

Web portal nogometne navijačke skupine

Golemović, Mario

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:584795>

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

Download date / Datum preuzimanja: **2024-11-01**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni studij računarstva

WEB PORTAL NOGOMETNE NAVIJAČKE SKUPINE

Završni rad

Mario Golemović

Osijek, 2024.

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. POSTOJEĆA RJEŠENJA	2
2.1. Kohorta Osijek	2
2.2. Torcida – KN Torcida Split	3
2.3. Horde zla – Udruženje navijača FK Sarajevo	4
3. TEHNOLOGIJE ISKORIŠTENE ZA IZRADU WEB APLIKACIJE	5
3.1. HTML	6
3.2. CSS	7
3.3. JavaScript	8
3.4. Vue.js	9
3.5. Express.js	10
3.6. MongoDB	11
3.7. Visual Studio Code	12
4. FUNKCIONALNOST WEB APLIKACIJE	13
4.1. Baza podataka	13
4.2. Prijava i registracija	14
4.3. Odjava	17
4.4. Administrator	18
4.5. Korisnici	21
5. KORIŠTENJE WEB APLIKACIJE	25
5.1. Pokretanje aplikacije	25
5.2. Početni prikaz stranice	26
5.3. Raznolikost stranice	27
5.4. Uloga administratora i prijavljenih korisnika	33
5.5. Opis mogućnost aplikacije	34
6. ZAKLJUČAK	39
LITERATURA	40
SAŽETAK	41
ABSTRACT	42

1. UVOD

Cilj napravljene aplikacije je da se pojednostavi organizacija odlaska na utakmice i okupi sve ljubitelje nogometnog kluba "Real Madrid", bez obzira na njihovu životnu dob. Nogomet, kao najpopularniji sport na svijetu, privlači raznoliku publiku, od ljudi koji prate samo najbolje utakmice najvećih klubova bez pretjerane ljubavi prema određenom klubu do obitelji kojima je ljubav prema određenom klubu smisao života. Često se suočavamo s poteškoćama u organizaciji takvih putovanja, kao što su pronalazak odgovarajućih autobusa, rezervacija smještaja i nabava karata. Ova web aplikacija je osmišljena kako bi riješila te probleme te korisnicima pružila jednostavno iskustvo planiranja odlaska na utakmicu te uvid u novosti i događanja vezana za sami klub. Kako bi se riješili ovi problemi, ovaj rad koristi niz tehnologija, uključujući HTML, CSS, Vue.js i Express.js, dok se za bazu podataka koristi MongoDB. Kombinacija ovih tehnologija omogućila je stvaranje web portala koji daje sveobuhvatno korisničko iskustvo. Korištenje Visual Studio Code-a kao razvojne okoline osigurava efikasno razvijanje i održavanje projekta.

Poglavlja su strukturirana u četiri glavna dijela. Prvi dio obuhvaća analizu postojećih rješenja i web stranica namijenjenih ljubiteljima drugih nogometnih klubova kako bi se stekao bolji uvid u ovakav način okupljanja ljubitelja određenog kluba. Drugo poglavlje detaljno razmatra korištene tehnologije za razvoj web aplikacije. Treće poglavlje opisuje funkcionalnosti koje web aplikacija pruža svojim korisnicima, dok se u četvrtom poglavlju analizira uporaba aplikacije kroz perspektive administratora i članova stranice.

1.1. Zadatak završnog rada

Glavni cilj ovog završnog rada je stvaranje web portala koji okuplja nogometnu navijačku skupinu. Portal omogućava registraciju članova, objavu različitih članaka kao i informacije o svom nogometnom klubu, organizaciju dobrotvornih akcija i odlaska na utakmice, prijevoza i slično. Administrator je zadužen za objavu članaka, praćenje komentara registriranih korisnika. Svi koji se žele pridružiti navijačkoj skupini na mreži moraju se registrirati na portalu. Registrirani korisnici mogu komentirati i biti aktivno uključeni u navijačku skupinu.

2. POSTOJEĆA RJEŠENJA

Ovo poglavlje analizira postojeće web portale namijenjene nogometnim navijačkim skupinama. U fokusu su sljedeći portali:

- Kohorta Osijek
- Torcida – KN Torcida Split
- Horde zla – Udruženje navijača FK Sarajevo

2.1. Kohorta Osijek

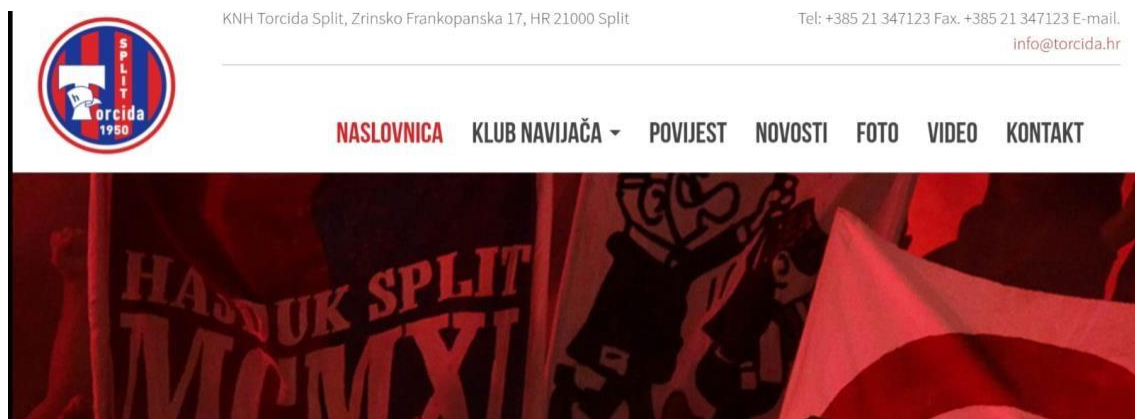
Na početnoj stranici Kohorte Osijek nalazi se pregledna navigacija koja obuhvaća galeriju slika, kontakt informacije, najnovije vijesti te druge korisne rubrike. Galerija slika pruža uvid u prošla događanja i aktivnosti organizirane od strane Kohorte Osijek, omogućujući posjetiteljima da dožive atmosferu kroz vizualni prikaz. U rubrici Kontakt, posjetitelji će pronaći relevantne informacije o načinima komunikacije s Kohortom Osijek. Bilo da je riječ o pitanjima, podršci ili suradnji, kontakt informacije olakšavaju korisnicima da stupe u kontakt s odgovornima unutar zajednice. Rubrika Novosti redovito se ažurira s najnovijim informacijama o projektima, događanjima ili drugim značajnim aktivnostima koje su trenutačno u tijeku. To pruža posjetiteljima kontinuiran uvid u događanja unutar Kohorte Osijek. Uz to, korisnici mogu istraživati i druge dijelove stranice, ovisno o njihovim interesima i potrebama. Na slici 2.1. prikazana je glavna stranica Kohorte Osijek. [1].



Slika 2.1. Početni prikaz glavnog dijela stranice Kohorta Osijek.

2.2. Torcida – KN Torcida Split

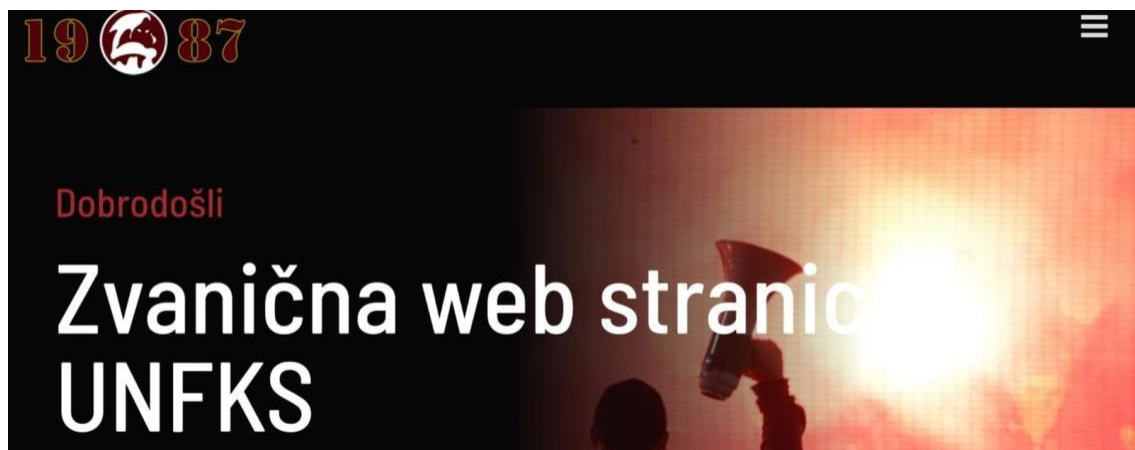
Ova platforma pruža uvid u dinamičan svijet strasti, zajedništva i ljubavi prema Hajduku. Galerija slika na stranici otkriva intenzitet emocija koje se proživljavaju na stadionu, izvan terena i tijekom raznih događanja koje se organiziraju. Svaka slika je priča za sebe, zabilježena u duhu prave sportske strasti. U rubrici Kontakt, posjetitelji će pronaći sve potrebne informacije za uspostavu komunikacije s članovima Torcide. Bez obzira na to jeste li strastveni navijač, potencijalni član ili jednostavno imate pitanje. Novosti prate puls događanja unutar KN Torcida Split. Ovdje se mogu saznati najnovije informacije, planirani događaji i projekti koji oblikuju identitet ove strastvene navijačke zajednice. Svi dijelovi stranice pružaju uvid u povijest, vrijednosti i ciljeve Torcide. Ova stranica nije samo mjesto za informacije, već i prozor u svijet gdje se rađaju nezaboravni trenuci navijanja i ljubavi prema klubu. Torcida Split - mjesto gdje se srce navijača stopi s ritmom stadiona. Na slici 2.2. prikazan je glavni dio Torcidine stranice [2].



Slika 2.2. Početni prikaz glavnog dijela stranice Torcida – KN Torcida Split.

2.3. Horde zla – Udruženje navijača FK Sarajevo

Ova stranica pruža ekskluzivan uvid u živopisni svijet navijačke skupine koja s ponosom podržava svoj klub. U galeriji slika, zaronite u mnoštvo trenutaka snimljenih tijekom strastvenih navijačkih okupljanja na stadionu i izvan njega. Svaka fotografija nosi vlastitu priču, oživljavajući duh zajedništva i ljubavi prema nogometu. Kroz rubriku Kontakt, posjetitelji mogu pronaći sredstva komunikacije s predstavnicima Horde Zla Sarajevo. Ova otvorena linija komunikacije stoji na raspolaganju za sva pitanja, podršku ili izraze interesa za pridruživanje navijačkom pokretu. Rubrika Novosti redovito ažurira članove i simpatizere o najnovijim događajima unutar Horde Zla. Od najava budućih događanja do projekata koje podržavaju, posjetitelji će biti uvijek informirani o aktivnostima koje oblikuju karakter ove strastvene zajednice. Svi dijelovi stranice nude uvid u povijest, vrijednosti i ciljeve Horde Zla Sarajevo, stvarajući online prostor gdje se rađaju nezaboravni trenuci navijanja i posvećenosti klubu. Horde Zla Sarajevo - gdje svaki klik nosi ritam navijačke strasti. Na slici 2.3. prikazana je početna stranica Sarajevske navijačke skupine [3].



