

# Aplikacija za mobilne uređaje, od ideje do realizacije

---

Jurković, Kristina

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University North / Sveučilište Sjever**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:122:632097>

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

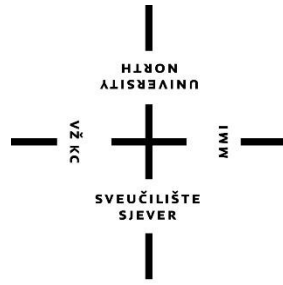
Download date / Datum preuzimanja: **2024-07-22**



Repository / Repozitorij:

[University North Digital Repository](#)





# Sveučilište Sjever

Završni rad br. 726/MM/2021

## **Aplikacija za mobilne uređaje, od ideje do realizacije**

**Kristina, 2930/336**

Varaždin, srpanj 2021. godine

# Prijava završnog rada

## Definiranje teme završnog rada i povjerenstva

ODJEL Odjel za multimediju

STUDIJ preddiplomski stručni studij Multimedija, oblikovanje i primjena

PRISTUPNIK Kristina Jurković

JMBAG 0336029102

DATUM 02.07.2021.

KOLEGIJ Web dizajn

NASLOV RADA Aplikacija za mobilne uređaje, od ideje do realizacije

NASLOV RADA NA ENGL. JEZIKU Mobile app, from idea to realization

MENTOR Robert Geček

ZVANJE doc.art. dr.sc.

ČLANOVI POVJERENSTVA

1. doc.dr.sc. Andrija Bemik - predsjednik
2. Snježana Ivančić Valenko, v.pred. - član
3. doc.art.dr.sc. Robert Geček - mentor
4. Nikolina Bolčević Horvatić, pred. - zamjenski član
5. \_\_\_\_\_

## Zadatak završnog rada

BROJ 726/MM/2021

OPIS

Mobilne aplikacije jedan su od najefikasnijih kanala komunikacije na tržištu. S obzirom na nedostatak cjenika u ugostiteljskim objektima, te primoranim ograničenjem međusobne interakcije, pokazala se potreba za brzim, a efikasnim rješenjem. Kao dobro rješenje pokazala se aplikacija za skeniranje QR koda preko kojega se putem mobilne aplikacije može vidjeti ponuda objekta, te naručiti hrana ili piće, kao i platiti, bez međusobne interakcije između gosta i osoblja. Samo je potrebno na svakom stolu osigurati QR kod kako bi svaki gost bio u mogućnosti skenirati ga i vidjeti ponudu kafića ili restorana.

Rad će uključivati:

Opis izrade aplikacije, definiciju i opis jezika i alata korištenih prilikom izrade aplikacije, te završnu verziju aplikacije izrađene koristeći opisane jezike i alate.

ZADATAK URUČEN

07.07.2021.





# Sveučilište Sjever

**Odjel za Ime odjela**

**Završni rad br. 726/MM/2021**

## **Aplikacija za mobilne uređaje, od ideje do realizacije**

**Student**

Kristina, 2930/336

**Mentor**

doc. art. dr. sc. Robert Geček

Varaždin, srpanj 2021. godine



## **Predgovor**

Ovim putem zahvaljujem profesoru doc.art.dr.sc. Robertu Gečeku na mentorstvu, podršci i pomoći tijekom pisanja ovog završnog rada, kao i na angažiranosti oko istoga. Svojom dostupnošću i brzim povratnim informacijama oko pitanja i nejasnoća koje sam imala bio je od velike pomoći i time pisanje ovog rada učinio lakšim i kvalitetnijim.

Također zahvaljujem i ostalim profesorima koji su se trudili kvalitetno prenijeti svoje znanje, te zbog kojih sam lakše odlučila u kojem području se želim usavršavati.

## Sažetak

Ideja završnog rada je da se kroz isti na konkretnom primjeru prikažu stečena znanja sa fakulteta. Od jezika će se koristiti HTML, CSS i JavaScript. Razvijati će se dizajn aplikacije koji je već prethodno osmišljen, te će po koracima biti objašnjen svaki dio aplikacije kao i kôd za taj dio. Prije svega će se čitatelje uvesti u teorijski dio aplikacije i korištenih tehnologija kako bi lakše razumjeli daljnje postupke i sami kôd. Koraci kodiranja biti će detaljno objašnjeni i prezentirani jasno i na što jednostavniji način. Kao rezultat završnog rada, čitatelju koji se ne razumije u kodiranje trebalo bi biti lako shvatljivo kako se koji dio dizajna razvio.

**Ključne riječi:** mobilna aplikacija, QR sustav, dizajn, kodiranje

## Summary

The idea of the final work is to show the acquired knowledge from the college on a concrete example through the same. HTML, CSS and JavaScript will be used as programming languages. The design of the app, which has already been previously designed, will be developed and each part of the app will be explained step by step, as well as the code for that part. First of all, readers will be introduced to the theoretical part of the app and the technologies used, to make it easier to understand further procedures and the code itself. The coding steps will be explained in detail and presented clearly and in as simply as possible. As a result of the final work, it should be easy for the reader who does not understand coding to understand how is every part of the design developed.

**Keywords:** mobile app, QR system, design, coding



## **Popis korištenih kratica**

**HTML**    HyperText Markup Language  
**CSS**     Cascading Style Sheet  
**NPM**    Node Package Manager

# Sadržaj

1.	Uvod .....	1
2.	HTML.....	2
2.1.	Razvoj HTML-a .....	3
2.2.	Struktura HTML dokumenta .....	4
2.3.	HTML komentari.....	5
3.	HTML5 .....	6
3.1.	Struktura HTML-a 5 .....	6
4.	CSS .....	7
4.1.	CSS verzije .....	8
5.	JavaScript .....	9
6.	Node.js.....	10
6.1.	NPM .....	10
7.	Ideja .....	11
8.	Problematika.....	12
9.	Proces izrade.....	13
10.	Kriteriji za uspjeh .....	14
11.	Ponašanje aplikacije iz perspektive korisnika .....	15
12.	Komponente koje bi aplikacija trebala imati .....	16
13.	Buduće nadogradnje .....	17
14.	Postavljanje projekta .....	18
14.1.	Osnove.....	18
14.2.	Vite .....	18
14.3.	Tailwind.....	19

15. Razvoj front-end dizajna aplikacije .....	20
15.1. Homepage .....	21
15.1.1. Kôd .....	22
Zaključak .....	45
Popis slika.....	46
Literatura .....	48
Prilozi .....	49

# 1. Uvod

Tema završnog rada je razvoj dizajna mobilne aplikacije koja omogućava da se preko skeniranja QR kôda može vidjeti ponuda objekta. Problem kod većine postojećih rješenja je ili nekvalitetan dizajn ili neispravnost rada aplikacije. Ova aplikacija riješiti će problem nedostajućih cjenika, kao i maksimalno ograničiti interakciju osoblja s gostima.

Svrha i cilj ovog rada bilo je primijeniti i prezentirati znanja stečena na fakultetu ponajviše iz kolegija „Web dizajn“ i „Programski alati 2“, no kod samog dizajna aplikacije bilo je potrebno primijeniti znanja iz kolegija „Grafički dizajn“ i „Dizajn korisničkog sučelja“. Za kodiranje korišten je uređivač Visual Studio Code.

Na početku završnog rada nalazit će se teorijski osvrt na tehnologije i jezike koji su korišteni prilikom izrade aplikacije, kao i detaljan opis i uvod u aplikaciju; od dizajna do njezine svrhe i budućih nadogradnji. Nakon teorijskog osvrta dati će se vizualni opis razvoja dizajna aplikacije, te dijelovi kôda za svaki opisani dio.

## 2. HTML

HTML ili Hypertext Markup Language je format podataka koji se koristi za kodiranje sadržaja i strukture web stranice. Drugim riječima, to je prezentacijski jezik za izradu web stranica. Sprema se u obične tekstualne datoteke s nastavkom .html, što znači da se HTML kôd može napisati i u običnom notepadu i spremiti kao .html datoteka i raditi će isto kao da je napisan u bilo kojem drugom programu namijenjenom za to (Notepad++, Visual Studio Code, NetBeans...), ali radi preglednosti i teže snalažljivosti nije preporučljivo pisati kôd u običnom notepadu.

Kada netko pristupi web stranici, HTML datoteka te web stranice se prenosi i dekodira u vizualnu sliku koju vidimo na ekranu. [1] HTML nije programski jezik jer njime ne možemo izvršavati nikakve zadatke poput operacije zbrajanja ili oduzimanja brojeva ili slično.

