Web portal za nogometne vijesti

Špiranec, Domagoj

Undergraduate thesis / Završni rad

2023

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://urn.nsk.hr/urn:nbn:hr:200:948771

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-10**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 10.07.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Domagoj Špiranec
Studij, smjer:	Programsko inženjerstvo
Mat. br. Pristupnika, godina upisa:	R4575, 28.07.2020.
OIB Pristupnika:	93617416590
Mentor:	prof. dr. sc. Krešimir Nenadić
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Web portal za nogometne vijesti
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rad:	Opisati funkcionalnosti koje treba imati web portal za nogometne vijesti. Kratko opisati dizajn baze podataka potrebne za rad web portala. Predvidjeti registraciju korisnika koji će moći pristupiti portalu i komentirati pojedine objave. Izraditi web portal s navedenim mogućnostima i opisati postupak izrade kao i tehnologije korištene u izradi portala. Tema rezervirana: Domagoj Špiranec.
Prijedlog ocjene završnog rada:	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	10.07.2023.
Datum potvrde ocjene od strane Odbora:	08.09.2023.

Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 10.09.2023.

Ime i prezime studenta:	Domagoj Špiranec
Studij:	Programsko inženjerstvo
Mat. br. studenta, godina upisa:	R4575, 28.07.2020.
Turnitin podudaranje [%]:	5

Ovom izjavom izjavljujem da je rad pod nazivom**: Web portal za nogometne vijesti**

izrađen pod vodstvom mentora prof. dr. sc. Krešimir Nenadić

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

WEB PORTAL ZA NOGOMETNE VIJESTI

Završni rad

Domagoj Špiranec

Osijek, 2023.

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 2

2. PREGLED PODRUČJA I SLIČNA RJEŠENJA 3

2.1. Nogometne-vijesti 3

2.2. Klik 4

2.3. 24sata 5

2.4. SkySports 7

2.5. Marca 9

3. TEHNOLOGIJE KORIŠTENE U IZRADI RADA 10

3.1. Microsoft Visual Studio Code 10

3.2. HTML i CSS 11

3.3. JavaScript 12

3.4. Google Firebase 13

3.4.1. Cloud Firestore 13

3.4.2. Firebase Authentication 14

4. POSTUPAK IZRADE APLIKACIJE 15

4.1. Kreiranje i pokretanje *JavaScript* web portala 15

4.2. Kreiranje *Firebase* baze podataka i autentifikacije 16

4.3. Izrada sučelja za kreiranje nove objave 18

4.4. Izrada glavne stranice web portala 21

4.5. Izrada opširnijeg prikaza pojedine objave 23

5. PRIKAZ RADA WEB PORTALA 26

5.1. Registracija ili prijava korisnika 26

5.2. Kreiranje nove objave 27

5.3. Početni zaslon web portala 28

5.4. Prikaz pojedine objave portala 29

6. ZAKLJUČAK 31

LITERATURA ..... Error! Bookmark not defined.

SAŽETAK ..... Error! Bookmark not defined.**3**

ABSTRACT ..... Error! Bookmark not defined.**4**

1. UVOD

Kada se govori o vijestima riječ je o brzim i točnim informacijama za bitna događanja u lokalnom području te velika događanja u svijetu. Sportske vijesti jedne su od glavnih potkategorija vijesti općenito. Pisanje sportskih vijesti usredotočeno je na izvještavanje o amaterskim i profesionalnim sportskim događanjima i vijestima. Velik broj ljudi na svijetu prati i uživa u sportu, a bez sumnje najpoznatiji sport današnjice je nogomet. Nogomet privlači milijune obožavatelja svojom kreativnošću, uzbuđenjem i nepredvidivošću. Nogomet se prvi put počeo igrati u Engleskoj 1863. godine, a mjenjanjem pravila je evoluirao kroz godine. Kako je nogomet napredovao, počeo je privlačiti medijsku pozornost i novinare. Prvi nogometni članak izašao je pod nazivom „*The Goal: The Chronicle of Football*“ davne 1873. godine. Iz velike svjetske popularnosti stvorena je potreba za kreiranjem nogometnih vijesti. Danas je sva pozornost na internetu pa su tako s vremenom internetski portali, televizijske emisije i društveni mediji zamijenili novinske članke, časopise i veliku većinu tiskanih medija. Neki od najpoznatijih globalnih sportskih kanala današnjice su *SkySports, The Guardian, Goal, Espn, Marca* i brojni drugi, dok na lokalnom području najaktivnije objavljuju kanali *Nogometne-vijesti.hr, Germanijak, 24sata* i *Jutarnji list*. Cilj uspješnog web portala za nogometne vijesti je brzo i točno doprinošenje informacija o nogometnim događanjima na web portal. Takva događanja mogu biti velike pobjede niže rangiranih nogometnih klubova ili reprezentacija, pobjednici finala za velike trofeje, rezultati, transferi igrača, ozljede, kontroverze, izjave za medije i slično. Nogometne vijesti su prozor u nogometni svijet, pružajući uvid u nevjerojatne priče koje se odvijaju izvan nogometnih travnjaka. Kroz ovaj završni rad bit će vizualizirana izrada web portala i njegovih funkcionalnosti objavljivanja, uređivanja i brisanja vijesti, prijave, registracije, komentiranja i prikazivanja vijesti. Osnovne funkcionalnosti koje web portal o nogometnim novostima treba nuditi s korisničke strane su funkcionalnosti prijave i odjave iz web portala, pregleda u cijelu objavu i komentiranja objave. S druge strane, administrator ima dodatne funkcionalnosti kreiranja, uređivanja i brisanja objave. Korisnici koji nisu prijavljeni u web portal neće moći pogledati niti komentirati objavu. U drugom poglavlju bit će prikazana i objašnjena slična rješenja ovog zadatka kroz nekoliko primjera koji postoje na webu. Uz svaki primjer bit će navedeni prednosti i nedostaci u odnosu na ostala rješenja te sličnosti s ovim radom. U trećem poglavlju bit će kratko objašnjene sve tehnologije korištene u izradi ovog rada i kratki primjeri njihove funkcije. U četvrtom poglavlju prikazat će se tijek izgradnje funkcionalnosti web portala za

nogometne vijesti uz isječke napisanih kodova i objašnjenja. U zadnjem, odnosno petom poglavlju prikazat će se prikaz rada web portala i objašnjenje funkcija koje se koriste na određenim web adresama.

1.1. Zadatak završnog rada

Opisati funkcionalnosti koje treba imati web portal za nogometne vijesti. Kratko opisati dizajn baze podataka potrebne za rad web portala. Predvidjeti registraciju korisnika koji će moći pristupiti portalu i komentirati pojedine objave. Izraditi web portal s navedenim mogućnostima i opisati postupak izrade kao i tehnologije korištene u izradi portala.

2. PREGLED PODRUČJA I SLIČNA RJEŠENJA

Web portali za vijesti dostupni na internetu većinom se baziraju na sličnoj konstrukciji i sličnim osnovnim funkcionalnostima koje su potrebne za kompletan web portal. Iz tog razloga ovaj web portal za nogometne vijesti ne razlikuje se puno od drugih sličnih rješenja. Upravo ta rješenja uzeta su kao izvor i motivacija za kreiranje web portala s osnovnim funkcionalnostima objavljivanja vijesti za prikaz korisniku.

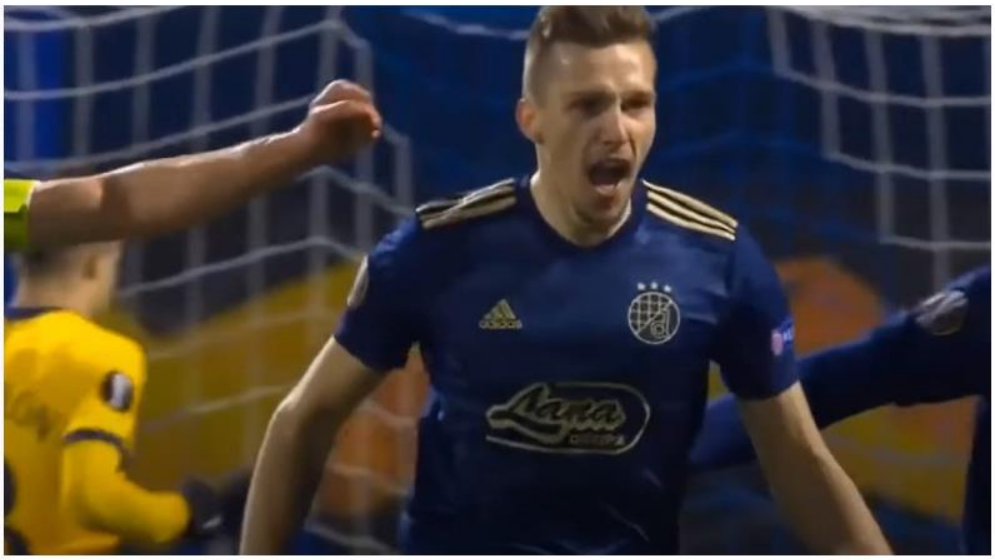
Izrađen je po modelu *One Web Server with One Database.* Kako i samo ime kaže, imat će jedinstveni server koji je pokrenut i podržava velik broj istovremeno prijavljenih korisnika. Također radi s jednom bazom podataka u koju su spremljene bitne informacije unesene od korisnika ili admina.

Arhitektura ovog web portala formirana je po modelu SPA (engl. *Single Page Application*) koji je idealan za ovaj rad zbog sprječavanja prekida u korisničkom sučelju. SPA ne zahtijeva učitavanje potpuno novih stranica svaki put kada korisnik izvrši novu radnju. Umjesto toga, bez potrebe za ponovnim učitavanjem stranice ažurirani sadržaj pruža se trenutnoj stranici. Iz tog razloga SPA se koristi skoro posvuda. Primjeri aplikacija koji koriste ovu strukturu su *Gmail, Google Maps, Airbnb, Netflix, Pinterest, Paypal* i mnogi drugi.

2.1. Nogometne-vijesti

Nogometne-vijesti jedan je od najpoznatijih web portala za nogometne vijesti na području Hrvatske. Vrlo je relevantan i sadrži vijesti za više kategorija nogometa. Svakoj vijesti (objavi) moguće je pristupiti klikom na gumb „SAZNAJTE VIŠE“ koji učitava novi dokument s jedinstvenom identifikacijskom oznakom koji poprima oblik naslova objave. Sadrži mogućnost dijeljenja svake objave na društvenim mrežama *Facebook, Twitter* i *Google+* kao što je prikazano na slici 2.1. U navigacijskoj traci ugrađena je funkcionalnost pregleda vijesti po katergorijama, odnosno ovisno o tome jesu li vijesti na Hrvatskoj, Europskoj ili svjetskoj razini.