Slika 2.3. Početni prikaz glavnog dijela stranice Horde zla – Udruženje navijača FK Sarajevo.

3. TEHNOLOGIJE ISKORIŠTENE ZA IZRADU WEB APLIKACIJE

“Backend“ i “frontend“ su dva ključna dijela jedne web aplikacije, a njihova suradnja omogućuje da web stranica ili aplikacija pravilno funkcionira.

“Frontend“ predstavlja korisničko sučelje koje korisnici vide i s čime interagiraju. Ovo uključuje sve vizualne elemente poput teksta, slika, gumba i izbornika. “Frontend“ programeri koriste jezike poput JavaScripta, HTML-a i CSS-a kako bi stvorili atraktivno i interaktivno korisničko iskustvo.

S druge strane, “backend“ predstavlja "mozak" web aplikacije. To je skriveni dio sustava koji se brine za obradu podataka, upravljanje bazom podataka, autentikaciju korisnika i sve druge logičke operacije koje su potrebne za funkcioniranje aplikacije. “Backend“ prima zahtjeve od “frontend“ dijela, obrađuje ih, pristupa bazi podataka ako je potrebno, te zatim šalje odgovore ili informacije “frontendu“.

Ukratko, “frontend“ se bavi onim što korisnici vide i kako s aplikacijom interagiraju, dok “backend“ upravlja logikom i podacima iza scene. Kada korisnik, na primjer, klikne gumb na “frontendu“, “frontend“ šalje zahtjev “backendu“, koji tada izvršava potrebne operacije, obrađuje podatke i šalje odgovor natrag “frontendu“, koji ga zatim prikazuje korisniku. Ova međusobna komunikacija između “frontend“ i “backend“ dijela omogućuje potpuno funkcioniranje web aplikacije. [4].

3.1. HTML

HTML (engl. *Hypertext Markup Language*) je jezik označavanja koji služi za strukturiranje sadržaja na web stranicama. Koristi posebne oznake (engl. tags) za definiranje različitih elemenata na stranici, poput naslova, paragrafa, slika i poveznica. HTML omogućuje programerima da organiziraju informacije na jasan način, čime se omogućuje preglednicima da pravilno interpretiraju i prikažu te informacije korisnicima. Osim toga, HTML se često kombinira s drugim tehnologijama poput CSS-a za stiliziranje i JavaScripta za dodavanje interaktivnosti na web stranicama. Logo HTML-a nalazi se na slici 3.1.[5].



Slika 3.1. *HTML logo.*

3.2. CSS

CSS (engl. *Cascading Style Sheets*) je jezik za stiliziranje web stranica. Primjenjuje se za definiranje izgleda elemenata koji su označeni u HTML dokumentu. CSS omogućava prilagodbu boje, veličine, rasporeda i drugih vizualnih svojstava web stranica. Stilovi se primjenjuju putem selektora koji određuju koji će se elementi stilizirati. Na primjer, možete stilizirati sve paragrafe na stranici ili samo određeni element s jedinstvenim identifikatorom.

CSS pruža fleksibilnost i odvaja prezentaciju od strukture HTML-a. To znači da možete promijeniti izgled više stranica samo mijenjajući jedan CSS dokument. Osim toga, CSS podržava koncept kaskadnosti, gdje se stilovi primjenjuju prema određenom redosljedu i mogu se naslijediti od roditeljskih elemenata.

Ukratko, CSS omogućava dizajniranje privlačnih i vizualno ugodnih web stranica, čineći ih korisnički prilagodljivima i estetski ugodnima. Logo CSS-a nalazi se na slici 3.2. [6].



Slika 3.2. Logo CSS-a.

3.3. JavaScript

JavaScript je svestrani programski jezik koji igra ključnu ulogu u razvoju interaktivnih i dinamičnih web stranica. Ova tehnologija omogućuje programerima stvaranje korisničkih iskustava koja reagiraju na korisničke akcije, ažuriraju se dinamički i komuniciraju s poslužiteljem bez potrebe za osvježavanjem cijele stranice.

Jedna od ključnih značajki JavaScripta je sposobnost manipulacije DOM-om (engl. *Document Object Model*), što omogućuje programerima da pristupaju, mijenjaju i ažuriraju elemente na web stranici. Osim toga, JavaScript podržava asinkrono izvršavanje koda, što znači da se određeni dijelovi koda mogu izvršavati neovisno o ostatku aplikacije, čime se poboljšava performanse.

JavaScript se često koristi za validaciju obrazaca, animacije, rad s AJAX-om (engl. *Asynchronous Java Script And XML*) za dinamičko učitavanje podataka te za upravljanje događajima kao što su klikanje mišem ili pritiskanje tipki. Također, uz Node.js, JavaScript se može koristiti i na serverskoj strani, čime se omogućava cjeloviti razvoj web aplikacija koristeći jedan programski jezik.

Svojom fleksibilnošću i moćnim mogućnostima, JavaScript je postao ključni alat u svijetu web razvoja, pridonoseći dinamici i interaktivnosti modernih web stranica. Logo JavaScripta prikazan je na slici 3.3. [7].



Slika 3.3. Logo JavaScripta

3.4. Vue.js

Vue.js je JavaScript razvojni okvir koji se primjenjuje za izgradnju korisničkih sučelja. Ovaj razvojni okvir naglašava jednostavnost integracije i fleksibilnost, čineći ga popularnim izborom za razvoj jednostraničnih aplikacija SPA (engl. *Single-page application*) i složenijih korisničkih sučelja.

Jedna od ključnih prednosti Vue.js-a je njegova lakoća u učenju i upotrebi. Sintaksa Vue.js-a je jasna i slična HTML-u, što olakšava razvoj, održavanje i proširenje projekata. Vue.js se može koristiti postupno, što znači da se može integrirati u postojeće projekte bez potrebe za temeljitim preuređivanjem koda.

Vue.js pruža modularnu arhitekturu, gdje se aplikacija sastoji od komponenti koje se mogu ponovno koristiti. Ove komponente omogućuju organiziranje koda na logičan način, poboljšavaju čitljivost i pojednostavljuju održavanje.

Još jedna važna značajka Vue.js-a je dvosmjerno povezivanje podataka, poznato kao "two-way data binding". To znači da se promjene u korisničkom sučelju automatski odražavaju u podacima i obrnuto, čime se smanjuje potreba za ručnim upravljanjem stanjem aplikacije.

Uz to, Vue.js zajednica je aktivna, a razvojni okvir se redovito ažurira, pružajući podršku za nove tehnologije i pristupajući najnovijim trendovima u razvoju web aplikacija. Sve ove značajke čine Vue.js snažnim alatom za razvoj modernih i efikasnih korisničkih interfejsa. Logo Vue.js-a nalazi se na slici 3.4. [8].



Slika 3.4. *Vue.js logo.*

3.5. Express.js

Express.js je minimalistički i fleksibilni web razvojni okvir za Node.js, dizajniran za olakšavanje izrade web aplikacija i API-ja (engl. *Application Programming Interface*). Ovaj razvojni okvir naglašava jednostavnost, performanse i mogućnost brze integracije s drugim paketima i modulima Node.js ekosustava.

Express.js olakšava usmjeravanje, upravljanje zahtjevima i odgovorima, te obradu “middleware-a“. Middleware su funkcije koje se izvršavaju tijekom obrade zahtjeva, omogućujući prilagodbu i proširenje funkcionalnosti aplikacije. Express.js također pruža jednostavnu sintaksu za rad s HTTP zahtjevima, parametrima usmjeravanja i stvaranje dinamičkih web stranica.

Jedna od prednosti Express.js-a je njegova otvorenost prema integraciji s različitim pogonskim sustavima predložaka, bazama podataka i drugim komponentama. To omogućuje programerima da odaberu tehnologije koje najbolje odgovaraju potrebama njihovog projekta.

Express.js se često koristi u kombinaciji s drugim tehnologijama, poput MongoDB za baze podataka, Angular ili React za korisnička sučelja. Ovaj pristup omogućuje izradu potpunog web stoga koristeći JavaScript na obje strane - na klijentskoj (frontend) i poslužiteljskoj (backend) strani.

Sa svojom jednostavnom i proširivom arhitekturom, Express.js je postao popularan izbor za brz i efikasan razvoj web aplikacija i API-ja u Node.js ekosustavu. Logo Node.js-a nalazi se na slici 3.5. [9].



Slika 3.5. Node.js logo.

3.6. MongoDB

MongoDB je dokumentna, NoSQL baza podataka koja se ističe svojom fleksibilnošću i sposobnošću rada s nestrukturiranim ili polustrukturiranim podacima. Ova baza podataka često se koristi u aplikacijama koje zahtijevaju skalabilnost i brz pristup podacima.

MongoDB koristi BSON (Binary JSON (engl. *JavaScript Object Notacion*)) format za pohranu podataka, što znači da podaci unutar baze podataka imaju oblik JSON dokumenata. Svaki dokument predstavlja zaseban entitet i može sadržavati različite vrste podataka, uključujući nizove, ugniježdene dokumente i druge složene strukture.

Jedna od glavnih prednosti MongoDB-a je njegova sposobnost horizontalnog skaliranja, što znači da se može lako proširivati dodavanjem novih čvorova u klaster. Ova skalabilnost čini MongoDB popularnim rješenjem za velike i rastuće aplikacije.

MongoDB nudi bogat upitni jezik koji omogućuje napredno pretraživanje, filtriranje i analizu podataka. Također podržava indekse, što povećava brzinu pretraživanja i upita, posebno kod velikih količina podataka.

Osim toga, MongoDB podržava replikaciju i razdvajanje baze na manje dijelove zbog poboljšanja otpornosti na kvarove i raspodjelu opterećenja.

U web razvoju, MongoDB se često koristi kao “backend“ baza podataka u kombinaciji s Express.js i Node.js. Ova kombinacija, poznata kao MEAN (MongoDB, Express.js, Angular, Node.js) stog, omogućuje izradu cjelovitog web stoga koristeći JavaScript na svim razinama aplikacije. Logo MongoDB-a nalazi se na slici 3.6. [10].



Slika 3.6. MongoDB logo.

3.7. Visual Studio Code

Visual Studio Code, razvijen od strane Microsofta, predstavlja integrirano razvojno okruženje koje se izdvaja po svojoj jednostavnosti korištenja, podršci za različite programske jezike te fleksibilnosti. Ovaj besplatni IDE (engl. *Integrated development environment*) otvorenog koda pruža intuitivno korisničko sučelje, prilagodljive opcije i bogat ekosustav proširenja.

