

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 649

**RAZVOJ PROGRESIVNE WEB APLIKACIJE ZA PRAĆENJE
DNEVNE PONUDE U MENZAMA TE NUTRITIVNIH
VRIJEDNOSTI HRANE**

Luka Habuš

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 649

**RAZVOJ PROGRESIVNE WEB APLIKACIJE ZA PRAĆENJE
DNEVNE PONUDE U MENZAMA TE NUTRITIVNIH
VRIJEDNOSTI HRANE**

Luka Habuš

Zagreb, lipanj 2022.

ZAVRŠNI ZADATAK br. 649

Pristupnik: **Luka Habuš (0036517946)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentorica: prof. dr. sc. Lea Skorin-Kapov

Zadatak: **Razvoj progresivne web aplikacije za praćenje dnevne ponude u menzama te nutritivnih vrijednosti hrane**

Opis zadatka:

U današnje vrijeme sve je češća uporaba aplikacija koje potiču zdrave životne navike. Primjeri takvih aplikacije su one koje potiču korisnike na svakodnevno vježbanje, redovito spavanje, zdravu prehranu i sl. Aplikacije mogu biti dostupne putem web preglednika, a također mogu biti instalirane na samom uređaju pri čemu nije potrebna konekcija na Internet. Progresivne web aplikacije (engl. Progressive Web Apps, PWA) nameću se kao trend koji omogućava korisnicima da lako i brzo preko weba pronalaze aplikacije, dodaju ih na svoje zaslone, primaju obavijesti te ih koriste i kada nisu spojeni na Internet. Ovim karakteristikama korisnici imaju dojam kao da koriste mobilnu aplikaciju, no bez potrebne prethodne instalacije. U ovom radu potrebno je razviti progresivnu web aplikaciju za praćenje dnevne ponude u studentskim menzama te nutritivnih vrijednosti hrane. Nadalje, aplikacija treba omogućiti praćenje gužve u menzama te osobnu potrošnju novaca. Cjelokupna aplikacija se razvija u timu, a cilj ovoga rada je razviti i integrirati korisničko sučelje navedene aplikacije. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Rok za predaju rada: 10. lipnja 2022.

Sadržaj

Uvod	1
1. Progresivne web aplikacije (PWA)	2
1.1. Serviseri	2
1.2. Registracija serviseri	3
1.3. Predmemoriranje	3
1.4. Manifest web aplikacije	4
1.5. Mogućnosti PWA	4
2. Programsko rješenje izrade progresivne web aplikacije “Što se jede?”	5
2.1. Opis i cilj izrade aplikacije	5
2.2. Korištene tehnologije	6
2.2.1. TypeScript	6
2.2.2. HTML	6
2.2.3. CSS	7
2.2.4. Node.js	7
2.2.5. React	8
2.2.6. jQuery	8
2.2.7. Tailwind CSS	8
2.2.8. Workbox	9
2.2.9. Visual Studio Code	9
2.2.10. Google Chrome	9
2.2.11. Figma	10
2.3. Implementacija i razvoj aplikacije	11
2.3.1. Web dizajn dokument	11
2.3.2. Dohvaćanje i prikaz podataka	13

2.3.3.	Početna stranica	14
2.3.4.	Kalkulator energijskog unosa	16
2.3.5.	Stranica zdravih recepata.....	17
2.3.6.	Stranica za registraciju i prijavu u sustav	18
2.3.7.	Planovi za budući razvoj aplikacije	19
	Zaključak	20
	Literatura	21
	Sažetak.....	22
	Summary.....	23

Uvod

U današnje vrijeme sve je češća uporaba aplikacija koje potiču zdrave životne navike. Primjeri takvih aplikacije su one koje potiču korisnike na svakodnevno vježbanje, redovito spavanje, zdravu prehranu i sl. Aplikacije mogu biti dostupne putem web preglednika, a također mogu biti instalirane na samom uređaju pri čemu nije potrebna konekcija na Internet. Progressivne web aplikacije (engl. Progressive Web Apps, PWA) nameću se kao trend koji omogućava korisnicima da lako i brzo preko weba pronalaze aplikacije, dodaju ih na svoje zaslone, primaju obavijesti te ih koriste i kada nisu spojeni na Internet. Ovim karakteristikama korisnici imaju dojam kao da koriste mobilnu aplikaciju, no bez potrebne prethodne instalacije.

U ovom radu se opisuje postupak izrade progresivne web aplikacije putem koje će se moći pregledavati dnevna ponuda u studentskim menzama i nutricionističke vrijednosti hrane te računati energijski unos. U prvom poglavlju rada objašnjen je pojam i mogućnosti progresivnih web aplikacija (PWA). Opisane su osnovne karakteristike i postupak za realizaciju PWA. U drugom poglavlju opisano je programsko rješenje izrade progresivne web aplikacije. Opisana je ideja rada i koncept idejnog rješenja. Zatim su opisane korištene tehnologije i vanjske biblioteke potrebne za izradu aplikacije, sama implementacija te daljnje mogućnosti proširenja aplikacije. Glavna ideja završnog rada je stvoriti jednostavnu funkcionalnu web aplikaciju koja u budućnosti može biti nadograđena novim značajkama.

1. Progressivne web aplikacije (PWA)

Progressivne web aplikacije (PWA) su web aplikacije koje koriste progresivno poboljšanje kako bi korisnicima pružile pouzdanije iskustvo, koriste nove mogućnosti za pružanje integriranijeg iskustva i koje se mogu instalirati [1]. Budući da se radi o web-aplikaciji, može se koristiti na bilo kojem uređaju samo s jednom bazom koda. Jednom instalirana, PWA izgleda kao svaka druga aplikacija:

- Ima ikonu na početnom zaslonu
- Pojavljuje se kada tražite aplikacije na uređaju
- Otvara se u samostalnom prozoru, potpuno odvojenom od korisničkog sučelja web preglednika
- Ima pristup višim razinama integracije s OS-om, na primjer, rukovanje URL-om ili prilagođavanje naslovne trake
- Može raditi izvan mreže

1.1. Serviseri

Serviseri (engl. service workers) su temeljni dio progresivnih web aplikacija. Omogućuju brzo učitavanje (bez obzira na mrežu), izvanmrežni pristup, push obavijesti i druge mogućnosti.

Korisnici očekuju da će se aplikacije pokrenuti na sporim ili nestabilnim mrežnim vezama, pa čak i kada su izvan mreže. Očekuju da će sadržaj s kojim su nedavno imali interakciju, poput medijskih zapisa ili ulaznica, biti dostupan i upotrebljiv. Kada zahtjev nije moguć, očekuju da će im aplikacija reći umjesto tihog neuspjeha ili rušenja. A korisnici žele sve to brzo. Kao što možemo vidjeti u studiji Milisekunde čine milijune [2], čak i poboljšanje vremena učitavanja od 0,1 sekunde može poboljšati konverziju do 10%. Ukratko: korisnici očekuju da će progresivna web aplikacija biti pouzdana i zato imamo servisere.