Velik nedostatak ovog web portala je što nema funkcionalnost prijave niti komentiranja objava nego je fokusiran samo na doprinos vijesti i informacija o zbivanjima u nogometu. Iz tog razloga korisnici na ovom web portalu nemaju veliku ulogu. Korisnike više privlači odlazak na portal ako imaju uvid u mišljenja drugih o nekoj temi.

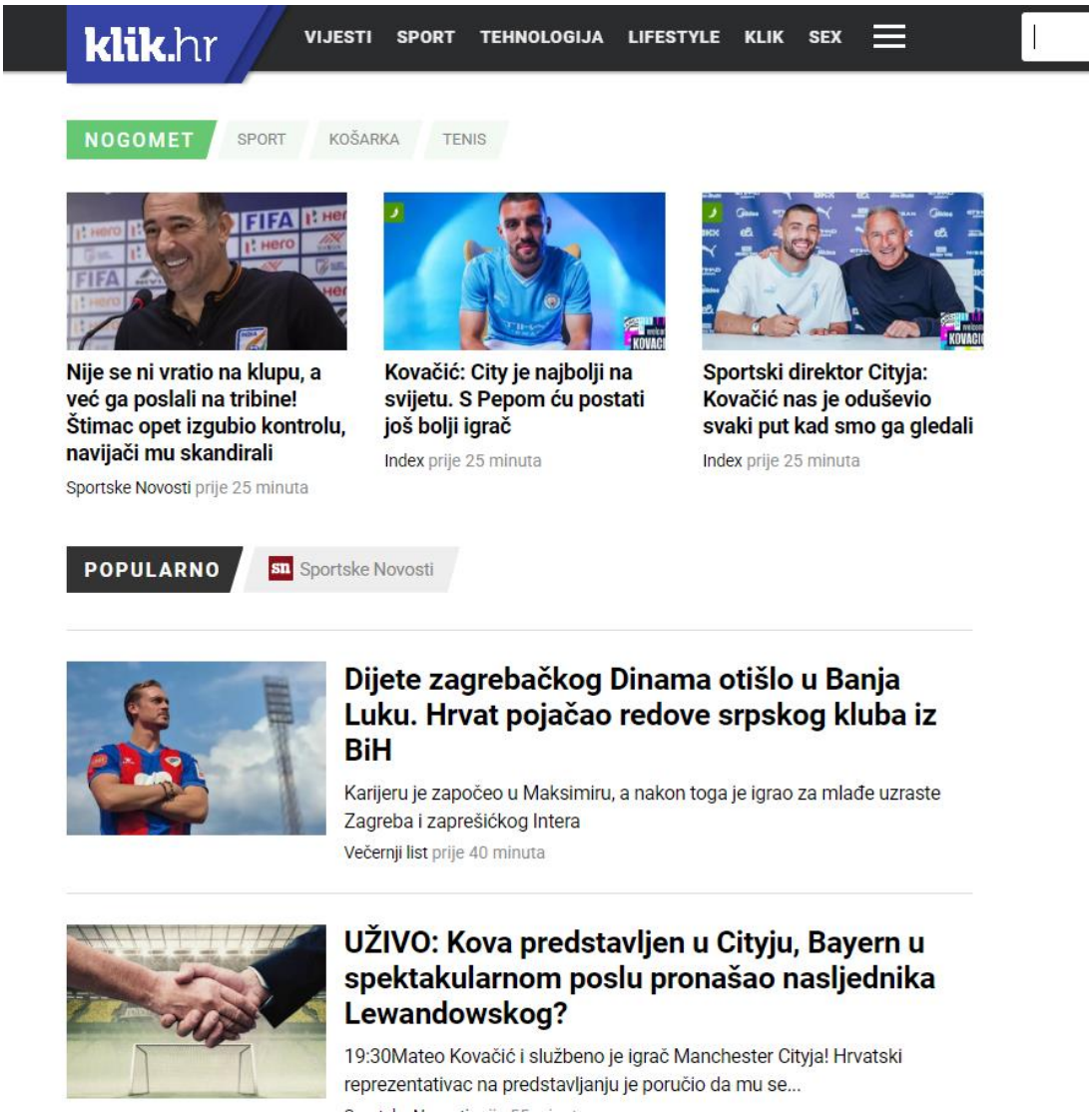


Slika 2.1. Naslovna stranica portala „Nogometne-vijesti“

Izvor: [1]

2.2. Klik

Klik poseban je po tome što ne posjeduje svoje objave nego članke koji predstavljaju poveznice s funkcijom preusmjeravanja korisnika na druge web portale. Zbog preusmjeravanja na druge portale bit će duže čekanje odgovora na HTTP (engl. *Hyper Text Transfer Protocol*) zahtjev i uspostavljanja veze. Uz glavnu temu nogometa, ovaj portal nudi vijesti o košarci i tenisu, a izvan sporta i brojne druge teme. Osim toga, portal nema ostalih značajnih funkcionalnosti pa iz tog razloga nije jako popularan. Na slici 2.2. prikazan je izgled poretka kategorija i objava na ovom web portalu.



Slika 2.2. Naslovna stranica portala „Klik“

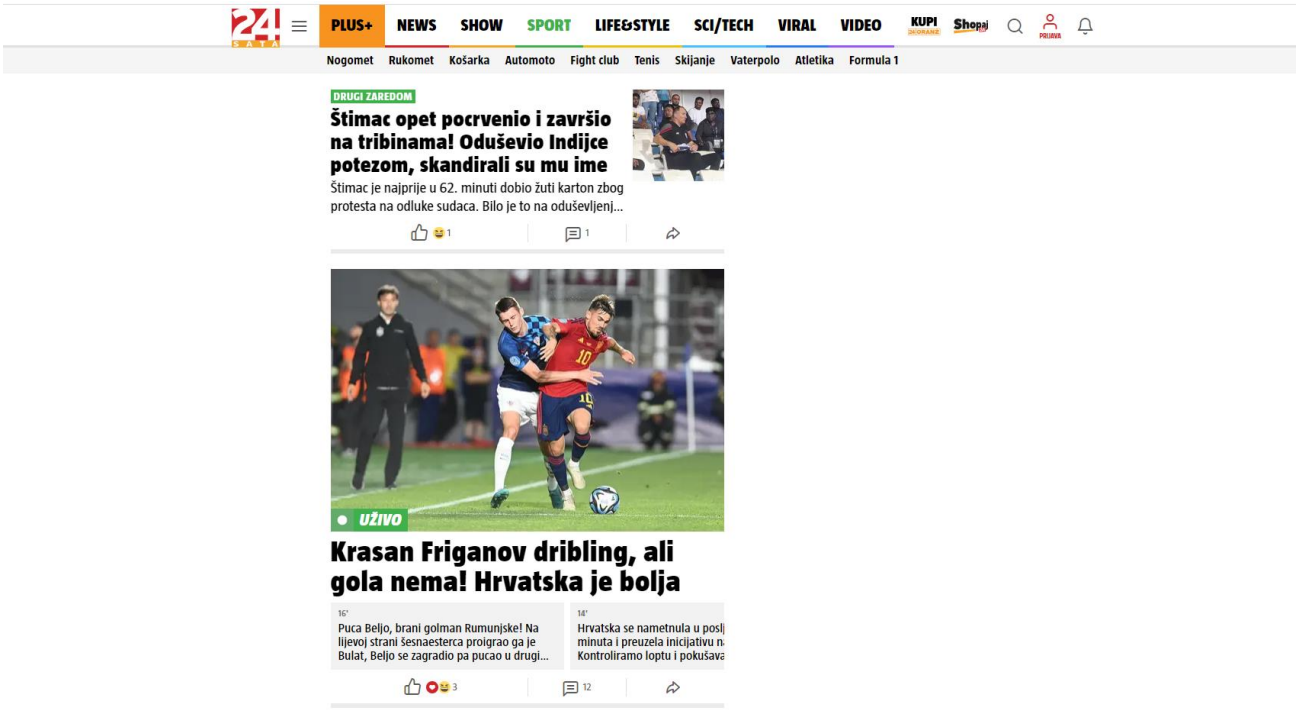
Izvor: [2]

2.3. 24sata

24sata jedan je od najpoznatijih i najposjećenijih portala ovog stoljeća. Krenuo je s radom početkom 2008. godine kada je uglavnom izvještavao o životnim temama. Godinama su se stvarale nove kategorije i sada je bez dvojbe najšire kategoriziran veliki portal u Hrvatskoj.

Portal 24sata ima ugrađene brojne funkcionalnosti pa tako korisnici imaju mogućnost reagirati s različitim emojijima, komentirati svaku od objava. Kako bi korisnici mogli koristiti te funkcionalnosti, potrebna je registracija i prijava u sustav kako bi se ime korisnika prikazivalo uz objavljen komentar. Podijeliti objavu na društvene mreže može se i bez registracije i prijave jer za to je potrebna samo prijava u određenu društvenu mrežu.

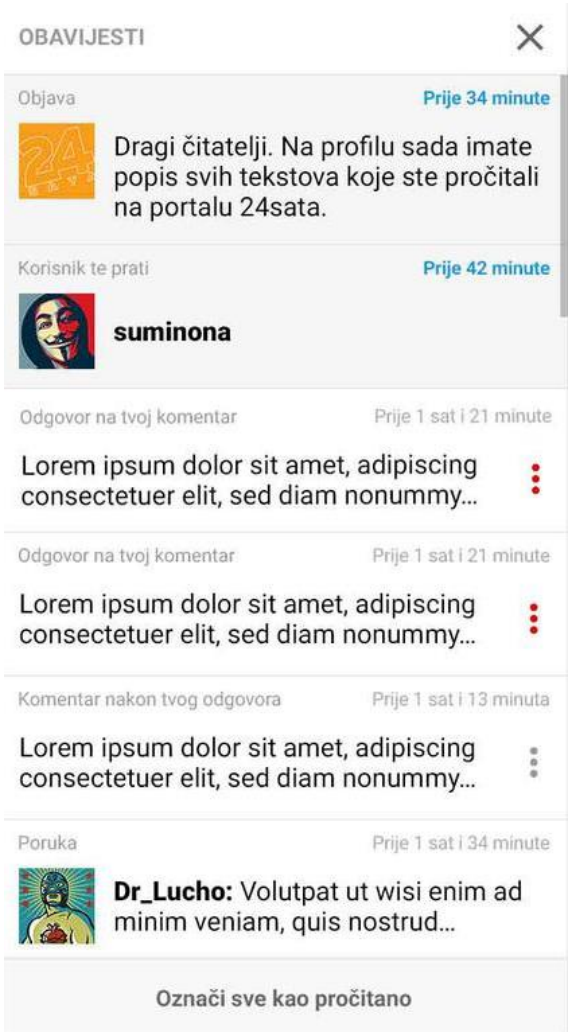
Nedostatak ovog portala je što izgled naslovne stranice s objavama izgleda prazno, to jest sadržaj je postavljen samo na užu sredinu stranice dok je sa strane neispunjeno i neiskorišteno. Na slici 2.3. prikazan je izgled naslovne stranice o nogometnim vijestima na portalu 24sata. Ispod objave vidi se broj reakcija i komentara.



