Web aplikacija za upravljanje projektima

Varga, Dario

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:200:050279

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-14**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Stručni studij

WEB APLIKACIJA ZA UPRAVLJANJE PROJEKTIMA

Završni rad

Dario Varga

Osijek, 2023.

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

Obrazac Z1S: Obrazac za imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Osijek, 12.05.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime Pristupnika:	Dario Varga
Studij, smjer:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. Pristupnika, godina upisa:	A 4063, 12.07.2011.
OIB Pristupnika:	52628001010
Mentor:	Marina Peko, dipl. ing.
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Robert Šojo, mag. ing. comp.
Član Povjerenstva 1:	Marina Peko, dipl. ing.
Član Povjerenstva 2:	mr. sc. Željko Štanfel
Naslov završnog rada:	Web aplikacija za upravljanje projektima
Znanstvena grana završnog rada:	**Procesno računarstvo (zn. polje računarstvo)**
Zadatak završnog rada	Kreirati web aplikaciju za upravljanje projektima u timu. Aplikacija sadrži mogućnost registracije, prijave i odjave korisnika, dodavanje profilne slike, kreiranje, editiranje i brisanje projekta, komentiranje projekta u vidu chat-a, dodavanje dnevnika projekta i praćenje progresa.
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	12.05.2023.

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, 13.07.2023.

Ime i prezime studenta:	Dario Varga
Studij:	Preddiplomski stručni studij Elektrotehnika, smjer Informatika
Mat. br. studenta, godina upisa:	A 4063, 12.07.2011.
Turnitin podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za upravljanje projektima**

izrađen pod vodstvom mentora Marina Peko, dipl. ing.

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:

SADRŽAJ

1. UVOD 1

1.1. Zadatak završnog rada 1

2. PREGLED PODRUČJA 2

2.1. Slack ..... **Error! Bookmark not defined.**

2.2. Asana ..... **Error! Bookmark not defined.**

2.3. Basecamp 4

3. KORIŠTENE TEHNOLOGIJE I ALATI 5

3.1. HTML ..... **Error! Bookmark not defined.**

3.2. CSS ..... **Error! Bookmark not defined.**

3.3. JavaScript ..... **Error! Bookmark not defined.**

3.4. React 7

3.5. Firebase ..... **Error! Bookmark not defined.**

3.6. Visual Studio Code ..... **Error! Bookmark not defined.**

4. FUNKCIONALNOST APLIKACIJE 10

4.1. Kreiranje novog projekta 10

4.2. Postavljanje Firebasea 11

4.3. Autentifikacija 12

4.3.1. Navigacija stranicama koristeći React Router Dom biblioteku 13

4.3.2. Registracija i prijava korisnika 14

4.3.3. Odjava korisnika 15

4.4. Značajke aplikacije 15

4.4.1. Kreiranje projekta 15

4.4.2. Komentiranje projekta 17

4.4.3. Izmjena projekta 18

4.4.4. Brisanje projekta 19

5. KORIŠTENJE WEB APLIKACIJE 20

5.1. Početna stranica prije prijave korisnika 20

5.2. Registracija 21

5.3. Početna stranica nakon što se korisnik prijavi/registrira 21
5.4. Kreiranje projekta 22
5.5. Prikaz projekta i komentiranje projekta 23
5.6. Izmjena projekta 24
5.7. Bilješke projekta 25
5.8. Odjava 26
6. ZAKLJUČAK 27
LITERATURA 28
SAŽETAK 29
ABSTRACT 30

1. UVOD

Biti projekt menadžer ili biti u projekt menadžment timu ne mora nužno biti kompliciranа stvar, svakako ne ako za to postoje aplikacije koje pojednostavljuju tu ulogu. Kako postoje aplikacije za upravljanje projektima, možemo lagano i jednostavno brinuti o svom timu, zadacima i rokovima bez da stalno budemo u uredu. Štoviše, ne mora se potrošiti puno novaca da bi uopće koristili većinu njih. Ovakve aplikacije sadrže alate i tehnike koji su potrebni da bi naše projekte uspješno izvršili.

U drugom poglavlju opisan je pregled područja sličnih rješenja/aplikacija, treće poglavlje sadrži opise korištenih tehnologija, četvrto poglavlje je funkcionalnost web aplikacije i peto poglavlje je korištenje web aplikacije.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je kreirati web aplikaciju za upravljanje projektima u timu. Aplikacija sadrži mogućnost registracije, prijave, odjave, dodavanje profilne slike, kreiranje projekta, komentiranje projekta, izmjenu projekta, dodavanje dnevnika projekta, mogućnosti vidjeti tko je online i slično. Ukratko su opisane tehnologije koje su korištene. Uz pomoć slika detaljno je objašnjen način korištenja aplikacije od registracije pa sve do same odjave.

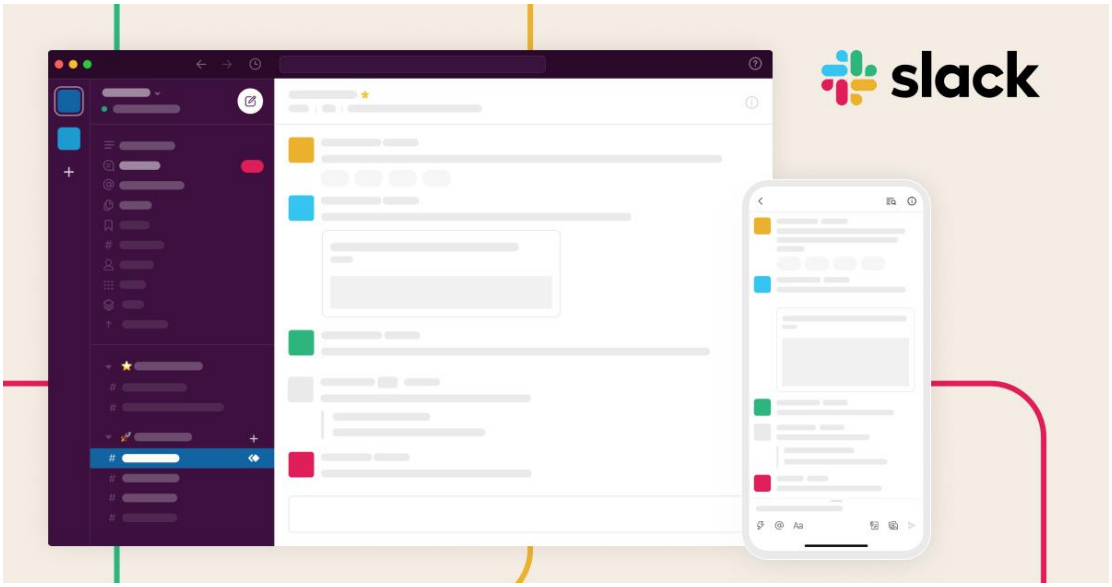
2. PREGLED PODRUČJA

U ovom poglavlju predstavljene su trenutno najpopularnije aplikacije koje služe za upravljanje projektima odnosno međusobno komuniciranje između članova tima ili na projektu, a to su:

- Slack
- Asana
- Basecamp

2.1. Slack

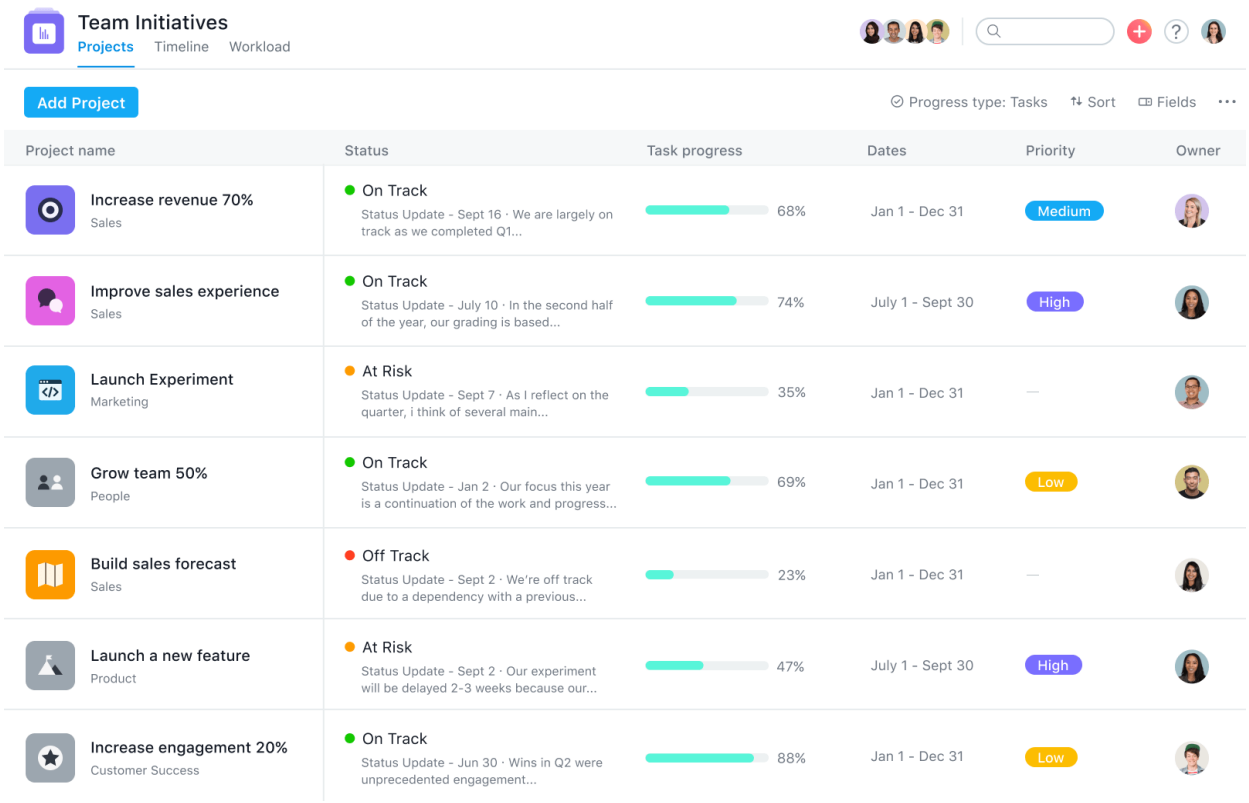
Slack je aplikacija za razmjenu poruka koju koristi skoro svaka firma. To je aplikacija putem koje se šalju poruke i ostali sadržaji. Slack u realnom vremenu šalje poruke i sadržaje. Postoje dvije vrste razgovora (engl. *chats*) na Slack-u a to su direktne poruke (engl. *direct message*), koristi se za razgovor 1 na 1 te kanali (engl. *channels*) odnosno grupni razgovori. Slack svakom timu pomaže da komunicira puno efikasnije tako zamjenjujući email, tekstne poruke i drugo. Postoji mobilna i desktop verzija aplikacije te tako omogućuje da međusobno komuniciramo i koordiniramo naše zadatke bez obzira gdje se nalazimo. U početku, Slack je zapravo bio interni komunikacijski alat za firmu Tiny Speck koju je vodio Stewart Butterfield dok su radili na online video igri *Glitch*. Tek u kolovozu 2013. godine, Slack je postao dostupan za sve.



Slika 2.1. *Slack u desktop i mobilnoj verziji.*

2.2. ASANA

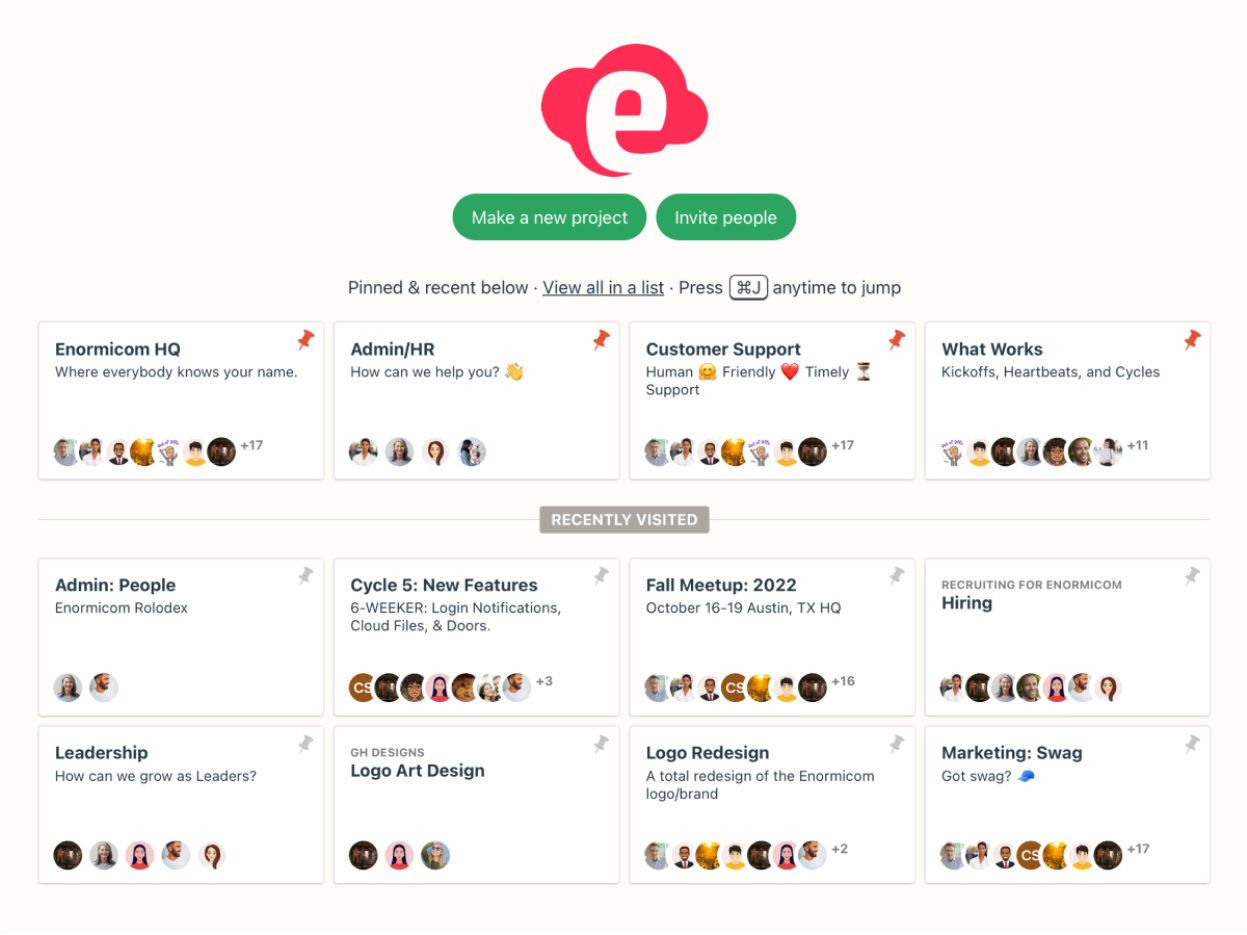
ASANA je trenutno najpopularnija aplikacija za upravljanje projektima. ASANA je prilagodljiv sustav za upravljanje sadržajem (engl. *content management system*) na radnom mjestu koji je dizajniran tako da pomaže širokom spektru kompanija kako bi organizirale svoje potrebe. Pomoću ASANA-e daju se zadaci specificirani za pojedince tako da bi svi znali tko radi na čemu. Sadrži mogućnost zadavanja početka projekta i krajnji rok projektu. Također radnici u timu mogu komunicirati između sebe na projektu. Koristi se za web i mobilno upravljanje radom (engl. *work management*). Inicijalno, ASANA Inc. je osnovana 2008. godine od strane suosnivača Facebook-a Dustin Moskovitza i inženjera Justin Rosenstein-a. ASANA je rađena pomoću TypeScript-a/JavaScript-a te nekoliko tisuća React komponenti. Na slici 2.2. prikazana je početna stranica web aplikacije ASANA-e.



Slika 2.2. *Početna stranica ASANA-e.*

2.3. Basecamp

Basecamp, kao i ASANA, je trenutno jedna od najpopularnijih aplikacija za upravljanje projektima. Ova aplikacija trenutno ima preko tri milijuna registriranih korisnika. Basecamp je alat za suradnju koji pomaže timovima da ostanu u kontaktu, dopušta korisnicima da kreiraju projekte, prate napredak projekta te upravljaju zadacima. Basecamp je osnovan 1999. godine ali se isprva zvao 37signals te se tek 2014. godine preimenovao u Basecamp i u to vrijeme je bila najpopularnija aplikacija.



Slika 2.3. *Početna stranica Basecamp-a.*

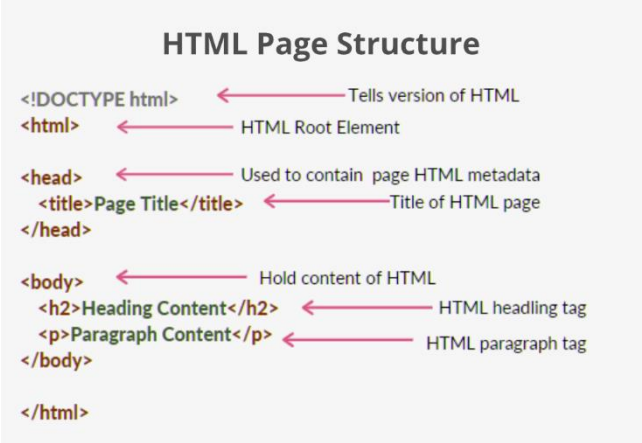
3. KORIŠTENE TEHNOLOGIJE I ALATI

Za izradu moderne web aplikacije potrebne su brojne tehnologije i alati. Primarni alati upotrijebljeni u izradi ovoga rada su:

- HTML
- CSS
- JavaScript
- React.js
- Firebase
- Visual Studio Code.