Sposoban podržati više programskih jezika, kao što su JavaScript, Python, Java i mnoge druge, VS Code je postao popularan odabir među programerima različitih profila. Integracija s Git-om olakšava upravljanje verzijama i timskim radom, dok inteligentni sustav preporuka i alati za uklanjanje pogrešaka doprinose efikasnosti u razvoju.

Unutar VS Code-a ističe se i aktivna zajednica korisnika te mogućnost prilagodbe putem proširenja, pružajući programerima potrebne resurse za poboljšanje produktivnosti. Ovo razvojno okruženje ima sveobuhvatan pristup podršci različitih tehnologija, čime osigurava široku primjenu u raznim razvojnim scenarijima. Logo Visual Studio Code-a nalazi se na slici 3.7. [11].



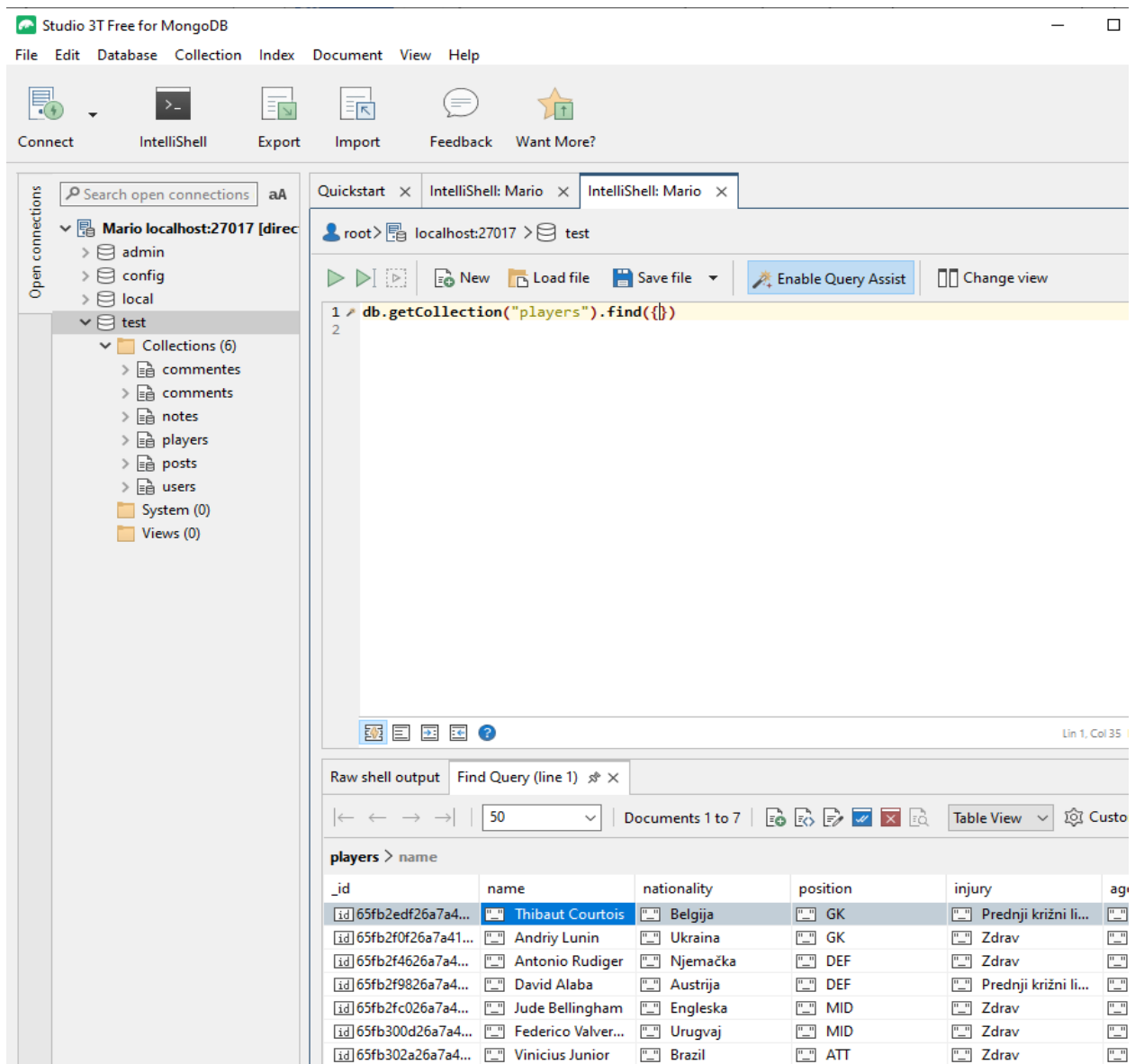
Slika 3.7. *Visual Studio Code logo.*

4. FUNKCIONALNOST WEB APLIKACIJE

U ovom poglavlju detaljno su opisane funkcionalnosti koje su implementirane u izradi web portala za nogometnu navijačku skupinu.

4.1. Baza podataka

Za postavljanje MongoDB baze podataka korišteni su alati *Docker* i *Robo 3T*. U bazi podataka bilo je potrebno napraviti šest tablica, a to su *commentes*, *comments*, *notes*, *players*, *posts* i *users*. Na slici 4.1. nalazi se prikaz tablica koji se koriste za aplikaciju.



The screenshot displays the Studio 3T Free for MongoDB interface. On the left, the 'Open connections' sidebar shows a connection to 'Mario localhost:27017 [direct]'. Underneath, a tree view lists collections: 'admin', 'config', 'local', and 'test'. The 'test' collection is expanded, showing sub-collections: 'Collections (6)', 'comments', 'comments', 'notes', 'players', 'posts', 'users', 'System (0)', and 'Views (0)'. The main workspace shows the 'IntelliShell: Mario' tab with a query editor containing the command: `1 db.getCollection("players").find({})`. Below the editor, the 'Raw shell output' section shows a table view of the 'players' collection. The table has columns: '_id', 'name', 'nationality', 'position', 'injury', and 'ag'. The data rows are as follows:

_id	name	nationality	position	injury	ag
[id] 65fb2edf26a7a4...	Thibaut Courtois	Belgija	GK	Prednji križni li...	
[id] 65fb2f0f26a7a41...	Andriy Lunin	Ukraina	GK	Zdrav	
[id] 65fb2f4626a7a4...	Antonio Rudiger	Njemačka	DEF	Zdrav	
[id] 65fb2f9826a7a4...	David Alaba	Austrija	DEF	Prednji križni li...	
[id] 65fb2fc026a7a4...	Jude Bellingham	Engleska	MID	Zdrav	
[id] 65fb300d26a7a4...	Federico Valver...	Urugvaj	MID	Zdrav	
[id] 65fb302a26a7a4...	Vinicius Junior	Brazil	ATT	Zdrav	

Slika 4.1. Prikaz tablica u Robo 3T

4.2. Prijava i registracija

Da biste pristupili svim funkcionalnostima na stranici, korisnik se mora prijaviti ili registrirati. Ako korisnik prvi put posjećuje stranicu, potrebno je izvršiti registraciju. Registracija se vrši klikom na link u navigacijskoj traci, nakon čega se otvara stranica koja sadrži obrazac u koji je potrebno upisati korisničko ime, email i lozinku, a zatim stisnuti na gumb Registriraj.

Slika 4.2 prikazuje kod registracije.

```
static async signup({ username, email, password }) {
  try {
    if (!username || !email || !password) {
      return "Required fields are missing";
    }
    if (typeof username !== "string" || typeof email !== "string" || typeof password !== "string") {
      return "Required fields are not string";
    }
    if (username.length < 3 || username.length > 50) {
      return "Please write a username not longer than 50 characters and not less than 3 characters";
    }
    if (!validator.isEmail(email)) {
      return "Not a valid email format";
    }
    if (password.length < 8 || password.length > 50) {
      return "Please write a password not longer than 50 characters and not less than 8 characters!";
    }
    const existingEmail = await userModel.findOne({ email: email });
    const existingUserName = await userModel.findOne({ username: username });
    if (existingEmail) {
      throw "EMAIL ALREADY EXISTS";
    }
    if (existingUserName) {
      throw "USERNAME ALREADY EXISTS";
    }
    const hashedPassword = await bcrypt.hash(password, 10);
    const result = await userModel.create({
      email: email,
      password: hashedPassword,
      username: username,
    });
    const token = jwt.sign(
      { email: result.email, id: result.id },
      SECRET_KEY,
      { expiresIn: "3h" }
    );
    return {
      username: result.username,
      email: result.email,
      token,
    };
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.2. Prikaz nodejs koda za registraciju.

Korisnik koji se prethodno registrirao na stranicu može odabrati opciju "Prijava" u navigacijskoj traci kako bi se ponovno prijavio na stranicu. Za prijavu je potrebno unijeti ispravnu e-mail adresu, zaporku te kliknuti na gumb "Prijava". Slika 4.3. prikazuje kod prijave.

```
static async signin({ email, password }) {
  try {
    if (!email || !password) {
      return "Required fields are missing";
    }
    if (typeof email !== "string" || typeof password !== "string") {
      return "Required fields are not string";
    }
    const existingUser = await userModel.findOne({ email: email });
    if (!existingUser) {
      throw "USER NOT FOUND";
    }
    const matchPassword = await bcrypt.compare(
      password,
      existingUser.password
    );
    if (!matchPassword) {
      throw "INVALID PASSWORD";
    }
    const token = jwt.sign(
      {
        role: existingUser.role,
        email: existingUser.email,
        id: existingUser.id,
      },
      SECRET_KEY,
      { expiresIn: "3h" }
    );
    return {
      email: existingUser.email,
      role: existingUser.role,
      id: existingUser.id,
      token,
    };
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.3. Prikaz nodejs koda za prijavu.

Prilikom prijave koriste se jwt tokeni. Na slici 4.4. vidljiv je kod koji predstavlja middleware funkciju u Node.js aplikaciji koja koristi JSON Web Token za autentifikaciju korisnika.

```
const jwt = require("jsonwebtoken");
const SECRET_KEY = "NOTESAPI";
module.exports = (req, res, next) => {
  try {
    let token = req.headers.authorization;
    if (token) {
      token = token.split(" ")[1];
      let user = jwt.verify(token, SECRET_KEY);
      req.userId = user.id;
      req.email = user.email;
    } else {
      return res.status(401).json({ message: "Unauthorized user" });
    }
    next();
  } catch (error) {
    console.log(error);
    return res.status(401).json({ message: "Unauthorized user" });
  }
};
```

Slika 4.4. Prikaz nodejs koda za dodjeljivanje tokena.