## 2.1. Razvoj HTML-a

HTML je nastao iz problematike dijeljenja radova i rezultata istraživanja kojom su se susretali znanstvenici u CERN-u (Europska organizacija za nuklearno istraživanje), što je 1989. navelo Tim Berners-Lee-a na razmišljanje o tome kako to riješiti. Do tada je svaki od znanstvenika imao svoje računalo, svoj način pisanja i program za pisanje kojeg je preferirao, stoga kada bi netko zatražio dokument svojeg kolege, trebao je ili naučiti koristiti drugo računalo ili program u kojem je kolega pisao ili je trebao pretvoriti tu datoteku kako bi odgovarala njegovom programu. Tim Berners Lee je smatrao da bi taj problem riješio hipertekstualni sistem, ali pošto su do tada svi bili prekompleksni za tako nešto, počeo je dizajnirati svoj jednostavni hipertekstualni jezik baziran na SGML-u (Standard Generalized Markup Language).

Na Božić, 1990., pokrenut je WorldWideWeb preglednik i server gdje su svi mogli objavljivati svoje dokumente u standardnom formatu kojeg svatko može pronaći i čitati preko preglednika. 1993. su Tim Berners Lee i Dae Raggett napisali prvu probnu verziju HTML-a koja se zvala Hypertext Markup Language Ver 1.0 i trebala je postati standard, ali nije uspjela zbog prevelikog rasta broja preglednika. Tako je HTML 1.0 ubrzo zamijenjen HTML-om 2.0 jer je i dalje postojala potreba za standardom.

1994. Tim Berners Lee je napustio CERN i organizirao W3C (World Wide Web Consortium koji se i danas bavi standardizacijom tehnologija korištenih na webu. W3C je pokušao standardizirati HTML svojom 3.0 inačicom koja je imala mogućnost tablica. HTML 4 je predstavljen u prosincu 1997. gdje se standard „čistio“ od suvišnih oznaka.

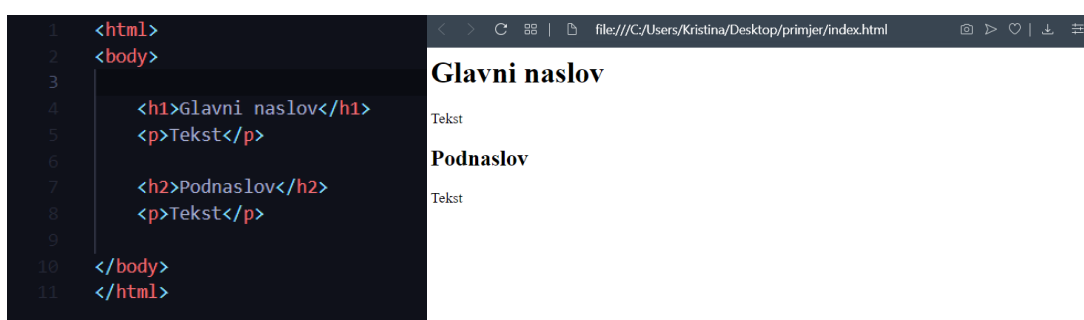
W3C je radio i na drugim standardima, ali i na novoj inačici HTML-a. Odlučili su da budućnost leži u XML-u (Extensible Markup Language) koji je bio dista sličan HTML-u. Ono po čemu se XHTML razlikuje od HTML-a je to što kod XHTML-a greške ne prolaze neprimjetno i to je nosilo stroža pravila i druga razika je nadogradivost u smislu mogućnosti dodavanja novih elemenata koje samo treba opisati u odvojenoj datoteci i povezati ju sa svojim glavnim dokumentom. 2000. je XHTML predstavljen kao standard. [2]

## 2.2. Struktura HTML dokumenta

Struktura dokumenta je vrlo bitna kako bi čitatelji mogli shvatiti poruku koju želimo prenijeti i kako bi se mogli lakše snalaziti po dokumentu. Struktura se sastoji većinom od naslova, podnaslova, nekakvog teksta i slika.

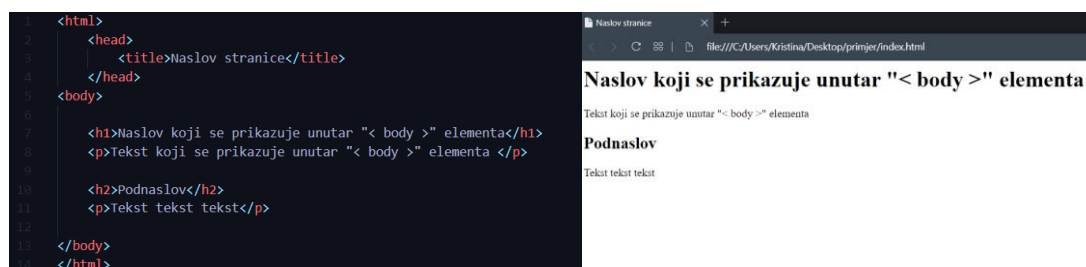
Kako bi opisali strukturu HTML dokumenta, koristimo HTML elemente koji se stavljaju unutar tagova (< element />). Glavni elementi su <html>, <head>, <body> i <title>.

Primjer ispod prikazuje najosnovniji HTML kôd i njegov rezultat.



Slika 1. Osnovni HTML kod i njegov prikaz u pregledniku

Sve što je unutar <body> elementa prikazuje se u web pregledniku. Unutar <head> elementa stavljaju se informacije o stranici i to se ne prikazuje u web pregledniku. Sadržaj koji se nalazi unutar <title> elementa prikazuje se na vrhu preglednika, na kartici.



Slika 2. Prikaz sadržaja unutar <body>, <head> i <title> elemenata

## 2.3. HTML komentari

HTML komentari dobra su praksa ponajviše radi preglednosti kôda i lakše snalažljivosti u kôdu. Oni pružaju dodatnu informaciju o nekoj liniji kôda i dobra su pomoć nekome drugome ako želi pogledati nečiji kôd. Oni nisu vidljivi korisnicima u pregledniku, mogu ih vidjeti samo osobe koje pregledavaju kôd. Također ako iz nekog razloga ne želimo u pregledniku prikazati neki dio kôda, umjesto da taj dio obrišemo, možemo ga samo staviti pod komentar u slučaju da nekada opet zatrebamo taj dio kôda i poželimo ga prikazati. [3]

Komentari se stavljaju unutar sljedećih znakova: `<!-- ovdje ide komentar -->`.

```
<h1>Lorem ipsum</h1>
<!-- ovdje slijedi paragraf -->
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Pellentesque quis metus sed risus imperdiet finibus. Phasellus
lobortis ornare elementum. Ut dolor nulla, accumsan finibus
eleifend quis, rutrum a quam.</p>
<!-- <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Pellentesque quis metus sed risus imperdiet finibus. Phasellus
lobortis ornare elementum. Ut dolor nulla, accumsan finibus
eleifend quis, rutrum a quam.</p> -->
```

## Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque quis metus sed risus imperdiet finibus. Phasellus lobortis ornare elementum. Ut dolor nulla, accumsan finibus eleifend quis, rutrum a quam.

Slika 3. HTML komentari



## 3. HTML5

HTML5 (Hypertext Markup Language 5) je najnovija verzija HTML verzije 1.0, te nastoji reflektirati potrebe sadašnjih i budućih web stranica. Nasljeđuje veliku većinu značajki od svojih prethodnika, što znači da veliki dio HTML-a 5 radi i u starijim i u novijim preglednicima. Službeno je postao standardom 28.10.2014. godine.