Slika 2.3. Naslovna stranica portala „24sata“

Izvor: [3]

Ima funkcionalnost obavještavanja prijavljenih korisnika o temama koje su korisnici označili kao njihov interes, također i obavijesti o objavama na kojima su korisnici objavili komentar. Slika 2.4. prikazuje izgled kutije unutar koje korisnicima dolaze obavijesti.



Slika 2.4. Kutija s obavijestima

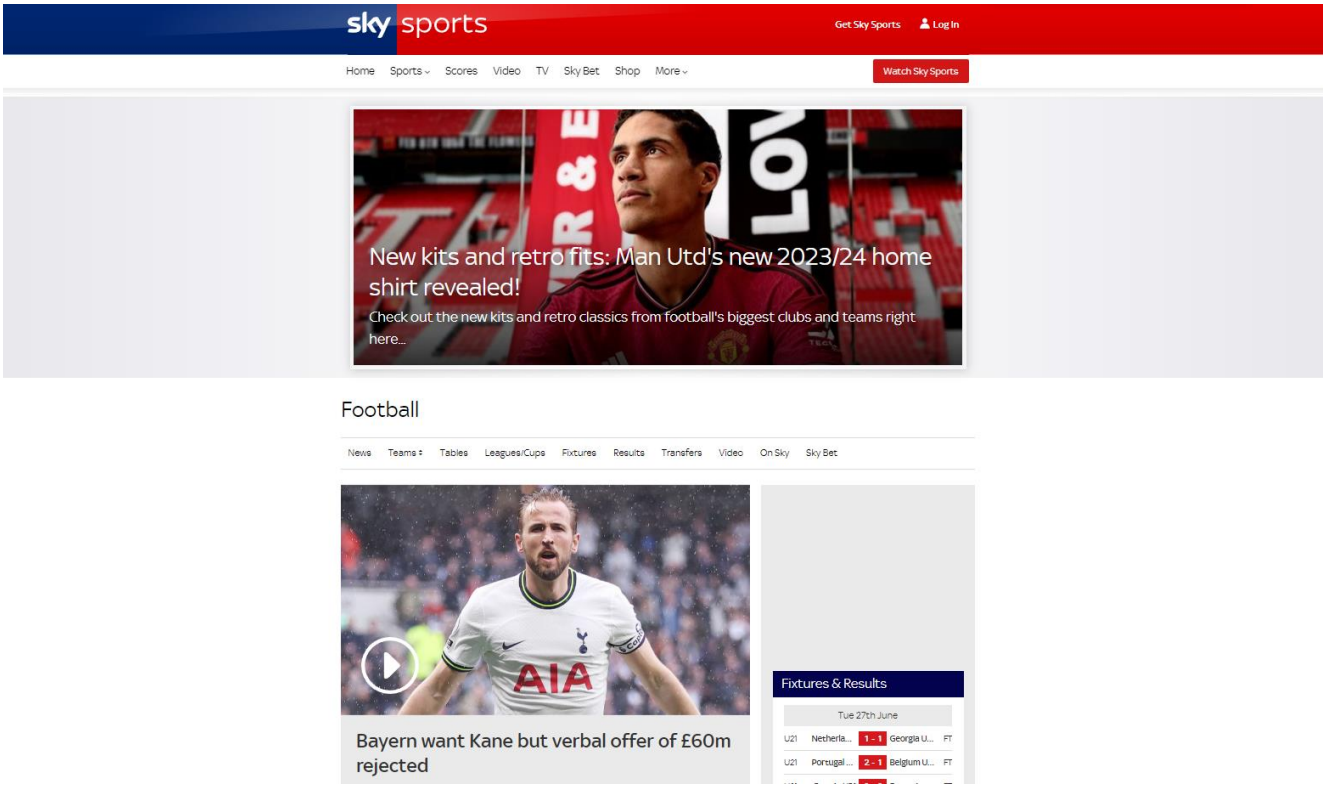
Izvor: [3]

2.4. SkySports

SkySports poznat je po svojim britanskim televizijskim kanalima na satelitskoj televiziji i jedan je od najpoznatijih globalnih sportskih portala. Donosi vijesti o više od 15 sportova diljem svijeta. Kako je već poznato, nogomet je najpoznatiji i najviše investiran sport na svijetu i iz tog razloga prva i najveća kategorija na Skysportsu predstavlja nogomet.

Na naslovnoj stranici nogometnih vijesti odmah se primjećuje velik izbor selektora za vijesti. Neki od selektora su: videozapisi, transferi, tablice, nadolazeće utakmice itd. Također vijesti je moguće filtrirati ovisno jesu li vijesti u obliku videozapisa, važne novosti ili sveukupne novosti.

Klikom na pojedinu objavu korisnika se šalje na web adresu čija adresa ima oblik formiran od naziva objave i jedinstvenog identifikacijskog ključa. Na toj web adresi prikazan je veliki skup informacija vezanih za temu te objave. Cjelokupni izgled ovog web portala je odličan i vrlo uređen uz to što sadrži brojne animacije i pokrete. Na ovom portalu moguće je prijaviti se u SkySports sustav ako su korisnici pretplaćeni na SkySports televiziju. Nakon uspješne prijave korisnici čak mogu gledati SkySports kanale. Na slici 2.5. prikazan je izgled naslovne stranice nogometnih vijesti s ovog portala.



Slika 2.5. Naslovna stranica portala „SkySports“

Izvor: [4]

Velik nedostatak ovog sportskog portala je što ima prikaz na prekomjerni broj sponzoriranih linkova. Na određenim web adresama ovog portala postoji više takvih linkova nego pravih nogometnih objava što je neprivlačno korisnicima.

Također, ovaj portal ima vrlo detaljno zaglavlje u kojem su navedeni svi njihovi sportski kanali, partneri i dodatne web stranice.

2.5. Marca

Marca je vodeći španjolski web portal koji se prvenstveno fokusira na nogomet. Ima fokus na španjolskim momčadima i reprezentaciji Španjolske. Stilski je vrlo uredan i pregledan, a također ima primamljive prijelaze i efekte.

Nudi mogućnost pretplate na dnevne novine i časopise te mogućnost prijave na portal. Prijavom korisnici pristaju na opciju obavještavanja o vijestima za koji imaju interes. Nakon uključivanja pretplate korisnik ima mogućnost komentiranja pojedine objave, a ako komentar skupi najviše oznaka „upvote“ bit će prikazan pod oznakom „#1“ čime će se isticati u usporedbi s drugima.

Za svaku objavu također se prikazuje detaljni skup informacija o vijesti koju prenosi ovaj web portal. Ako je moguće, uz određene objave sa strane su prikazane prve postave momčadi iz ekipa za koju se najavljuje utakmica ili ako se radi o konačnom rezultatu susreta. Uz samu postavu nudi se i detaljna statistika utakmice kako bi korisnici analizirali ili na osnovu statistike komentirali objavu. Slikom 2.6. prikazana je jedna od vijesti sa nogometnog portala Marca.

Slika 2.6. Primjer vijesti na nogometnog portala „Marca“

Izvor: [5]

3. TEHNOLOGIJE KORIŠTENE U IZRADI RADA

Web portal je izrađen u uređivaču izvornog koda *Microsoft Visual Studio Code,* kod je pisan u prezentacijskom jeziku za izradu web stranica HTML (engl. *HyperText Markup Language*), stilskom jeziku CSS (engl. *Cascading Style Sheets*) i skriptnom programskom jeziku *JavaScript*. Za spremanje podataka o korisnicima i objavama korištena je platforma *Google Firebase*. Na toj platformi za izgradnju baze podataka korišten je *Cloud Firestore,* a za čuvanje informacija o korisnicima *Firebase Authentication.*

3.1. Microsoft Visual Studio Code

Visual Studio Code (skraćeno *VS Code*) snažan je uređivač izvornog koda koji je dostupan za Windows, Linux i macOS platforme. Ima ugrađenu podršku za *JavaScript, TypeScript* i *Node.js*. Također, ima ogromnu količinu mogućnosti proširenja za druge jezike i okruženja. Najčešća proširenja su *C++, C#, Java, Python, PHP, Go, .NET*. *Visual Studio Code* je kao „mini-verzija" integriranog razvojnog okruženja *Microsoft Visual Studio*, ali je korišten u svrhe ovog završnog rada zbog efikasnijeg pisanja koda u *JavaScript*-u koji omogućuje sve bitne funkcionalnosti. Na slici 3.1. prikazan je *Visual Studio Code* logo.

Slika 3.1. *Microsoft Visual Studio Code* logo

Izvor: [6]

3.2. HTML i CSS

HTML je standardni prezentacijski jezik za web stranice. To je jezik označavanja hipertekstualnih dokumenata koji se objavljuju preko World Wide Web servisa (WWW). „Hypertext“ referira na veze koje povezuju web stranice jednu s drugom, odnosno jednog dokumenta s drugim dokumentom. Te veze su osnovni aspekt weba.

HTML ima mogućnost imenovanja informacijskih izvora prema jedinstvenoj shemi *URI* (engl. *Universal Resource Identifier*) tako da svaki informacijski izvor raspoloživ na webu (HTML dokument, slika, videozapis...) ima svoju adresu.

Konstrukcija HTML-a sastoji se od posebnih elemenata: (Izvor: [7]) <nav>, <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, , <ul>, <li>, <ol> , , < img>, <aside>, <audio>, <video>, <output> i mnogi drugi. Ti elementi koriste se za kreiranje odlomaka, tablica, slika, poveznica, listi itd. Postoje svojstva koja se mogu pridružiti tim elementima a ona se zovu atributi. Primjer je prikazan na slici 3.2.:

```
<title id="formTitle"></title>
```

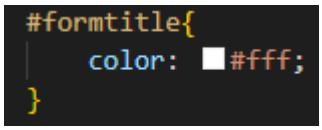
Slika 3.2. Primjer HTML koda

U ovom primjeru napisan je element title koji ima atribut id s vrijednosti „formTitle“.