Kada aplikacija zatraži resurs koji pokriva opseg serviser, uključujući i kada je korisnik izvan mreže, serviser presreće zahtjev, djelujući kao mrežni proxy. Zatim može odlučiti treba li poslužiti resurs iz predmemorije putem API-ja za pohranu predmemorije, iz mreže kao što bi se inače događalo bez serviser, ili ga stvoriti iz lokalnog algoritma. To vam omogućuje da pružite iskustvo slično onome koje pruža platformska aplikacija. Može raditi čak i potpuno izvan mreže.

1.2. Registracija serviser

Prije nego što serviser preuzme kontrolu nad stranicom, ona mora biti registrirana za tu progresivnu web aplikaciju. To znači da će prvi put kada korisnik dođe na aplikaciju mrežni zahtjevi ići izravno na poslužitelj jer serviser još nema kontrolu nad stranicama.

Nakon provjere podržava li preglednik Service Worker API, PWA može registrirati serviser. Kada se učita, serviser se postavlja između PWA i mreže, presrećući zahtjeve i servirajući odgovarajuće odgovore.

Serviser se može registrirati na sljedeći način:

```
if ('serviceWorker' in navigator) {
    navigator.serviceWorker.register("/serviceworker.js");
}
```

1.3. Predmemoriranje

Predmemorija smanjuje ovisnost aplikacije o mrežnim uvjetima. Uz dobru upotrebu predmemorije, web aplikacija se može učiniti dostupnom izvan mreže i u bilo kojem mrežnom stanju.

API za pohranu predmemorije dostupan je iz različitih konteksta:

- Kontekst prozora (glavna dretva progresivne web aplikacije)
- Serviser
- Svi drugi serviseri koje koristite

Jedna od prednosti upravljanja predmemorijom pomoću serviser je ta što njegov životni ciklus nije vezan za prozor, što znači da ne blokira glavnu dretvu.

1.4. Manifest web aplikacije

Manifest web aplikacije je JSON datoteka koja definira kako se PWA treba tretirati kao instalirana aplikacija, uključujući izgled i dojam te osnovno ponašanje unutar operacijskog sustava. Manifest može uključivati osnovne informacije kao što su naziv aplikacije, ikona i boja teme te napredne postavke, kao što su željena orijentacija i prečaci za aplikacije.

1.5. Mogućnosti PWA

Progresivne web aplikacije mogu raditi više toga od prikazivanja sadržaja na zaslonu ili povezivanja s web uslugama. PWA mogu pristupati datotekama iz datotečnog sustava, mogu komunicirati s međuspremnikom sustava, pristupiti hardveru koji je povezan s uređajem i još mnogo toga. U nastavku su navedene neke od najvažnijih mogućnosti koje možete koristiti pomoću progresivne web aplikacije:

- Lokalno spremanje datoteka u predmemoriju pomoću Cache API-ja
- Izvršavanje zadataka u dretvama pomoću servisera
- Upravljanje mrežnim zahtjevima s različitim strategijama u serviseru
- WebAssembly za izvršavanje niskorazinskog kompajliranog koda radi izvedbe
- Komunikacija u stvarnom vremenu, korištenjem WebRTC API-ja
- Geolokacija dobiva lokaciju korisnika putem različitih davatelja usluga, kao što su satelit ili Wi-Fi putem Geolocation API-ja
- Kamera i mikروفon primaju strujanje medija s kamera i mikroфона pomoću sučelja medijskih uređaja
- Senzori prikupljaju informacije u stvarnom vremenu s akcelerometra, žiroskopa, magnetometra
- Biometrijska provjera autentičnosti (kao što je prepoznavanje lica ili otiska prsta)
- Slanje obavijesti pomoću push notifikacija

2. Programsko rješenje izrade progresivne web aplikacije “Što se jede?”

U ovom poglavlju će biti opisani glavni koraci izrade progresivne web aplikacije “Što se jede?” za praćenje dnevne ponude u studentskim menzama te nutritivnih vrijednosti hrane, korištene tehnologije te glavne funkcionalnosti aplikacije.

2.1. Opis i cilj izrade aplikacije

Programska implementacija predstavlja progresivnu web aplikaciju (PWA) koja korisnicima omogućava pregled dnevne ponude u menzama u Zagrebu, Splitu i Varaždinu. Unutar same aplikacije implementiran je kalkulator za izračun energijskog unosa hrane i stranica zdravih recepata čime se korisnike potiče na zdravu prehranu. Sama aplikacija je funkcionalna prototipna verzija koja se može koristiti na mobilnim uređajima i računalima kroz web preglednik ili instalacijom. U budućnosti ima potencijal za nadogradnju novim značajkama poput praćenja potrošnje novca i gužve u menzi te gemifikacije zdrave prehrane.

Aplikacija je podijeljena u četiri stranice. Na samom početku imamo stranicu za registraciju korisnika i prijavu. Nakon uspješne prijave korisnik se preusmjeruje na stranicu za pregled dnevne ponude hrane gdje može odabrati željeni grad i menzu. Sveukupno može birati između 3 grada i 15 studentskih menzi. Uz svako jelo prikazuju se i nutritivne vrijednosti tog jela (energija, ugljikohidrati, proteini i masti) te gumbi za dodavanje ili uklanjanje jela s tanjura. Nakon što je korisnik odabrao što će jesti može pregledati izračun ukupnog energijskog unosa te može isprazniti tanjur u slučaju da želi uzeti nešto drugo. Posljednja značajka je stranica zdravih recepata gdje su nabrojani svi potrebni sastojci i objašnjen je postupak pripreme određenog jela.

Ova aplikacija nastala je u sklopu Lumen Development natjecanja [3]. To je najveće studentsko natjecanje u izradi web i mobilnih aplikacija u Hrvatskoj. Cilj natjecanja je da na konkretnom slučaju studenti primjene znanje, pokažu svoje programerske, dizajnerske i prezentacijske vještine te se istaknu na tržištu rada. Tijekom natjecanja održavaju se

konzultacije i radionice koje natjecateljima omogućuju komunikaciju s najpoznatijim IT poduzećima u Hrvatskoj i tako imaju priliku dobiti povratnu informaciju na izrađenu aplikaciju. Radili smo u timu od 4 osobe, pri čemu sam se primarno fokusirao na izradu korisničkog sučelja i završili smo u polufinalu natjecanja.

