest godina. U slučaju da je korisnik mlađi od šesnaest, potreban je pristanak roditelja ili skrbnika koji pristaju na to da se u ime maloljetnika obvezuju na propisane uvjete korištenja aplikacije.

Korisnik jamči autorstvo nad sadržajem koji objavljuje. Objave korisnika moraju biti u skladu sa svim zakonskim i pravnim propisima komunikacije i objave sadržaja. Registrirani se korisnici obvezuju da svoju lozinku neće dijeliti s trećim osobama. Pri odabiru nadimka ili korisničkoga imena, zabranjeno je birati imena javnih ili privatnih osoba, robnu marku ili riječima koje bi za pojedine mogle djelovati uvredljivo.

Korisniku nije dopušteno korištenje vlastitoga korisničkoga računa tako da prouzroči štetu ili naruši uslugu i ugled aplikacije. Također, korisnik ne smije ni na koji način kršiti zakonske propise u vidu nanošenja bilo kojega oblika štete trećoj strani, odnosno drugim korisnicima. Korisnik je odgovoran za sadržaj i vrstu sadržaja koji objavljuje sukladno zakonskim propisima i pravilima korištenja aplikacije.

Također, korisnik je odgovoran za istinitost i točnost informacija, te se kreatori aplikacije ograđuju od osobnih stavova korisnika. Nastavno, korisnicima je strogo zabranjeno postavljanje linkova koji bi pri otvaranju na bilo koji način mogli naštetiti uređajima drugih korisnika.

U slučaju kršenja navedenih pravila, korisnici će biti sankcionirani privremenom ili trajnom zabranom prijave u korisnički profil.

5. ZAKLJUČAK

U današnjem dinamičnom okruženju brze tehnološke evolucije, razvoj aplikacija igra ključnu ulogu u unapređenju različitih segmenata društvenoga i poslovnoga života. Moderna društva sve više ovise o tehnološkim rješenjima koja olakšavaju organizaciju, komunikaciju i učinkovito upravljanje raznovrsnim aktivnostima. Završni rad usmjerio se na istraživanje i praktičnu primjenu koncepta razvoja aplikacije za vođenje navijačkih skupina korištenjem Reacta, razvojnoga okvira za kreiranje korisničkih sučelja.

Analizom stručnih izvora, uočeni su ključni aspekti u kreiranju interaktivnih i dinamičkih korisničkih sučelja. Praktična implementacija rezultirala je stvaranjem aplikacije za vođenje navijačkih skupina. Aplikacija je omogućila organizaciju događanja unutar skupina, međusobnu komunikaciju članova te olakšala praćenje ključnih informacija. Izrađeni praktični segment potvrdio je važnost primjene teorijskoga znanja u rješavanju stvarnih problema.

Tehnologije poput Reacta igraju ključnu ulogu u razvoju suvremenih aplikacija. Mogućnost stvaranja ponovno upotrebljivih komponenata omogućava poduzetni i učinkoviti razvoj, što je izraženo i u praktičnoj izradi aplikacije. Rad naglašava ne samo važnost teorijskoga razumijevanja tehnologije već i njegovu primjenu u stvarnom svijetu, dok teži k unapređenju korisničkoga iskustva i organizacije unutar navijačkih skupina.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU

Bana Josipa Jelačića 22/a, Čakovec

IZJAVA O AUTORSTVU

Završni/diplomski rad isključivo je autorsko djelo studenta te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, internetskih i drugih izvora) bez pravilnog citiranja. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom i nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Nikola Jančec (ime i prezime studenta) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog rada pod naslovom : WEB APLIKACIJA ZA VOĐENJE NAVIJAČKIH SKUPINA U NOGOMETU te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:



(vlastoručni potpis)

LITERATURA

- [1] Rouse, M. (2022). Programming Tool. Dostupno na: <https://www.techopedia.com/definition/8996/programmingtool#:~:text=A%20programming%20tool%20may%20be,as%20a%20software%20development%20tool>. (Pristupljeno: 01.08.2023.)
- [2] Geeks for Geeks. What is Software Tool?. Dostupno na: <https://www.geeksforgeeks.org/what-is-software-tool/>. (Pristupljeno: 25.07.2023.)
- [3] Tuama, D. Code Institute. Visual Studio Code (VS Code): What It Is and Why Developers Love It. Dostupno na: <https://codeinstitute.net/global/blog/what-is-vs-code/> (Pristupljeno: 10.02.2024.)
- [4] Rouse, M. (2016). JavaScript. Dostupno na: <https://www.techopedia.com/definition/3929/javascript-js>. (20.07.2023.)
- [5] Cindrić, S. (2020). Što je JavaScript kod web stranica?. Dostupno na: <https://znatko.com/9456/sto-je-javascript>. (Pristupljeno: 26.02.2022)
- [6] Developer Mozilla. Org. What is JavaScript?. Dostupno na: <https://developer.mozilla.org/en->

US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. (Pristupljeno: 20.02.2022.)

- [7] Ragala, BK. (2022). What are the Pros and Cons of React. Dostupno na: <https://www.knowledgehut.com/blog/web-development/pros-and-cons-of-react#what-is-react-js?%C2%A0>. (Pristupljeno: 01.08.2023.)
- [8] React.js. (2023). Top Industry Verticals Industry verticals where React JS is being used. Dostupno na: <https://www.similartech.com/technologies/react-js>. (Pristupljeno: 25.08.2023).
- [9] Geeks for Geeks. What are the advantages of React.js ?. Dostupno na: <https://www.geeksforgeeks.org/what-are-the-advantages-of-react-js/>. (Pristupljeno: 20.08.2023)
- [10] Ragala, BK. (2023). What Is a React Component?. Dostupno na: <https://www.knowledgehut.com/blog/web-development/react-component#what-are-the-usesof-class-components?%C2%A0>. (Pristupljeno: 17.08.2023.)
- [11] React. Start a New React Project. Dostupno na: <https://react.dev/learn/start-a-new-reactproject> (Pristupljeno: 10.08.2023.)
- [12] Hamilton, T. (2023). What is Application Testing?. Dostupno na: <https://www.guru99.com/application-testing.html>. (Pristupljeno: 10.08.2023.)
- [13] Softwarwe Testing Help. (2023). Application Testing – Into The Basics Of Software Testing. Dostupno na: <https://www.softwaretestinghelp.com/application-testing-into-thebasics-of-software-testing/>. (Pristupljeno: 07.08.202

POPIS SLIKA I GRAFOVA

Slika 1: Primjena Reacta po industrijama	5
Slika 2: Korisničko sučelje aplikacije	11
Slika 3: Registracija korisnika	12
Slika 4: Verifikacijska poruka putem e-maila korisnika	12
Slika 5: Validacija pri registraciji novih korisnika	13
Slika 6: Korisničko sučelje Internet trgovine	14
Slika 7: Kupovina u webshopu	14
Slika 8: Shema cijele baze	15
Slika 9: Primjer rezultata u tražilici pri pretraživanju pojmova	16
Slika 10: Primjer rezultata u tražilici pri pretraživanju korisnika	17
Slika 11: Primjer rezultata u tražilici pri pretraživanju po datumu	17
Slika 12: Sučelje administratora	19
Slika 13: Dijagram procesa	20