Na stranici za prijavu nalazi se opcija "Zaboravljena lozinka", koja korisnicima omogućuje da otvore stranicu gdje mogu unijeti svoju e-mail adresu i postaviti novu lozinku.. Zatim se klikom na gumb potvrđuje nova zaporka. Na slici 4.5. vidljiv je kod za ažuriranje zaporka.

```
static async resetPassword({ email, newPassword }) {
  try {
    if (!email || !newPassword) {
      return "Email and new password are required";
    }

    const existingUser = await userModel.findOne({ email });

    if (!existingUser) {
      return "User with this email does not exist";
    }

    const hashedPassword = await bcrypt.hash(newPassword, 10);

    await userModel.findByIdAndUpdate(existingUser._id, {
      password: hashedPassword,
    });

    return "Password reset successfully";
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.5. Prikaz nodejs koda za ažuriranje lozinke

4.3. Odjava

Kada završe s radom na aplikaciji, administratori i prijavljeni korisnici mogu se odjaviti sa stranice. Time se poništava token. Kod za odjavu nalazi se na slici 4.5.

```
import { defineStore } from 'pinia'
import jwt_decode from 'jwt-decode'

export const useUserStore = defineStore('user', {
  state: () => {
    const token = localStorage.getItem('token')
    const user = token ? jwt_decode(token) : false
    return {
      user
    }
  },
  getters: {
    userEmail() {
      return this.user ? this.user.email : null;
    },
    UserId() {
      return this.user ? this.user.id : null;
    },
  },
  actions: {
    signin(user) {
      this.user = user
      localStorage.setItem('token', user.token)
    },
    signout() {
      this.user = false

      localStorage.removeItem('token')
    }
  },
  persist: true
})
```

Slika 4.5. Prikaz koda za odjavu.

4.4. Administrator

Administrator prilikom prijave na stranicu prolazi kroz iste korake kao i ostali korisnici. Nakon prijave, administratoru se prikazuje drugačija navigacijska traka i ima pristup različitim funkcionalnostima. U navigacijskoj traci administrator vidi dodatne gumbове *Dodaj igrača*, *Dodaj utakmicu* i *Dodaj novost*. Klikom na gumb *Dodaj igrača*, administrator ima mogućnost dodavanja novih igrača i njihovih statističkih podataka u bazu. Prilikom uspješnog dodavanja igrača, igrač se izlistava iz baze na stranici za igrače. Klikom na gumbове *Dodaj utakmicu* i *Dodaj novost*, administrator ima mogućnost dodavanja novih obavijesti vezanih za klub i organizaciju odlaska na utakmice u bazu podataka. Prilikom uspješnih dodavanja obavijesti one se izlistavaju iz baze na stranicama *Novosti i Utakmice*. Tada se uz podatke o igračima i novostima izlistanih iz baze prikazuje i mogućnost uređivanja i brisanja. Kod za prepoznavanje administratorskog korisničkog računa nalazi se na slici 4.6.

```
const jwt = require("jsonwebtoken");
const SECRET_KEY = "NOTESAPI";
module.exports = (req, res, next) => {
  try {
    let payload = jwt.decode(req.headers.authorization.split(" ")[1]);
    if (payload.email !== "mariogolemovic12@gmail.com") {
      return res.status(401).json({ message: "Ne moze!" });
    }

    next();
  } catch (error) {
    console.log(error);
    return res.status(401).json({ message: "Unauthorized user" });
  }
};
```

Slika 4.6. Prikaz koda za prepoznavanje adminskeg korisničkog računa.

Kod za dodavanje administratorskih ovlasti nad određenim funkcijama nalazi se na slici 4.7.

```
const express = require("express");
const {
  getPost,
  getPosts,
  createPost,
  deletePost,
  updatePost,
} = require("../controllers/PostController");
const auth = require("../middlewares/auth");
const checkRole = require("../middlewares/checkRole");
const postRouter = express.Router();

postRouter.get("/", getPost);

postRouter.get("/posts", getPosts);

postRouter.post("/", auth, checkRole, createPost);

postRouter.delete("/", auth, checkRole, deletePost);

postRouter.put("/", auth, checkRole, updatePost);

module.exports = postRouter;
```

Slika 4.7. Prikaz koda za dodavanje adminskih ovlasti nad funkcijama.

Kod za kreiranje objava od strane administratora nalazi se na slici 4.8.

```
static async createPost({ title, description, userId }) {
  try {
    if (!title || !description) {
      return "Required fields are missing";
    }
    if (typeof title !== "string" || typeof description !== "string") {
      return "Required fields are not string";
    }
    if (title.length < 5 || title.length > 50) {
      return "Please write a title not longer than 50 characters";
    }
    if (description.length < 20 || description.length > 255) {
      return "Please write a description not longer than 255 characters";
    }
  }

  const userPost = await postModel.create({
    title,
    description,
    userId,
  });

  return {
    title: userPost.title,
    description: userPost.description,
    userId: userPost.userId,
    id: userPost.id,
  };
} catch (error) {
  console.log(error);
  throw error;
}
```

Slika 4.8. Prikaz koda za kreiranje objava.

Kod za ažuriranje objava od strane administratora nalazi se na slici 4.9.

```
static async updatePost({ title, description, id }) {
  try {
    if (!title || !description || !id) {
      return "Required fields are missing";
    }
    if (
      typeof title !== "string" ||
      typeof description !== "string" ||
      typeof id !== "string"
    ) {
      return "Required fields are not string";
    }
    if (title.length < 5 || title.length > 50) {
      return "Please write a title not longer than 50 characters";
    }
    if (description.length < 20 || description.length > 255) {
      return "Please write a description not longer than 255 characters";
    }

    const post = await postModel.findByIdAndUpdate(
      { _id: id },
      {
        title: title,
        description: description,
      }
    );

    if (!post) {
      return "Something went wrong, check the id";
    }

    return {
      id: post.id,
      title: post.title,
      description: post.description,
    };
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.9. Prikaz koda za ažuriranje objava.

Kod za brisanje objava od strane administratora nalazi se na slici 4.10.

```
static async deletePost(id) {
  try {
    const post = await postModel.findByIdAndDelete(id);

    if (!post) {
      return "Something went wrong, check the id";
    }

    return "Great job! Your post is successfully deleted";
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.10. Prikaz koda za brisanje objava.

4.5. Korisnici

Svaki posjetitelj stranice ima mogućnost čitanja objava te uvid u statističke detalje pojedinog igrača, kao i sve detalje vezane za stranicu, ali nema mogućnost komentiranja objava sve dok se nije registrirao. Kod za dohvaćanje objava nalazi se na slici 4.11.

```
static async getPosts() {
  try {
    const posts = await postModel.find().sort({createdAt : 'desc'});
    if (!posts) {
      return "Something went wrong";
    }

    return posts.map((post) => {

      return {
        id: post.id,
        title: post.title,
        description: post.description,
        userId: post.userId,
        createdAt: post.createdAt,
        updatedAt: post.updatedAt,
      };
    });
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.11. Prikaz koda za dohvaćanje objava..

Svaki korisnik ima mogućnost ulaska u detalje pojedine objave gdje se obavijest vidi u potpunosti kao i komentari registriranih korisnika. Kod za dohvaćanje pojedine objave nalazi se na slici 4.12.

```
static async getPost(id) {
  try {
    const post = await postModel.findById(id);

    if (!post) {
      return "Something went wrong, check the id";
    }
    return {
      id: post.id,
      title: post.title,
      description: post.description,
      userId: post.userId,
      createdAt: post.createdAt,
      updatedAt: post.updatedAt,
    };
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.12. Prikaz koda za dohvaćanje pojedine objave.

Kod za dohvaćanje komentara nalazi se na slici 4.13.

```
static async getCommentesByPost(postID) {
  try {
    const userPost = await postModel.findById({ _id: postID });

    if (!userPost) {
      return "Post is not aviable";
    }

    const returnValue = {
      id: userPost._id,
      title: userPost.title,
      description: userPost.description,
      userId: userPost.userId,
      createdAt: userPost.createdAt,
      updatedAt: userPost.updatedAt,
      commentes: [],
    };

    const userCommentes = await commenteModel.find({ postID }).populate('userId', 'username');
    userCommentes.forEach((userCommente) => {
      returnValue.commentes.push({
        userId: userCommente.userId,
        id: userCommente.id,
        username: userCommente.userId.username,
        commente: userCommente.commente,
        createdAt: userCommente.createdAt,
        updatedAt: userCommente.updatedAt,
      });
    });

    function compare(a, b) {
      return b.createdAt - a.createdAt;
    }
    returnValue.commentes.sort(compare);

    return returnValue;
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.13. Prikaz koda za dohvaćanje komentara.

Registracija na stranici omogućuje korisnicima da postanu članovi jednostavnim postupkom. To je ključan korak za omogućavanje komentiranja objava vezanih za novosti u klubu i organizaciju odlaska na utakmice. Kod za kreiranje komentara nalazi se na slici 4.14.

```
static async postCommente({ commente, postID, userId }) {
  try {
    if (!commente || !postID) {
      return "Required fields are missing";
    }
    if (typeof commente !== "string" || typeof postID !== "string") {
      return "Required fields are not string";
    }
    if (commente.length < 3 || commente.length > 100) {
      return "Please write a comment not longer than 100 characters";
    }
    const post = await postModel.findById(postID);
    if (!post) {
      return "Post does not exist";
    }
    const userCommente = await commenteModel.create({
      commente,
      postID,
      userId,
    });

    return {
      commente: userCommente.commente,
      postId: userCommente.postID,
      userId: userCommente.userId,
      id: userCommente.id,
    };
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

Slika 4.14. Prikaz koda za kreiranje komentara.

Svaki korisnik može brisati isključivo svoj komentar, dok administrator ima mogućnost brisanja svih komentara. Kod za brisanje komentara nalazi se na slici 4.15.

```
static async deleteCommente(id, userId, requestingUserEmail) {
  try {
    const commente = await commenteModel.findById(id);

    if (!commente) {
      return "Comment not found";
    }

    if (
      commente.userId.toString() === userId ||
      requestingUserEmail === 'mariogolemovic12@gmail.com'
    ) {
      const deletedCommente = await commenteModel.findByIdAndDelete(id);

      if (!deletedCommente) {
        return "Something went wrong, check the id";
      }

