

RAZVOJ APLIKACIJE ZA VOĐENJE KNJIŽNICE KORIŠTENJEM VUE.JS I FIREBASE TEHNOLOGIJA

DEVELOPMENT OF A LIBRARY APPLICATION USING VUE.JS AND FIREBASE TECHNOLOGIES

Mario Vidović¹, Sanja Kraljević²

¹Tehničko veleučilište u Zagrebu, Vrbik 8, Zagreb, Hrvatska, student

²Tehničko veleučilište u Zagrebu, Vrbik 8, Zagreb, Hrvatska

SAŽETAK

U ovom radu prikazan je postupak razvoja aplikacije za vođenje online knjižnice te su opisane i komentirane korištene tehnologije. Cilj je stvoriti intuitivnu i jednostavnu aplikaciju za katalogizaciju i pretraživanje knjižničke građe, te omogućiti korisnicima da rezerviraju knjige i prate svoje posudbe. Opisani su kriteriji odabira tehnologija kao i glavne funkcionalnosti aplikacije. Aplikacija koristi Firebase kao bazu podataka za pohranu podataka o knjigama i novostima, ali i kao sustav za autentifikaciju korisnika. Za izradu prednjeg dijela aplikacije korišten je radni okvir Vue.js. Rad obuhvaća proces izrade te prikazuje gotov proizvod, uključujući funkcionalnosti aplikacije, vrste korisnika aplikacije te korištenje digitalne članske iskaznice za jednostavniju identifikaciju korisnika.

Ključne riječi: *Vue, Firebase, e-knjižnica, rute, komponente*

ABSTRACT

In this paper, the process of developing an application for managing an online library is presented, and the technologies used are described and commented on. The goal is to create an intuitive and simple application for cataloging and searching library materials, and to enable users to reserve books. The criteria for selecting technologies as well as the main functionalities of the application are described. The application uses Firebase as a database to store data about

books and news, but also as a user authentication system. The Vue.js framework was used to create the front end of the application. The paper shows the creation process, including the functionality of the application, the types of application users and the use of a digital membership card for easier user identification.

Keywords: *Vue, Firebase, e-library, router, components*

1. UVOD

1. INTRODUCTION

U vrijeme kada ljudi većinu svoga vremena provode na mobitelima ili za računalom, nerijetko se pojavljuju aplikacije s ciljem olakšavanja svakodnevnog života. Aplikacija za online rezervaciju knjiga u knjižnici je jedna od njih, a u praktičnom dijelu rada je razvijena takva aplikacija. Njezin cilj je omogućiti korisnicima jednostavnu i brzu rezervaciju knjiga koje žele čitati, bez potrebe za dugim čekanjem u redovima, s različitih uređaja (primjerice s računala, mobitela ili tableta). Ova aplikacija omogućuje knjižnicama da automatiziraju proces rezervacije knjiga i time poboljšaju svoju učinkovitost, te smanje količinu administrativnog posla.

U posljednjih 10 godina, web stranice i web aplikacije su postigle veću dinamiku i mogućnosti zahvaljujući programskom jeziku JavaScript. [1] JavaScript je jedan od najpopularnijih programskih jezika na svijetu i koristi se za prikazivanje sadržaja u web pregledniku, kao

jedna od ključnih tehnologija World Wide Web-a uz HTML i CSS. Povećanje korištenja JavaScripta na stranicama i aplikacijama dovelo je do toga da se velika količina koda preselila iz servera u preglednike što je rezultiralo velikim brojem linija JavaScript koda i datoteka povezanih s HTML i CSS datotekama, bez dobre organizacije. Upravo zbog ovog razloga se sve više razvojnih inženjera odlučuje na korištenje nekog od JavaScript radnih okvira za izgradnju korisničkog sučelja i organizaciju datoteka i podataka. Neki od popularnih su React.js, Angular i Vue.js, svaki sa svojim prednostima i nedostacima te sličnostima i razlikama. Prije odabira radnog okvira, potrebno je napraviti analizu ponuđenih opcija kako bi se utvrdilo koja najbolje odgovara korisničkim potrebama.