3.1. HTML

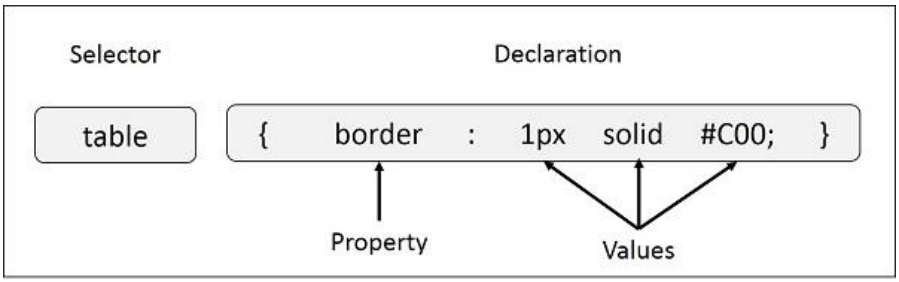
HTML (engl. *Hyper Text Markup Language*) je jezik koji se koristi za dizajniranje web stranica koristeći *markup* jezik te prikazujući ih na *World Wide Webu* (www) [1]. HTML je zapravo kombinacija *Hypertext-a* i *Markup* jezika. *Hypertext* definira poveznicu između web stranica, a *Markup* jezik definira tekstni dokument unutar oznaka (engl. *tags*) koje definiraju strukturu web stranice. HTML sadrži oznake i atribute koji se koriste za dizajniranje web stranica. Također postoji mogućnost povezivanja više stranica koristeći hiper linkove (engl. *hyperlinks*). Osnovna struktura HTML stranice sadrži sve bitne građevne elemente (npr. deklaraciju tipa dokumenta, HTML, *head*, *title* i *body* elemente) na kojima su sve web stranice napravljene. Prvu verziju HTML-a napisao je Tim Berners-Lee 1993. godine. Od tada je postojalo više verzija HTML-a. Najuspješnija verzija je bila 4.01 do 2012. godine kada je izašla 5.0 verzija te postala standardna praksa u izrađivanju web stranica. Za izradu ove aplikacije također je korištena HTML 5.0 verzija.



Slika 3.1. *Prikaz HTML strukture na web stranici.*

3.2. CSS

CSS (engl. *Cascading Style Sheets*) je jednostavan jezik koji je namijenjen za dizajniranje izgleda web stranica [2]. Koristeći CSS, kontrolira se izgled, font ili boja teksta, razmak između paragrafa, pozadinske slike ili boje, izgled dizajna, varijacije u izgledu na različitim uređajima i veličinama ekrana te ostalih efekata. CSS je jednostavan za naučiti i razumjeti a omogućuje veliku kontrolu nad izgledom HTML ili XML (engl. *Extensible Markup Language)* dokumenta. Neke prednosti CSS-a su što se može napisati za jedan dokument te isti iskoristiti za više različitih dokumenata. Možemo definirati stil za svaki pojedini HTML element te ga primijeniti na koliko god web stranica. Ostale neke prednosti bi bile, lako održavanje, kompatibilnost za više uređaja, globalni web standardi. CSS dodajemo u dokument na tri načina: *inline styling, internal style sheet* i *external style sheet*. Hakon Wium Lie se smatra začetnikom CSS-a kojeg je kreirao 1994. godine. Do sada su izašle samo tri verzije, a za ovu aplikaciju se koristio CSS3 koji je izašao 1999. i koriste ga većina organizacija diljem svijeta.



Slika 3.2. *Primjer CSS sintakse.*

3.3. Javascript

Javascript (JS) je dinamički programski jezik [3]. „Lagan“ je i najčešće se koristi kao dio web stranica, čije implementacije omogućuju skripti na strani klijenta međudjelovanje sa korisnikom i stvaranje dinamičkih stranica. Interpretira se kao objektno-orijentirani programski jezik. Povijesno gledajući web stranice su bile statične slično kao stranica u knjizi no sa dolaskom Javascripta i tehnologije na strani preglednika (engl. *browser-side technology*) omogućena je izrada dinamičkih web stranica. Koristeći Javascript, preglednici mogu reagirati na korisnikovu interakciju i promijeniti raspored sadržaja na stranici. Danas, većina web stranica bi bile statične bez Javascripta, odnosno imale bi samo dinamičke izmjene kao što su animacije koje su moguće uz CSS. Javascript jezik kreirao je Brendan Eich 1995. godine u samo 10 dana. Prvotno se zvao Mocha no tek kasnije Javascript. Današnja verzija Javascripta odnosno Ecmascripta je verzija 6 koja je korištena i za ovu aplikaciju.

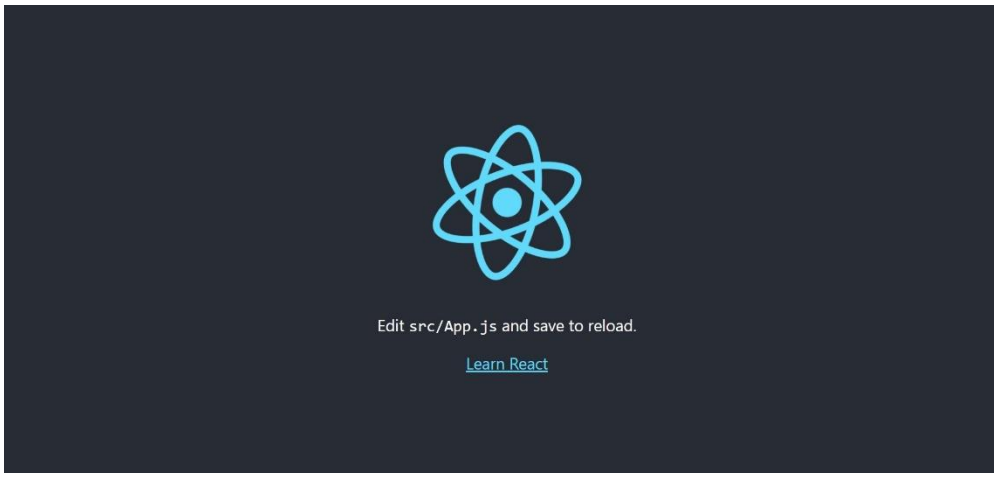
```
const num1 = 7;
const num2 = 9;

//multiply numbers
const product = num1 * num2;

//display product
console.log("The product of " + num1 + " and " + num2 + " is: " + product);
```

Slika 3.3. *Primjer kako pomnožiti dva broja u JS-u.*

3.4. React

React je Javascript biblioteka (engl. *library*) koja pomaže developerima u izradi korisničkih sučelja [4]. React je postao popularan zbog jednostavnosti i fleksibilnosti te ga koriste kompanije kao što su Meta, Airbnb, Netflix i druge. Primarni cilj React-a je graditi brza, skalabilna i pojednostavljena korisnička sučelja. No da bi se koristio React, treba biti dobro upoznat sa Javascriptom. Trenutno je React najpopularniji Javascript *library* i koristi ga više od 88% frontend developera [5]. Kao što je navedeno ranije, HTML je ono što daje strukturu web stranci, tako u React-u postoji Javascript XML (JSX) s kojim se opisuje struktura stranice. U React-u se zapravo

stranice dijele na manje komponente te se onda spajaju u jednu glavnu datoteku. Tim putem zapravo se pravi nešto što se zove aplikacija na jednoj stranici (engl. *Single Page Application*). Web stranice prije koje su se radile samo putem Javascripta, svaki puta kada bi korisnik kliknuo na neki link na stranici da ga odvede na drugi dio stranice, korisnik bi napravio zahtjev prema serveru te mu server vraća tu stranicu za kojom je napravio zahtjev. U tom slučaju bi dolazilo do toga da se stranica učitava i po nekoliko sekundi ovisno o brzini interneta i odgovoru servera. Međutim sa React-om koji je *SPA*, on nakon korisnikovog klika na link uspijeva presresti taj zahtjev i ne šalje ga na server nego učitava sadržaj stranice dinamički u pregledniku. Stoga su stranice brže i bolje za korisnike. Jordan Walke, inženjer u Meta-i, kreirao je React 2011. godine. No tek 2013. je otvoren kod za sve. Trenutna verzija Reacta je 18.2.0 koja je ujedno korištena za izradu ovog projekta.