      return { success: true };
    } else {
      return "Unauthorized to delete this comment";
    }
  } catch (error) {
    console.log(error);
    throw error;
  }
}
```

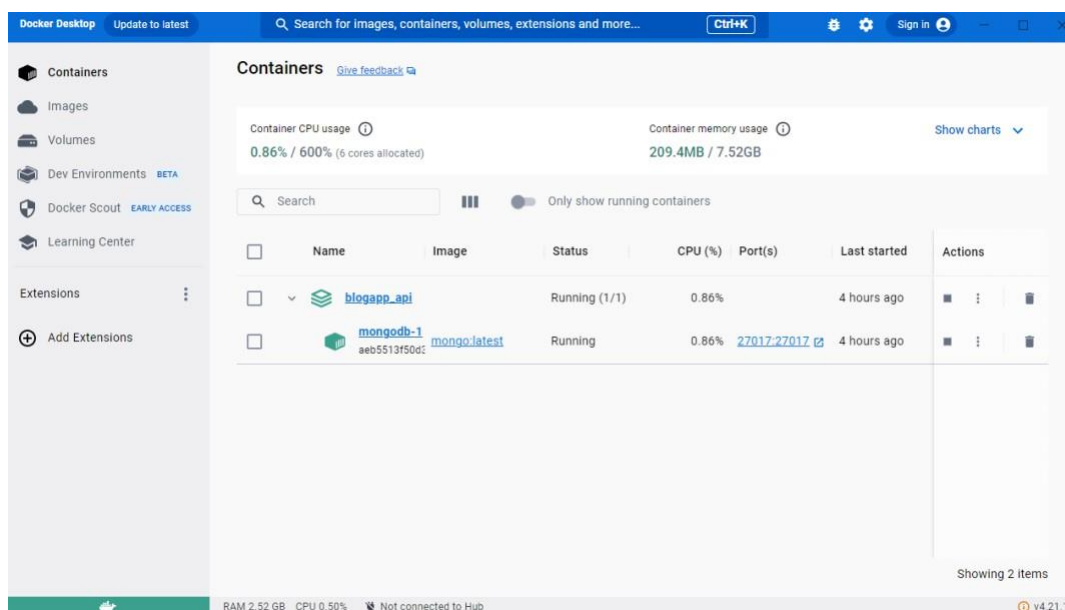
Slika 4.15. Prikaz koda za brisanje komentara.

5. KORIŠTENJE WEB APLIKACIJE

Web portal za ljubitelje nogometnog kluba "Real Madrid" organiziran je s različitim razinama pristupa: administratorima, članovima i posjetiteljima. Administratorima su dostupni specifični pregledi i funkcionalnosti koje ostali korisnici nemaju. Članovi i posjetitelji portala mogu pregledavati sadržaj stranica, no samo registrirani članovi imaju mogućnost komentiranja objava. Ako posjetitelji žele komentirati, moraju se registrirati. Administrator ima ključnu ulogu u sustavu jer može unositi nove igrače s njihovim statistikama, objavljivati novosti o klubu te organizirati odlaske na utakmice. Također, administrator ima ovlasti za uređivanje i brisanje unesenih informacija.

5.1. Pokretanje aplikacije

Za pokretanje aplikacije potrebno je osigurati Docker i Robo 3T instalirane na računalu. Ovi alati su ključni za uspostavljanje okruženja u kojem će Mongo DB baza podataka raditi. Pokretanje Mongo DB baze se postiže pokretanjem Docker kontejnera pomoću naredbe “ docker-compose -f local_database.yaml up -d“ u terminalu. Kada je baza pokrenuta, sljedeći korak je pokretanje “backend“ i “frontend“ dijelova aplikacije također pomoću naredbi u Visual Studio Codeu “Backend“ se pokreće naredbom “ npm run start“, a “frontend“ se pokreće naredbom “ npm run dev“. “Backend“ će biti pokrenut na određenom portu, dok će frontend biti dostupan na drugom portu. Slika 5.1. prikazuje Docker kontejner.

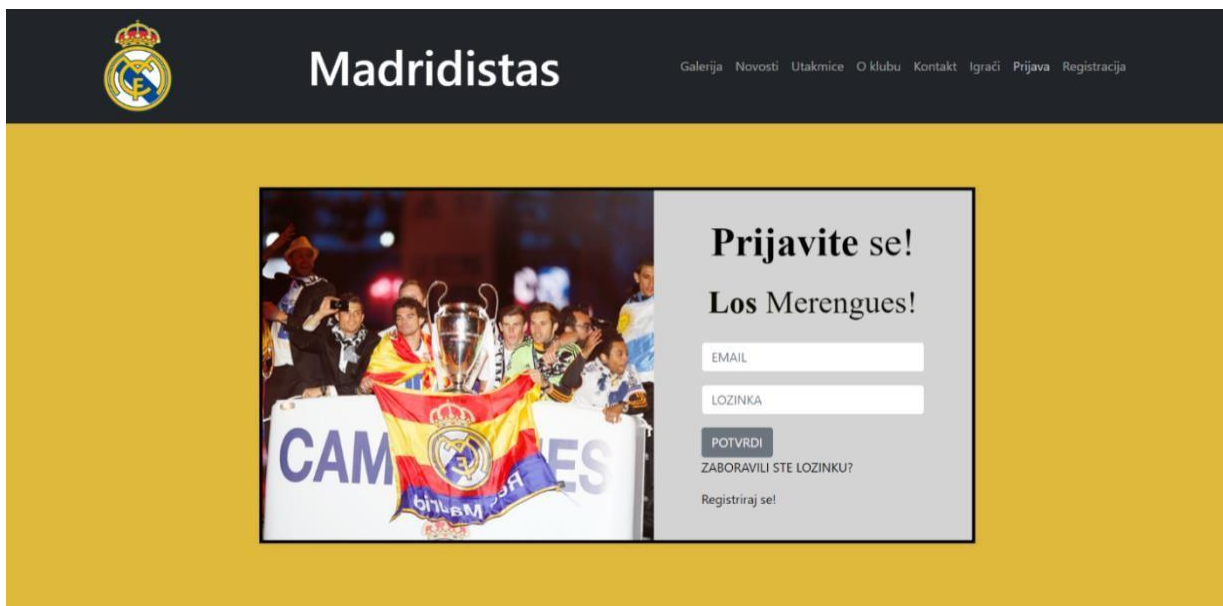


Slika 5.1. Prikaz pokrenutog Docker kontejner

5.2. Početni prikaz stranice

Početna stranica sastoji se od navigacijske trake i obrasca za prijavu već registriranih korisnika. Navigacijska traka, osim naslova “Madridistas“ i logotipa nogometnog kluba “Real Madrid“, sadrži putanje prema stranicama *Galerija*, *Novosti*, *Utakmice*, *O klubu*, *Kontakt*, *Igrači*, *Prijava* i *Registracija*. Kad se korisnik prijavi, na navigacijskoj traci umjesto komponenti *Prijava* i *Registracija* nalazi se nova komponentna *Odjava*. Odlaskom na bilo koju drugu stranicu iz navigacijske trake, navigacijska traka ostaje uvijek prisutna u vrhu stranice.

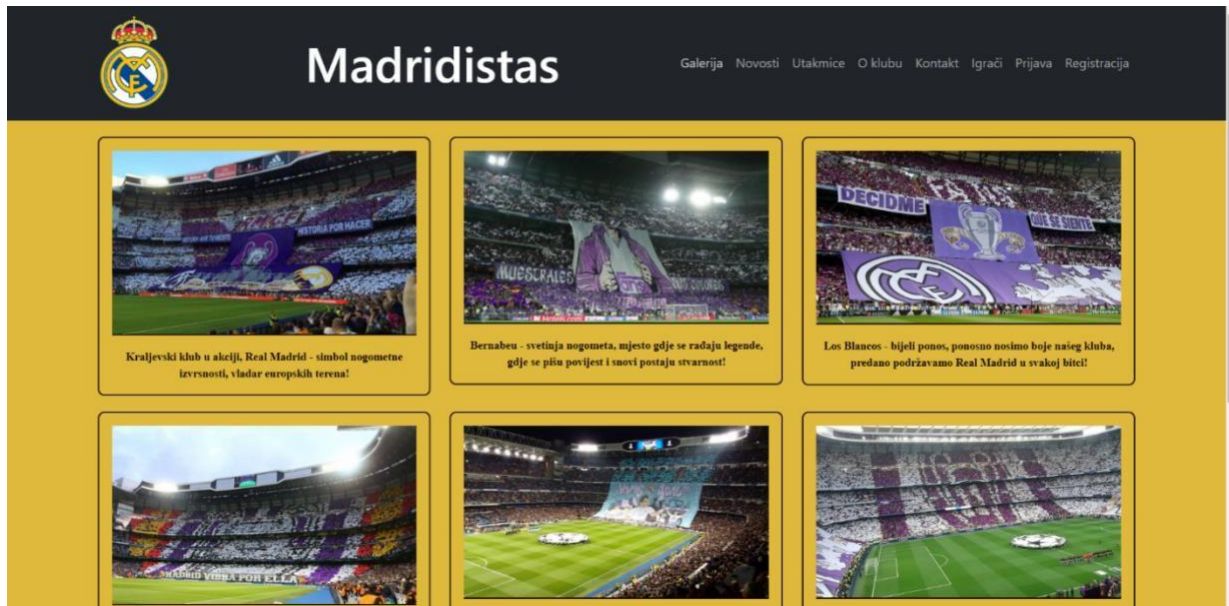
Ispod navigacijske trake nalazi se forma za prijavu članova koja se sastoji od e-pošte i lozinke. Ukoliko korisnik nema kreiran profil, ispod obrasca za prijavu se nalazi poveznica za registraciju članova. Ukoliko član ima kreiran profil, ali je zaboravio svoju lozinku, poveznica za oporavak lozinke se također nalazi ispod obrasca za prijavu. Slika 5.2. prikazuje početni prikaz stranice.



Slika 5.2. Početni prikaz stranice.

5.3. Raznolikost stranice

Odlaskom na putanju *Galerija* u navigacijskoj traci, korisnik odlazi na stranicu gdje su prikazane neke od najboljih fotografija s utakmica na stadionu "Santiago Bernabeu". Pregledu stranice mogu pristupiti svi korisnici, uključujući članove, posjetitelje i administratora. Prikaz stranice prikazan je na slici 5.3.



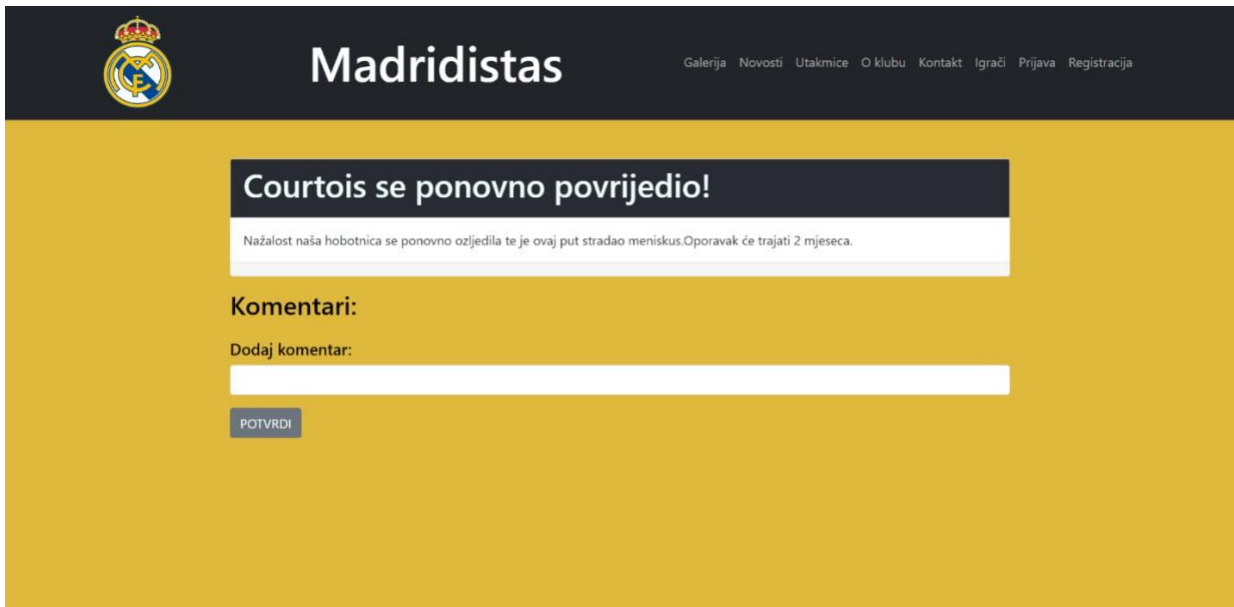
Slika 5.3. Prikaz stranice Galerija.

Stranica *Novosti* izlistava sve članke o novostima vezanim za klub dodanima od strane administratora. Stranica se sastoji od navigacijske trake, liste članaka s prikazanim naslovima pojedinog članka te kratki sadržaj članka. Klikom na naslov pojedinog članka, korisnik odlazi na posebnu stranicu gdje se nalaze detalji članka zajedno s komentarima korisnika. Prikaz stranice *Novosti* nalazi se na slici 5.4.



Slika 5.4. Prikaz stranice Novosti.

Prikaz stranice gdje se nalaze detalji članka nalazi se na slici 5.5.



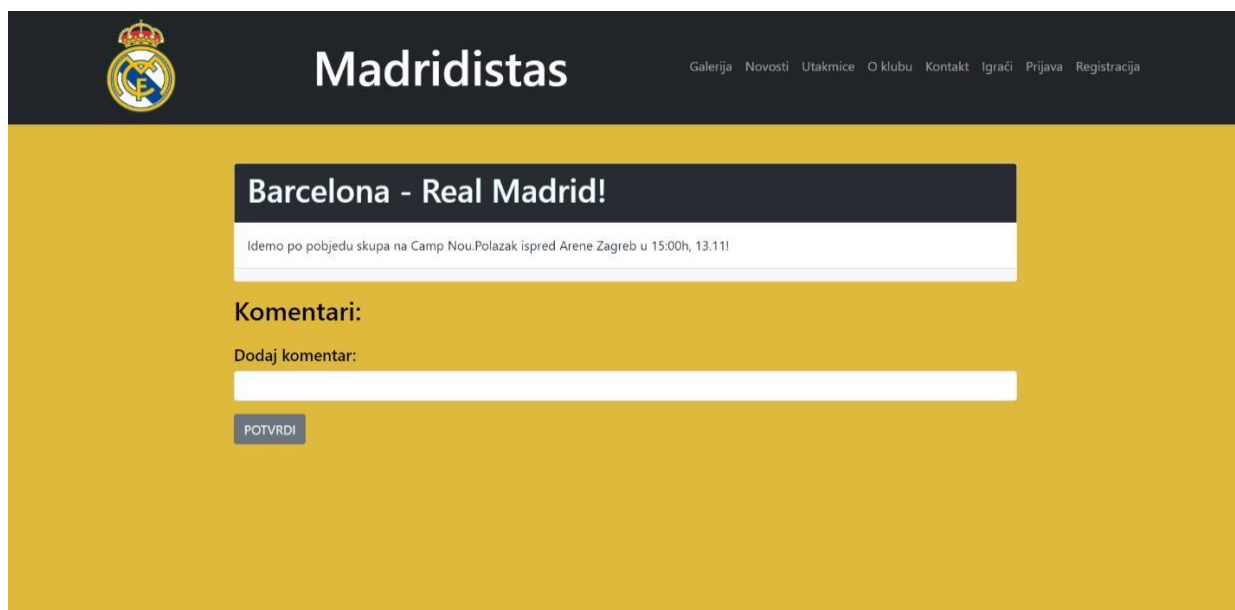
Slika 5.5. Prikaz stranice s detaljima članka.