Neke od novouvedenih sintaksi su <video>, <audio> i <canvas>, ali je i integriran SVG (Scalable Vector Graphics) za vektorske grafike na webu te MathML za matematičke formule. HTML5 omogućuje stvaranje igara, izgradnju mobilnih aplikacija i još mnogo toga.  
[4]

### 3.1. Struktura HTML-a 5

- Razlika između strukture HTML-a 5 i HTML-a:

```
<!doctype html>
<html lang="hr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="style.css">
    <title>Moj portfolio</title>
  </head>
  <body>
    <script src="/js/scripts.js"></script>
  </body>
</html>
```

Slika 4. Struktura HTML-a 5

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Naslov</title>
5   </head>
6   <body>
7     <h1>Lorem ipsum</h1>
8     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
9       Duis eu viverra neque.
10      Vestibulum vel rhoncus arcu, ac condimentum neque.</p>
11   </body>
12 </html>
```

Slika 5. Struktura HTML-a

## 4. CSS

CSS (Cascading Style Sheets) je stilski jezik koji se koristi za definiranje stilova web stranica, uključujući njihov izgled, dizajn, varijacije prikaza različitih uređaja i veličine zaslona itd. CSS jezik omogućuje upravljanje stilovima fontova, boja teksta, pozadinskih slika, veličina zaslona, razmaka između paragrafa i još mnogo toga.

CSS jezik omogućava da se napisani kôd može ponovno koristiti kada je god potrebno. Isti list stilova može se koristiti za različite HTML stranice. Također se može definirati stil za svaki HTML element, te ga primijeniti na više web stranica. Više nije potrebno svaki put pisati attribute HTML tagova, već se napiše jedno CSS pravilo za tag koje se zatim primjeni na ostale koji se pojavljuju.

### World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

#### [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

#### [Help](#)

on the browser you are using

#### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,[X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )

#### [Technical](#)

Details of protocols, formats, program internals etc

#### [Bibliography](#)

Paper documentation on W3 and references.

#### [People](#)

A list of some people involved in the project.

#### [History](#)

A summary of the history of the project.

#### [How can I help ?](#)

If you would like to support the web..

#### [Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

*Slika 6. Prva web stranica, bez CSS-a*

## 4.1. CSS verzije

- CSS 1 – prva verzija CSS-a izašla je kao preporuka od strane W3C (World Wide Web Consortium) u prosincu 1996. godine. Opisuje jezik i jednostavne modele vizualnog oblikovanja HTML oznaka.
- CSS 2 – druga verzija postaje preporuka W3C u svibnju 1998 . godine, te je izgrađena na prethodnoj verziji. Sadrži podršku za stilove koji su specifični za medije, poput zvučnih uređaja, ispisa, pozicioniranja elemenata, tablica i fontova za preuzimanje.
- CSS 3 – treća verzija CSS-a postala je preporuka W3C u lipnju 1999. godine. Izgrađena je na prethodnoj verziji, te sadrži podijeljenu dokumentaciju poznatu kao moduli. Svaki modul ima novo proširenje koje je definirano u drugoj verziji CSS-a. CSS 3 moduli također imaju stare specifikacije i proširenja koja ima CSS 2, to uključuje selektore, pozadinu, obrube, vrijednosti slika, efekti teksta itd. [5]

## 5. JavaScript

JavaScript jezik 1995. predstavljen je kao način dodavanja programa na web stranice u Netscape Navigator pregledniku. Omogućio je moderne web aplikacije s kojima se može direktno komunicirati bez da se za svaku radnju stranica mora ponovno učitati. Pomoću JavaScripta je također moguće izvoditi različite oblike interaktivnosti i vještina. Kada se jezik počeo koristiti i u drugim web preglednicima, napisan je standardni dokument koji opisuje način na koji bi JavaScript jezik trebao raditi i to se naziva ECMAScript (European Computer Manufacturers Association) standard, prema Ecma International organizaciji koja je izvršila standardizaciju.

Bilo je nekoliko verzija JavaScript jezika. ECMAScript 3 bila je široko podržana verzija između 2000. i 2010. godine. U to vrijeme radilo se na verziji 4 koja je trebala imati brojna radikalna poboljšanja i proširenja jezika, no promjena široko korištenog jezika na tako radikalna način pokazala se politički teškom, te je rad na verziji 4 napušten 2008. godine, što je dovelo do verzije 5 koja je bila manje ambiciozna i napravljena su samo neka kontroverzna poboljšanja. Verzija 5 izašla je 2009. godine. 2015. godine izašla je verzija 6 u kojoj je bilo veliko ažuriranje koje je uključivalo neke ideje planirane za verziju 4. Od tada su svake godine nova, mala ažuriranja.

S obzirom na to da se JavaScript jezik svake godine razvija, preglednici to moraju stalno pratiti, što znači da stariji preglednici možda neće podržavati sve značajke. JavaScript jezik se osim na web preglednicima koristi i u nekim bazama podataka poput MongoDB i CouchDB za skriptiranje i upite. Pored ostalih platformi za programiranje na radnoj površini i poslužiteljima, Node.js pruža okruženje za programiranje JavaScript jezika izvan preglednika.

## 6. Node.js

Node.js je serverska JavaScript platforma i izvršava JavaScript kôd izvan web preglednika. Omogućuje jednostavnu izgradnju brzih mrežnih aplikacija. Jedan od problema sa sustavima za pisanje koji komuniciraju putem mreže je upravljanje unosom i izlazom, to jest čitanjem i upisivanjem podataka u mrežu i s mreže, te tvrdog diska. Premještanje podataka zahtijeva vrijeme, a pametno raspoređivanje može biti od velikog značaja pri brzini reagiranja sustava na korisnika ili mrežne zahtjeve. Često je kod takvih programa korisno asinkrono programiranje koje programu omogućuje istodobno slanje i primanje podataka sa i na više uređaja, bez sinkronizacije.

U početku je svrha Node.js sustava bila lakše i prikladnije asinkrono programiranje. To je jedan od rijetkih programskih jezika koji nema ugrađen način za unos i izlaz. Dakle, JavaScript se može uklopiti u Node-ov pristup unosu i izlazu bez da završi s dva nedosljedna sučelja.

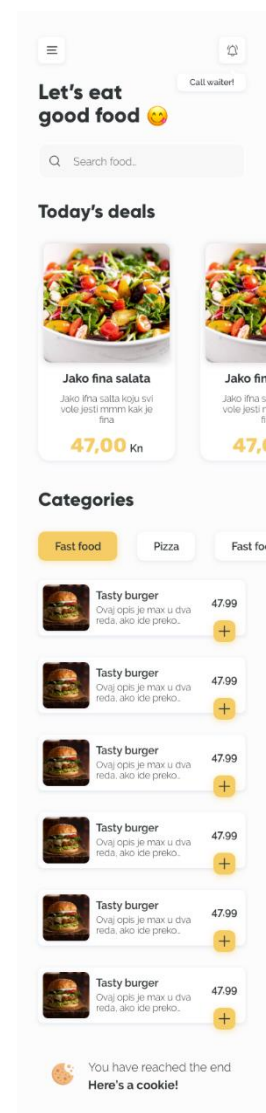
### 6.1. NPM

NPM je online usluga na koju se mogu preuzimati i učitavati paketi, ali je također i program (u paketu s Node.js) koji pomaže instalirati te pakete i upravljanje njima. Nakon što se Node instalira na računalo, dobije se i npm naredba koja se može koristiti za interakciju s ovim spremištem. NPM se uglavnom koristi za preuzimanje paketa. [6]

## 7. Ideja

Ideja je bila izraditi mobilnu aplikaciju za kafiće i restorane koja omogućava gostima da jednostavno skeniraju QR kôd na stolu i putem web preglednika vide ponudu objekta.

Ugostitelji samostalno mogu kreirati svoje cjenike i printati QR kodove, potpuno besplatno.



Slika 7. Ilustracija ideje

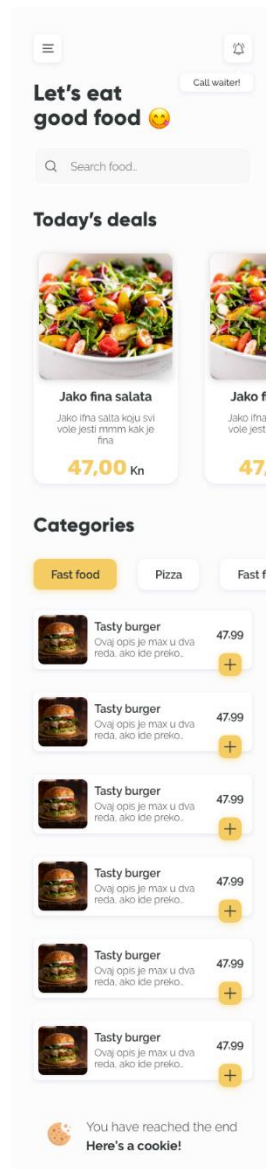
## 8. Problematika

Potrebno je čisto, elegantno rješenje za restorane, a mnoga postojeća rješenja su nekvalitetnog dizajna i puna „bugova“ što se tiče pozadinskih slika i ostalih neestetskih izbora dizajna.

Budući da su sukladno s propisima o pandemiji fizički cjenici u restoranima i kafićima zabranjeni, ova aplikacije ne samo da rješava problem nedostajućih cjenika, već i maksimalno ograničava količinu interakcije s osobljem. Uz naručivanje putem aplikacije, osoblje ne treba dolaziti do svakog stola kako bi zaprimilo narudžbe, već sve vide automatski preko aplikacije i na njima je samo da donesu narudžbu.

## 9. Proces izrade

Izrađivati će se prototip koji će budući kupci moći koristiti kao demo verziju. Proces izrade podijeljen je u tri faze: dizajn, razvoj front-end dizajna i razvoj funkcionalnosti front-end-a.



Slika 8. Dizajn aplikacije

Kako bi se ubrzao proces razvoja aplikacije, razvoj front-end dizajna odvojiti će se od razvoja front-end funkcionalnosti. Programer koji će biti zaslužan za razvoj funkcionalnosti primiti će html/css/js gotovu aplikaciju te će ju morati povezati s API uslugom i učiniti onaj zadnji korak, a to je da aplikacija zapravo radi.



## 10. Kriteriji za uspjeh

Kriteriji koji moraju biti zadovoljeni kako bi se ovaj projekt smatrao uspješnim su sljedeći:

- Gotova aplikacija trebala bi biti jednostavna za upotrebu i dizajn postojan
- Svaka korisnikova interakcija treba imati povratnu informaciju u obliku animacije
- Sve datoteke i dokumentacija korištena za ovaj projekt trebaju biti organizirane i smještene na jedno mjesto kako bi svima koji rade na projektu bile dostupne

## **11. Ponašanje aplikacije iz perspektive korisnika**

Ovaj projekt namijenjen je upotrebi unutar restorana i kafića, te olakšanom naručivanju. Svaki stol biti će opremljen jedinstvenim QR kodom koji odgovara stolu u POS (Point Of Sale) softveru.

Gosti mogu listati, pretraživati i pročitati više o proizvodima i hrani koja se poslužuje. Također mogu dodavati i uklanjati narudžbu, kao i plaćati. Svaki proizvod ima niz svojstava koja se mogu primijeniti; slike, kratki i dugi opisi, cijene, veličine, količine, alergeni itd.

## 12. Komponente koje bi aplikacija trebala imati

- Popis proizvoda
- Pretraživanje proizvoda
- Istaknuti proizvodi
- Pozovi konobara
- Košarica za kupovinu
- Opis proizvoda
- Povezani proizvodi
- Višejezičnost
- O nama i kontakt stranica

## 13. Buduće nadogradnje

- Plaćanje kreditnom karticom
- Alergeni
- Status narudžbe
- Veličine proizvoda (npr. Pizza – mala, srednja, velika...)
- Povezani proizvodi koji se najviše prodaju (Pod kategorijom „Pizza“, npr. pica i prilozi)
- Prilagodljive narudžbe (dodaci za pizzu, sadržaj burgera)
- Lojalnost
- Korisnički računi
- Naručivanje unutar i izvan restorana i kafića
- Podrška za pristup dizajnu

## 14. Postavljanje projekta

Prije samog početka kodiranja potrebno je napraviti par osnovnih koraka i postaviti sve što je potrebno kako bi se moglo početi s kodiranjem aplikacije. U nastavku slijedi kronološki slijed koraka.

### 14.1. Osnove

Instalirati Visual Studio Code i preferirane dodatke (plugin). Preporučljivo *prettier* za formatiranje i *emmet* kako bi se stvari ubrzale.

Zatim je potrebno instalirati node.js kako bi mogli koristiti npm kao paket.

Nakon što je node.js instaliran, na računalu se kreira radna mapa u kojoj će se nalaziti projekt. Mape je najbolje kreirati sljedećim redoslijedom: Projekt -> Aplikacija -> Aplikacija za restorane i kafiće

### 14.2. Vite

Vite je alat za izradu čiji je cilj pružiti brže iskustvo u razvoju modernih web projekata.

- Za početak potrebno je otvoriti „terminal“ u Visual Studio Code-u unutar mape koja je kreirana u koraku prije, te provjeriti da li je npm instaliran, a to se provjerava sa komandom: `npm -v`
- Vite projekt postavlja se sa komandom: `npm init @vitejs/app`
- Dalje je potrebno slijediti instrukcije i odabrati *vanilla* project, ime se postavlja po želji
- Sa komandom `npm run dev` provjerava se da li Vite radi
- CTRL (ili ALT) + klik na IP adresu za otvaranje projekta u pregledniku

## 14.3.Tailwind

- Sljedeći korak je instaliranje Tailwinda i svih njegovih komponenti sa komandama:  