U ovom radu obrađuju se progresivni JavaScript okvir Vue.js i Firebase, platforma za razvoj mobilnih i web aplikacija koju je razvio i koju održava Google. Prikazat će se povezivanje i usklađivanje ovih dviju tehnologija kako bi aplikacija funkcionirala, te će se opisati proces razvoja kao i koraci izrade aplikacije uključujući izradu komponenti i baze podataka, kao i autentifikacija i validacija korisnika i administratora. Razlog za korištenje navedenih tehnologija je taj što njihova kombinacija pruža jednostavan i brz razvoj aplikacije, usluge poslužitelja (eng. *hosting*) i autentifikacijski sustav, ali i veliku zajednicu koja pruža razne module i pakete.

2. USPOREDBA RADNIH OKVIRA ZA RAZVOJ PREDNJEG DIJELA WEB APLIKACIJE

2. COMPARISON OF FRONT-END FRAMEWORKS FOR WEB APPLICATION DEVELOPMENT

2.1. REACT.JS

2.1. REACT.JS

React.js je biblioteka za JavaScript koju su razvili inženjeri Facebooka u svrhu izgradnje i održavanja aplikacija, a odlikuju ju dva bitna svojstva – skalabilnost i održivost. React.js omogućuje programerima izgradnju web

aplikacija koje se mogu efikasno ažurirati i prikazivati.

Iako ima mnoge značajke radnog okvira, mnogi su protiv mišljenja da je React.js kompletan radni okvir. Službena stranica ga navodi kao JavaScript biblioteku za izradu korisničkih sučelja.[2]

Iako postoje nedostaci, React.js je jedan od najsigurnijih radnih okvira za programere, pogotovo ako žele napraviti web aplikaciju sa dugoročnom podrškom, velikom brzinom i kvalitetnim funkcijama.[3] Jedan od razloga zbog kojeg će ga početnici teže naučiti je prelazak sa pisanja HTML i CSS koda na JSX, iako je ova mogućnost opcionalna.

2.2. ANGULAR.JS

2.2. ANGULAR.JS

AngularJS je JavaScript radni okvir koji se koristi za izradu web aplikacija kojeg je razvio Google, a omogućava izradu dinamičkih, interaktivnih i kompleksnih aplikacija. Dozvoljava korištenje standardnog HTML-a i CSS-a, ali isto tako proširuje mogućnosti HTML-a. Često se koristi za izradu kompleksnih web aplikacija s mnogo funkcionalnosti. Angular komponente su jako korisne i relativno ih je jednostavno koristiti što je jedan od razloga zašto je danas jedan od najpoznatijih JavaScript radnih okvira.

Data binding i *dependency injection* eliminiraju potrebu za pisanjem puno koda, te omogućavaju brzo i kvalitetno razvijanje aplikacija. Da bi neka aplikacija bila potpuna, ona mora imati i svoj pozadinski dio (eng. *back-end*) i bazu. AngularJS to ne pokriva pa se u kombinaciji s njime mora koristiti još neka tehnologija, primjerice Java, Ruby, C# i slične.

2.3. VUE.JS

2.3. VUE.JS

Vue.js je progresivni JavaScript radni okvir za izradu web aplikacija koji pruža alate za izradu interaktivnih komponenti koje se mogu lako integrirati u postojeće projekte. Fokusira se na pružanje jednostavnog API-ja za rad s podacima i korisničkim sučeljem, što ga čini popularnim

izborom za izradu SPA (*single-page* aplikacija) i PWA (progresivnih web aplikacija). Temelji se na konceptu komponenti, što omogućuje modularnost i ponovno korištenje koda.