Odlaskom na stranicu *Utakmice* u navigacijskoj traci, slično kao i na stranici *Novosti*, nalazi se lista koja sadrži objave vezane za organizaciju odlaska na pojedine utakmice. Svaka objava sadrži naslov i kratki sadržaj. Klikom na naslov članka koji, prikazuje koja utakmica je u pitanju, odlazimo na novu stranicu gdje se nalaze detalji organizacije i komentari članova stranice. Prikaz stranice gdje se nalazi lista utakmica nalazi se na slici 5.6.



Slika 5.6. Prikaz stranice s utakmicama.

Prikaz stranice gdje se nalaze detalji oko organizacije za odlazak na utakmice nalazi se na slici 5.7.



Slika 5.7. Prikaz stranice s detaljima organizacije.

Stranica *O klubu* daje uvid u veliku i bogatu povijest Real Madrida. Sadrži sve najbitnije informacije vezane za povijest kluba, njegove navijače, stadion na kojem igra te postignute uspjehe. U donjem dijelu stranice nalaze se poveznice na službene društvene mreže i stranice nogometnog kluba "Real Madrid". Prikaz stranice *O klubu* nalazi se na slici 5.8.



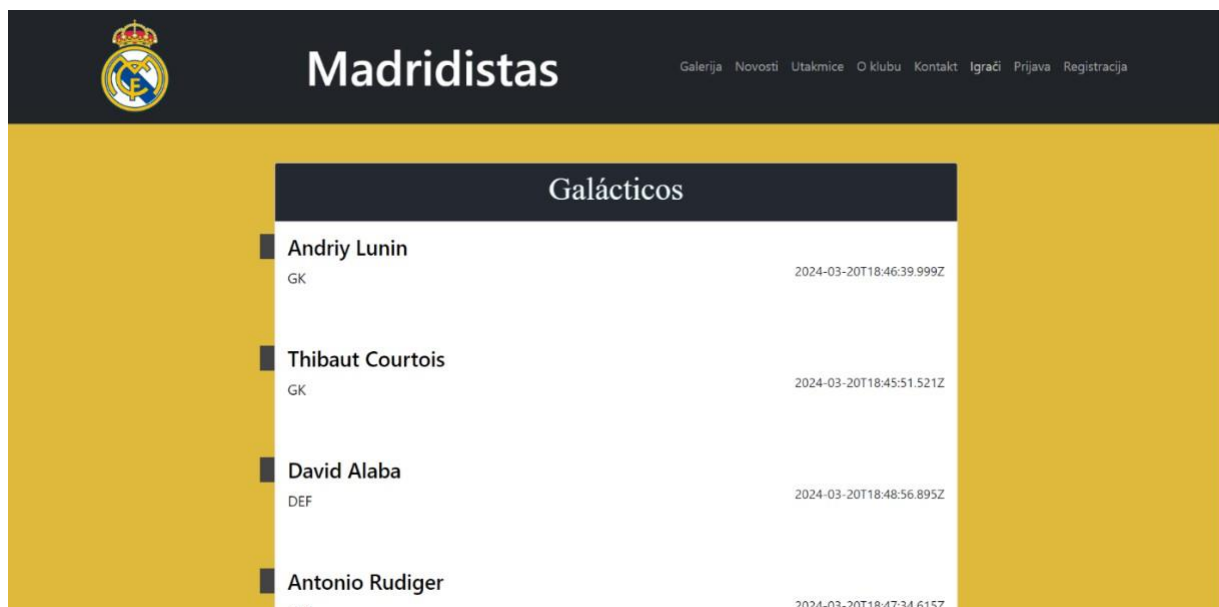
Slika 5.8. Prikaz stranice *O klubu*.

Kontakt stranica pruža sve potrebne informacije o administratoru. Prikazuje osobne detalje pomoću kojih članovi grupe mogu izravno kontaktirati administratora. Također sadrži poveznice za administratorove društvene mreže. Prikaz stranice *Kontakt* nalazi se na slici 5.9.



Slika 5.9. Prikaz stranice *Kontakt*.

Stranica *Igrači* sadrži listu svih aktivnih igrača koji igraju za nogometni klub “Real Madrid“. Lista prikazuje imena igrača te ispod imena poziciju na kojoj taj igrač igra. Igrači su sortirani po pozicijama na kojima igraju krenuvši od golmana prema napadu. Klikom na ime pojedinog igrača otvara se nova stranica koja sadrži sve statističke detalje vezane za odabranog igrača. Prikaz stranice gdje se nalazi lista s igračima nalazi se na slici 5.10.



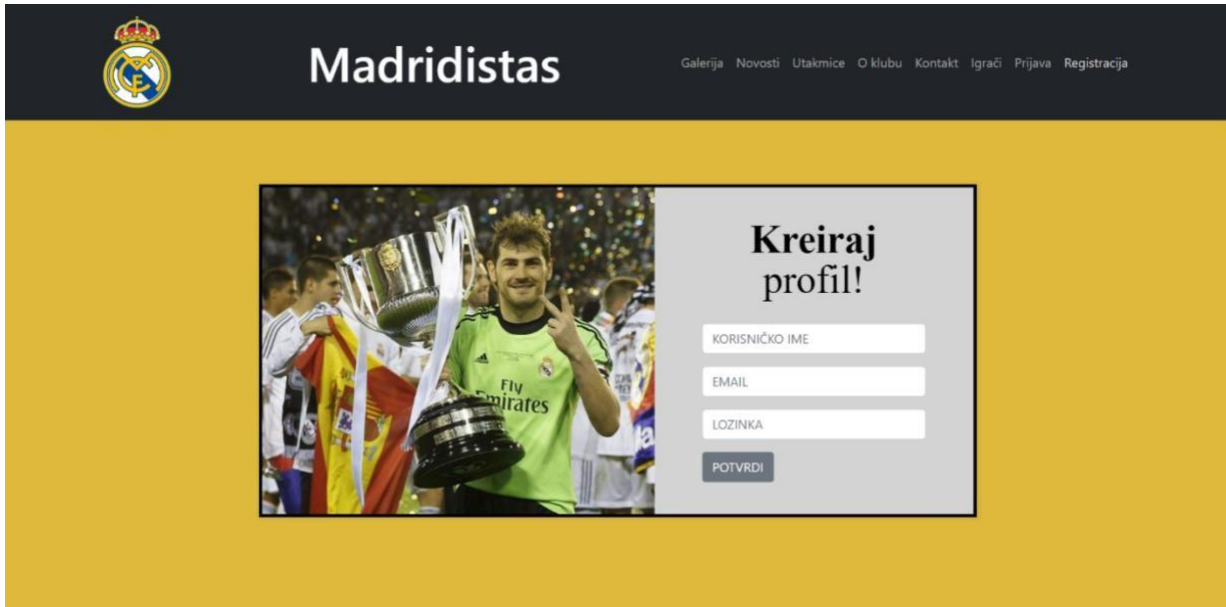
Slika 5.10. Prikaz stranice Igrači.

Prikaz stranice gdje se nalaze statistički detalji pojedinog igrača nalazi se na slici 5.11.



Slika 5.11. Prikaz stranice s detaljima o igraču.

Stranica *Registracija* sadrži obrazac za registriranje novih članova. Obrazac zahtjeva od korisnika unos korisničkog imena, e-pošte i lozinke. Nakon unosa potrebnih podataka korisnik klikom na gumb “POTVRDI“ završava s procesom registracije. Prikaz stranice gdje se nalazi obrazac za registraciju nalazi se na slici 5.12.



Slika 5.12. Prikaz stranice *Registracija*.

5.4. Uloga administratora i prijavljenih korisnika

Uloga administratora pisanje i objavljivanje članaka vezanih za novosti u nogometnom klubu “Real Madrid“, članaka vezanih za organizaciju oko odlazaka na utakmice te unos statističkih podataka vezanih za igrače kluba. Također, ima mogućnost brisanja i ažuriranja svih članaka, igrača i njihovih statistika. Kao administrator, može pisati komentare, ali i brisati neprimjerene komentare drugih članova. Prijavljeni korisnici mogu komentirati članke, za razliku od posjetitelja portala. Ako zaborave svoju zaporku, imaju opciju oporavka iste.

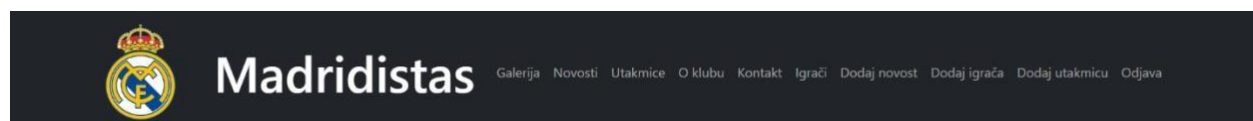
5.5. Opis mogućnost aplikacije

Neprijavljeni korisnici mogu samo pregledavati sadržaj prikazan na aplikaciji. Prijavljeni korisnici imaju dodatnu mogućnost komentiranja objava. Slika 5.13. prikazuje komentiranje članka.



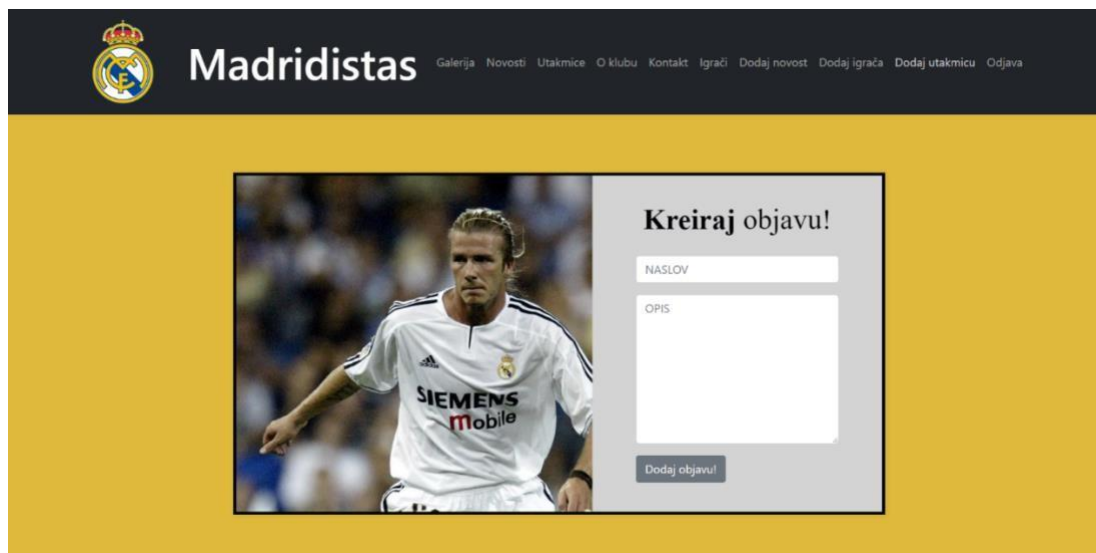
Slika 5.13. Prikaz komentiranja članka

Administrator u svojoj navigacijskoj traci ima mogućnost odlaska na stranice *Dodaj novost*, *Dodaj igrača* i *Dodaj utakmicu*. Na tim stranicama administrator objavljuje novosti, organizira odlazak na utakmice te dodaje i manipulira podacima vezanim za igrače. Nakon što administrator unese željeni tekst te potvrdi objavu pritiskom na gumb, nova objava se izlistava na stranici *Novosti*, *Igrači* ili *Utakmice*, ovisno o tome što administrator želi dodati. Slika 5.14. prikazuje navigacijsku traku administratora.



Slika 5.14. Prikaz navigacijske trake administratora