```
npm install -D tailwindcss@latest postcss@latest autoprefixer@latest --save-dev  
  
npx tailwindcss init -p
```
- Iz datoteka *main.js* i *style.css* briše se postojeći sadržaj i u *style.css* dodaje se sljedeće:  

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```
- Obično bi bilo potrebno pokrenuti prilagođenu naredbu unutar datoteke *package.json* za kako bi se kompilirao CSS ("`build:css`": "`tailwind build style.css -o dist/style.css`"), no sa Vite alatom to je obavljeno automatski

Ovim korakom je završeno postavljanje svega što je potrebno za početak kodiranja, te sada slijedi kodiranje aplikacije.

## 15. Razvoj front-end dizajna aplikacije

U daljnjem tekstu opisivati će se po koracima kako je razvijen dizajn aplikacije. Aplikacija sadržava sljedeće stranice:

- Homepage
- Product page
- Order page
- Search
- About us
- Swipe to confirm
- Order in progress

Od sedam navedenih stranica, radi opsega rada opisivat će se samo jedna stranica – početna (Homepage). Ista je također i glavna stranica, a sve ostale napravljene su prema njoj uz potrebne izmjene.

## 15.1. Homepage

Početna stranica sadržavat će sljedeće komponente:

**Padajući izbornik sa jezicima** gdje će korisnik moći odabrati na kojem jeziku želi da mu se aplikacija prikazuje.

**Call waiter** (pozovi konobara) **gumb** otvara komponentu „call waiter“ koja se otvara prilikom učitavanja stranice i nakon pet sekundi nestaje. Pojavljuje se i pritiskom na gumb, te nestaje pritiskom bilo gdje na stranicu.

**Glavni naslov** „Let's eat good food“ trebao bi biti učitani pomoću JavaScripta.

**Okvir za pretraživanje** (Search bar) je div na koji se može kliknuti te vodi na drugu stranicu koja je stvarni okvir za pretraživanje, ali u fokusu.

**Today's deals** prikazuje proizvode u carouselu sa najviše 8 proizvoda.

**Footer** (podnožje) sadrži tri gumba; ono što je istaknuto označava da je aktivno, kućica vodi do naslovne stranice (a to je ova stranica), košarica vodi do stranice sa košaricom za kupovinu, a knjiga vodi na About us stranicu.

**Categories** (kategorije) su navedene u carouselu koji se može listati horizontalno i odabrana kategorija je istaknuta. Klik na neku od kategorija trebao bi izazvati animaciju kako bi označio promjenu. Unutar kategorija je prikaz proizvoda koji spadaju pod tu kategoriju. Cijeli div sa proizvodom je „klikajući“ i vodi na „Product page“, osim ako se klikne na gumb sa ikonom plusa koji korisniku daje povratnu informaciju u obliku animacije i automatski dodaje jedan proizvod u košaricu.



### 15.1.1. Kôd

Unutar `<head>` tagova nalazi se sve ono što se postepeno stavlja u kôd , ovisno o tome što je u tom trenutku bilo potrebno. Ovdje se nalaze biblioteke, css stilovi, javascript i ostalo što je potrebno za rad i izgled aplikacije. Za sada je stranica prazna zato što unutar `<body>` tagova nema nikakvog sadržaja.

Odmah na početku valja napomenuti da je dobra praksa stavljanje komentara u kôd radi preglednosti, ali i lakše snalažljivosti, kako onog koji kodira tako i onoga koji će pregledavati ili izmjenjivati kôd .

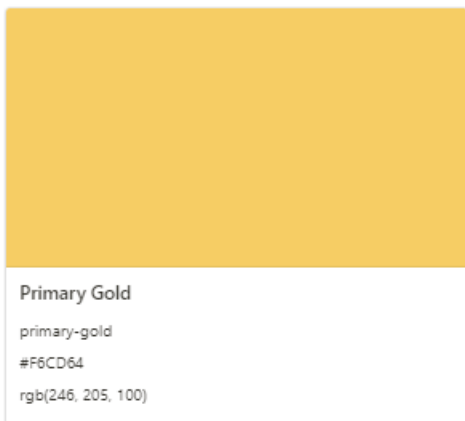
```
<!DOCTYPE html>
<html lang="en" >
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="favicon.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="/style.css" />
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"
      integrity="sha512-
894YE6QWD5I59HgZOGReFYm4dnWc1Qt5NtYSaNCOP+u1T9qYdvdihz0PPSiiqn+/+3e7Jo4EaG7Tu
bfWGUrMQ=="
      crossorigin="anonymous"
    ></script>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/animejs/3.2.1/anime.min.js"
      integrity="sha512-
z40Uqw38qNLpn1libAN9BsoDx6nbNFio51A6CuTp9N1k83b89hgyCVq+N5FdBJptINztxN1Z3SaKSK
US5UP60Q=="
      crossorigin="anonymous"
    ></script>
    <script src="https://unpkg.com/@popperjs/core@2"></script>
    <script src="https://unpkg.com/tippy.js@6"></script>
    <script type="module" src="/main.js"></script>
    <script src="/dropdown.js"></script>
    <title>Vite App</title>
  </head>
```

Slika 9. Prikaz kôda unutar `<head>` tagova

Uz dizajn, dobivene su i palete boja koje su se morale koristiti prilikom razvoja dizajna aplikacije, kao i obitelj fontova koji će se koristiti - Gilroy, Raleway, Rubik.

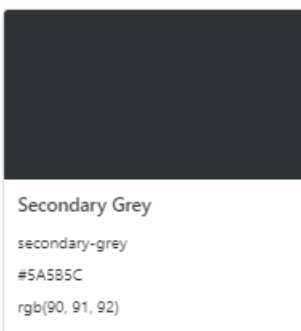
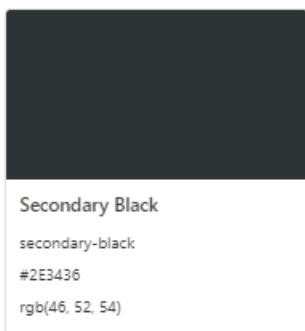
### Primary Palette

---



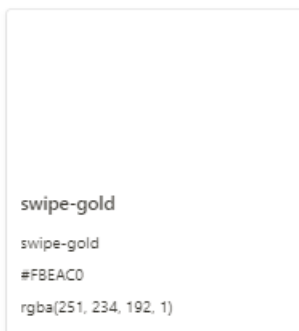
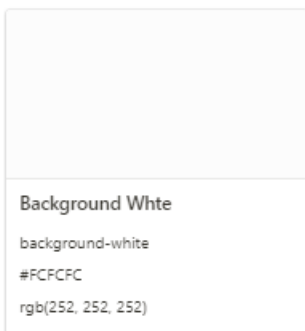
### Secondary Palette

---



### Other

---



Slika 10. Paleta boja

S obzirom na to da se unaprijed znalo, na osnovu dizajna, koje će se boje i fontovi koristiti, isti su odmah definirani unutar *tailwind.config.js* datoteke. To je Tailwind-ova datoteka u kojoj se mogu definirati bilo kakve prilagodbe. Svaki dio konfiguracijske datoteke je opcionalan, tako da je potrebno samo navesti što se želi promijeniti. Također neki elementi sadržavaju sjene kori sjene pa su i one odmah definirane.