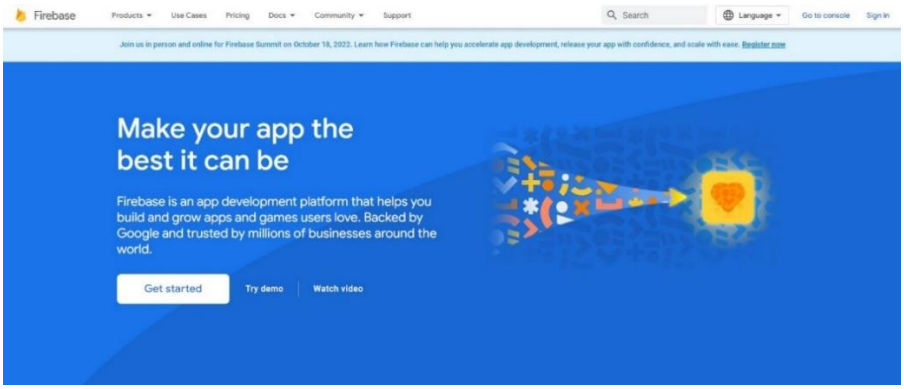


Slika 3.4. *Početna stranica u pregledniku nakon kreiranja React aplikacije*

3.5. Firebase

Firebase je platforma za razvoj aplikacija koja programerima omogućuje izradu web, Android i iOS aplikacija [6]. Firebase nudi mnoge značajke koje ga predstavljaju kao glavni pozadinski razvojni alat (engl. *backend development tool*) za web i mobilne aplikacije. Smanjuje nam radno opterećenje i vrijeme razvoja aplikacije. Savršen je alat za izradu prototipova. Firebase je jednostavan, lagan te prepoznat alat u industriji. Jedne od značajki što nam Firebase nudi su baza podataka u stvarnom vremenu, autentifikacija, pohrana podataka, usluge poslužitelja i drugo,

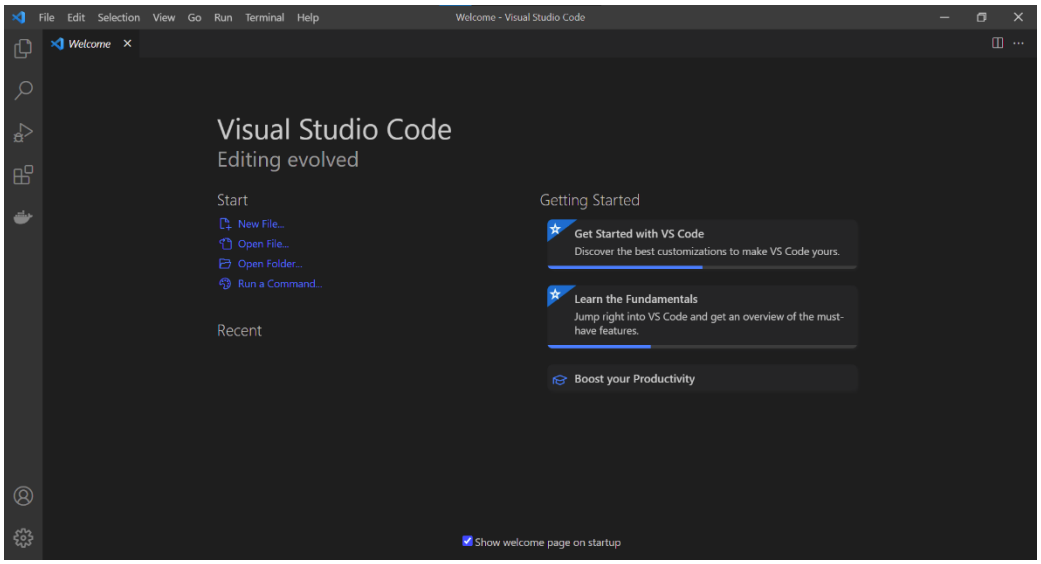
a svime se upravlja s jedne platforme. Napravljen 2011. kao *chat API* (engl. *Application Programming Interface*) te kupljen od strane Google-a 2014. godine.



Slika 3.5. *Početna stranica Firebase-a.*

3.6. Visual Studio Code

Visual Studio Code je besplatan, lagan, ali snažan uređivač za pisanje koda koji se može pokrenuti na desktopu ili na web-u i dostupan je za Windows, macOS, Linux i Raspberry Pi OS [7]. Dolazi sa ugrađenom podrškom za Javascript, Typescript i Node.js te ima bogati spektar ekstenzija za ostale programske jezike kao što su C++, C#, Java i drugi. Između ostalog VS Code podržava tzv. *IntelliSense* dovršavanje koda za varijable, metode i uvezene module. Microsoft ga je prvi puta najavio 2015. godine, a 2016. godine je pušten na web. Trenutno je najpopularniji među programerima vjerojatno zato što se redovno ažurira, prilagodljiv je te ima veliku podršku za proširenja od svojih ostalih konkurentata.



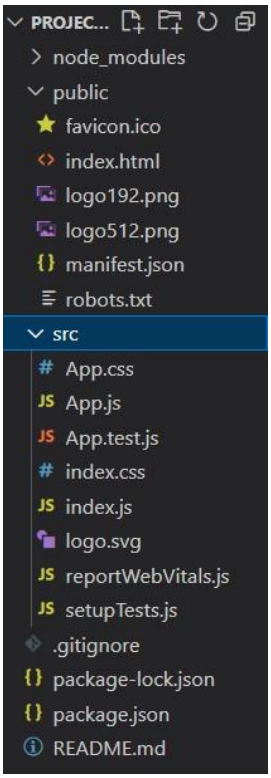
Slika 3.6. *Naslovnica dobrodošlice unutar VS Code-a.*

4. FUNKCIONALNOST APLIKACIJE

U ovome poglavlju bit će opisane funkcionalnosti korištene u web aplikaciji te postupak kreiranja novog React projekta.

4.1. Kreiranje novog React projekta

Projekt u React-u se u terminalu (engl. *command prompt*) kreira putem naredbe *npx create-react-app ime-projekta*. Naredba kreira tri mape i četiri datoteke. Mape čine *node_modules*, *public* i *src*, a datoteke *.gitignore*, *package.json*, *README.md* i *package-lock.json*. Sljedeća slika pokazuje kako to izgleda u VSC-u.



Slika 4.1. *Početni izgled React projekta unutar VSC-a.*

Npx create-react-app je najbolji način za jednostavno inicijaliziranje React projekta. Može se i specificirati verzija koja se želi instalirati tako da se na kraj naredbe napiše @ i broj verzije.

Create-react-app (CRA) služi samo za proces stvaranja aplikacija, a ne za cijeli razvojni ciklus kao na primjer kod Angular-a. CRA koristi građevni alat *webpack* koji služi za postavljanje predloška za aplikaciju i automatski će generirati datoteke i mape te tako uštedjeti vremena u samoj kreaciji projekta.

Nakon izvršavanja CRA naredbe i ulaska u mapu projekta može se vidjeti gore navedene mape i datoteke. *Package.json* je datoteka gdje se vidi *dependency* lista, popis skripti i lista podržanih preglednika.

Mapa *public* sadrži sva statična sredstva kao što su HTML, CSS, JavaScript i medijske datoteke. Datoteka *index.html* je jedina HTML datoteka koja je potrebna za stvaranje *Single Page Application* (SPA). Ova datoteka ima vrlo jednostavnu strukturu budući da unutar *body* elementa sadrži samo jedan *div* element sa ID-om *„root“*. Kada se aplikacija sastavlja (engl. *bundle*), react komponente će „tražiti“ taj *div* te će se na njega spojiti kako bi se aplikacija prikazala.

Mapa *src* je mjesto gdje se zapravo nalazi izvorni kod (engl. *source code*). To je također mapa čije će datoteke *webpack* procesuirati za produkciju. Unutar *src* mape nalazi se *index.js* datoteka koja formira korijensku datoteku svih JavaScript datoteka i prikazuje korijen svih React komponenti, koje bi trebale biti sastavljene unutar *App.js* datoteke.

Datoteka *App.css* sadrži CSS stilove koji se učitavaju unutar *App.js*. Unutar datoteke *index.css* se definiraju globalni CSS stilovi za cijelu aplikaciju.

Datoteka *.gitignore* sadrži listu datoteka koje ne želimo postaviti na naš *git* repozitorij, a to prema zadanim postavkama, uključuje *node_modules* mapu koja sadrži i sprema sve pakete koje naš projekt koristi. *README.md* datoteka sadrži dokumentaciju aplikacije, a *package-lock.json* sadrži *dependency* listu koji su instalirani te njihovu verziju.

4.2. Postavljanje Firebase-a

Ako se kod postavlja na Github, tada postoji opasnost od mogućeg neovlaštenog korištenja *Firebase* parametara. Parametre je potrebno zaštititi pomoću varijabilnih okolina. U tom slučaju se kreira *.env* datoteka u korijenskom direktoriju. Poslije toga u *.gitignore* datoteku dodajemo *.env* kako se ne bi postavila online na Github. Sljedeći korak je dodavanje potrebnih podataka u *.env* datoteku kako bi spajanje na Firebase bilo moguće. Ovdje je API ključ najbitnija stavka, dok su

ostali podaci prikazani na sljedećoj slici. Vrijednosti svih varijabilnih okolina (engl. *environment variables*) nisu prikazane radi sigurnosti.

Slika 4.2. *Datoteka u kojoj se nalaze varijable okoline*.

Potom slijedi inicijalizacija Firebase aplikacije, odnosno povezivanje *frontend* dijela aplikacije s poslužiteljem. Povezivanje aplikacije je vidljivo na slici 4.4.

Slika 4.3. *Inicijalizacija Firebase aplikacije*.