2.2. Korištene tehnologije

Za izradu progresivne web aplikacije koristilo se nekoliko tehnologija. Za programsku implementaciju korišten je programski jezik TypeScript, prezentacijski jezik HTML, stilski jezik CSS, izvršno okruženje Node.js, JavaScript biblioteke React i jQuery te radni okvir Tailwind CSS. Za programiranje aplikacije koristio se uređivač izvornog koda Visual Studio Code, a testiranje se provodilo unutar Google Chrome web preglednika. Web dizajn dokument izrađen je pomoću kolaborativnog alata za izradu prototipa Figma, a PWA funkcionalnosti su omogućene pomoću Workbox biblioteka.

2.2.1. TypeScript

TypeScript je programski jezik koji je razvio i održava Microsoft. To je strogi sintaktički nadskup JavaScripta i jeziku dodaje neobavezno statičko tipkanje. Budući da je to nadskup JavaScripta, postojeći JavaScript programi su također valjani TypeScript programi. TypeScript je proizašao iz nedostataka JavaScripta za razvoj aplikacija velikih razmjera kako u Microsoftu tako i među njihovim vanjskim korisnicima. Izazovi u radu sa složenim JavaScript kodom doveli su do potražnje za prilagođenim alatima kako bi se olakšao razvoj komponenti u jeziku [4].

2.2.2. HTML

HTML je kratica za HyperText Markup Language, što znači prezentacijski jezik za izradu web stranica. HTML jezikom oblikuje se sadržaj i stvaraju se hiperveze hipertekst dokumenta. HTML je jednostavan za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i tome što je

od početka bio zamišljen kao besplatan i tako dostupan svima. Prikaz hipertekst dokumenta omogućuje web preglednik. Temeljna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertekst dokument. Pri tome se nastoji da taj dokument izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ [5].

2.2.3. CSS

CSS je kratica od (eng.) Cascading Style Sheets. Radi se o stilskom jeziku, koji se rabi za opis prezentacije dokumenta napisanog pomoću markup jezika (npr. HTML).

Kako se web razvijao, prvotno su u HTML ubacivani elementi za definiciju prezentacije (npr. tag), ali je dovoljno brzo uočena potreba za stilskim jezikom koji će HTML osloboditi potrebe prikazivanja sadržaja (što je prvenstvena namjena HTML-a) i njegovog oblikovanja (čemu danas služi CSS). Drugim riječima, stil definira kako prikazati HTML elemente. CSS-om se uređuje sam izgled i raspored stranice [6].

2.2.4. Node.js

Node.js je asinkrono JavaScript izvršno okruženje koje omogućava izvršavanje JavaScript koda izvan web-preglednika. Node.js dopušta programerima da koriste JavaScript za pisanje alata naredbenog retka i za izvođenje skripti na strani poslužitelja za proizvodnju dinamičkog sadržaja web stranice prije nego što se stranica pošalje u korisnikov web preglednik. Posljedično, Node.js predstavlja paradigmu "JavaScript posvuda" koja objedinjuje razvoj web-aplikacije oko jednog programskog jezika, umjesto različitih skriptnih jezika na strani poslužitelja i klijenta. JavaScript je jedini jezik koji Node.js izvorno podržava, ali su dostupni mnogi jezici za prevođenje u JS. Kao rezultat toga, Node.js aplikacije mogu biti napisane i u TypeScript-u.

Node.js radi na petlji događaja s jednom dretvom, koristeći neblokirajuće ulazno/izlazne pozive, dopuštajući mu podršku za desetke tisuća istodobnih veza bez troškova promjene konteksta dretve. Dizajn dijeljenja jedne dretve među svim zahtjevima koji koriste uzorak promatrača namijenjen je za izgradnju visoko konkurentnih aplikacija, gdje svaka funkcija koja izvodi ulazno/izlaznu obradu mora koristiti povratni poziv [7].

2.2.5. React

React (također poznat kao React.js ili ReactJS) besplatna je JavaScript biblioteka otvorenog koda za izgradnju korisničkih sučelja temeljenih na komponentama korisničkog sučelja. Održava ga Meta (bivši Facebook) i zajednica pojedinačnih programera i tvrtki. React se može koristiti kao baza u razvoju jednostraničnih, mobilnih ili poslužiteljskih aplikacija s radnim okvirima poput Next.js. Međutim, React se bavi samo upravljanjem stanja i prikazivanjem tog stanja DOM-u, pa stvaranje React aplikacija obično zahtijeva korištenje dodatnih knjižnica za usmjeravanje, kao i određene funkcionalnosti na strani klijenta.

React se pridržava paradigme deklarativnog programiranja. Programeri dizajniraju poglede za svako stanje aplikacije, a React ažurira i generira komponente kada se podaci promijene. Ovo je u suprotnosti s imperativnim programiranjem.

React kod napravljen je od entiteta koji se nazivaju komponente. Ove komponente mogu se koristiti višestruko puta. Komponente se mogu prikazati određenom elementu u DOM-u pomoću React DOM biblioteke. Prilikom renderiranja komponente, moguće je proslijediti vrijednosti između komponenti kroz "rekvizite" (engl. props).

React Hooks su funkcije koje programerima omogućuju da se "prikluče" na značajke React stanja i životnog ciklusa iz komponenti funkcija [8].

2.2.6. jQuery

jQuery je JavaScript biblioteka koja čini stvari poput obilaženja i manipulacije HTML dokumenata, rukovanja događajima i animacije mnogo jednostavnijim uz API jednostavan za korištenje koji radi u mnoštvu preglednika [9].

2.2.7. Tailwind CSS

Tailwind CSS je uslužni CSS okvir za brzu izgradnju prilagođenih korisničkih sučelja. To je vrlo prilagodljiv CSS okvir niske razine koji vam daje sve građevne blokove koji su vam potrebni za izradu dizajna po mjeri. Ne nameće specifikacije dizajna ili kako bi vaša stranica trebala izgledati. Jednostavno spajate male komponente kako biste

konstruirali jedinstveno korisničko sučelje. Tailwind uzima CSS datoteku, obrađuje ovu CSS datoteku preko konfiguracijske datoteke i proizvodi izlaz [10].

2.2.8. Workbox

Workbox je skup modula koji pojednostavljaju uobičajene interakcije serviseru kao što su usmjerenje i predmemorija. Svaki modul se bavi specifičnim aspektom razvoja serviseru. Workbox ima za cilj učiniti korištenje serviseru što lakšim, a istovremeno dopušta fleksibilnost za prilagodbu složenih zahtjeva aplikacija gdje je to potrebno [11].

2.2.9. Visual Studio Code

Visual Studio Code je Microsoftov uređivač izvornog koda za Windows, Linux i macOS. Može se koristiti s raznim programskim jezicima, uključujući Javu, JavaScript, Go, Node.js, Python, C++. Temelji se na Electron radnom okviru, koji se koristi za razvoj Node.js web aplikacija.