```
module.exports = {
  purge: [],
  darkMode: false, // or 'media' or 'class'
  theme: {
    extend: {
      fontFamily: {
        gilroybold: ["Gilroy-Bold", "sans-serif"],
        gilroyheavy: ["Gilroy-Heavy", "sans-serif"],
        gilroylight: ["Gilroy-Light", "sans-serif"],
        raleway: ["Raleway", "sans-serif"],
        rubik: ["Rubik", "sans-serif"],
      },
      zIndex: {
        "-10": "-10",
      },
      colors: {
        "background-white": "#FCFCFC",
        "primary-gold": "#F6CD64",
        "secondary-black": "#2E3436",
        "secondary-grey": "#5A5B5C",
        "shadow-color": "#737DAF",
        "search-grey": "#F8F8F8",
        transparent: "rgba(0, 0, 0, 0.30)",
        "swipe-gold": "#FBEAC0",
      },
      boxShadow: {
        customShadow: "0px 2px 6px rgba(115, 125, 175, 0.14)",
        customShadow2: "0px 2px 7px rgba(115, 125, 175, 0.25)",
        goldShadow: "0px 4px 10px rgba(246, 205, 100, 0.45)",
        cardshadow: "0px 0px 3px rgba(115, 125, 175, 0.2)",
      },
    },
  },
  variants: {
    extend: {},
  },
  plugins: [],
};
```

Slika 11. *tailwind.config.js* datoteka sa bojama, fontovima i sjenama

Pošto Google Fontovi ne podržavaju fontove koji će se koristiti, postavljeni su ručno. Postoji više metoda koje se mogu koristiti prilikom postavljanja fontova, no ovdje je korištena @import metoda. Fontovi su učitani na samom vrhu *style.css* datoteke.

```
@import url("http://fonts.cdnfonts.com/css/gilroy-  
bold?styles=20876,20877,20878");  
@import url("https://fonts.googleapis.com/css2?family=Raleway:wght@100;200;300  
;400;500;600;700;800;900&display=swap");  
@import url("https://fonts.googleapis.com/css2?family=Rubik:wght@300;400;500;6  
00;700;800;900&display=swap");  
  
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

Slika 12. *style.css* datoteka sa fontovima

## BODY

Nakon toga tijelu stranice zadana je bijela boja pozadine, te div-u koji sadži jezike, zvono, naslove i okvir za pretraživanje, definirana je maksimalna širina „md“ što znači da će se aplikacija na svakoj veličini ekrana prikazivati u veličini za mobilne uređaje.

## GLAVNI DIO

S obzirom na to da se Tailwind-ovi kontejneri ne centriraju sami automatski i nemaju nikakav ugrađeni horizontalni odmak (u daljnjem tekstu: padding), zadano je da su horizontalne margine automatske. Vertikalni razmak između kontejnera je 5px i padding također 5px.

## JEZICI

Za izbornik korišten je `<select>` element koji se koristi za kreiranje padajućih izbornika. Atribut „name“ potreban je za pozivanje podataka forme, ako se atribut *name* izostavi, podaci s padajućeg izbornika neće biti poslani.

Nadalje su dodane potrebne klase za oblikovanje, poravnanje i sami izgled izbornika.

## ZVONO

Zvonu je definiran id „*myButton*“ koji ga povezuje s JavaScriptom i na taj način omogućuje interakciju s korisnikom pritiskom na gumb koji ima malu kratku animaciju kada se pritisne.

Animacija je takva da se prilikom učitavanja stranice automatski pojavi okvir sa porukom „*Call waiter!*“ i nakon dvije sekunde nestane. No također se ista animacija prikazuje i pritiskom na gumb zvona i nestaje prilikom klikanja bilo gdje na stranicu.

Za okvir s porukom korišten je Tippy.js. Tippy nudi već gotove elemente poput okvira koji iskaču, izbornika i padajućih izbornika. Nudi mnogo različitih stilova, animacija, pozicioniranja, tranzicija i ostalo.

Za okvir s porukom korištena je svijetla temu (bijeli okvir sa crnim tekstom), animacija '*scale-extreme*', vrsta strelice je zaobljena, okvir je pozicioniran ispod zvona i trajanje mu je 0,2 sekunde prilikom pojavljivanja i isto toliko dok nestaje.

Sve njegove komponente moraju se učitati na samom vrhu koda unutar '*main.js*' datoteke.

```
import tippy from "tippy.js";
import "tippy.js/dist/tippy.css";
import "tippy.js/themes/light.css";
import "tippy.js/animations/scale.css";
import "tippy.js/animations/scale-subtle.css";
import "tippy.js/animations/scale-extreme.css";
//Call waiter
const p = document.querySelector("#myButton");
tippy(p, {
  placement: "bottom",
  animation: "scale-extreme",
  arrowType: "round",
  theme: "light",
  content: "Call waiter!",
  arrow: true,
  duration: [200, 200],
});
p._tippy.show();

setTimeout(() => {
  p._tippy.hide();
}, 2000);
});
```

Slika 13. *main.js* datoteka sa Tippy elementima za okvir s porukom

## GLAVNI NASLOV

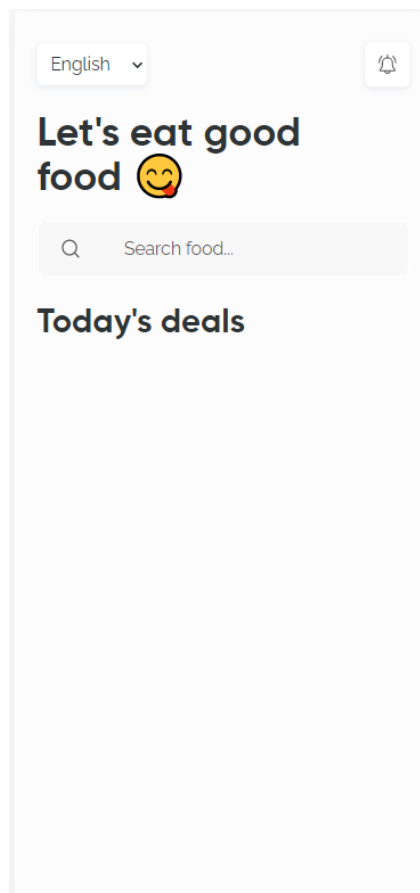
Glavni naslov postavljen je unutar <h1> taga koji i služi za označavanje glavnog i najbitnijeg naslova.