4.3. Autentifikacija

Autentifikacija je pojam koji se odnosi na proces dokazivanja da je neka činjenica ili neki dokument originalan. U informatici se ovaj pojam obično povezuje s dokazivanjem korisnika. Obično korisnik dokazuje svoj identitet davanjem svojih vjerodajnica, odnosno dogovorenih informacija koje dijele korisnik i sustav. Ova aplikacija daje mogućnost registracije (engl. *sign up*), prijave (engl. *login*) i odjave (engl. *logout*). Ne registrirani i ne prijavljeni korisnik ne može pristupiti određenim rutama, a ako je korisnik prijavljen onda ne može pristupiti *login* i *sign up* rutama jer se smatra da je korisnik prijavljen te je prošao cijeli taj proces registracije i prijave. Nakon što se korisnik registrira odnosno prijavi, on je preusmjeren na početnu stranicu. Ukoliko

neprijavljeni korisnik pokuša pristupiti rutama kojima samo prijavljeni korisnici mogu pristupiti, on biva preusmjeren na stranicu prijave.

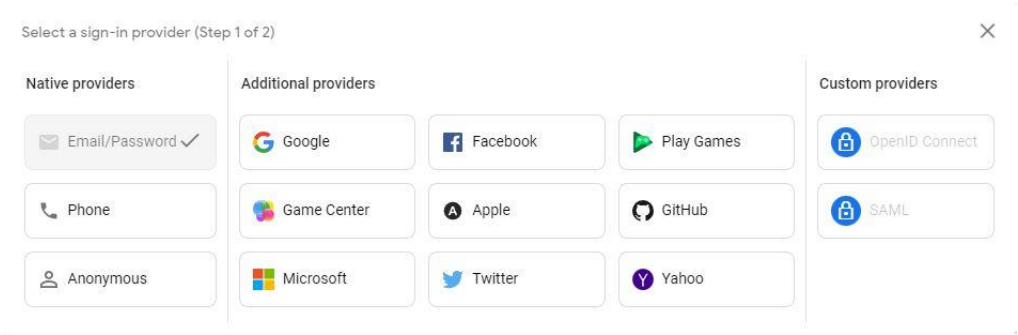
4.3.1. Navigacija stranicama koristeći React Router Dom biblioteku

Implementirana navigacija je kreirana korištenjem *React Router Dom* biblioteke. Biblioteka se instalira upisujući naredbu *npm install react-router-dom* u terminal. Unutar aplikacije postoji nekoliko putanja (engl. *URL*). Na sljedećoj slici je prikazan kod koji se koristi za preusmjeravanje korisnika u aplikaciji.

```
25              <Routes>
26                <Route
27                  path="/"
28                  element={user ? <Dashboard /> : <Navigate to="/login" />}
29                />
30                <Route
31                  path="/create"
32                  element={user ? <Create /> : <Navigate to="/login" />}
33                />
34                <Route
35                  path="/login"
36                  element={!user ? <Login /> : <Navigate to="/" />}
37                />
38                <Route
39                  path="/signup"
40                  element={!user ? <Signup /> : <Navigate to="/" />}
41                />
42                <Route
43                  path="/projects/:id"
44                  element={user ? <Project /> : <Navigate to="/login" />}
45                />
46              </Routes>
```

Slika 4.4. *Kod za preusmjeravanje korisnika.*

U Firebase-u postoji više načina pomoću kojih se korisnik prijavljuje sustav. Neki od njih su preko Google-a, Facebooka ili čak Githuba. U ovom slučaju to je preko e-maila i lozinke koju korisnik sam odabere prilikom registracije. Navedene mogućnosti prikazane su na slici 4.5.



Slika 4.5. *Odabir načina prijave putem Firebase-a.*

4.3.2. Registracija i prijava korisnika

Registracija korisnika je prvi korak u korištenju ove aplikacije. Sama registracija sastoji se od forme sa tri polja za e-mail, lozinku i korisničko ime te polje za odabir profilne slike. Ako su sva polja popunjena ispravno korisnik će uspješno biti registriran te preusmjeren na početnu stranicu, u suprotnom ostaje na registracijskoj formi te mora pogledati što mora ispraviti što je naznačeno greškama u formi. Svi korisnikovi podaci su spremljeni u *Firebase* bazu podataka. Prilikom prijave korisnik opet ispunjava formu koja se sastoji od polja za e-mail i polja za lozinku te se podaci uneseni u polja uspoređuju sa onima u bazi podataka. Ukoliko uneseni podaci odgovaraju onima iz baze podataka, korisnik je uspješno prijavljen te preusmjeren na početnu stranicu.

```
11  const login = async (email, password) => {
12    setError(null);
13    setIsPending(true);
14
15    //sign the user in
16    try {
17      const res = await projectAuth.signInWithEmailAndPassword(email, password);
18
19      // update online status
20      await projectFirestore
21        .collection("users")
22        .doc(res.user.uid)
23        .update({ online: true });
```

Slika 4.6. *Dio koda odgovoran za prijavu korisnika.*

4.3.3. Odjava korisnika

Tipka za odjavu korisnika odjavljuje korisnika iz aplikacije i preusmjerava ga na stranicu za prijavu. Za akciju odjave korisnika korištena je *Firebase* metoda *signout()*. Na sljedećoj slici prikazan je dio koda za odjavu korisnika.

```
11  const logout = async () => {
12    setError(null);
13    setIsPending(true);
14
15    //sign the user out
16    try {
17      const { uid } = user;
18      await projectFirestore
19        .collection("users")
20        .doc(uid)
21        .update({ online: false });
22
23      await projectAuth.signOut();
```

Slika 4.7. *Dio koda odgovoran za odjavu korisnika.*

4.4. Značajke aplikacije

Funkcionalnosti kao što su dodavanje profilne slike prilikom registracije, kreiranje projekta, komentiranje projekta, izmjena projekta, brisanje vlastitih projekata su sve značajke koje su zapravo srž ove aplikacije. Sve funkcionalnosti će biti pojašnjene u narednim poglavljima.

4.4.1. Kreiranje projekta

Samo prijavljeni korisnik ima pristup za kreiranje projekta. Ukoliko korisnik pokuša otići na *URL* za kreiranje projekta a da nije prijavljen, aplikacija ga preusmjerava na *login* stranicu. Pritiskom tipke za kreiranje projekta pokreće se funkcija *handleSubmit* koja provjerava i radi nekoliko stvari. Provjerava jesu li sva polja ispravno upisana te nakon provjere stvara projekt

objekt sa određenim svojstvima. Ako je sve u redu tj. nema greške sa *response* objektom, kreira se projekt te se preusmjerava korisnika na početnu stranicu.

```
45  const handleSubmit = async (e) => {
46    e.preventDefault();
47    setFormError(null);
48
49    if (!category) {
50      setFormError("Please select a project category.");
51      return;
52    }
53
54    if (assignedUsers.length < 1) {
55      setFormError("Please select at least 1 user for the project.");
56      return;
57    }
58
59    const createdBy = {
60      displayName: user.displayName,
61      photoURL: user.photoURL,
62      id: user.uid,
63    };
```

Slika 4.8. *Dio koda odgovoran za kreiranje projekta*

```
65    const assignedUsersList = assignedUsers.map((u) => {
66      return {
67        displayName: u.value.displayName,
68        photoURL: u.value.photoURL,
69        id: u.value.id,
70      };
71    });
72
73    const project = {
74      name: name,
75      details: details,
76      category: category.value,
77      dueDate: timestamp.fromDate(new Date(dueDate)),
78      comments: [],
79      createdBy: createdBy,
80      assignedUsersList: assignedUsersList,
81      progressBar: 0,
82    };
83
84    await addDocument(project);
85    if (!response.error) {
86      navigate("/");
87    }
88  };
```

Slika 4.9. *Dio koda odgovoran za kreiranje projekta.*

4.4.2. Komentiranje projekta

Na slici ispod prikazana je metoda za dodavanje komentara. Kada korisnik napiše komentar u područje teksta i pritisne dugme za dodavanje komentara pokreće se prikazana metoda. Metoda je asinkrona što znači da se izvodi asinkrono sve dok ne naiđe na *await* te tada stane i čeka dok se *await* zadatak ne izvrši. Metoda izvršava dvije stvari. Prvo stvara *commentToAdd* objekt sa određenim svojstvima, točnije stvara komentar objekt koji će zapravo biti korisnikov komentar, a druga stvar ažurira dokument u bazi gdje na niz komentara dodaje novi komentar. Ako je sve u redu, područje teksta se resetira te komentar postane vidljiv.

```
15  const handleSubmit = async (e) => {
16    e.preventDefault();
17
18    const commentToAdd = {
19      displayName: user.displayName,
20      photoURL: user.photoURL,
21      content: newComment,
22      createdAt: timestamp.fromDate(new Date()),
23      id: Math.random(),
24    };
25
26    await updateDocument(project.id, {
27      comments: [...project.comments, commentToAdd],
28    });
29    if (!response.error) {
30      setNewComment("");
31    }
32  };
33
```