Kreirao ga je Evan You koji je radio na AngularJS aplikacijama s ciljem stvaranja uspješnijih i efikasnijih aplikacija. Ono što ga čini posebnim je njegova kombinacija elemenata Angular-a i React.js-a.

Vue.js koristi HTML kao temeljni jezik za opis komponenti i njihovog ponašanja. Ovo se odražava u sintaksi koja kombinira HTML s JavaScriptom i posebnim Vue naredbama. Komponente se obično sastoje od tri dijela: predložak (eng. *template*), skripta (eng. *script*) i (eng. *style*).

Neki od nedostataka ovog radnog okvira su ti što je relativno mlad i što se brzo razvija, zbog čega velika količina dostupnih materijala na internetu vrlo brzo postane zastarjela što razvojnim inženjerima otežava proces učenja.

3. USPOREDBA FIREBASEA I DRUGIH BAZA PODATAKA

3. COMPARISON OF FIREBASE AND OTHER DATABASES

3.1. MYSQL

3.1. MYSQL

MySQL je sustav za upravljanje relacijskim bazama podataka, koji radi na svim platformama poput Linuxa, Windowsa, Unixa i ostalima. Njegova pouzdanost, stabilnost i jednostavnost ga čine najpopularnijim sustavom za upravljanje bazom podataka.

MySQL je dizajniran kako bi bio kompatibilan s različitim operacijskim sustavima i omogućio rad s velikim količinama podataka. Zbog svoje rasprostranjenosti, ima veliku zajednicu korisnika i dostupnost velikog broja dodatnih alata i rješenja.

Međutim, MySQL ima neke nedostatke, kao što je ograničenje u skalabilnosti u okruženjima s visokim opterećenjem. Ako će aplikacija koju će korisnici koristiti s vremenom sadržavati

veliku količinu podataka, preporuča se korištenje MySQL-a kao baze podataka. Postoji mogućnost za nastajanjem konflikata ako se podacima pokuša pristupiti iz više aplikacija istovremeno.[8]

3.2. MONGODB

3.2. MONGODB

MongoDB je NoSQL baza podataka otvorenog koda, razvijena za visoke performanse, visoku dostupnost i jednostavnu skalabilnost. Koristi dokument-orijentiran model podataka, što znači da podaci nisu pohranjeni u tablicama kao u relacijskim bazama podataka, već u obliku dokumenata, primjerice u JSON datotekama. To omogućuje fleksibilnost u strukturiranju podataka i brzu izmjenu strukture bez potrebe za promjenom tablica.

Osim fleksibilnosti, prednosti MongoDB-a uključuju i veliku skalabilnost (podržava razdvajanje podataka na više servera), visoku performansu (omogućuje brzu izmjenu podataka i pretraživanje), te podršku za različite programske jezike.

Ne podržava povezivanje podataka kao što podržavaju relacijske baze podataka. Iako se podaci mogu rasporediti na više servera, to može utjecati na performansu i usporiti izvršavanje.[9]