## OKVIR ZA PRETRAŽIVANJE

Okvir za pretraživanje zapravo je „klikajući“ div koji otvara novu stranicu u kojoj se zatim može pretraživati. Kako bi se okvir za pretraživanje isticao u odnosu na bijelu pozadinu, definirana mu je siva boja pozadine i bijeli, tanki obrub.

## TODAY'S DEALS

Sljedeći je naslov „Today's deals“ i taj će dio prikazivati današnju ponudu jela prikazanu preko listajućih kartica sa slikom jela, nazivom, kratkim opisom te cijenom.



Slika 14. Komponente za jezike, zvono, naslovi, okvir za pretraživanje

```

<body class="bg-background-white">
  <div class="max-w-md mx-auto space-y-5 px-5">
    <!-- TopBar -->
    <!--Languages-->
    <div class="flex justify-between items-center mt-7">
      <div>
        <select
          name="foo"
          id="my-select"
          class="p-2 flex justify-center font-raleway text-base text-
secondary-black items-center rounded-md bg-white shadow-customShadow"
        >
          <option value="" data-placeholder>English</option>
          <option value="1">Hrvatski</option>
          <option value="2">Espanol</option>
          <option value="3">Deutsch</option>
        </select>
      </div>
      <!--Bell-->

      <div
        id="myButton"
        class="button baton w-10 h-10 flex justify-center items-
center rounded-md bg-white shadow-customShadow"
      >
        
      </div>
    </div>
    <!---END TopBar-->

    <!---Main title-->
    <div
      class="flex flex-wrap w-9/12 font-gilroybold text-4xl text-secondary-
black"
    >
      <h1>Let's eat good food 🍷</h1>
    </div>

    <!---Search bar-->
    <div onclick="location.href='#'">
      <div
        class="relative pl-5 my-2 p-3 flex flex-row space-x-10 items-
center bg-search-grey rounded-lg border-2 border-white"
      >
        <div>
          
        </div>
        <div>

```

Slika 15. Prikaz kôda za naslov, zvonu, jezike i okvir za pretraživanje

## KARTICE S PONUDOM JELA

Pod kategorijom „Today's deals“ nalaze se kartice koje prikazuju današnju ponudu jela. Za početak je postavljen jedan div koji će u sebi sadržavati sve ostale divove sa karticama. U glavnom divu upotrijebljen je *flex flex-nowrap* kako bi se spriječilo da se kartice koje prelaze granicu kontejnera premjeste u novi red, pošto je ovdje cilj da sve kartice budu u jednom redu i da se mogu listati horizontalno. Zatim je ubačen *overflow* koji kontrolira na koji način element rukuje sadržajem koji je prevelik za kontejner. U ovom slučaju odabran je *overflow-x-auto* da bi se omogućilo horizontalno pomicanje kartica. Sa umetanjem *space-x-10* određen je horizontalni razmak od 10px između svake kartice.

Kako bi slika „izlazila“ izvan područja div-a, stavljena je negativna gornja margina *-mt-10*. Da bi se sačuvalo izgled i kvaliteta slike, korišten je *object-cover* i time spriječeno deformiranje slike i gubitak kvalitete.

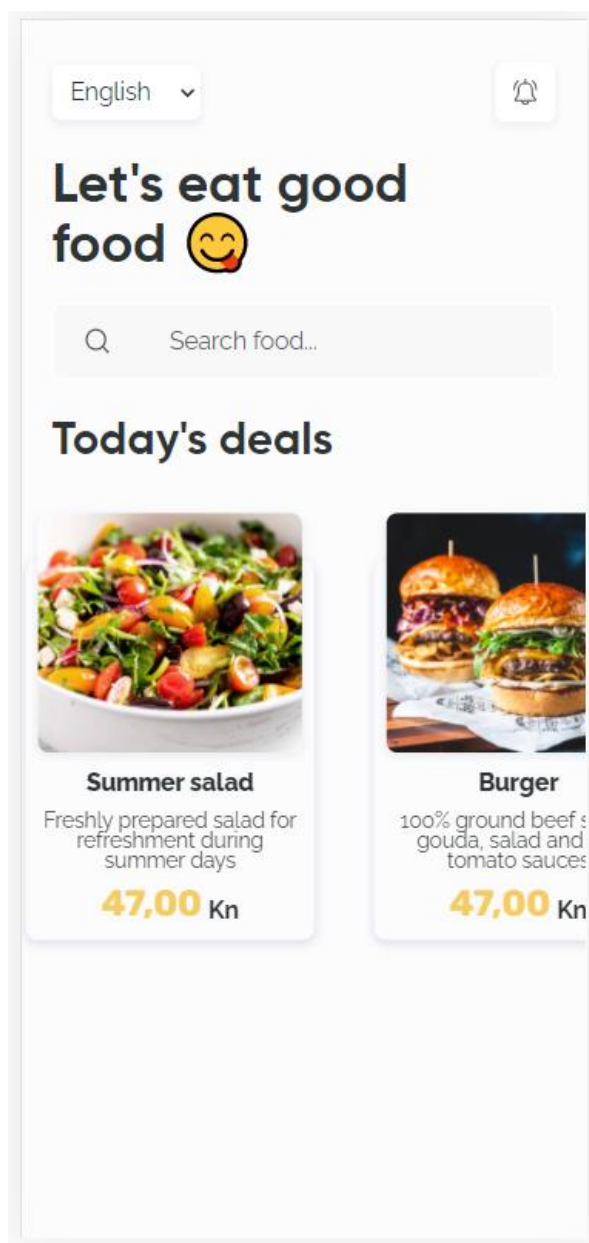
Naziv jela, opis i cijena idu u posebne divove zbog lakšeg poravnanja, sve mora biti jedno ispod drugog. Sav sadržaj je horizontalno centriran pomoću *flex items-center* i pozicioniran vertikalno sa *flex-col* koji služi za vertikalno pozicioniranje sadržaja. Vertikalni razmak između sadržaja je 1.5px, što postižemo sa *space-y-1.5*.

Dio koji prikazuje cijenu također je omotan u poseban div i podijeljen je na dva paragrafa (<p>) zbog različitog oblikovanja. Prvi paragraf sadržava iznos cijene i nešto je veći od sadržaja u drugom paragrafu. Boja je žuta i font podebljan, dok na sadržaj drugog paragrafa (Kn) nije primijenjeno ništa osim gornje margine koja iznosi 2px kako tekst ne bi bio u ravnini iznosa cijene nego nešto malo ispod toga.

Ovim je završena prva kartica sa prikazom jela i još je potrebno napraviti dvije, no pošto se rade na isti način kao i prva onda je najjednostavnije samo kopirati prvu i zalijepiti kôd još dva puta i zatim samo promijeniti sadržaj (naziv jela, sliku, opis i cijenu).