Uključuje osnovnu podršku za većinu uobičajenih programskih jezika, poput isticanja sintakse, podudaranja zagrada, formatiranje koda. Visual Studio Code također se isporučuje s inteligentnim dovršavanjem koda za JavaScript, TypeScript, JSON, CSS i HTML, kao i podršku za otklanjanje pogrešaka za Node.js. Podršku za dodatne jezike mogu pružiti besplatno dostupna proširenja.

Umjesto projektnog sustava, omogućuje korisnicima otvaranje jednog ili više direktorija, koji se zatim mogu spremati u radne prostore za buduću ponovnu upotrebu. To mu omogućuje da radi kao uređivač koda koji ne ovisi o jeziku [12].

2.2.10. Google Chrome

Google Chrome je web preglednik koji razvija američka tvrtka Google. Stabilna inačica dostupna je za Microsoft Windows, Mac OS X i Linux operacijske sustave. Chrome omogućuje korisnicima da sinkroniziraju svoje oznake, povijest i postavke na svim uređajima s instaliranim preglednikom slanjem i primanjem podataka putem

odabranog Google računa, koji zauzvrat ažurira sve prijavljene instance Chromea. Za web programere, Chrome ima preglednik elemenata koji korisnicima omogućuje da pogledaju DOM i vide što čini web stranicu. Od svibnja 2012. Google Chrome je web preglednik s najvećim udjelom na tržištu [13].

2.2.11. Figma

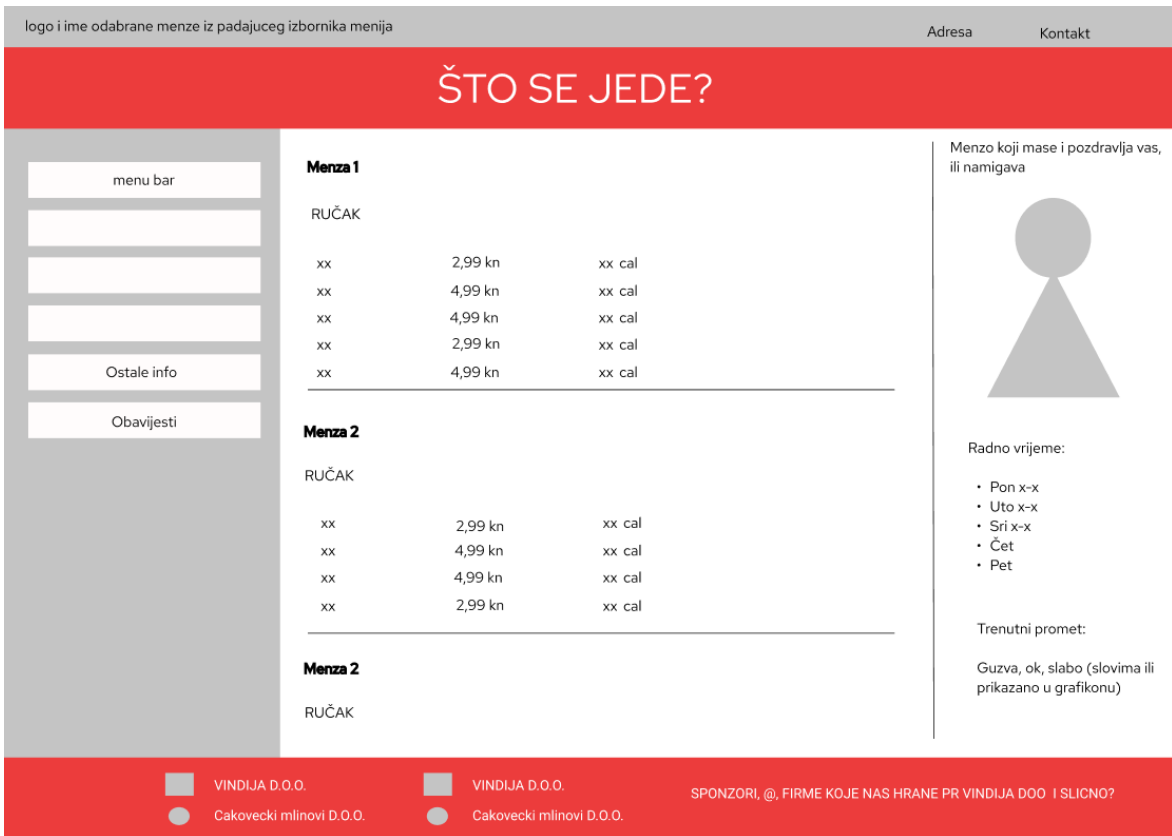
Figma je uređivač vektorske grafike i alat za izradu prototipa koji se prvenstveno temelji na webu, s dodatnim izvanmrežnim značajkama koje omogućuju desktop aplikacije za macOS i Windows. Mobilna aplikacija Figma za Android i iOS omogućuje pregled i interakciju s Figma prototipima na mobilnim uređajima u stvarnom vremenu. Skup značajki Figue usredotočen je na korištenje u korisničkom sučelju i dizajnu korisničkog iskustva, s naglaskom na suradnju u stvarnom vremenu [14].