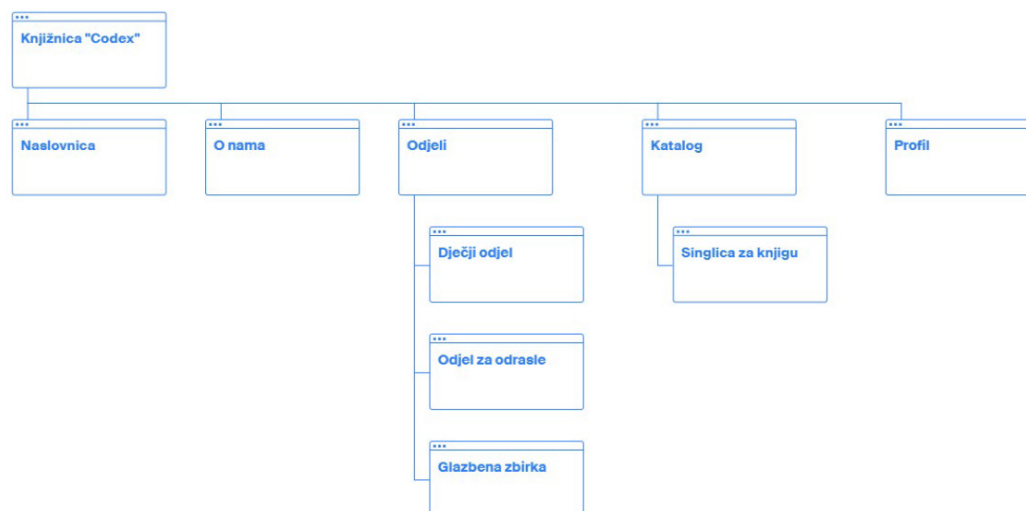
3.3. FIREBASE

3.3. FIREBASE

Firestore je platforma za razvoj mobilnih i web aplikacija koja pruža razne usluge za razvoj aplikacija poput baze podataka, autentifikacije, poslužitelja i *push* obavijesti. Firestore omogućava razvojnim inženjerima da se fokusiraju na izradu korisničkog sučelja i funkcionalnosti, umjesto na održavanje i skaliranje infrastrukture.

Ova platforma pruža sposobnost izrade aplikacija bez potrebe za upravljanjem poslužiteljima i pisanjem API-ja, jer sama platforma predstavlja kombinaciju poslužitelja i baze podataka. Sve se može prilagoditi potrebama aplikacije, što omogućuje jednostavniju i bržu izradu.

Jedan od glavnih razloga odabira Firestore kao baze podataka za ovaj rad, je taj što Vue.js ima



Slika 1
Hijerarhijska
struktura
aplikacije

Figure 1
Hierarchical
structure of the
application

dobro razvijenu podršku za Firebase, i obrnuto. Kombinacija Vue.js radnog okvira i Firebase platforme pruža jednostavno stvaranje dinamičnih web aplikacija s minimalnom količinom koda, zahvaljujući komponentama koje olakšavaju rad s oba okruženja. Iako se Vue.js može koristiti i s drugim bazama podataka, korištenje Firebasea pruža dodatnu efikasnost, jednostavnost i brzinu u izvršavanju upita. [10]

Firestore je izuzetno napredno BaaS (eng. *Backend as a Service*) poslužiteljsko rješenje koje se jednostavno instalira i pruža jednostavan pristup podacima, datotekama i informacijama. Razvija ga Google, što znači da ima izvrsnu podršku, visoku sigurnost i ne zahtijeva konfiguraciju poslužitelja.

Firestore baza podataka ima podatke pohranjene u JSON formatu, što može dovesti do poteškoća pri prelasku na drugo rješenje kada se jednom izgradi aplikacija s njom. Ograničene mogućnosti upita i indeksiranja također mogu utjecati na brzinu i skalabilnost baze podataka i aplikacije u cjelini. [11]

4. RAZVOJNI PROCES I TEHNIČKI ASPEKTI

4. DEVELOPMENT PROCESS AND TECHNICAL ASPECTS

U početku rada na projektu, potrebno je instalirati Node.js – *open-source* platformu napravljenu pomoću JavaScripta, koja omogućava korištenje jedinstvenog jezika između *front-end* i *back-*

end dijela aplikacije. Nakon instalacije Vue.js paketa i kreiranja projekta, prvi korak je kreiranje bitnih komponenti, dijelova aplikacije koji će se ponovno koristiti na različitim mjestima, kao što su navigacija i podnožje. [12]