Slika 4.10. *Implementacija koda za dodavanje komentara.*

4.4.3 Izmjena projekta

Na slici ispod prikazana je metoda za izmjenu projekta. U slučaju da se mijenja neki dio projekta, korisnik ima pristup izmjeni projekta. Ovdje se, kao i pri kreiranju projekta, provjerava jesu li sva polja ispravno popunjena. Ako nisu, prikazat će se greška u formi koja će uputiti korisnika kako da ju ispravi. Pritiskom na dugme *Edit project* otvara se novi prozor sa formom za izmjenu projekta gdje se mogu promijeniti svi podaci projekta. Kada je korisnik napravio izmjene koje želi, pritiskom na tipku *Save project* spremaju se sve izmjene i zatvara se prozor za izmjene.

```
40  const handleSubmit = async (e) => {
41    e.preventDefault();
42    setFormError(null);
43
44    if (!category) {
45      setFormError("Please select a project category.");
46      return;
47    }
48
49    if (assignedUsers.length < 1) {
50      setFormError("Please select at least 1 user for the project.");
51      return;
52    }
53
54    const assignedUsersList = assignedUsers.map((u) => {
55      return {
56        displayName: u.value.displayName,
57        photoURL: u.value.photoURL,
58        id: u.value.id,
59      };
60    });
```

Slika 4.11. *Implementacija koda za izmjenu projekta.*

```
62      await updateDocument(project.id, {
63        name: name,
64        details: details,
65        dueDate: timestamp.fromDate(new Date(dueDate)),
66        category: category.value,
67        assignedUsersList: assignedUsersList,
68      });
69
70      if (!response.error) {
71        closeEdit();
72      }
73    };
```

Slika 4.12. *Implementacija koda za izmjenu projekta.*

4.4.4 Brisanje projekta

Kada je projekt gotov, kreator projekta ima mogućnost obrisati projekt. Pritiskom na dugme *Delete project* se pokreće *handleDelete* metoda koja briše projekt iz baze te više ne postoji u aplikaciji. Nakon toga, korisnika se preusmjerava na početnu stranicu.

```
 8    const handleDelete = () => {
 9      deleteDocument(project.id);
10      navigate("/");
```

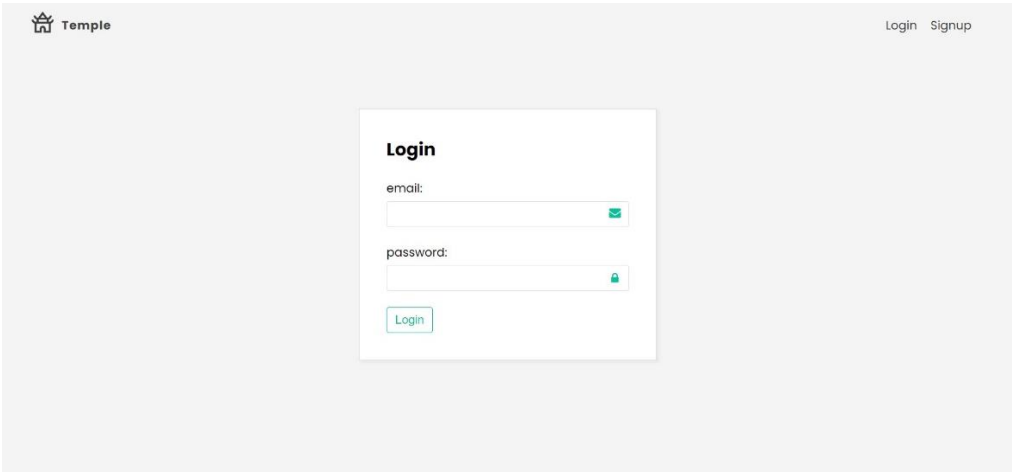
Slika 4.13. *Implementacija koda za brisanje projekta.*

5. KORIŠTENJE WEB APLIKACIJE

Ovo poglavlje će prikazati sve funkcionalnosti koje aplikacija pruža prilikom korištenja. Prikazat će se način korištenja aplikacije od same prijave korisnika, kreiranja projekta, komentiranje projekta te odjave korisnika.

5.1. Početna stranica prije prijave korisnika

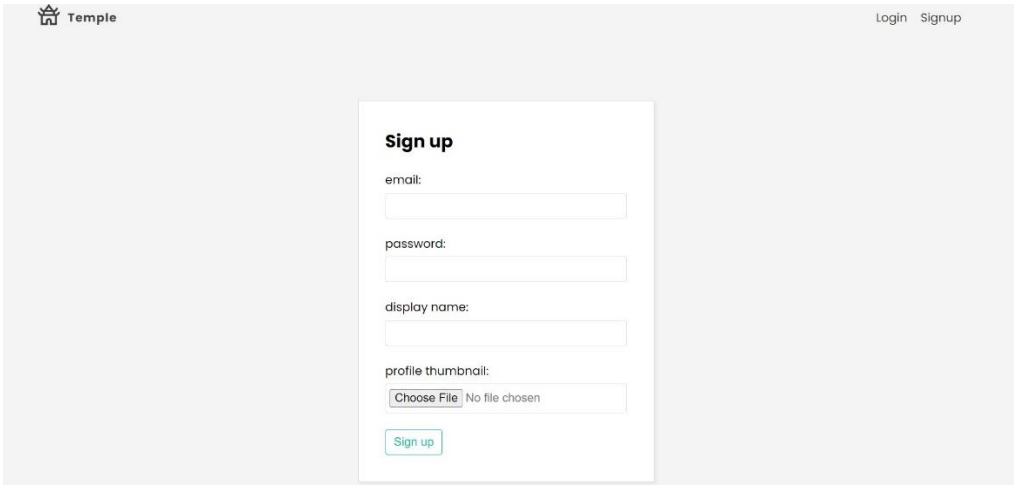
Kada korisnik prvi puta pristupi aplikaciji otvori mu se stranica za prijavu. Ako korisnik već ima napravljen račun tada treba ispuniti polja za prijavu i pritisnuti tipku *Login* te će biti preusmjeren na *Dashboard*. Ukoliko korisnik nema račun tada pritiskom na tipku *Signup* odlazi na stranicu za registraciju.



Slika 5.1. *Izgled početne stranice korisnika koji nije prijavljen*.

5.2. Registracija

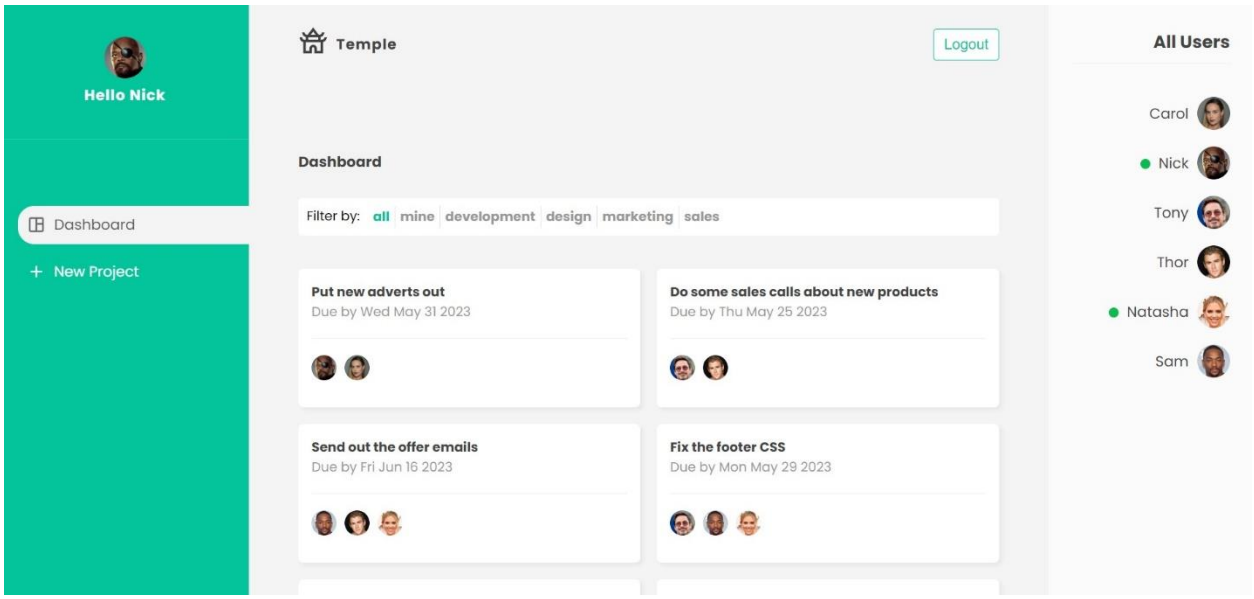
Na stranici za registraciju, korisniku je prikazana forma koju je potrebno ispuniti. Polja koja treba ispuniti su e-mail, lozinka, korisničko ime koje će se prikazati unutar aplikacije te odabir profilne slike. Ukoliko jedno od polja nije pravilno ili uopće nije popunjeno, u formi će se prikazati greška koja će uputiti korisnika kako da je ispravi. Ako je sve ispravno napravljeno, pritiskom na tipku *Sign up* kreira se novi korisnički račun te se korisnika preusmjerava na *Dashboard*.