Za definiranje načina kako se prikazuju pojedini elementi koristi se stilski jezik CSS. Prvi korak uređivanja prikaza elemenata HTML-a je uključivanje CSS-a na jedan od tri moguća načina:

- unutrašnji stil (u zaglavlju samog HTML dokumenta)
- vanjski stil (zaseban vanjski dokument)
- inline stil (unutar samog ciljanog elementa u HTML-u)

Kako bi se pristupilo elementima iz HTML-a koristeći CSS primjenjuju se atributi elemenata iz HTML-a. CSS će za atribut id definirati način prikaza samo za taj jedinstveni element, dok za atribut class definira stil za grupu elemenata. U nastavku na slici 3.3. dan je primjer stilizacije elementa <title>:



Slika 3.3. Primjer CSS koda

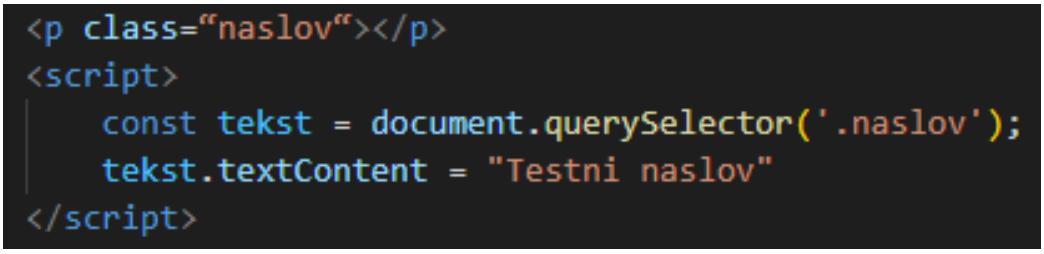
Ovim CSS kodom postavljena je boja teksta elementa title s id-jem „formtitle“ na crvenu.

3.3. JavaScript

JavaScript je programski jezik koji web stranici odnosno web portalu omogućuje interaktivnost koje su uobičajene također i u internetskim videoigrama kada se pritisnu gumbi. Nudi dinamički stil prikaza, animacije, tranzicije i funkcionalnosti.

JavaScript je razvijen kako bi bio jednostavan za početnike te je kroz godine razvijen i programeri su ubacili brojne alate povrh osnovnog *JavaScript* jezika. Ovaj programski jezik dopušta uporabu velikog broja sučelja za programiranje aplikacija, skraćeno API (engl. *Application Progrramming Interface*), te oni omogućuju dinamičko stvaranje HTML dokumenta i ugrađivanje CSS stilova (Izvor: [9]).

Isto kao i CSS, ovaj programski jezik može se pisati unutar samog HTML dokumenta uz navođenje posebnog elementa „<script>“ *(*[8] *Simpson, 2016: 134)*. Zbog veće preglednosti, ovaj pristup se izbjegava te se kod piše u zasebne .js dokumente. Iz zasebnog dokumenta također može se dohvatiti svaki element napisan u .html datoteci koji je definiran klasom, imenom ili id-jem. Bitno je naglasiti da ako se dohvaća iz zasebnog dokumenta, potrebno je napisati liniju koda koja upućuje na pripadnu .js datoteku. Neke od metoda koje *JavaScript* nudi za dohvaćanje su *querySelector(), getElementById()* i *getElementByName().* Primjer .html koda i dohvaćanje tog elementa prikazan je na slici 3.4.



Slika 3.4. Primjer koda kojim se dohvaća element p

U gore prikazanom isječku koda unutar <script> elementa napisan je kod za dohvaćanje HTML elementa p i postavljanje njegovog teksta na „Testni naslov“. Logo HTML-a, CSS-a i JS-a prikazani su na slici 3.5.

Slika 3.5. HTML, CSS i JS logo

Izvor: [10]

3.4. Google Firebase

Google Firebase je platforma koja sadrži skup razvojnih alata koji su temeljeni na oblaku i programerima pomažu pri lakšoj izradi, održavanju i implementaciji web i mobilnih aplikacija.

Prijava i kreiranje projekata na ovoj platformi je potpuno besplatna, a za svaki alat koji se koristi, *Firebase* nudi specificiranu analitiku. Uz to nudi mogućnost spremanja podataka u NoSQL bazu podataka, mogućnost čuvanja podataka o korisnicima koji se prijavljuju na aplikaciju putem predefinirane metode za prijavu pod nazivom „Sign-in method“.

3.4.1. Cloud Firestore

Cloud Firestore je NoSQL baza podataka smještena u oblaku kojima mogu pristupiti web i mobilne aplikacije putem izvornog paketa za razvijanje softvera (SDK) (Izvor: [10]). Podaci koji su spremljeni u bazi podataka sadrže mapiranja za unesene vrijednosti. Mapiranja omogućuju povezivanje različitih modela podataka koji zajedno čine cjelinu. Ti dokumenti podržavaju brojne tipove podataka kao što su int, string, Date, float, nizovi, polja, liste i brojni drugi. U suštini,

strukture podataka koje su potrebne za rad aplikacije moguće ih je kreirati i podržane su na *Cloud Firestore*.

Podacima spremljenim u ovu bazu podataka može se izravno pristupiti postavljanjem određenih upita (engl. *query*). Također moguće je i poželjno zaštititi pristup ovim podacima s autentifikacijskim i sigurnosnim pravilima.

3.4.2. Firebase Authentication

Firebase Authentication je autentifikacijska značajka na platformi *Firebase* kojom se stvara mogućnost znanja o korisniku te o točno kojem korisniku se radi u radom trenutku (Izvor: [11]). Korisnike web aplikacije moguće je prijaviti u stvorenu *Firebase* aplikaciju korištenjem FirebaseUI-ja ili korištenjem *Firebase Authentication* SDK-a. Prijava je moguća na više načina, a jedna od najlakših metoda je *Google sign-in.*

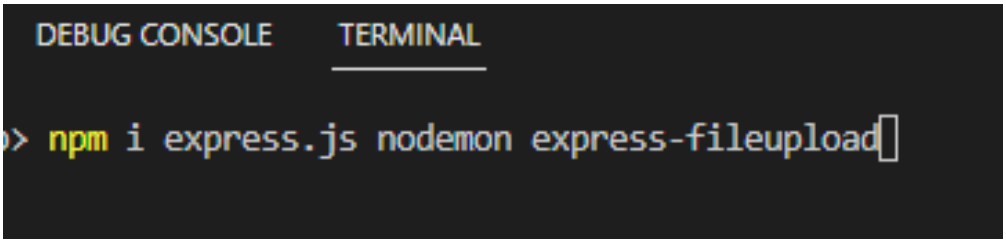
4. POSTUPAK IZRADE APLIKACIJE

U ovom poglavlju bit će prikazan slijed kreiranja web portala za nogometne vijesti. Prije početka izrade, potrebno je odabrati pogodno razvojno okruženje i programski jezike. Za ovu namjenu, odabran je *Visual Studio Code* i programski jezik *JavaScript* (u nastavku označen kao JS) koji se pokazao kao najefikasniji za ovaj rad.

4.1. Kreiranje i pokretanje *JavaScript* web portala

Za pokretanje JS web portala potrebno je napisati nekoliko bitnih naredbi s kojim se omogućuje povezivanje i rad aplikacije na webu. Prva od njih je *npm init* s kojom se inicijalizira upravitelj paketa za JS odnosno *npm*. S tom naredbom kreirana je *package.json* datoteka. U toj datoteci nalaze se naziv, verzija i ovisnosti, to jest opisni i funkcionalni podaci o VS Code projektu.

Zatim, potrebno je instalirati pakete koji su potrebni za rukovanjem potrebnih funkcionalnosti, a to su *express.js* i *express-fileupload.* Ova radnja se vrši komandnom linijom koja je prikazana na slici 4.1:



Slika 4.1. Kod za instaliranje potrebnih paketa

Instalacijom *express-fileupload* omogućena je funkcionalnost spremanja unesenih datoteka s web stranice u projekt kojoj se zatim može pristupiti pomoću metode objekta *req*: *req.files.* Također je stvorena je zaključana verzija datoteke *package.json* pod imenom *package-lock.json*. Ova datoteka zaključava sve instalirane ovisnosti na određenu verziju.

Za osposobljavanje portala za rad na webu potrebna je datoteka *index.js* u koju se „uvoze“ svi paketi koji su prethodno instalirani. Datoteka *index.js* prikazana je na slici 4.2. Uvoz paketa se postiže funkcijom *express* te se ti pakedi pridodaju aplikaciji da ih koristi metodom *use*. Na kraju se postavlja „port za slušanje“ na vijednost 4000 na koju će se pristupati web portalu.

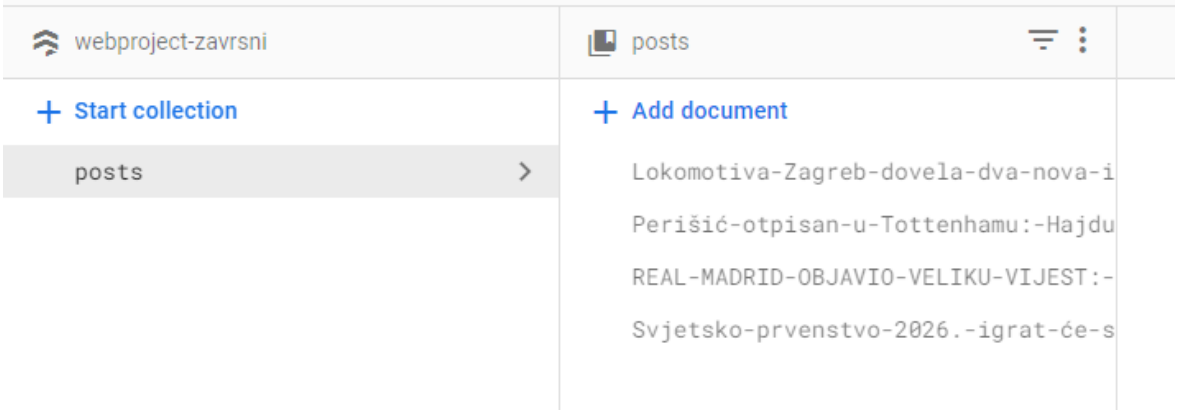
```
JS index.js > ...
1   const express = require('express')
2   const fileUpload = require('express-fileupload')
3   const path = require('path')
4
5   const initial_path=path.join(__dirname, "public")
6
7   const app = express()
8   app.use(express.static(initial_path))
9   app.use(fileUpload())
10
11
12  app.get('/', (req,res) => {
13      res.sendFile(path.join(initial_path, "home.html"))
14  })
15
16  app.listen("4000", () => {
17      console.log("Server started on port 4000")
18  })
```

Slika 4.2. Kod u *index.js*

Sada je web portal spreman za pokretanje i to se postiže naredbom *npm start* u terminalu.