Sada to izgleda ovako:



*Slika 16. Kartice s ponudom jela*

```

<!--Food card-->

<div class="flex flex-nowrap overflow-x-auto space-x-10 px-5 pt-16 pb-5">
  <!-- ... -->
  <div>
    <div
      id="1"
      class="baton p-2 top-9 border-background-white w-48 rounded-lg shadow-
customShadow2"
    >
      
      <div
        class="flex items-center w-full flex-col font-raleway space-y-
1.5 text-secondary-black"
      >
        <p class="font-extrabold text-center">Summer salad</p>
        <p class="text-center text-sm leading-none">
          Freshly prepared salad for refreshment during summer days
        </p>
        <div class="flex flex-wrap space-x-1">
          <p class="text-2xl text-primary-gold font-rubik font-bold">
            47,00
          </p>
          <p class="mt-2"><b>Kn</b></p>
        </div>
      </div>
    </div>
  </div>
</div>

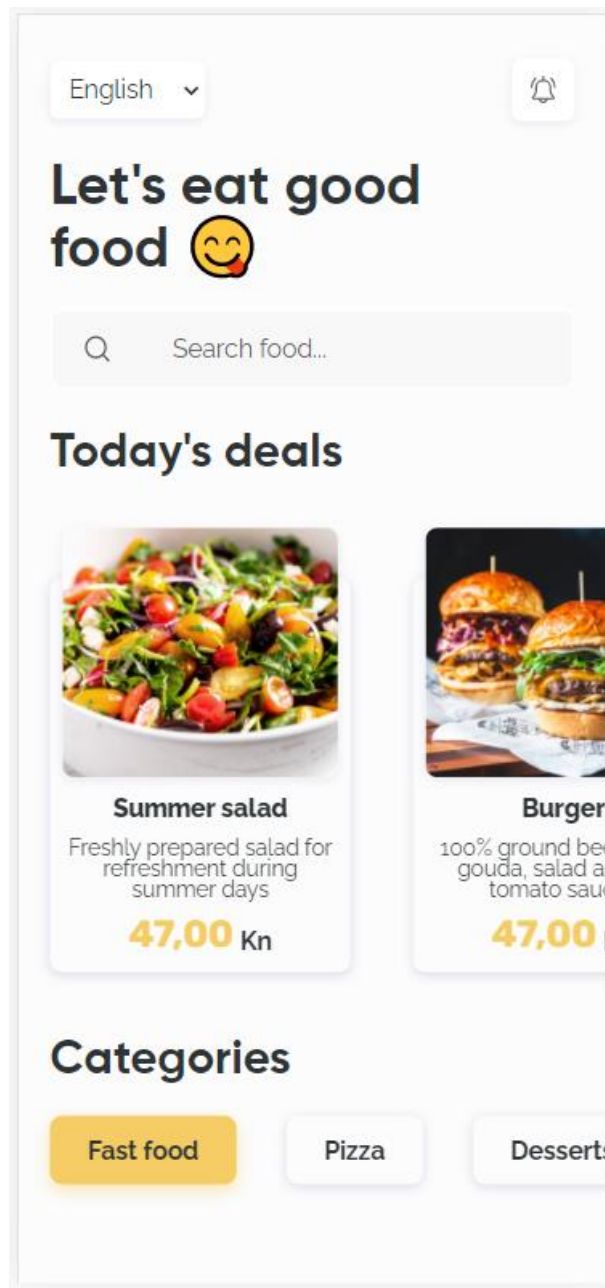
```

*Slika 17. Prikaz kôda za kartice s ponudom jela*

## KATEGORIJE

Pod stavkom Kategorije (Categories) nalaze se kategorije hrane. Ovdje su to 'Fast food', 'Pizza', 'Desserts'. Klikom na pojedinu kategoriju prikazuje se popis jela koja spadaju pod istu. Kategorija na kojoj se korisnik trenutno nalazi istaknuta je žutom bojom, klikom na drugu kategoriju ističe se ona koja je odabrana.

Popis kategorija također je moguće listati horizontalno kao što je moguće i sa karticama hrane. Potrebno je dobiti sljedeće:



Slika 18. Kategorije

Sam naslov nalazi se u posebnom div-u kojem je maksimalna širina zaslon mobilnog uređaja. Horizontalna margina postavljena je na automatski, vertikalni razmak 5px i unutarnji horizontalni odmak (padding) također je 5px. Stil naslova identičan je stilu prošloga naslova, stoga je moguće samo kopirati kôd za stil koji je već primijenjen.

## KARTICE S NAZIVOM KATEGORIJE

Sve tri kartice omotane su u jedan, glavni div kojemu sam stavila klasu *flex-nowrap* kako bi kartice bile u jednom redu unatoč tome što prelaze granice kontejnera, na taj način kartice koje prelaze granicu ne budu se premjestile u novi red, što će omogućiti horizontalno listanje kartica dodavanjem još *overflow-x-auto*. Razmak između svakog div-a sa karticom je 8px, unutarnji odmak (padding) 5px i širina je auto; to znači da preglednik sam izračuna i odabere širinu elementa.

Prva kategorija je Fast food i ona je istaknuta žutom bojom, što označava da se korisnik trenutno nalazi na istoj. Kategorija je označena tako što je div-u dodijeljena predefinirana žuta pozadinska boja pomoću klase *bg-primary-gold*. Sjena je zlatne boje, div-u je zaobljenje postavljeno na 'lg' (large). Korištena je klasa *whitespace-nowrap* da bi se spriječilo umotavanje teksta u element. Na taj način će novi redovi i razmaci biti sažeti. Unutarnji horizontalni odmak div-a je 6px, a vertikalni 2.5px.

Druge dvije kategorije (Pizza i Desserts) iste su kao i prva, stoga je kôd prve kategorije kopiran i zalijepljen dva puta, samo je iz druge dvije uklonjena klasa *bg-primary-gold* pošto ta klasa označava da je neka kategorija odabrana, a ne mogu biti sve tri kategorije odabrane u isto vrijeme.

```

<!--Subtitle 2-->
  <div class="max-w-md mx-auto space-y-5 px-5">
    <div
      class="flex flex-wrap w-9/12 font-gilroybold text-3xl text-
secondary-black"
    >
      <h1>Categories</h1>
    </div>
  </div>

<!--Category card 1-->
<div class="flex flex-nowrap overflow-x-auto space-x-8 p-5 w-auto">
  <div
    id="c1"
    class="baton flex items-center w-auto whitespace-nowrap font-
raleway text-secondary-black px-6 py-2.5 bg-primary-gold shadow-
goldShadow rounded-lg"
  >
    <p class="font-bold text-center text-base">Fast food</p>
  </div>

  <!--Category card 2-->
  <div
    id="c2"
    class="baton flex items-center w-auto whitespace-nowrap font-
raleway text-secondary-black px-6 py-2.5 shadow-customShadow2 rounded-lg"
  >
    <p class="font-bold text-center text-base">Pizza</p>
  </div>

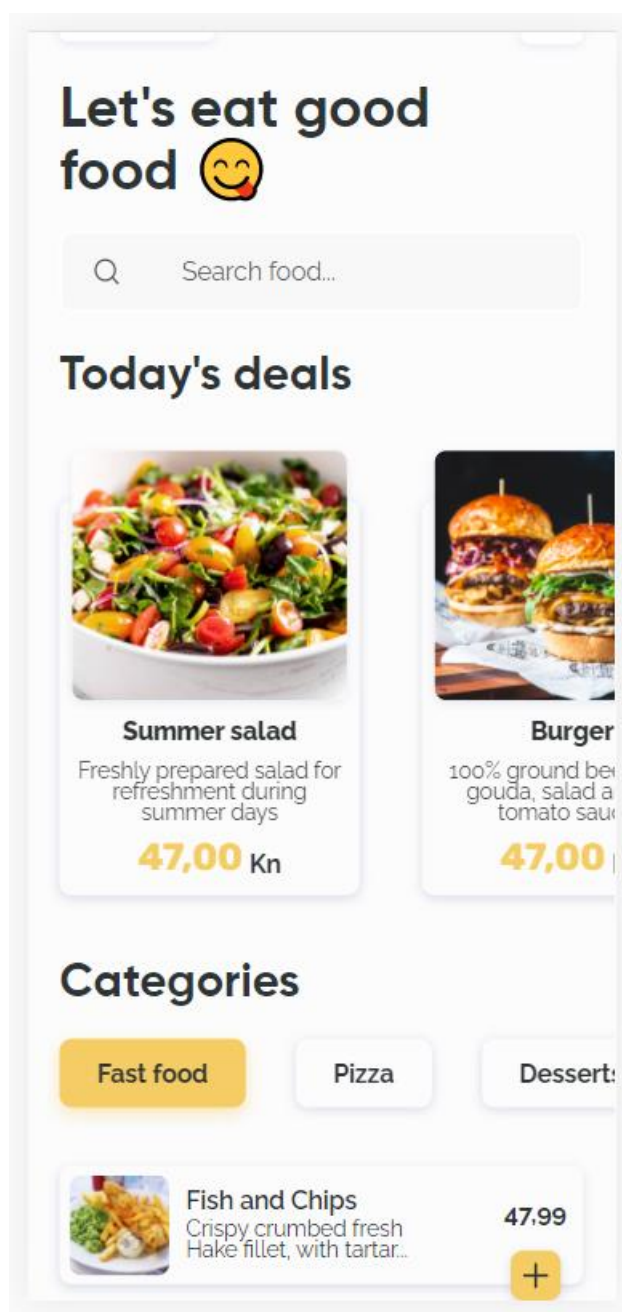
  <!--Category card 3-->
  <div
    id="c3"
    class="baton flex items-center w-auto whitespace-nowrap font-
raleway text-secondary-black px-6 py-2.5 shadow-customShadow2 rounded-lg"
  >
    <p class="font-bold text-center text-base">Desserts</p>
  </div>
</div>