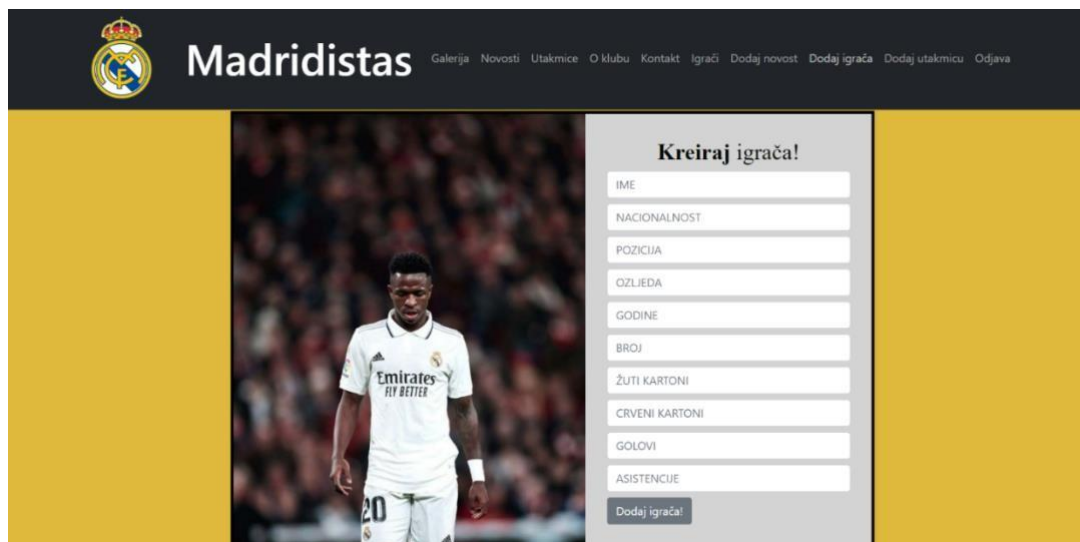
Slika 5.15. prikazuje stranicu za dodavanje članka vezanog za organizaciju odlaska na utakmice.



The screenshot shows the Madridistas website header with the club logo and navigation links: Galerija, Novosti, Utakmice, O klubu, Kontakt, Igrači, Dodaj novost, Dodaj igrača, Dodaj utakmicu, Odjava. The main content area features a large image of a Real Madrid player in a white kit with 'SIEMENS Mobile' on the front. To the right of the image is a form titled 'Kreiraj objavu!' with the following fields: 'NASLOV' (Title), 'OPIS' (Description), and a 'Dodaj objavu!' (Add article) button.

Slika 5.15. Prikaz dodavanja članka o organizaciji

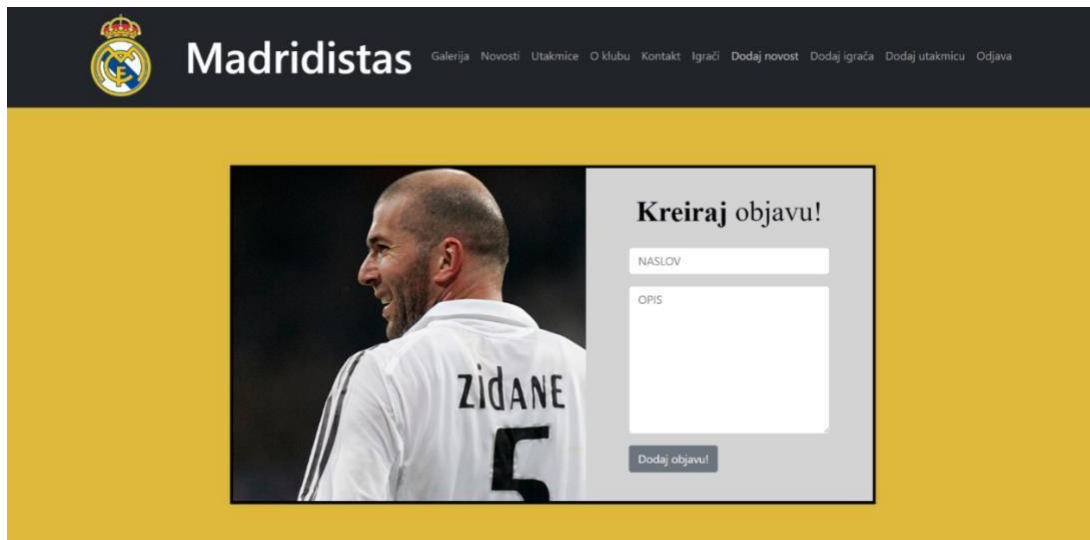
Slika 5.16. prikazuje stranicu za dodavanje novog igrača.



The screenshot shows the Madridistas website header with the club logo and navigation links: Galerija, Novosti, Utakmice, O klubu, Kontakt, Igrači, Dodaj novost, Dodaj igrača, Dodaj utakmicu, Odjava. The main content area features a large image of a Real Madrid player in a white kit with 'Emirates FLY BETTER' on the front. To the right of the image is a form titled 'Kreiraj igrača!' with the following fields: 'IME' (Name), 'NACIONALNOST' (Nationality), 'POZICIJA' (Position), 'OZLJEDA' (Injury), 'GODINE' (Age), 'BROJ' (Number), 'ŽUTI KARTONI' (Yellow Cards), 'CRVENI KARTONI' (Red Cards), 'GOLOVI' (Goals), 'ASISTENCJE' (Assists), and a 'Dodaj igrača!' (Add player) button.

Slika 5.16. Prikaz dodavanja novog igrača

Slika 5.17. prikazuje stranicu za dodavanje novog članka vezanog za novosti u klubu.



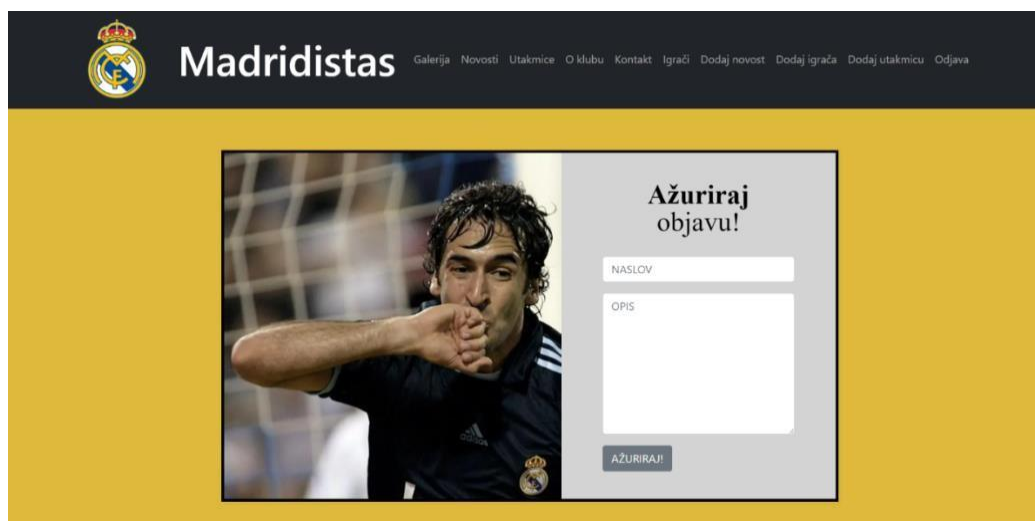
Slika 5.17. Prikaz dodavanja novosti

Unesene članke i igrače, administrator može ažurirati i brisati po potrebi. Nakon odabira opcije ažuriranja, administrator odlazi na drugu stranicu gdje unosi nove podatke objave, te se objava automatski ažurira u listi. Ukoliko administrator odabere opciju brisanja objave, nakon potvrđivanja da ju želi obrisati, objava se briše iz liste. Slika 5.18. prikazuje gumbe za ažuriranje i brisanje objava vidljive samo administratoru.



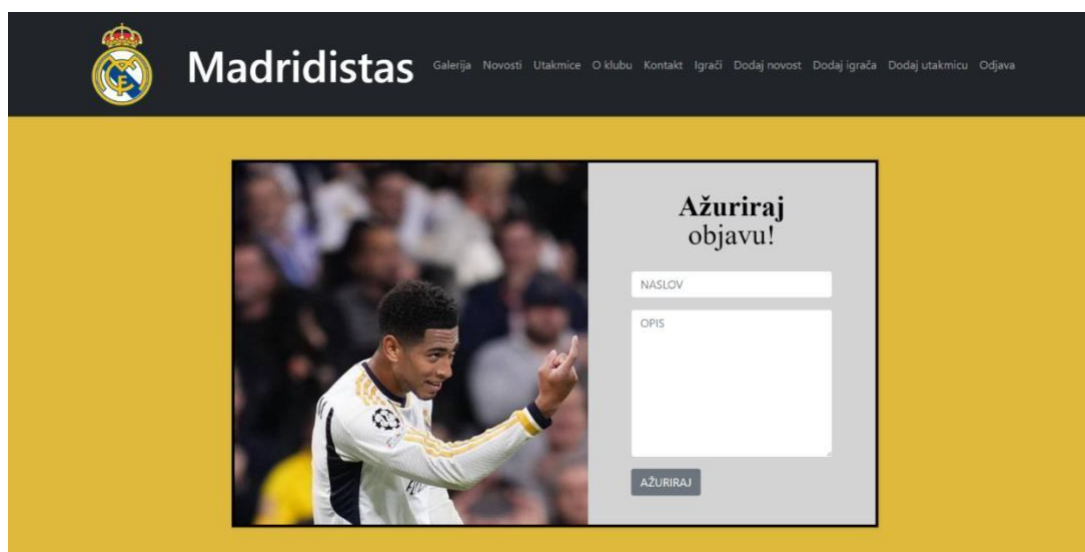
Slika 5.18. Prikaz dugmadi za ažuriranje i brisanje

Slika 5.19. prikazuje stranicu za ažuriranje članka vezanog za organizaciju odlaska na utakmice.



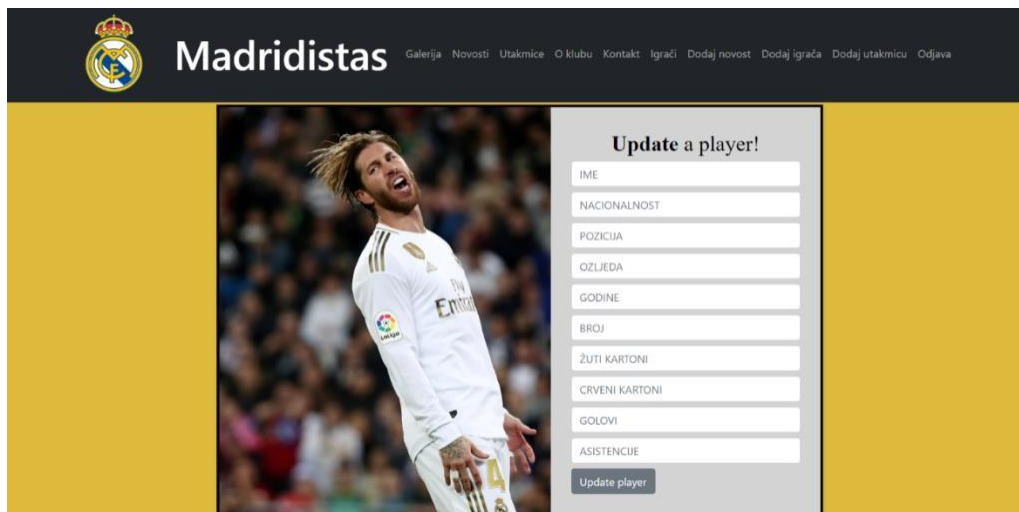
Slika 5.19. Prikaz stranice za ažuriranje članka za organizaciju

Slika 5.20. prikazuje stranicu za ažuriranje članka vezanog za novosti u klubu.



Slika 5.20. Prikaz stranice za ažuriranje novosti

Slika 5.21. prikazuje stranicu za ažuriranje statistike igrača.



The screenshot shows the Madridistas website interface. At the top, there is a dark header with the Real Madrid logo and the word "Madridistas" in white. To the right of the logo, there is a navigation menu with links: "Galerija", "Novosti", "Utakmice", "O klubu", "Kontakt", "Igrači", "Dodaj novost", "Dodaj igrača", "Dodaj utakmicu", and "Odjava". Below the header, the main content area is split into two parts. On the left, there is a large image of a Real Madrid player in a white kit, looking upwards with his mouth open. On the right, there is a form titled "Update a player!". The form contains several input fields for player information: "IME", "NACIONALNOST", "POZICIJA", "OZLJEDA", "GODINE", "BROJ", "ŽUTI KARTONI", "CRVENI KARTONI", "GOLOVI", and "ASISTENCIJE". At the bottom of the form is a button labeled "Update player".

Slika 5.21. Prikaz stranice za ažuriranje igrača

6. ZAKLJUČAK

U okviru ovog završnog rada razvijen je web portal posvećen navijačima nogometnog kluba "Real Madrid". Portal omogućuje organizaciju putovanja na utakmice i pruža najnovije vijesti o klubu. Za komentiranje objava potrebna je prijava na stranicu. Prilikom registracije, korisnikova zaporka se šifrira radi veće sigurnosti računa. Portal nudi jednostavan prikaz novosti i komunikacije, što ga čini pristupačnim za sve posjetitelje. Također, sadrži brojne upute na različitim stranicama kako bi korisnici mogli lako navigirati kroz portal.