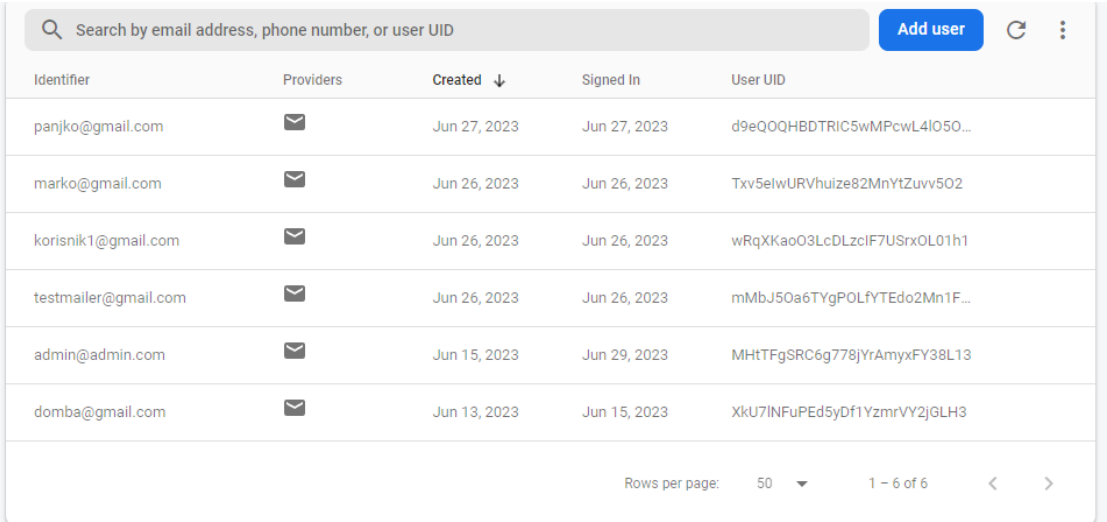
4.2. Kreiranje *Firebase* baze podataka i autentifikacije

Za uspješno rukovanje podacima koji će se koristiti potrebna je baza podataka. *Cloud Firestore* je korišten u ovu svrhu jer nudi bazu podataka smještenu u oblaku. Svakim dodavanjem novih podataka baza se sinkronizira pomoću slušatelja u stvarnom vremenu. Za početak rada kreiran je *Firebase* te je u njemu izgrađena baza podataka. Na slici 4.3. prikazan je model baze podataka pod nazivom „*posts*“ i u njoj nekoliko dokumenata koji predstavljaju vijesti.



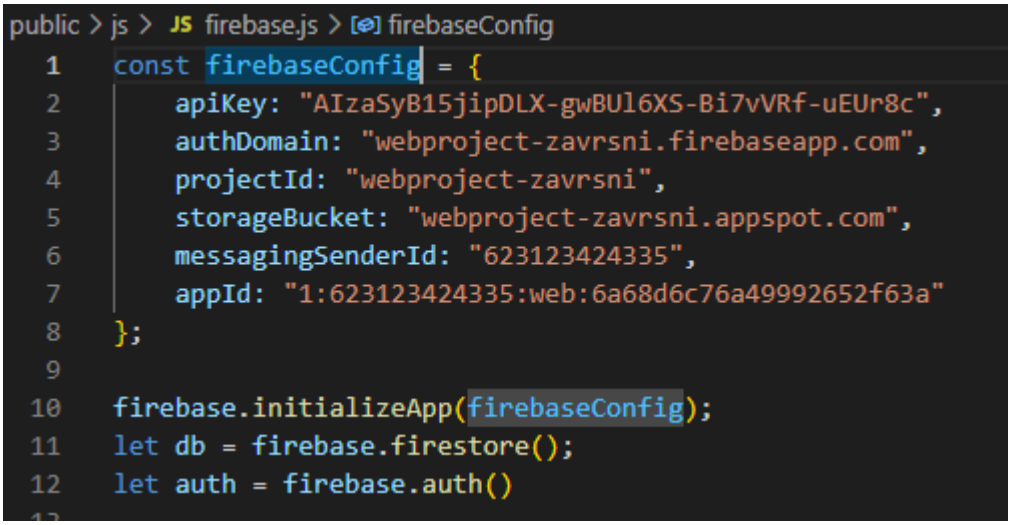
Slika 4.3. *Firebase* model baze podataka „posts“

Uz bazu podataka, za ovu aplikaciju potreba je registracija i prijava korisnika u portal. To se postiže izgradnjom autentifikacije na *Firebase*-u koji nudi više metoda prijave kao što su standardna email/password prijava, Facebook, Gmail, Apple, Twitter i brojni drugi. Kao najefikasnija i najjednostavnija opcija odabrana je prijava putem e-maila i zaporke. Prikaz kako izgleda kreirana tablica autentifikacije dana je slikom 4.4.



Slika 4.4. Tablica s podacima autentificiranih korisnika portala

Povezivanje *Firebase*-a s aplikacijom moguće je provedbom metode inicijalizacije aplikacije s konfiguracijskim podacima *Firebase* projekta koji je kreiran. Najbitniji od tih podataka su API ključ, ID projekta i ID aplikacije. Provedba ove metode prikazana je na slici 4.5.



Slika 4.5. Pohrana zajedničke konfiguracije u aplikaciji

Varijabla *db* sada predstavlja instancu na *Firebase* bazu podataka, a varijabla *auth* instancu na *Firebase* autentifikaciju. Ove varijable mogu se koristiti u svim dijelovima aplikacije uz navođenje elementa <script src="js/firebase.js"></script>.

Firebase nudi ugrađene metode autentifikacije korisnika kao što su *createUserWithEmailAndPassword* za kreiranje novog računa korisnika u portalu. Ako korisnik već ima stvoren račun, metoda koja se koristi za prijavu je *signInWithEmailAndPassword*. Unutar tih metoda predaju se argumenti e-mail i zaporka.

4.3. Izrada sučelja za kreiranje nove objave

Stvaranje objava na portalu i spremanje podataka u *Firebase* vrši se pomoću zasebne web adrese, odnosno sučelja za kreiranje nove objave. Ovoj web adresi može pristupiti samo administrator portala jer je on nadležan za objavljivanje vijesti.

Cjelokupni dizajn ovog sučelja definiran je opisnim jezikom CSS koji stilski opisuje elemente iz datoteke *editor.html*, odnosno pretvara jednostavne HTML elemente u uređene i atraktivne elemente s ciljem privljačenja korisnika. Dizajniran je tako da se prvo unosi naslovna slika objave koja se proteže cijelom širinom stranice, zatim odabir jedne od tri vrste nogometnih vijesti ovisno o regiji, naslov, tekst objave i po želji dodatne slike uz tekst. Kako bi ovo bilo moguće, instaliran je *express-fileupload* na samom početku izrade ovog web portala. Tom instalacijom omogućen je prijenos slikovnih datoteka na portal i spremanje u bazu podataka. Nije moguće postavljanje nove objave bez unosa naslovne slike.

U glavnoj JS datoteci *index.js* definirana je metoda spremanja unesene slike u mapu „uploads" koja je sastavni dio ove aplikacije. Prikaz koda te metode dan je slikom 4.6.

```
app.post('/upload', (req, res) => {
    const file = req.files.image;
    let date = new Date();
    // image name
    let imagename = date.getDate() + date.getTime() + file.name;
    // image upload path
    let path = 'public/uploads/' + imagename;

    // create upload
    file.mv(path, (err, result) => {
        if(err){
            throw err;
        } else{
            // our image upload path
            res.json(`uploads/${imagename}`)
        }
    })
})
```

Slika 4.6. Metoda *post* za spremanje slika unesenih s web poslužitelja

Objekt *req* predstavlja zahtjev koji je poslan iz metode *uploadImage* i u ovom slučaju je to polje podataka. Iz polja podataka uzima se ime slike te se zatim preoblikuje i konačno sprema u mapu *uploads* pomoću objekta *res* koji predstavlja odgovor na zahtjev.

Za svaku unesenu sliku program iz datoteke *editor.js* poziva tu post metodu i predaje joj podatke formirane pomoću *FormData()* klase kao što je prikazano u liniji 23 na slici 4.7. Kao odgovor na taj zahtjev dobiva se podatak spremljen u mapu *uploads* i šalje ih se dalje u metodu *addImage*. Ovaj proces prikazan je na slici 4.7.

```
20  const uploadImage = (uploadFile, uploadType) => {
21      const [file] = uploadFile.files;
22      if(file && file.type.includes("image")){
23          const formdata = new FormData();
24          formdata.append('image', file);
25          fetch('/upload', {
26              method: 'POST',
27              body: formdata
28          }).then(res => res.json())
29          .then(data => {
30              if(uploadType=="image"){
31                  addImage(data, file.name)
32              } else{
33                  bannerPath = `${location.origin}/${data}`;
34                  banner.style.backgroundImage = `url("${bannerPath}")`;
35              }
36          })
37      } else{
38          alert("upload Image only");
39      }
40  }
41
42  const addImage = (imagepath, alt) => {
43      let currentPosition = articleField.selectionStart;
44      let textToInsert = `\r![${alt}](${imagepath})\r`;
45      articleField.value = articleField.value.slice(0, currentPosition) + textToInsert + articleField.value.slice(currentPosition);
46  }
```

Slika 4.7. Metode *uploadImage* i *addImage*

U metodi *addImage* kreirana je varijabla *currentPosition* kojoj se dodjeljuje vrijednost trenutne pozicije kursora u pisanju teksta artikla. Na njegovu poziciju nadodaje se modificirani tekst koji u sebi sadrži adresu na kojoj je spremljena unesena slika u mapi „uploads“. U tom trenutku ovaj modificirani tekst nije poprimio oblik slike već će ga poprimiti tek nakon klika na gumb „Objavi“.