Sljedeći korak je planiranje i kreiranje strukture aplikacije. Strukturu određuje način kako je sadržaj grupiran, povezan i predstavljen korisnicima. Dobro strukturirana aplikacija omogućuje lakše snalaženje korisnika na njoj, a Google će moći bolje kategorizirati URL-ove. Da bi se rute postavile unutar projekta, korišten je Vue Router – JS biblioteka za izgradnju korisničkog sučelja za Vue.js aplikaciju. Ona omogućava navigaciju za SPA definiranje ruta i njihovo mapiranje na određene komponente. Nakon toga, putanje do stranica se definiraju korištenjem `<router-link>` atributa u HTML-u. Potrebno ga je instalirati pomoću `npm`-a i uvesti u projekt unutar `main.js` datoteke na pomoću `import router from '@router'`. Prilikom definiranja novih ruta, osim putanje potrebno je definirati naziv rute i komponentu na koju se ta stranica odnosi. Zadani naziv će se prikazivati u naslovu kartice preglednika kada je stranica otvorena. Kako je to organizirano u aplikaciji, prikazano je na slici 1.

U idućem koraku planira se upravljanje stanjima kao što su registracija i prijava korisnika, dohvaćanje podataka o prijavljenom korisniku, preusmjeravanje korisnika ukoliko pokuša bez autentifikacije pristupiti stranici koja to očekuje, odjave i slično. Za navedeno se koristi Vuex. Vuex funkcionira tako da stvara skladište (eng. *store*),

gdje se čuvaju svi podaci aplikacije. Komponente u aplikaciji komuniciraju s ovim skladištem putem promjena (eng. *mutations*) i radnji (eng. *actions*). Promjene se koriste za izmjenu stanja u skladištu, dok se radnje koriste za izvođenje složenih radnji uz asinkrono dohvaćanje podataka. [13] Ukratko, Vuex omogućava centralizirano upravljanje stanjem aplikacije, što pomaže u osiguravanju da su podaci uvijek sinkronizirani i ažurni, te olakšava razvoj Vue.js aplikacija.

Nadalje se postavlja baza podataka Firebase. Pri tome je obavezno kreiranje računa na službenoj Firebase stranici nakon čega je moguće stvoriti projekt kroz konzolu. U sučelju preglednika koje nudi potrebne alate za upravljanje projektom, unose se podatci o platformi na kojoj će se aplikacija prikazivati, te se dobiva kod koji sadrži sve potrebne podatke za baze koje će aplikacija koristiti – Firestore (za pohranu podataka o knjigama, novostima i rezervacijama) i Firebase Storage (za pohranu fotografija knjiga). Korištenjem npm-a instalira se Firebase i inicijalizira se u aplikaciji.

Firestore je usluga pohrane podataka i upravljanja dokumentima u oblaku, a u aplikaciji je korištena za pohranu fotografija knjiga. Kako bi se povezali podaci iz Storagea s onima iz tablice „books“, aplikacija prvo pohranjuje medij u bazu, čime se dobiva poveznica na njega i sprema u varijablu. Nakon toga, navedena varijabla se sprema u tablicu kroz funkciju `addBook()`. Kako bi se medij pohranio nakon odabira, poziva se funkcija `uploadImage` u kojoj je definiran Firebase Storage i putanja unutar koje će se medij pohraniti. Kada se pohrani, poveznica se sprema u varijablu `this.coverImageUrl`.

5. FUNKCIONALNI OPIS APLIKACIJE

5. FUNCTIONAL DESCRIPTION OF THE APPLICATION

Od početne stranice očekuje se da privuče pažnju korisnika prilikom posjeta njoj, te da prikaže najbitnije informacije o stranici poput predloženih knjiga, novosti ili nadolazećih događaja. Sadržaj na naslovnici se prikazuje zahvaljujući *HomeView.vue* datoteci u kojoj su sadržane sve sekcije koje

se nalaze na naslovnici. Za prikaz predloženih knjiga se koristi `swiper.js` – dodatak temeljen na JavaScript programskom jeziku koji pomaže u stvaranju slidera za web stranice. U njemu su definirani nazivi četiri predložene knjige, njihove slike i autori, te gumb koji vodi na stranicu s više informacija o odabranoj knjizi.