2.3. Implementacija i razvoj aplikacije

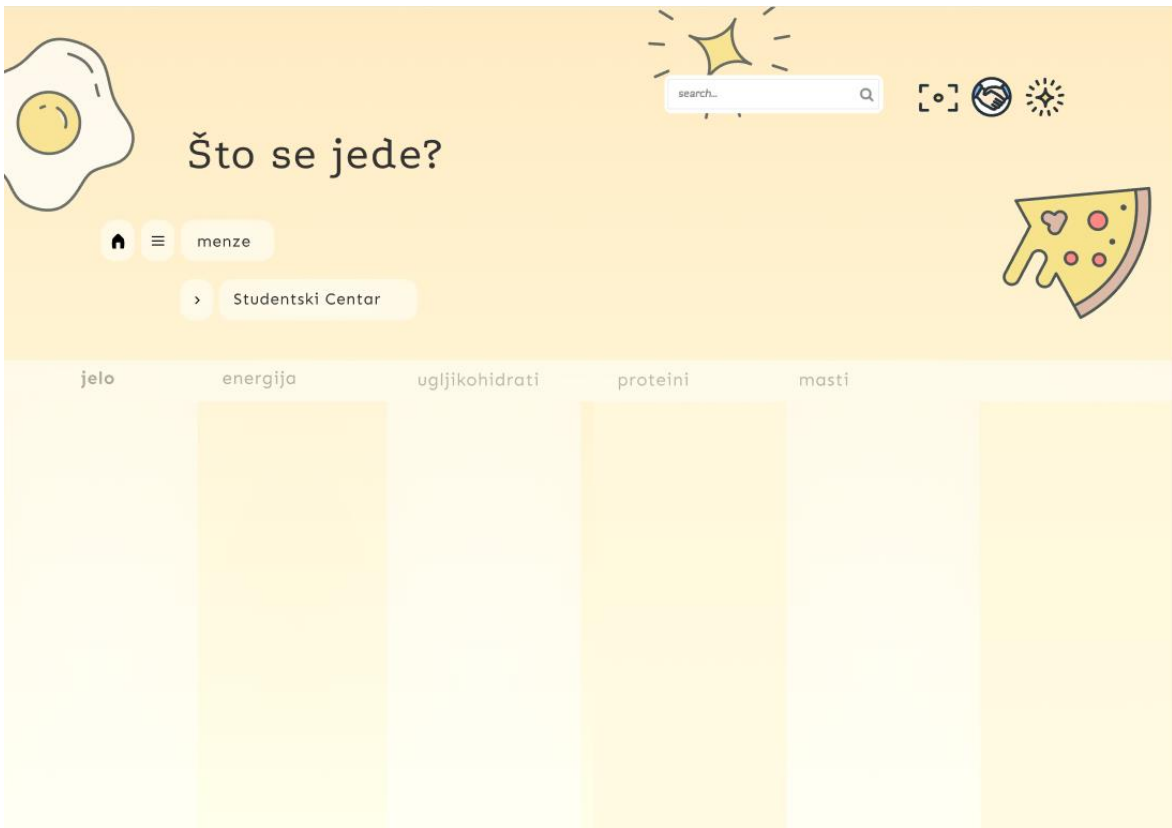
Na samom početku razvoja aplikacije “Što se jede?” napravljen je web dizajn dokument koji je služio kao referenca za izradu korisničkog sučelja. Paralelno s time radilo se na poslužiteljskoj strani dohvata podataka u stvarnom vremenu i stvaranju baze podataka nutricionističkih vrijednosti hrane. Također se radilo na dizajnu lika zvanog Menzo koji bi služio kao maskota aplikacije. Planirano je dodavanje mogućnosti igranja jednostavnih igara s Menzom u svrhu gemifikacije zdrave prehrane. Nakon što je implementiran dohvat podataka s poslužiteljske strane izrađena je početna stranica gdje korisnik ima mogućnost odabira grada u kojem se nalazi i željene menze te prikaz dnevne ponude hrane. To je omogućilo daljnji razvoj kalkulatora energijskog unosa. Također je dodana stranica s prikazom zdravih recepata i na kraju su implementirane stranice za registraciju i prijavu u sustav.

2.3.1. Web dizajn dokument

Web dizajn dokument služi kao referenca za sve uključene u razvoj i izrađen je pomoću Figma. Na njemu su vidljive funkcionalnosti koje web aplikacija mora imati. Tijekom razvoja se odustalo od nekih značajki zbog manjka vremena, a samim time se mijenjao i web dizajn dokument. U nastavku može se vidjeti inicijalni dizajn početne stranice (Slika 1) i završna verzija dizajna početne stranice (Slika 2). Izbačen je sav višak informacija i ostavljene su samo najbitnije funkcionalnosti odabira grada i željene studentske menze.



Slika 1. Inicijalni dizajn početne stranice



Slika 2. Završna verzija dizajna početne stranice

2.3.2. Dohvaćanje i prikaz podataka

Kako bi u svakom trenutku prikazivali točnu dnevnu ponudu hrane napisane su funkcije za dohvat podataka iz različitih menzi ovisno o odabranom gradu (Slika 3). Također, izrađena je baza podataka nutricionističkih vrijednosti za 240 različitih jela (Slika 4). Podaci su dobiveni iz aplikacije NutriMenza [15]. Za svako dohvaćeno jelo s web stranice menze vrši se provjera nalazi li se jelo u bazi nutricionističkih vrijednosti. Ponekad se zna dogoditi tipfeler navedenih jela i stoga se provjera radi računanjem Levenshteinove udaljenosti [16]. Levenshteinova udaljenost između dvije riječi najmanji je broj uređivanja od jednog znaka potreban za promjenu jedne riječi u drugu. Jelo s najmanjom navedenom udaljenosti prikazuje se korisniku (Slika 5).

```
62
63 async function scrape_menu_zg(): Promise<Canteen[]> {
64   const scrape_cassandra = async () => {
65     const res = await fetch("https://www.cassandra.hr/studentski-menu/");
66     const html = await res.text();
67     const dom = new JSDOM(html);
68
69     const data =
70       dom.window.document.querySelector("article)?.textContent || "";
71     const dishes = data
72       .split("\n")
73       .filter((l) => /^[^\s]+/.test(l))
74       .filter((v, i, a) => a.indexOf(v) === i)
75       .filter((dish) => dish.indexOf("MENU") === -1)
76       .filter((dish) => dish.indexOf("202") === -1)
77       .map(addNutritionalData);
78
79     return {
80       name: "Cassandra",
81       menu: [{ name: "Ručak", dishes }],
82     };
83   };
84 }
```

Slika 3. Funkcija za dohvaćanje dnevne ponude u restoranu Cassandra

```
379   {
380     "dish": "Špageti bolonjez",
381     "energy": 654.2,
382     "carbohydrates": 83.7,
383     "proteins": 34.2,
384     "fat": 22.2,
385     "category": "Glavna jela"
386   },
```

Slika 4. Prikaz jela u bazi nutricionističkih vrijednosti

Što se jede? 🍴 📄

Jelo	Energija	Ugljikohidrati	Proteini	Masti		
Grah varivo	419.8 kcal	52.4 g	19.6 g	16.1 g	Ukloni	Dodaj
Kobasice	384.1 kcal	8.9 g	30.1 g	25.7 g	Ukloni	Dodaj
Krem juha od gljiva	49.5 kcal	6.9 g	2.3 g	1.3 g	Ukloni	Dodaj
Kruh (3šnite)	195.9 kcal	23 g	2.7 g	10.5 g	Ukloni	Dodaj
Pohani sojini medaljoni	521.9 kcal	44.4 g	43.6 g	18 g	Ukloni	Dodaj

Slika 5. Prikaz jela na početnoj stranici

2.3.3. Početna stranica

Na vrhu početne stranice napisan je naziv menze i grad. Ispod se nalaze padajući izbornici za odabir menze i grada (Slika 6). Od gradova korisnik može odabrati Zagreb, Split ili Varaždin (Slika 7). Ovisno o gradu, korisnik može odabrati sljedeće studentske menze:

- Zagreb: Cassandra, Odeon
- Split: objekt jednostavnih usluga STOP, restoran Spinut, restoran FGAG, restoran FESB, restoran Indeks, restoran Kampus, restoran Ekonomski fakultet, restoran Kantun C (Slika 8)
- Varaždin: studentski restoran Varaždin, studentski restoran Čakovec, studentski restoran Koprivnica, studentski restoran Križevci, studentski restoran Bjelovar

Nakon toga slijedi prikaz jela i nutricionističkih vrijednosti (energija, ugljikohidrati, proteini, masti) te gumbi za dodavanje jela u tanjur ili uklanjanje. Gumbi mijenjanju boju ako prijedemo mišem preko. Gumb „Ukloni“ poprima crvenu boju, dok gumb „Dodaj“ zelenu (Slika 9). Na dnu se nalazi izbornik za sortiranje hrane po meniju, abecedi, proteinima i kalorijama (Slika 10).