Klikom na gumb „Objavi“, odnosno u kodu *publishButton*, pokreće se funkcija *addEventListener* prikazana na slici 4.8. za taj gumb. U ovoj funkciji se generira nasumični id za objavu, formira se oblik prikaza datuma objave, dohvaća se odabrani radijski gumb (engl. radio button) i kreira se prazna lista komentara. Metodom *db.collection(„posts“).doc(docName).set()* podaci se spremaju u *Firestore* bazu podataka.

```
publishBtn.addEventListener('click', () => {
    if(postTitleField.value.length && articleField.value.length){

        let docName;
        //generiranje id
        let letters = 'abcdefghijklmnopqrstuvwxyz';
        let postTitle = postTitleField.value.split(" ").join("-");
        let id = '';
        for(let i = 0; i < 4; i++){
            id += letters[Math.floor(Math.random() * letters.length)];
        }

        if(postID[0] == 'editor'){
            //postavljanje imena objavi
            docName = `${postTitle}-${id}`;
        } else{
            docName = decodeURI(postID[0])
        }

        let date = new Date();

        const optionButtons = document.getElementsByName('option');
        for (i = 0; i < optionButtons.length; i++) {
            if (optionButtons[i].checked)
                var checkedOption = optionButtons[i].value;
        }

        const comments = []
        db.collection("posts").doc(docName).set({
            title: postTitleField.value,
            article: articleField.value,
            bannerImage: bannerPath,
            publishedAt: `${date.getDate()}. ${months[date.getMonth()]} ${date.getFullYear()}.`,
            author: auth.currentUser.email.split("@")[0],
            comments: comments,
            typeOfFootball: checkedOption
        })
        .then(() => {
            location.href = `/${docName}`;
        })
        .catch((err) => {
```

Slika 4.8. Funkcija *addEventListener* za gumb „Objavi“

4.4. Izrada glavne stranice web portala

Potrebno je strukturirati i dizajnirati naslovnu stranicu web portala. Prvi korak je kreiranje navigacijske trake u kojem su svi potrebni hiperlinkovi na druge web adrese. Glavna putanja koja vodi na naslovnu stranicu je označena znakom „/“ i postavljena je na logo ovog web portala.

Ovisno o tome je li korisnik prijavljen ili ne, bit će prikazani dodatni elementi s desne strane navigacijske trake. Ako korisnik nije prijavljen, bit će prikazan gumb za prijavu na skroz desnoj strani. Klikom na taj gumb portal ga vodi na novu web adresu s putanjom /auth. Na toj web adresi korisniku je ponuđena opcija prijave na postojeći račun ili registracije čime se kreira novi. Ovu funkcionalnost omogućio je Firebase. Nakon uspješne prijave korisnik je vraćen na naslovnu stranicu te onda navigacijska traka ima novi oblik. Umjesto gumba prijava sada je prikazan gumb za odjavu i tekst koji predstavlja korisničko ime prijavljenog korisnika. Navigacijska traka je prikazana kodom sa slika 4.9. i 4.10.

```html
<nav class="navbar">
    <a href="/"><img href="/" src="images/logo.PNG" class="logo" width=100 height=100 alt="NV logo"></a>
    <ul class="links-container">

    </ul>
</nav>
```

Slika 4.9. HTML kod za navigacijsku traku

```javascript
let ul = document.querySelector('.links-container')

auth.onAuthStateChanged((user) => {
    if(user){
        if(auth.currentUser.email == "admin@admin.com"){
            ul.innerHTML += `
            <li class="link-item"><a class="username">${auth.currentUser.email.split("@")[0]}</a></li>
            <li class="link-item"><a href="/editor" class="link">Nova objava</a></li>
            <li class="link-item"><a href="#" onclick="logoutUser()" class="link">Odjava</a></li>
            `
            const adminusername = document.querySelector(".username").style.color="red"
        }else{
            ul.innerHTML += `
                <li class="link-item"><a class="link">${auth.currentUser.email.split("@")[0]}</a></li>
                <li class="link-item"><a href="#" onclick="logoutUser()" class="link">Odjava</a></li>
            `
        }
    } else{
        ul.innerHTML += `
        <li class="link-item"><a href="/auth" class="link">Prijava</a></li>
        `
    }
})
```

Slika 4.10. JS kod za navigacijsku traku

Ovaj JS kod može se koristiti u bilo kojem HTML dokumentu koji sadrži element *links-container* pa je tako korišten u dva HTML dokumenta, odnosno dvije web adrese a to su *home.html* i *post.html*. Na slici 4.11. prikazan je kod u kojem se prvotno provjerava je li korisnik prijavljen te je li riječ o adminu, zatim se na dohvaćeni HTML element *links-container* operatorom „+=“dodaju elementi koji su potrebni ovisno o ishodu prijave.

Naslovna stranica portala započeta je velikim naslovom te zatim dodavanjem objava u element *post-section* kao što je prikazano u kodu sa slike 4.11. Traženi element dohvaćen je funkcijom *querySelector* i spremljen u konstantu pod nazivom *postSection* koji se kao parametar uvrštava u funkciju *showPosts*. Ova funkcija također provjerava je li korisnik prijavljen i na osnovu toga nadopunjuje prazan element s podelementima koji formiraju objavu, tj. nogometnu vijest. Svaka objava formira se podacima prethodno spremljenim u *Firestore* bazu podataka. Svaka pojedina objava ima element naslovne slike, teksta naslova, datuma objavljivanja, broj komentara i gumbe koji se prikazuju ovisno o statusu korisnika.

```
const showPosts = (post) => {
    let data = post.data();

    auth.onAuthStateChanged((user) => {
        if(user && user.email == "admin@admin.com"){
            postSection.innerHTML += `
            <div class="post-card">
                <img src="${data.bannerImage}" class="post-image" alt="">
                <p class="post-date">Objavljeno: ${data.publishedAt}</p>
                <img src="images/commenticon.png" class="comment-icon" alt="">
                <p class="comment-count">${data.comments.length}</p>
                <h1 class="post-title">${data.title.substring(0, 50) + '...'}</h1>
                <a href="/${post.id}" class="custom-btn btn-read">Saznaj više</a>
                <a href="/${post.id}/editor" onclick="editPost('${post.id}')" class="custom-btn btn-edit">uredi</a>
                <a href="#" onclick="deletePost('${post.id}')" class="custom-btn btn-delete">obriši</a>
            </div>
            `;
            i+=1;
        } else{
            postSection.innerHTML += `
            <div class="post-card">
                <img src="${data.bannerImage}" class="post-image" alt="">
                <img src="images/commenticon.png" class="comment-icon" alt="">
                <p class="comment-count">${data.comments.length}</p>
                <p class="post-date">Objavljeno: ${data.publishedAt}</p>
                <h1 class="post-title">${data.title.substring(0, 50) + '...'}</h1>
                <div onclick="readPost('${i}','${post.id}')" class="custom-btn btn-read">Saznaj više
                    <span class="popuptext" id="${i}">Potrebna je prijava</span>
                </div>
            </div>
            `;
            i+=1;
        }
    })
}
```

Slika 4.11. Kod za nadopunjavanje elementa *post-section*

U slučaju da je korisnik nije admin prikazan je samo gumb koji vodi na novu web adresu s putanjom /post.id gdje je izlistan kompletan sadržaj objave. Ako je situacija obratna, to jest ako je

korisnik zapravo admin, funkcija *showPosts* će za svaku objavu nadodati dva nova gumba koji nude opciju uređivanja i brisanja objave.

4.5. Izrada opširnijeg prikaza pojedine objave

Kao i kod svakog web portala s vijestima, jako je bitno dodati mogućnost da korisnik vidi detalje informacije koje ga zanimaju. S naslovne stranice moguće je posjetiti pojedinu objavu klikom na gumb „Saznaj više“ koji ga vodi na adresu s putanjom /${post.id}. Ova vrijednost predstavlja prethodno generirani slučajni identifikacijski ključ koji je jedinstven i spremljen unutar svoje objave u kolekciji.

Firebase referenci pristupa se pomoću metode *get()* uz kojeg se navodi ID od objave koja se traži. Nakon toga pokreće se proces postavljanja izgleda objave uz dohvaćanje svake vrijednosti iz baze podataka za tu specifičnu objavu. Nakon dohvaćanja svake vijesti poziva se funkcija *addArticle* koji redom slaže elemente podataka na ovu web adresu kao što je prikazano na slici 4.12.

```
const addArticle = (ele, data) => {
    data = data.split("\n").filter(item => item.length);
    //console.log(data);

    data.forEach(item => {
        //provjera headinga
        if(item[0] == '#'){
            let hCount = 0;
            let i = 0;
            while(item[i] == '#'){
                hCount++;
                i++;
            }
            let tag = `h${hCount}`;
            ele.innerHTML += `<${tag}>${item.slice(hCount, item.length)}</${tag}>`
        }

        //provjera formata slike
        else if(item[0] == "!" && item[1] == "["){
            let seperator;

            for(let i = 0; i <= item.length; i++){
                if(item[i] == "]" && item[i + 1] == "(" && item[item.length - 1] == ")"){
                    seperator = i;
                }
            }

            let alt = item.slice(2, seperator);
            let src = item.slice(seperator + 2, item.length - 1);
            ele.innerHTML += `
                <img src="${src}" alt="${alt}" class="article-image">
            `;
        }

        else{
            ele.innerHTML += `<p>${item}</p>`;
        }
    })
}
```

Slika 4.12. Funkcija *addArticle* za nadopunjavanje HTML koda na specifičnoj web adresi

Ovim kodom također se izvlači izvor slike kako bi ju bilo moguće uvrstiti u element *img*.

Nakon izlistanja svih podataka ostvarena je funkcionalnost komentiranja pojedine objave. Komentari se također spremaju u bazu podataka za specifični element kolekcije. Kreiran je blok element koji služi za pisanje komentara koji sadrži HTML elemente <*textarea*> za unos teksta komentara i dva button elementa <*button*> koji služe za objavu i brisanje teksta komentara. Nakon klika na gumb za objavu komentara pokreće se funkcija *addEventListener* za varijablu commentBtn kao na slici 4.13. Provodi se provjera u slučaju da korisnik nije prijavljen i automatski ga vodi na putanju */auth* na kojoj ima mogućnost prijave ili registracije. Zatim se provjerava je li uopće unesen tekst komentara za formulaciju polja podataka koji je potrebno spremiti u *Firebase*. Iz razloga što *Firebase* ne podržava ugniježđena polja potrebno je to polje pretvoriti u objekt. Za ključ se uzima ime autora komentara, a za vrijednost ključa uzima se tekst. U varijablu *commentSection* automatiski se zatim dodaju elementi koji formuliraju novi blok element predstavljajući objavljeni komentar. Tekst komentara se postavlja na prazan string za mogućnost objave novog komentara.

```
commentBtn.addEventListener('click', function(event) {
    event.preventDefault();
    auth.onAuthStateChanged((user) => {
        if(!user){
            location.href = `/auth`;
        }else{
            if(commentField.value.length){
                let date = new Date();
                const comment = [
                    ["name",auth.currentUser.email.split("@")[0]],
                    ["publishedAt",`${date.getDate()} ${months[date.getMonth()]} ${date.getFullYear()}`],
                    ["body",commentField.value]
                ]

                //pretvaranje nested arraya u object
                const commentObject = {}
                for(const item of comment){
                    const key=item[0];
                    const value=item[1];
                    commentObject[key]=value;
                }

                let postRef = db.collection("posts").doc(decodeURI(thispostID[0]))
                postRef.get().then(function(doc) {
                    if(doc.exists){
                        const commentsArray = doc.data().comments;
                        commentsArray.push(commentObject)
                        return postRef.update({comments:commentsArray})
                    }else{
                        location.replace("/");
                    }
                })
                let lastcommentSection = commentSection.innerHTML;
                commentSection.innerHTML = `
                    <div class="comments">
                        <div class="commentcontainer">
                            <form>
                                <h6 class="publishedDate"><span>${comment[1][1]}</span></h6>
                                <h2 class="comtitle">${comment[0][1]}:</h2>
                                <textarea type="text" class="commentbox">${comment[2][1]}</textarea>
                            </form>
                        </div>
                    </div>
                    ` + lastcommentSection;
                commentField.value = "";
            } else{
                console.log("Unesi tekst")
            }
```

Slika 4.13. Funkcija *addEventListener* za postavljanje komentara na objavu

Svaki komentar za specifičnu objavu bit će prikazan i to redom od najnovijeg do najstarijeg.