Obrasci za prijavu i registraciju korisnika poštuju dobre prakse ovih obrazaca u standardnim web aplikacijama, te se prilikom registracije validiraju slučajevi poput: postoji li već unesena e-mail adresa već postoji u bazi, je li unesena neispravna e-mail adresa, je li unesena lozinka preslaba (ako je kraća od 8 znakova) i je li se dogodila greška s bazom podataka.

Iz sigurnosnih razloga, prije pohrane u podataka u bazu, lozinka se sažme pomoću `bcrypt` algoritma. `Bcrypt` koristi varijabilnu duljinu hash vrijednosti što znači da se za svaku lozinku generira jedinstvena sažeta vrijednost (eng. *hash*), čime se napadačima onemogućava korištenje prethodno izračunate vrijednosti za probijanje novih lozinki.

Za autentifikaciju je dovoljno u Firebaseovu naredbu `signInWithEmailAndPassword` proslijediti unesenu e-mail adresu i lozinku. Ako se pojavi greška, `catch()` metoda će proći kroz moguće slučajeve i odraditi `commit`, tj. izvršiti mutaciju.

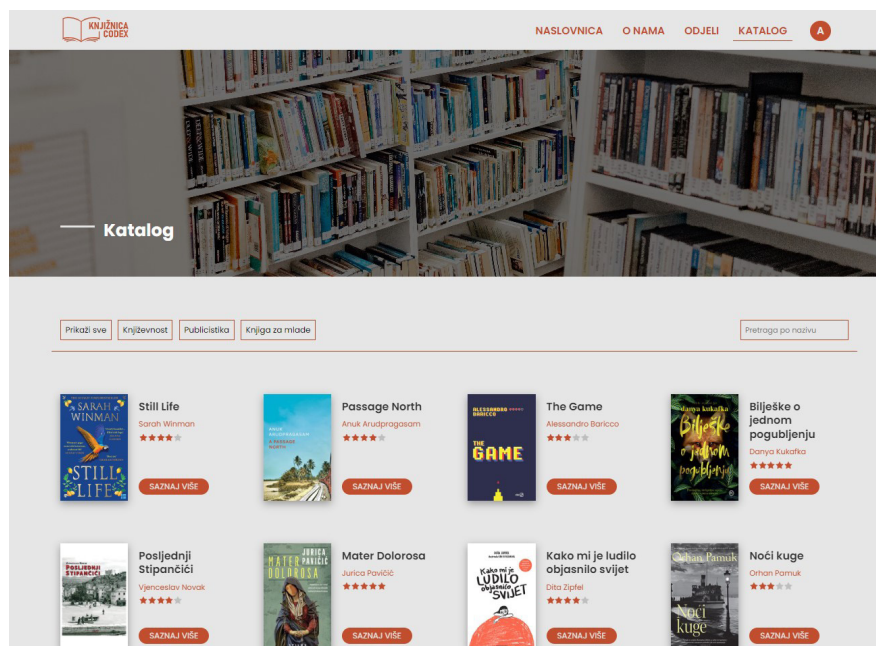
Jedini način da se stanje u centraliziranom skladištu Vuex-a promijeni je izvršavanjem mutacija (eng. *mutations*) pomoću određenih metoda. Vuex mutacije se odvijaju preko okidača, pa su tako izrađene mutacije koje se odnose na registraciju i prijavu korisnika.

Aplikacija razlikuje dvije vrste korisnika, a to su administratori odnosno knjižničari i korisnici odnosno članovi knjižnice. Za unos podataka u bazu knjiga razvijeno je sučelje putem kojeg administratori unose informacije o knjigama - naslove, sadržaje, imena autora, vrstu uveza i ostalo, što je prikazano na slici 2. Također je bitno stvoriti katalog koji će te iste knjige, i njihove informacije, ispisivati i omogućiti korisnicima da ih filtriraju po žanru ili pretraže prema nazivu knjige. Unesene knjige je moguće filtrirati prema njihovom žanru klikom na gumb s nazivom tog žanra, ali je također moguće pretraživati upisivanjem naslova željene knjige. Osim unosa