Što se jede? 🍷 📄

Cassandra, Zagreb

Cassandra ▾ Zagreb ▾

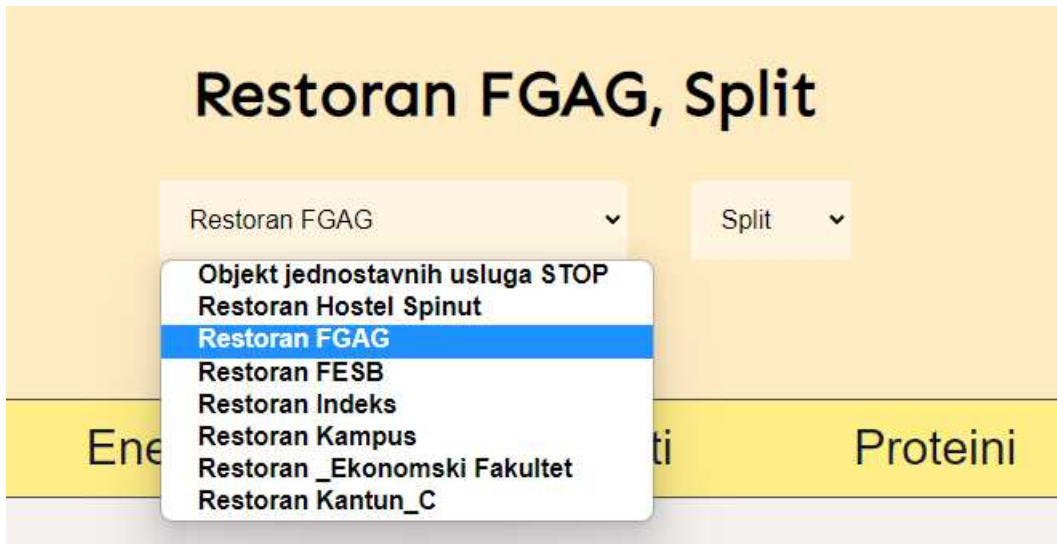
Ručak 🍕

Jelo	Energija	Ugljikohidrati	Proteini	Masti		
Grah varivo	419.8 kcal	52.4 g	19.6 g	16.1 g	Ukloni	Dodaj
Kobasice	384.1 kcal	8.9 g	30.1 g	25.7 g	Ukloni	Dodaj
Krem juha od gljiva	49.5 kcal	6.9 g	2.3 g	1.3 g	Ukloni	Dodaj

Slika 6. Početna stranica



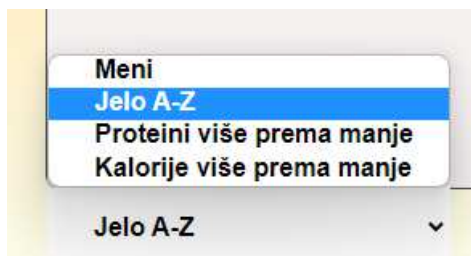
Slika 7. Odabir grada



Slika 8. Odabir menze



Slika 9. Gumbi "Ukloni" i "Dodaj"



Slika 10. Sortiranje prikaza jela

2.3.4. Kalkulator energijskog unosa

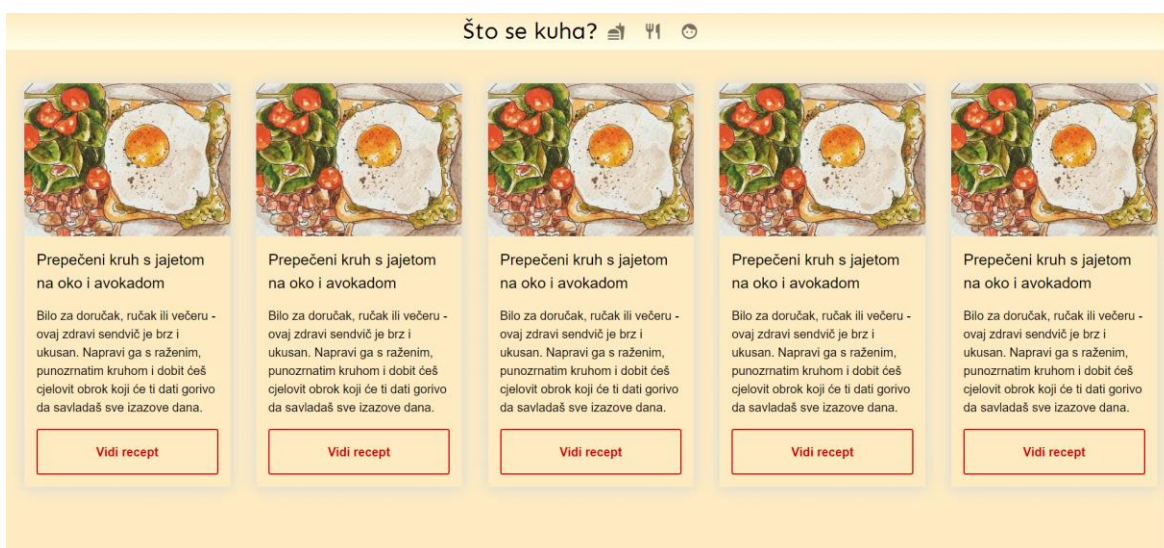
Nakon odabira željenih jela na početnoj stranici računa se suma nutricionističkih vrijednosti na stranici "Što je na tanjuru?". Suma je prikazana s konzistentnim stilovima u tabličnom prikazu pomoću Tailwind CSS-a, svi odabrani proizvodi smještaju se u lokalno pohranjeni niz `selectedDishes` uz pomoć `JSON.stringify` metode. Ovo omogućuje da se vrijednosti u njemu dinamično mijenjaju koristeći `React useState` hook. Odabrana jela mogu se pojedinačno uklanjati s tanjura, ili ako korisnik želi ukloniti sve proizvode - postoji opcija "Isprazni tanjur" koji vezanom funkcijom isprazni sve elemente iz `selectedDishes` niza (Slika 11).