Iako danas postoji mnogo web portala posvećenih obožavateljima različitih nogometnih klubova, razlike među njima su izrazito uočljive. Neki služe samo kao prikaz obavijesti vezanih za klub, dok neki imaju svoje trgovine te nude mogućnost kupovine raznih artikala. Ovaj portal služi da bi svojim članovima dao što bolji uvid u događanja koja se događaju u klubu te da im pomogne kako da uz skupinu lakše i jeftinije dođu do karata za željenu utakmicu te prijevoza.

Način na koji bi se ova aplikacija mogla unaprijediti je dodavanjem automatskih procesa slanja obavijesti na e-mail adresu radi izvršavanja različitih funkcija, kao što je primjerice slanje novih zaporki te obavijesti prilikom ažuriranja igrača na stranici te objavljivanja novosti.

LITERATURA

- [1] Kohorta Osijek, <https://www.kohorta.net>, 05.05.2024.
- [2] Torcida – KN Torcida Split, <https://www.torcida.hr>, 05.05.2024.
- [3] Horde zla – Udruženje navijača FK Sarajevo, <https://hordezla.ba>, 05.05.2024.
- [4] Difference between frontend and backend, <https://www.computerscience.org>, 07.05.2024.
- [5] HTML, <https://www.w3schools.com> , 07.05.2024.
- [6] CSS, <https://www.w3schools.com>, 07.05.2024.
- [7] JavaScript, <https://www.freecodecamp.org/news/what-is-javascript>, 07.05.2024.
- [8] Vue.js, <https://www.freecodecamp.org/news/vue-js-full-course> , 07.05.2024.
- [9] Express.js, <https://www.freecodecamp.org/news/tag/expressjs/>, 07.05.2024.
- [10] MySQL <https://www.freecodecamp.org/news/tag/mongodb/>, 07.05.2024.
- [11] Visual Studio Code, <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>, 10.05.2024.

SAŽETAK

Tema: Web portal nogometne navijačke skupine

Cilj ovog završnog rada je izrada web portala za organizaciju odlazaka na utakmice nogometnog kluba "Real Madrid" te objavljivanje raznih novosti vezanih za klub. Ovaj je web portal napravljen koristeći MongoDB za bazu podataka, HTML, CSS, Vue.js te Express.js. Web portal se pokreće u pregledniku preko Visual Studio Code. Nakon prijave na stranicu, korisnici imaju mogućnost komentiranja objava, te uvid u sve novosti vezane za klub i organizaciju za odlazak na utakmice. Administrator pak može brisati tuđe komentare koji se smatraju ne prikladnima. Također ima mogućnost unosa, brisanja i uređivanja objava vezanih za organizaciju za odlazak na utakmice, novosti vezanih za klub te aktivnih igrača i njihovih statistika. Administrator u svom izborniku vidi stranicu *Kontakt*, *Galerija*, *Novosti*, *Utakmice*, *O klubu*, *Igrači*, *Dodaj novost*, *Dodaj igrača*, *Dodaj utakmicu* i *Odjava*, a prijavljeni korisnici vide sve što i administrator osim stranica *Dodaj igrača*, *Dodaj utakmicu* i *Dodaj novost*.

Ključne riječi: baza podataka, navijački klub, organizacija, web portal, web tehnologije

ABSTRACT

Title: Web Portal for Football Fan Group

The goal of this final project is to create a web portal for organizing trips to matches of the football club "Real Madrid" and publishing various news related to the club. This web portal is built using MongoDB for the database, HTML, CSS, Vue.js, and Express.js. The portal runs in the browser through Visual Studio Code. After logging in, users have the option to comment on posts and access all news related to the club and organization of match trips. The administrator, on the other hand, can delete comments deemed inappropriate. They also have the ability to add, delete, and edit posts related to organizing match trips, news related to the club, active players and their statistics. In their menu, the administrator sees the following pages: Contact, Gallery, News, Matches, About the Club, Players, Add News, Add Player, Add Match, and Logout, while logged-in users see everything the administrator does except for the Add Player, Add Match, and Add News pages.

Keywords: *database, fan club, organization, web portal, web technologies*