The screenshot shows the 'Dodavanje knjige' form in the Knižnica Codex administrator interface. The form is titled 'Dodavanje knjige' and includes several input fields: 'Naziv knjige' (Book title), 'Autor' (Author), 'Vrsta' (Genre), 'Meki uvez' (Soft cover), 'Izdavač' (Publisher), 'Jezik' (Language), 'Stanje' (Status), 'ISBN', 'Cijena' (Price), 'Sadržaj' (Content), 'URL', and a file upload button labeled 'Choose File | No file chosen'. A red 'Dodaj knjigu' (Add book) button is at the bottom left. The background features a book cover with text.

Slika 2 Prikaz obrasca za dodavanje nove knjige

Figure 2 Book insert form



Slika 3 Prikaz stranice "Katalog"

Figure 3 Page „Catalog“

knjiga, administratori imaju mogućnost unosa objava. Obrazac za dodavanje objave sadrži polje za unos naslova i polje za unos sadržaja i gumb za predaju forme. Prilikom pohrane objave u bazu, automatski se sprema i vrijeme kada je novost unesena.

Aplikacija je napravljena s ciljem da svi posjetitelji mogu razgledati katalog i knjige u njemu, ali da samo registrirani i prijavljeni korisnici mogu rezervirati knjige, te je stoga potrebno provesti provjeru za prijavljenim korisnikom. Prijavljenim korisnicima će se prilikom pregleda pojedine knjige pojaviti i gumb za mogućnost rezervacije iste. Prijavljeni korisnici

moгу pregledavati i brisati vlastite rezervacije. Na slici 3 je prikaz kataloga neprijavljenom korisniku.

Kako bi se korisnike moglo jednostavnije identificirati, stvorena je digitalna iskaznica pomoću koje se korisnik može identificirati da preuzme svoje rezervirane knjige. Digitalna iskaznica je elektronički oblik članske iskaznice koja koristi QR kod kao način identifikacije korisnika, a primjer je vidljiv na slici 4. Kreira se kada se korisnik registrira u aplikaciju. Svi članovi kojima je draži „papirnati“ format, mogu zatražiti svoju iskaznicu te će ona za njih biti izrađena.

Za generiranje QR kodova u aplikaciji, korištena je komponenta QRCode.Vue koja se instalira putem npm-a.



Slika 4 Digitalna iskaznica korisnika
Figure 4 User's digital identity card

6. ZAKLJUČAK

6. CONCLUSION

U radu je prikazan uspješan razvoj aplikacije sa definiranim funkcionalnostima uporabom odabranih tehnologija. Vue.js je vrlo brz i jednostavan radni okvir, posebno za razvojne inženjere koji tek ulaze u svijet JS-a, koji je dobro dokumentiran i nudi mnogo mogućnosti. Osim što je Firebase odabran kao baza zbog svoje jednostavnosti u usporedbi s ostalim bazama podataka, dodatni razlog je taj što Vue.js omogućuje vrlo dobru implementaciju s Firebaseom i tako olakšava taj proces.

Kvaliteta i brzina rada ove aplikacije uvelike utječu na korisnikovu percepciju prema njoj i važni su faktori za njezinu popularnost i upotrebljivost. Danas je programiranje preko sučelja najpopularniji način za razvoj web aplikacija te je preporučljivo koristiti specijalizirana sučelja. U ovom projektu su, osim Vue.js radnog okvira, korišteni i Vuex store management i Firebase baza podataka kao izvor za pohranu podataka koji ne ovisi o lokalnom serveru. Vuex je korišten za lokalnu pohranu podataka, a Firebase kao izvor za dohvaćanje podataka.