Slika 5.2. *Prikaz stranice za registraciju.*

5.3. Početna stranica nakon što se korisnik prijavi/registrira

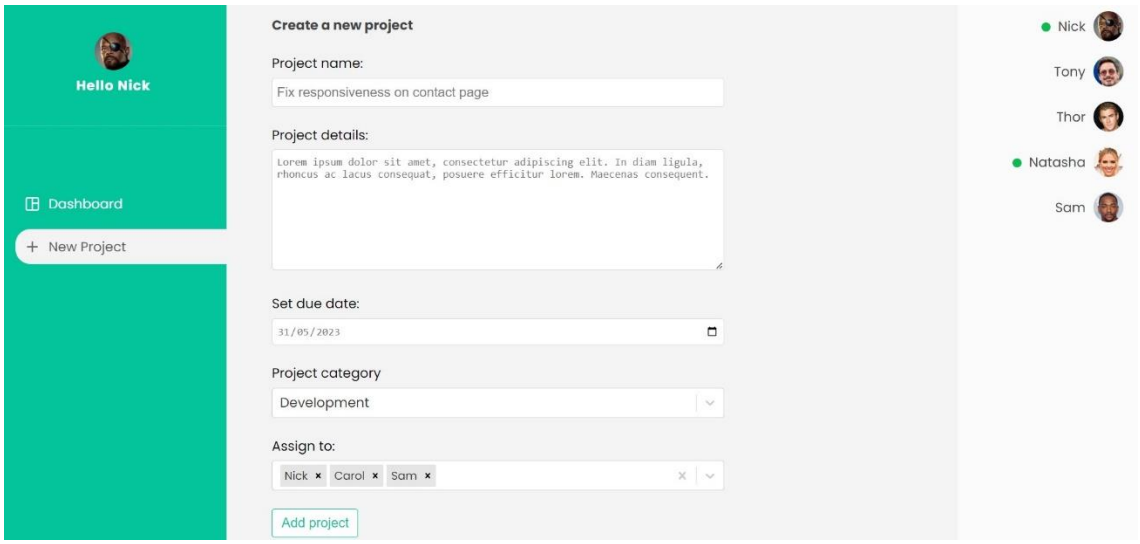
Nakon što se korisnik prijavi ili registrira, korisnik je preusmjeren na *Dashboard*. Sa lijeve strane korisnik može vidjeti svoju profilnu sliku, korisničko ime te tipke *Dashboard* i *New Project*. Svi projekt koji su napravljeni unutar tima mogu se filtrirati po kategorijama za lakše snalaženje korisnika. Sa desne strane su vidljivi svi registrirani korisnici te jesu li online.



Slika 5.3. *Početna stranica nakon prijave/registracije*.

5.4. Kreiranje projekta

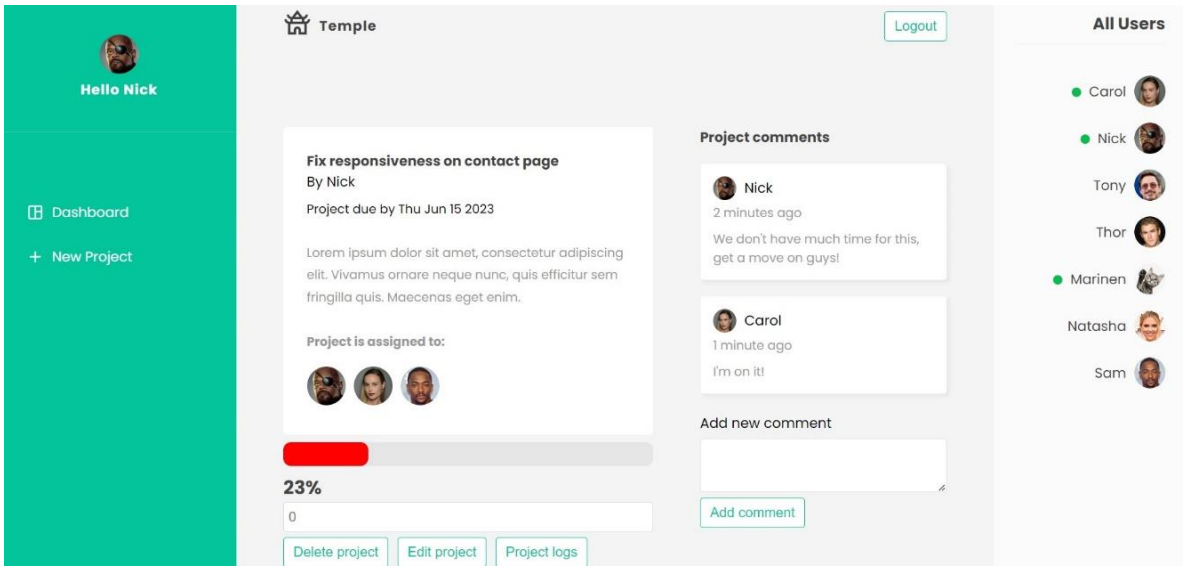
Nakon pritiska na dugme *New Project* otvara se forma za kreiranje projekta prikazana na slici 5.4. Korisnik ima nekoliko polja za ispuniti a to su ime projekta, opis projekta, krajnji datum izrade projekta, kategoriju projekta i odabir korisnika kojima će se projekt dodijeliti. Ako jedno ili više polja nije ispravno ili uopće popunjeno, prikazati će se greška u formi koja će uputiti kreatora projekta kako da je ispravi. Ako je sve ispravno popunjeno, pritiskom na dugme *Add project* stvara se novi projekt objekt u bazi te se preusmjerava korisnika na *Dashboard* gdje može vidjeti svoj novi projekt među već ostalim postojećim projektima.



Slika 5.4. *Forma za kreiranje projekta.*

5.5. Prikaz projekta i komentiranje projekta

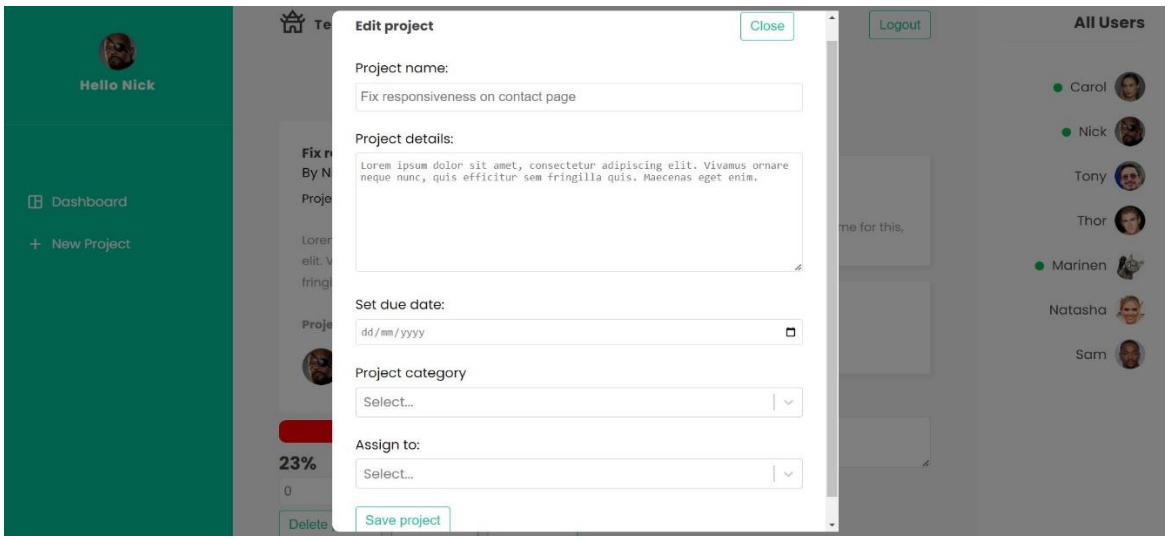
Nakon kreacije projekta, projekt se nalazi na *Dashboard* stranici te klikom na projekt se otvaraju detalji projekta prikazano na slici 5.5. Ovdje korisnik vidi sve što je u formi za kreiranje projekta upisao. Vidi se ime projekta, opis projekta, krajnji datum izrade projekta, kategorija projekta i lista korisnika kojima je projekt dodijeljen. Projekt se može komentirati te vidjeti tko je i kada ostavio komentar. Tipke *Delete project* i *Edit project* su vidljive samo kreatoru projekta. Tipka *Project logs* je vidljiva svim korisnicima koji otvore projekt. U *progress input*, koji je ujedno samo vidljiv kreatoru projekta, može se upisivati koliki je postotak izvršenosti projekta. Postotak izvršenosti projekta je vidljiv svim korisnicima. Pritiskom na tipku *Delete project* briše se projekt iz baze te više nije vidljiv unutar aplikacije.