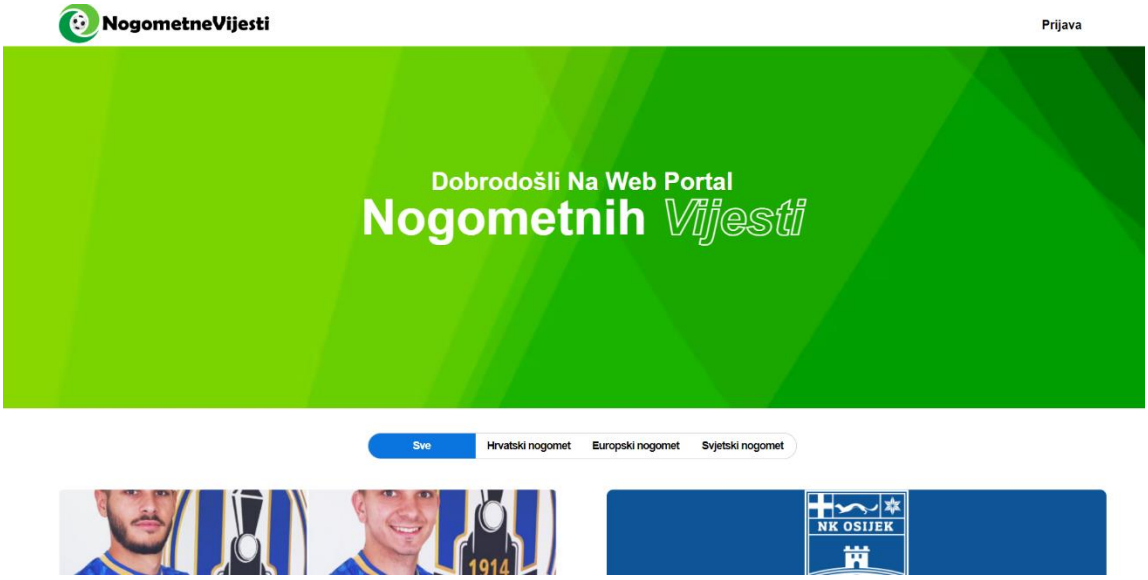
Nakon komentara dodana je funkcionalnost izlistanja svih ostalih objava s portala ako korisnik želi brzo doznati što više bez da se vraća na naslovnu stranicu. Kod je isti kao i kod izlistanja pojedine objave na naslovnoj stranici.

5. PRIKAZ RADA WEB PORTALA

U ovom poglavlju prethodno objašnjene funkcionalnosti bit će prikazane kroz slike redom kako je aplikacija kreirana.

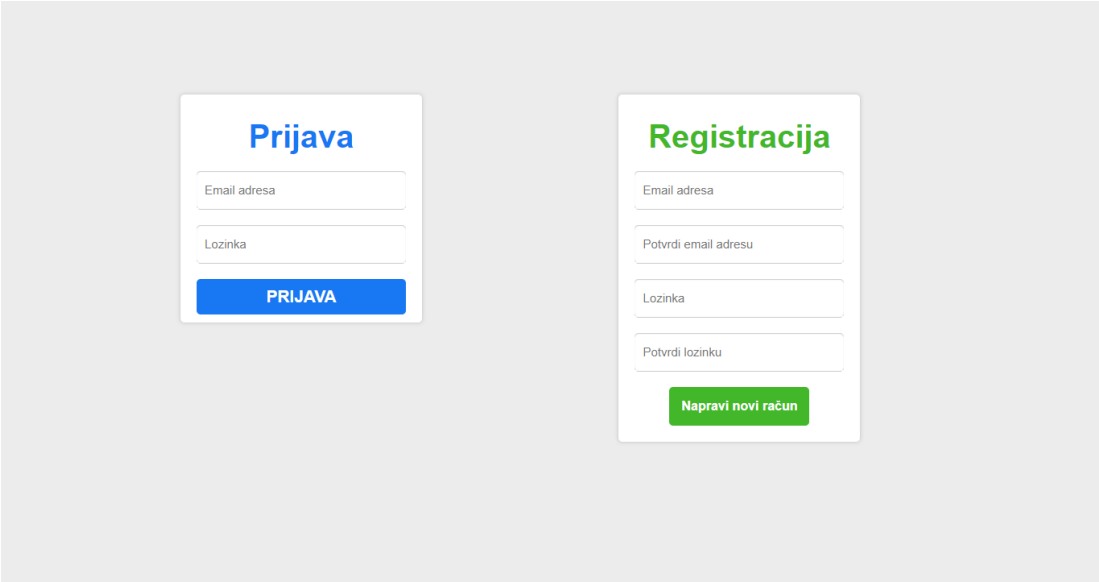
5.1. Registracija ili prijava korisnika

Prvi korak kod učitavanja aplikacije je pokretanje naredbe *npm start* kojom se server pokreće na portu 4000. Kako bi korisnik dospio na putanju za registraciju ili prijavu potrebno je pritisnuti gumb „Prijava“ u gornjem desnom kutu kao na slici 5.1.



Slika 5.1. Početni zaslon s navigacijskom trakom i gumbom „Prijava“

Klikom na gumb korisnika se šalje na putanju */auth* u kojem ima dva blokovna elementa: prijava i registracija. Ako korisnik već postoji u *Firebase* autentifikaciji prijavit će se svojim e-mailom i zaporkom. U suprotnom kreirat će novi račun i zbog mjera sigurnosti dvaput se traži e-mail i zaporka. Prikaz zaslona za autentifikaciju dan je na slici 5.2. *Firebase* automatski regulira je li e-mail pogrešnog oblika ili ako je zaporka prekratka.

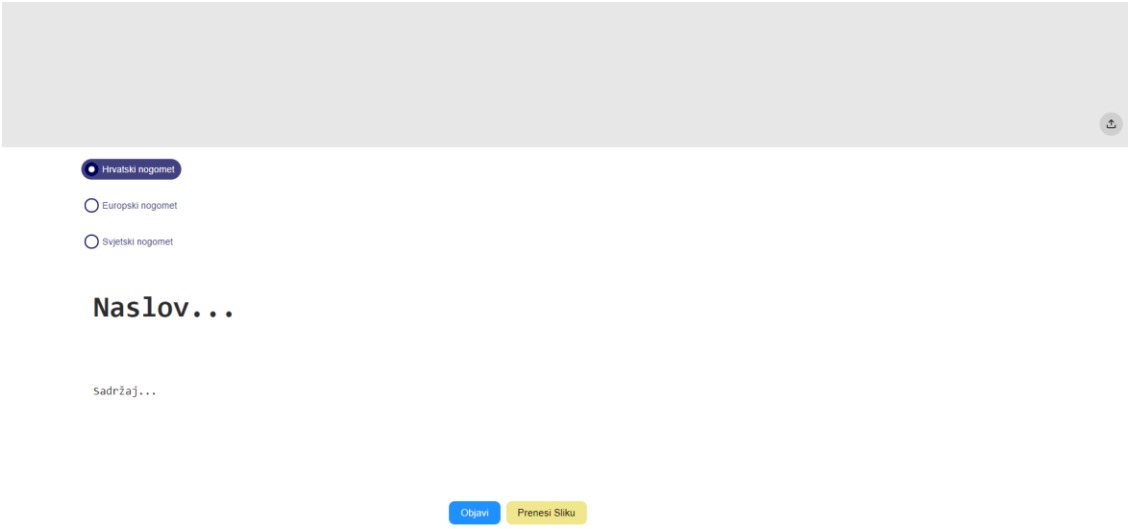


Slika 5.2. Zaslon za prijavu ili registraciju korisnika

Nakon uspješne prijave, korisnika se vraća na početni zaslon („/“).

5.2. Kreiranje nove objave

Za kreiranje objave koristi se putanja */editor*. Kreirati novu objavu može samo admin i nitko drugi, pokuša li netko nasilno pristupiti putanji vratit će ga na početni zaslon. Prečac do ove putanje ima samo admin kome se u navigacijskoj traci prikazuje gumb „Nova Objava“. Slika 5.3. prikazuje zaslon u kojem su prazna polja koja nadopunjuje admin.