Jedan od najvećih nedostaka tijekom procesa razvoja aplikacije jest manjak dokumentacije za specifične probleme. Iako nije razvijena višepatformska aplikacija, web verzija je

planirana da bude prilagodljiva različitim rezolucijama i veličinama zaslona. U budućem radu moguće je razmotriti mogućnost razvoja aplikacije za Android i iOS uređaje koja bi korisnicima olakšala i ubrzala proces rezervacije.

7. REFERENCE

7. REFERENCES

- [1.] „How the Front-End Industry Has Evolved Over the Past 10 Years“, In Plain English URL: <https://javascript.plainenglish.io/2022-front-end-tool-chain-ten-year-inventory-427d891d0e3d> (18.02.2023.)
- [2.] „What is React and Why should you learn it in 2022-23?“, AboutReact URL: <https://aboutreact.com/reactjs/> (15.11.2022.)
- [3.] „React.js Security Guide: Threats, Vulnerabilities, and Ways to Fix Them in 2023“, Relevant URL: <https://relevant.software/blog/react-js-security-guide> (18.02.2023.)
- [4.] Ken Williamson: „Learning AngularJS“, O'Reilly Media, Inc. (2015.)
- [5.] Hassan Djirdeh, Nate Murray, Ari Lerner: „Fullstack Vue“, Fullstack.io (2018.)
- [6.] „Learning VueJS – A VueJS Tutorial“, Great Learning URL: <https://www.mygreatlearning.com/blog/vuejs-tutorial> (27.11.2022.)
- [7.] Peters, K: „Vue.js review of 2017“, Medium URL: https://medium.com/@kevin_peters/what-happened-to-vue-js-in2017-aeaaa69c2c6f (15.11.2022.)
- [8.] „Upravljanje bazama podataka: Što je MySQL?“, HR Education URL: <https://hr.education-wiki.com/5876083-what-is-mysql> (15.11.2022.)
- [9.] „Upravljanje bazama podataka: Što je MongoDB?“, HR Education URL: <https://hr.education-wiki.com/5902372-what-is-mongodb> (17.11.2022.)
- [10.] You, E: „VueFire – Firebase meets Vue.js“, The Firebase Blog URL: https://firebase.blog/posts/2016/04/vuefire-firebase-meets-vuejs_0 (21.11.2022.)

- [11.] Umali, A: „Firebase vs MongoDB vs MySQL“
URL: <https://prezi.com/l8ixc0mtifhs/firebase-vs-mongodb-vs-mysql/>
(21.11.2022.)
- [12.] Kramberger, T.; Duk S.; Kovačević R. "Baze podataka", Manualia Polytechnici Studiorum Zagrebisensis, Tehničko veleučilište u Zagrebu, 2018. ISBN: 978-953-7048-70-9
- [13.] Tošić, K: „Što je Node.js, kako se koristi, koji su benefiti?“, PopArt Studio
URL: https://www.popwebdesign.net/popart_blog/2015/06/sta-je-node-js/
(24.11.2022.)
- [14.] Asaolu, E.: „Guide to Vuex – Centralized State Management for Vue Applications“, StackAbuse
URL: <https://stackabuse.com/guide-to-vuex-centralized-state-management-for-vue-applications> (10.12.2022.)

AUTORI · *AUTHORS*

• **Mario Vidović** - Student je preddiplomskog stručnog studija informatike na Tehničkom veleučilištu u Zagrebu. Stekao je iskustvo u razvoju web aplikacija kroz dosadašnje obrazovanje i druge projekte koji se odnose na razvoj cjelovitih aplikacija, uključujući razvoj prednjih i pozadinskih dijelova aplikacije. JavaScript bazirane aplikacije kao i web servisi čine njegove primarne interese.

Korespondencija · *Correspondence*

mario.vidovic@tvz.hr

• **Sanja Kraljević** - nepromijenjena biografija se nalazi u časopisu Polytechnic & Design Vol. 7., No. 4, 2019.

Korespondencija · *Correspondence*

sanja@tvz.hr