Slika 5.5. *Detalji kreiranog projekta.*

5.6. Izmjena projekta

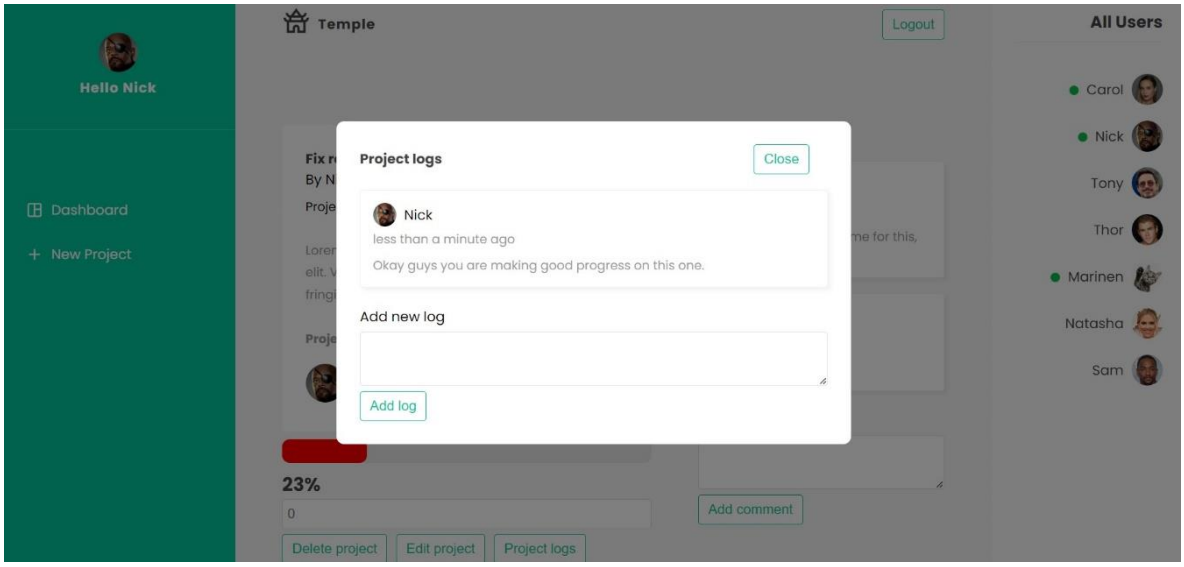
Pritiskom tipke *Edit project* otvara se modal sa formom za izmjenu projekta. Forma je identična onoj kao i za kreiranje projekta. Korisnik ovdje može promijeniti koja polja treba te nakon toga pritiskom na dugme *Save project* projekt se izmijenio sa novim podacima te se modal zatvara.



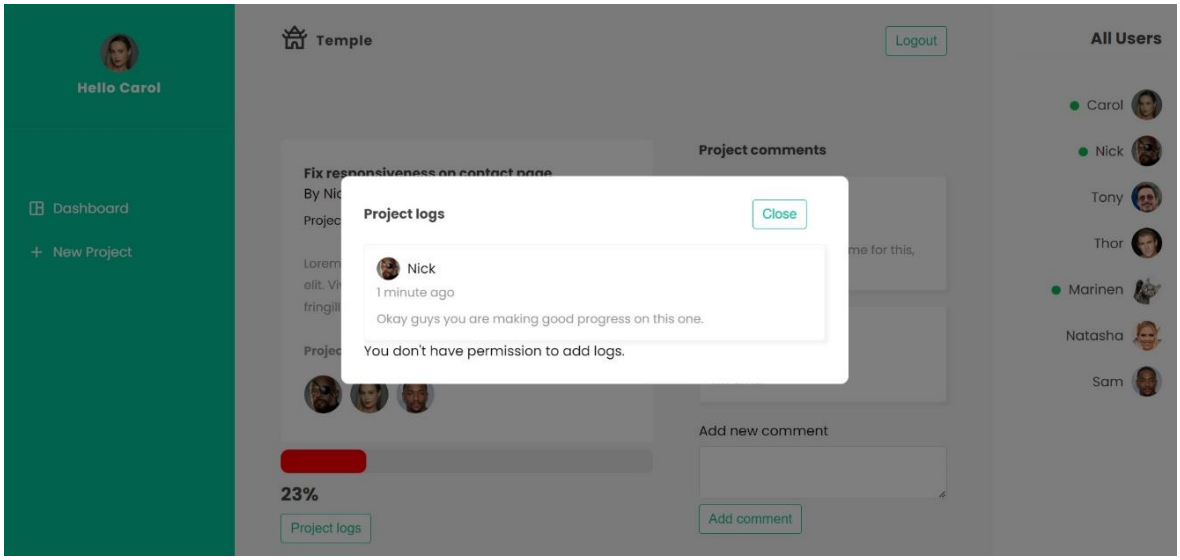
Slika 5.6. *Prikaz forme za izmjenu projekta.*

5.7. Bilješke projekta

Pritiskom na tipku *Project logs* otvara se modal gdje se vide svi dosad zabilježeni komentari kreatora projekta. Ovdje kreator upisuje što i kada se napravilo na projektu kako bi se lakše pratio napredak projekta. Treba naglasiti da samo kreator projekta ima pristup upisivanju bilješki dok ostali korisnici imaju samo pristup pregleda bilješki.



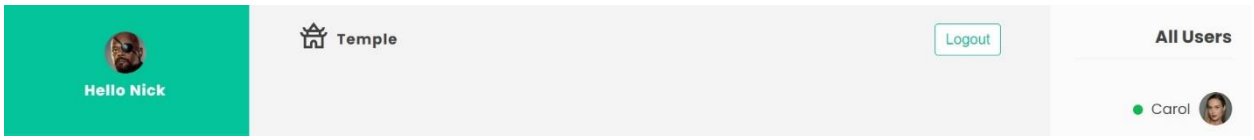
Slika 5.7. *Prikaz dnevnika projekta iz kreatorove perspektive.*



Slika 5.8. *Prikaz dnevnika projekta iz korisnikove perspektive.*

5.8. Odjava

Kada korisnik ne želi više koristiti aplikaciju, prilikom pritiska na tipku *Logout* koja se nalazi sa desne strane, korisnik se odjavljuje iz aplikacije te je preusmjeren na stranicu za prijavu. Nakon odjave korisnik nema pristup aplikaciji dok se ponovno ne prijavi.



Slika 5.9. *Prikaz tipke za odajvu.*

6. ZAKLJUČAK

Primarni zadatak ovog završnog rada bio je izraditi web aplikaciju koja bi poslužila timovima u firmi za lakšu komunikaciju oko projekata. Aplikacija je izrađena za sve koji žele brzo i efektivno komunicirati na izradi projekta. Aplikacija omogućuje izradu profila, prijave, pregled svih projekata, filtriranje projekata po kategorijama, kreiranje projekta, komentiranje projekta, izmjenu projekta, prikaz postotka izvršenosti projekta, pregled svih korisnika i jesu li online, zapisivanje dnevnika projekta te odjava iz aplikacije. Trenutno postoji podosta aplikacija za kreiranje projekata tako da bi se ova aplikacija mogla nadograditi sa nekolicinom stvari kao što su direktne poruke između korisnika, stvaranje grupnih razgovora, dobivanje notifikacije kada smo dodani na projekt i slično. Za izradu ove aplikacije potrebno je znanje *frontend* tehnologija HTML, CSS, JavaScript te React biblioteke i *backend* tehnologije Firebase.

LITERATURA

[1] What is HTML? : https://www.geeksforgeeks.org/html/ [02.04.2023]

[2] History of CSS : https://www.thecrazyprogrammer.com/2021/11/history-of-css.html [04.04.2023]

[3] What is JavaScript: https://aws.amazon.com/what-is/javascript/ [05.04.2023]

[4] What is React? : https://www.technigo.io/explained/what-is-react [07.04.2023]

[5] Why choose React development in 2022: The Ultimate Guide: https://www.uplers.com/blog/why-choose-react-web-development/ [07.04.2023]

[6] What is Google Firebase and Why Should You Use It?: https://www.makeuseof.com/what-is-google-firebase-why-use-it/ [10.04.2023]

[7] What is Visual Studio Code? Microsoft's extensible code editor: https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html [12.04.2023]

SAŽETAK

Tema: Web aplikacija za kreiranje projekta

U ovom završnom radu opisana je web aplikacija za upravljanje projektima. Aplikacija predviđena za korištenje između članova tima na projektu. Članovima nudi mogućnosti kao što su kreiranje projekta, komentiranje projekta, praćenja napretka projekta, brisanje projekta, izmjena projekta, filtriranje projekata po kategorijama, zapisivanje dnevnika projekta i slično. Kako bi sve od navedenog bilo dostupno, korisnici se prvo trebaju registrirati i prijaviti u aplikaciju. Neprijavljenom korisniku nije dopušteno da koristi aplikaciju te na svaki pokušaj korištenja je preusmjeren na stranicu za prijavu. Kod je napisan u Visual Studio Code-u. U nekoliko poglavlja su opisane korištene tehnologije za *frontend* kao što su HTML, CSS, JavaScript te React, a za *backend* Firebase.

Ključne riječi: backend, Firebase, frontend, React, upravljanje projektima

ABSTRACT

Topic: Web application for project management

In this paper, a web application for project management is described. An application is intended for use between team members on a project. It offers options to members such as creating a project, commenting on a project, monitoring project progress, deleting project, editing project, filtering projects by category, adding project logs etc. In order for all of the above to be available, users must first register and login in to the application. An unregistered user is not allowed to use the application and is redirected to the login page every time he tries to use it. The code is written in Visual Studio Code. Several chapters describe the technologies used for the interface, such as HTML, CSS, JavaScript and React and Firebase for the backend.

Key words: backend**,** Firebase, frontend, React, project management