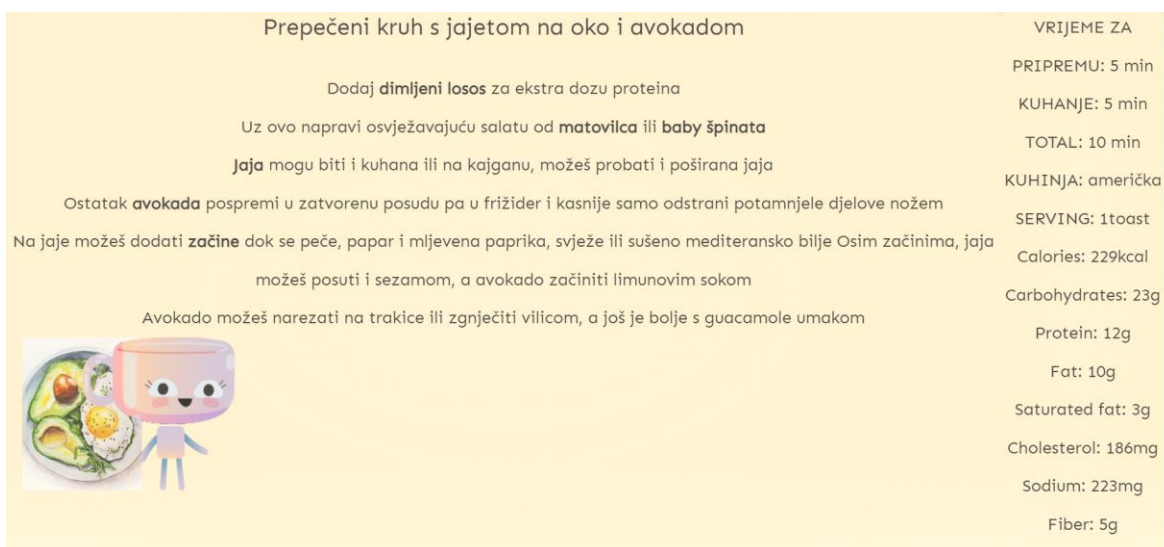
Slika 11. Kalkulator energijskog unosa

2.3.5. Stranica zdravih recepata

Ova stranica sadrži personalizirane kartične prikaze recepata na koje klikom možemo skočiti do odgovarajućeg opisa recepta (Slika 12). Navigacija je izgrađena pomoću react-router-dom, preciznije, rute su uspostavljene u tijelu Recipes.tsx komponente s odgovarajućim poveznicama koje se tada mogu primijeniti kroz cijelu aplikaciju. Trenutno sve kartice imaju isti sadržaj, ali se vrlo lako može prilagoditi različitim receptima u budućnosti. Prikazan je recept za prepečeni kruh s jajetom na oko i avokadom (Slika 13).



Slika 12. Kartični prikaz recepata



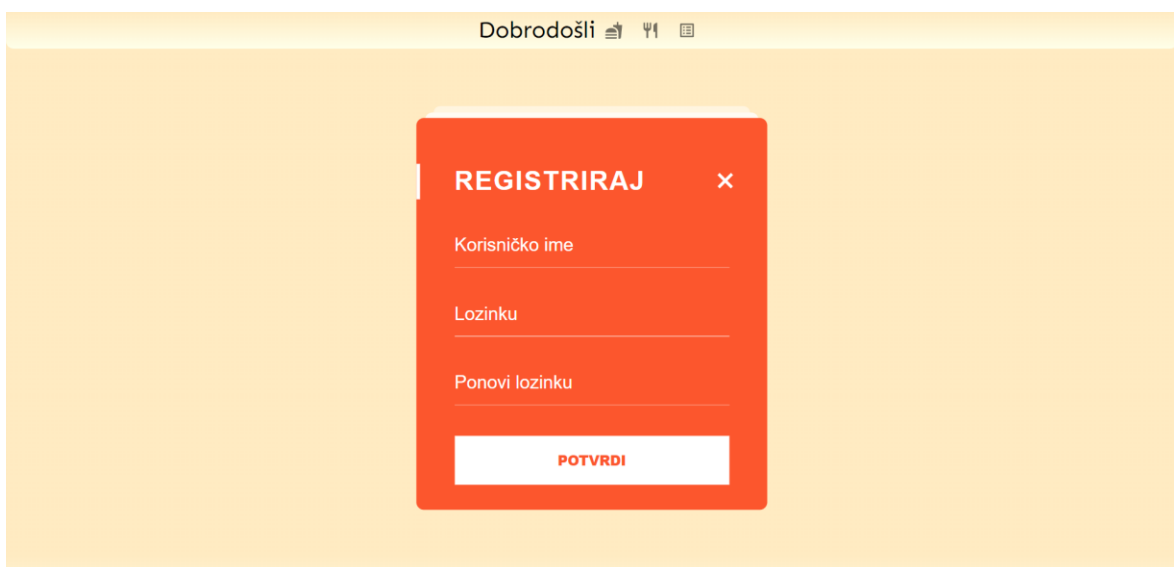
Slika 13. Primjer recepta

2.3.6. Stranica za registraciju i prijavu u sustav

Stranica za registraciju i prijavu u sustav sadrži kombinaciju SCSS-a i jQuery biblioteke s kojom je moguće napraviti komplicirane animacije funkcijama koje djeluju na SCSS stilove. To smo primijenili na + marker koji služi registraciji korisnika (Slika 14). Kad je pritisnut element se naizgled prelije laganom tranzicijom preko tijela forme za prijavu u sustav. Dodatne opcije za registraciju korisnika se prikazuju uz pomoć jQuery funkcije nakon što animacija stane (Slika 15).