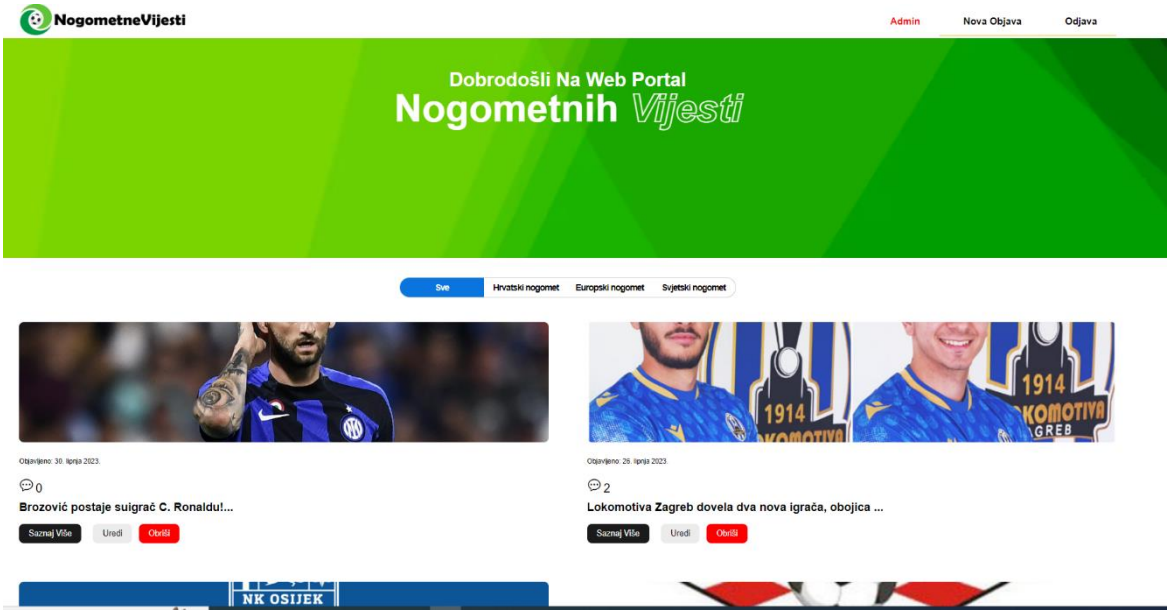


Slika 5.3. Zaslon za postavljanje nove objave

Na ovom zaslonu admin pomoću radio gumba bira hoće li pisati o hrvatskom, europskom ili svjetskom nogometu, zatim polje za unos naslova i sadržaja i na kraju gumbi za objavu ili prenošenje slike u opis objave.

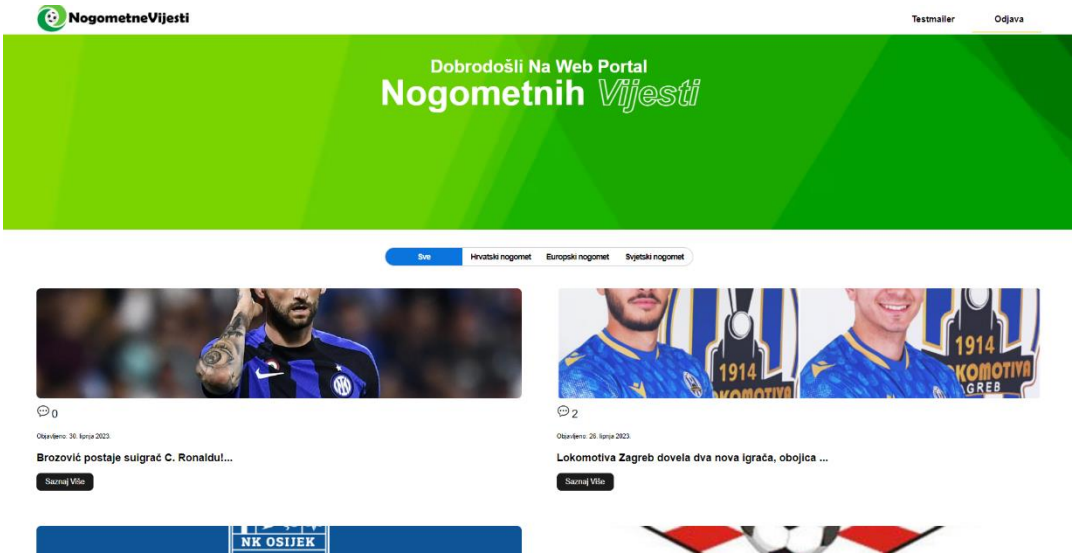
5.3. Početni zaslon web portala

Početni zaslon sastoji se od navigacijske trake, velikog naslova dobrodošlice uređenog uz pomoć CSS koda, filtra za tip nogometne vijesti i konačno izlistanja svih postojećih objava. Vijesti se mogu filtrirati po atributu *typeOfFootball* koji se unosi prilikom kreacije objave. Moguće je također filtrirati vijesti tako da se prikažu sve, samo hrvatske, samo europske ili samo svjetske vijesti. Defaultna vrijednost je prikazivanje svih vijesti kao na slici 5.4. U ovom prikazu je slučaj kad je korisnik prijavljen kao admin.



Slika 5.4. Početni zaslon web portala iz perspektive admina

Na slici 5.5. nije riječ o adminu pa se primjećuje razlika u navigacijskoj traci i gumbima ispod pojedine objave. Na ovaj način korisnik može samo čitati i filtrirati objave.



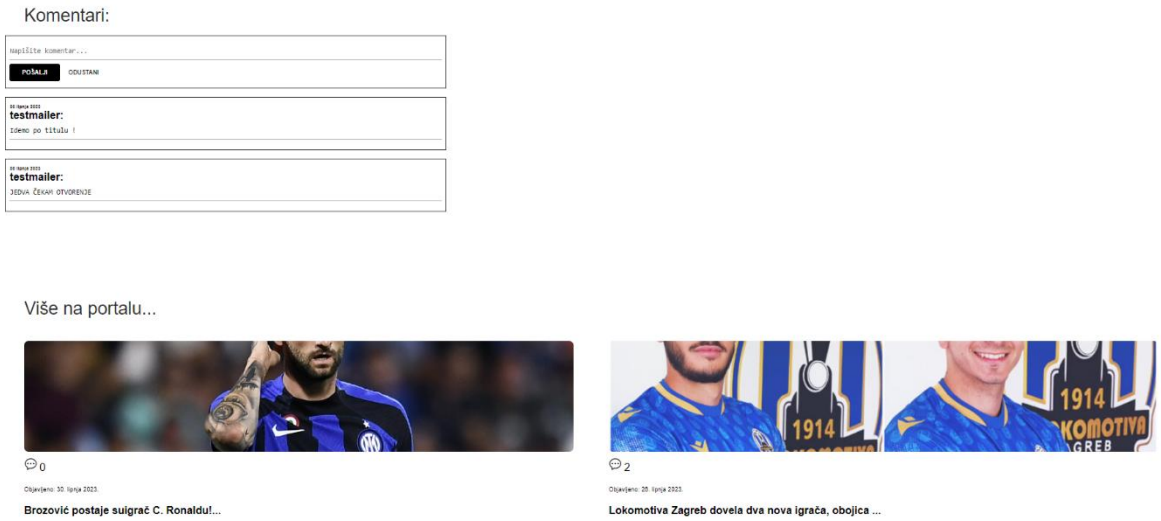
Slika 5.5. Početni zaslon web portala iz perspektive korisnika

5.4. Prikaz pojedine objave portala

Na ovom zaslonu prikazana je glavna slika objave u punoj širini, podebljani naslov objave, sadržaj uz mogućnost pojave slike koja pobliže opisuje vijest i komentari. Komentare može pisati bilo tko tko je prijavljen. Uz navedene komponente postoje i detalji kao što su datum postavljanja objave i pojedinog komentara, autora komentara te pri dnu izlistanje svih ostalih objava za one znatiželjne. Na slici 5.6. prikazan je primjer objave uz komponente naslovne slike, teksta naslova, datuma objave i sadržaja.

Slika 5.6. Prikaz objave vijesti za novi stadion NK Osijeka

Komentari su sastavni dio rasprave o svim vijestima na svijetu, pa tako i nogomentim. Ispod sadržaja svake pojedine objave izgenerirani su svi komentari povezani za identifikacijski ključ objave kao što je prikazano na slici 5.7. Pritiskom na gumb „Pošalji“ istog trenutka se komentar pojavljuje kao najnoviji i prvi u redu.



Slika 5.7. Prikaz izlistanja komentara i ostalih objava portala

6. ZAKLJUČAK

Cilj ovog završnog rada bio je kreirati web portal za nogometne vijesti. S razvojem današnjih tehnologija i poznavanjem programskih jezika za izradu web stranica proces izrade vrlo brzo napreduje. Proces izrade web aplikacije podložan je takozvanim *bug*-ovima, tj. greškama u pokretanju željenog koda te iz tog razloga svaki dio koda mora proći kroz testiranje kako bi stekli sigurnost u funkcionalnost aplikacije. Tijekom izrade web portala, također su se pojavljivali brojni tehnički izazovi kao što su učenje i korištenje novih programskih jezika i sučelja. Savladavanjem tih izazova stekao sam znanja i vještine u području web tehnologija te je u konačnici zadatak uspješno savladan implementacijom svih potrebnih funkcionalnosti koje web portali zahtijevaju te dodavanjem novih ideja koje, po određenim kriterijima, ovaj portal čine efikasnijim i lakše prilagodljivim. S pojavom novih tehnologija u budućnosti, web portal je izložen napretku, novim fukncionalnostima, kvalitetnijoj prilagodbi i usluzi korisnicima.

LITERATURA

[1] *Nogometne-vijesti* https://nogometne-vijesti.hr/ pristupljeno 23.6.2023.

[2] *Klik nogomet* https://klik.hr/nogomet pristuljeno 23.6.2023.

[3] *24sata nogomet* https://www.24sata.hr/nogomet pristupljeno 23.6.2023.

[4] *Skysports football* https://www.skysports.com/football/news pristupljeno 23.6.2023.

[5] *Marca football* https://www.marca.com/futbol.html pristupljeno 24.6.2023.

[6] *Visual Studio Code* https://visualstudio.microsoft.com/ pristupljeno 13.6.2023.

[7] Materijali s kolegija Osnove razvoja web i mobilnih aplikacija (prof. Dr. sc. Krešimir Nenadić) pristupljeno 23.6.2023.

[8] Knjiga „*Naučite JavaScript*“ autor: Kyle Simpson pristupljeno 15.6.2023.

[9] *JavaScript basics* https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics pristupljeno 19.6.2023.

[10] *Cloud Firestore* https://firebase.google.com/docs/firestore pristupljeno 20.6.2023.

[11] *Firebase Authentication* https://firebase.google.com/docs/auth pristupljeno 20.6.2023.

SAŽETAK

Nogometne vijesti su jedna od glavnih kategorija vijesti današnjice. Vođenje web portala za nogometne vijesti velik je izazov jer zahtijeva što brže i točnije informacije o stvarnim nedavnim zbivanjima u nogometnom svijetu. Također je vrlo bitno da web portal izgleda privlačno i estetski kako bi se postigla veća posjećenost i prilagodba korisnika. Funkcionalnosti web portala objašnjene su korak po korak i prikazane putem slika.

Ključne riječi: estetika, nogomet, vijesti, web portal

ABSTRACT

Football news are one of the main categories of news today. Running a web portal for football news is a big challenge because it requires fast and accurate information about real recent events in the football world. It is also very important that the web portal looks attractive and aesthetic in order to achieve greater traffic and user adaptation. The functionalities of the web portal are explained step by step and shown through images.

Key words: aesthetics, football, news, web portal