Slika 14. Forma za prijavu



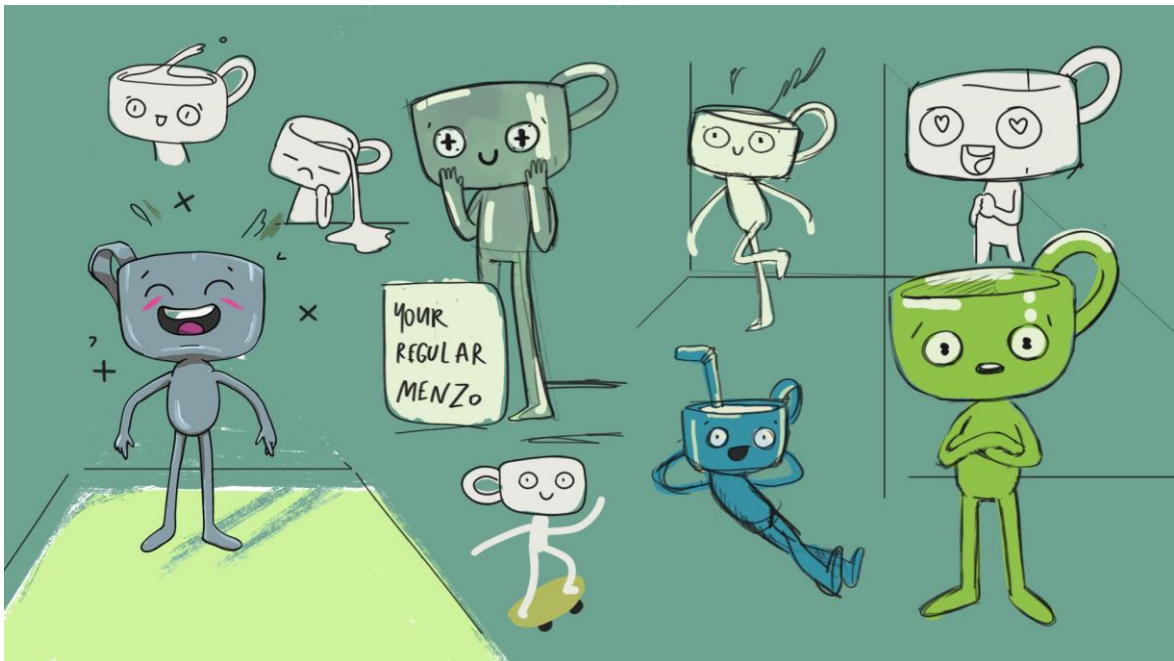
Slika 15. Forma za registraciju korisnika

2.3.7. Planovi za budući razvoj aplikacije

Aplikacija se može proširiti sljedećim značajkama koje se nisu stigle implementirati u trenutnoj fazi razvoja:

- Praćenje potrošnje novca u menzi
- Analiza energijskog unosa hrane kroz vrijeme i stvaranje statistike prehrambenih navika studenata
- Praćenje gužve u menzi
- Gemifikacija zdrave prehrane

Gemifikacija zdrave prehrane je zamišljena tako da korisnici imaju mogućnost igranja jednostavnih igara dok čekaju u redu za menzu. U igrama se pojavljuje lik Menzo koji je ujedno i njihov avatar. U slučaju da se korisnik nezdravo hrani kroz neko određeno vrijeme ne može igrati igre jer je Menzo umoran i to korisnika aplikacije potiče na zdravu prehranu. U nastavku se može vidjeti dizajn lika (Slika 16).



Slika 16. Lik Menzo

Zaključak

Progresivne web aplikacije (PWA) su web aplikacije koje koriste progresivno poboljšanje kako bi korisnicima pružile pouzdanije iskustvo, koriste nove mogućnosti za pružanje integriranijeg iskustva i koje se mogu instalirati.

U ovom završnom radu opisane su mogućnosti PWA te postupak izrade progresivne web aplikacije pomoću servisera (*service worker-a*). Koristeći se modernim web tehnologijama poput *NodeJS-a* i *React-a* napravljena je progresivna web aplikacija naziva “Što se jede?” koja služi pregledu dnevne ponude u studentskim menzama i nutricionističkih podataka te praćenju energijskog unosa hrane. Također je napravljena stranica zdravih recepata kako bi se korisnike potaknulo na zdraviju prehranu. Aplikacija ima puno potencijala za daljnji razvoj i upotrebu te je vrlo skalabilna i laka za održavanje.

PWA predstavlja novu eru razvoja web stranica, koja nadopunjuje dosad izrazito popularan koncept responzivnog web dizajna. Responzivne web stranice podigle su nivo očekivanja korisnika mobilnih i tablet uređaja. Osim apsolutne prilagođenosti svakoj veličini ekrana, korisnici su navikli ne osjetiti razliku između pretraživanja na mobilnim i desktop uređajima te zadržati svoje navike pretraživanja sa svog pametnog telefona. PWA je savršeni hibrid između nativne aplikacije i mobilnog weba.

Literatura

- [1] PWA, pristupljeno 9.6.2022., <https://web.dev/learn/pwa/progressive-web-apps/>
- [2] Milliseconds make millions, pristupljeno 9.6.2022., <https://web.dev/milliseconds-make-millions/>
- [3] Lumen Development, pristupljeno 9.6.2022., <https://www.estudent.hr/projekti/lumen-development>
- [4] TypeScript, pristupljeno 9.6.2022., <https://www.typescriptlang.org/>
- [5] HTML, pristupljeno 9.6.2022., <https://hr.wikipedia.org/wiki/HTML>
- [6] CSS, pristupljeno 9.6.2022., <https://hr.wikipedia.org/wiki/CSS>
- [7] Node.js, pristupljeno 9.6.2022., <https://en.wikipedia.org/wiki/Node.js>
- [8] React, pristupljeno 9.6.2022., <https://reactjs.org/>
- [9] jQuery, pristupljeno 9.6.2022., <https://jquery.com/>
- [10] Tailwind CSS, pristupljeno 9.6.2022., <https://tailwindcss.com/>
- [11] Workbox, pristupljeno 9.6.2022., <https://web.dev/learn/pwa/workbox/>
- [12] VS Code, pristupljeno 9.6.2022., <https://code.visualstudio.com/>
- [13] Google Chrome, pristupljeno 9.6.2022., https://hr.wikipedia.org/wiki/Google_Chrome
- [14] Figma, pristupljeno 9.6.2022., <https://www.figma.com/>
- [15] NutriMenza, pristupljeno 9.6.2022.,
http://www.pbf.unizg.hr/studenti/popularizacija_znanosti/nutrimenza
- [16] Levenshtein distance, pristupljeno 9.6.2022.,
https://en.wikipedia.org/wiki/Levenshtein_distance

Sažetak

U ovom radu objašnjen je pojam i mogućnosti progresivnih web aplikacija (PWA). Opisane su osnovne karakteristike i postupak za realizaciju PWA.

Izrađen je prototip progresivne web aplikacije koja služi pregledu dnevne ponude u studentskim menzama i nutricionističkih podataka te praćenju energijskog unosa hrane. Implementacija programskog rješenja realizirana je u Visual Studio Code uređivaču izvornog koda koristeći moderne web tehnologije NodeJS i React. Korisnike se pomoću aplikacije potiče na zdravije prehrambene navike.

Ključne riječi: progresivne web aplikacije, PWA, prehrana, zdravlje, NodeJS, React

Summary

This thesis explains the concept, characteristics and capabilities of Progressive Web Apps (PWA).

A Progressive Web App was developed that is used for monitoring the daily offer in student canteens and nutritional food values. The app can be used for calculating the energy intake of food and for looking up healthy recipes. It was developed using the Visual Studio Code editor using modern web technologies NodeJS and React. Users are encouraged to develop healthier eating habits.

Key words: Progressive Web Apps, PWA, food, health, NodeJS, React