```

Slika 19. Kôd za kartice kategorija

## POPIS JELA IZ ODABRANE KATEGORIJE

Sljedeći je na redu popis jela koja spadaju pod odabranu kategoriju. U ovom slučaju odabrana kategorija je 'Fast food' pa će se na popisu nalaziti jela koja su u ponudi, a spadaju pod brzu hranu. Prvo treba napraviti prvu karticu s popisa jela:



Slika 20. Popis jela iz odabrane kategorije

Kartica sadrži sliku jela, naziv i opis, cijenu i gumbić sa plusićem koji služi za dodavanje količine porcija.

Sljedeće kartice mogu se složiti na mnogo načina, no optimalan izbor u ovom slučaju bio bi korištenje rešetaka (grid-ova). Najprije su svi div-ovi umotani u glavni div kojemu je maksimalna širina zaslon mobilnog uređaja, margine su automatske, vertikalni razmak između svake kartice 8px i unutarnje margine glavnog div-a su 5px.

### **Prva kartica:**

Sastoji se od jednog div-a koji omotava ostale, zatim je podijeljeno u još tri div-a od kojih neki u sebi sadržavaju još div ili dva.

Prvi div koji omotava ostale određuje da se radi o rešetkama (grid-ovima) i govori da će se rešetka dijeliti na 4 stupca. To je određeno klasom *grid grid-cols-4*. Kartica ima oko sebe sjenu, unutarnji odmak sa svake strane je 1.5px. Zaobljena je i zaobljenje je veličine 'lg' (large).

Sada su unutar tog div-a definirana još tri div-a kojima su definirani stilovi i sadržaj. Prvom od ta tri div-a određen je raspon tako što mu je postavljena klasa *col-span-1*, što znači da taj div zauzima samo jedan stupac, i to prvi. U prvom stupcu nalazi se slika jela, ona je zaobljena i ima fiksnu visinu i širinu od 16px, kako se ne bi nepravilno rastegnula ili izrezala koristila se klasa *object-cover* da bude pravilno smještena unutar okvira.

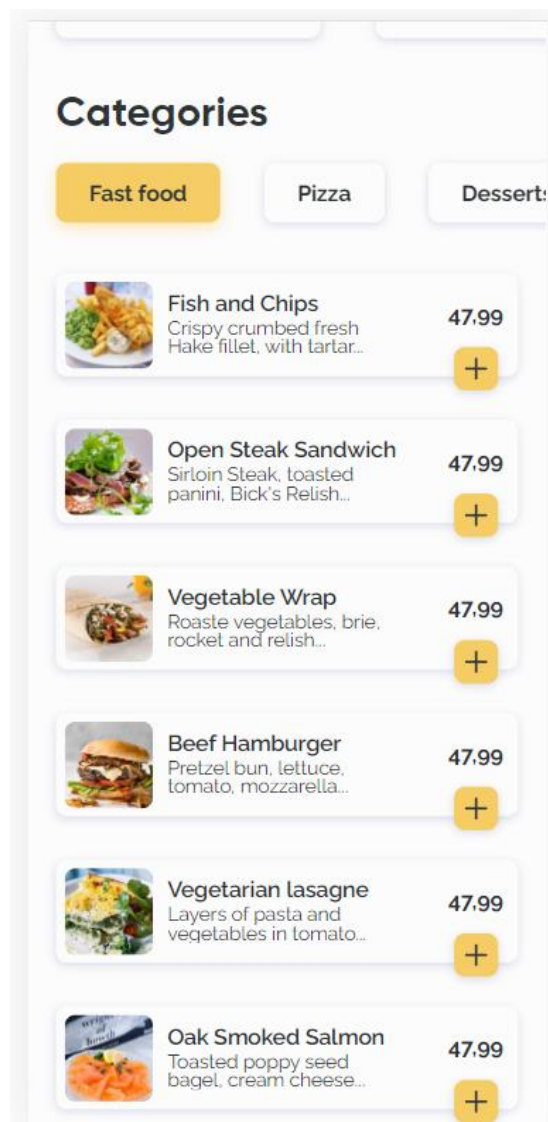
Sljedećem (drugom) div-u raspon je 2 stupca, znači da se postavlja klasa *col-span-2* i sada se taj div proteže kroz drugi i treći stupac. Negativnom lijevom marginom od 1.5px (*-ml-1.5*) povučen je div u lijevu stranu kako bi tekst bio bliže slici, iz razloga da bude vizualno ljepše. U taj div smješteni su naziv i opis jela od kojih je svaki dio u svom paragrafu. Paragrafu sa nazivom je font polu-podebljan, veličina teksta bazna, font *raleway* i poravnanje teksta lijevo. U drugom paragrafu sa opisom jela veličina teksta je postavljena na 'sm' (small), font također *raleway*, poravnanje teksta lijevo, maksimalna visina paragrafa je 7px, sama linija teksta postavljena je na 'none' (*leading-none*), te je korišten *overflow-hidden* kako tekst ne bi prelazio granicu paragrafa.

Raspon trećem div-u je 1 stupac i to je zadnji, četvrti od četiri stupaca koji su zadani u prvom, glavnom div-u. Njegova maksimalna visina je 5px i on je podijeljen na još dva poddiv-a. Prvi poddiv je cijena jela kojoj je poravnanje desno, gornja margina 3px, lijeva

marginu 9px, font *raleway* i podebljan. Drugi poddiv je gumb koji služi za dodavanje narudžbe i pritiskom na njega pojavljuje se animacija koja prikazuje „+1“, svakim pritiskom.

Div sadržava samo ikonu plusa. Širina mu je fiksna: 8px, gornja margina 3px, lijeva margina 10px, font je *raleway*, unutarnji razmak sa svake strane je 2px. Boja pozadine je predefinirana zlatna boja, gumb ima zaobljenje veličine 'lg' i sjenu oko sebe koja je definirana na početku kodiranja.

Nakon što je završena prva kartica, kôd za ostalih pet samo je kopiran i zalijepljen pet puta, te su zatim promijenjene slike, nazivi i opisi nakon čega je dobiveno sljedeće:



Slika 21. Kompletni popis jela iz odabrane kategorije



```

<!--Food List START-->
  <div class="max-w-md mx-auto space-y-8 px-5">
    <!--Card 1-->
    <div
      class="grid grid-cols-4 border-background-white shadow-
customShadow2 p-1.5 rounded-lg text-secondary-black"
    >
      <div class="col-span-1">
        
      </div>

      <div class="col-span-2 pt-1 -ml-1.5">
        <p class="font-semibold text-base font-raleway text-left">
          Fish and Chips
        </p>
        <p
          class="text-sm leading-none font-raleway text-left max-h-
7 overflow-hidden"
        >
          Crispy crumbed fresh Hake fillet, with tartar...
        </p>
      </div>

      <div class="col-span-1 max-h-5">
        <div class="justify-self-end mt-3 ml-9 font-raleway font-bold">
          47,99
        </div>

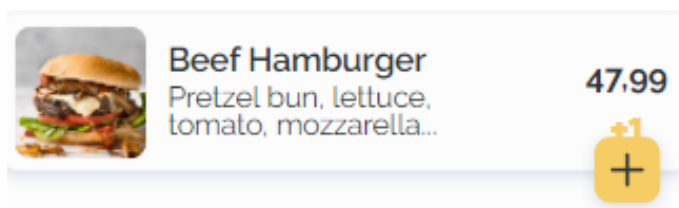
        <div
          id="ca1"
          class="baton baton-plus noselecto w-8 mt-3 ml-10 text-secondary-
black font-raleway p-2 bg-primary-gold shadow-customShadow2 rounded-lg"
        >
          
        </div>
      </div>
    </div>
  </div>
</div>

```

*Slika 22. Prikaz kôda za popis jela iz odabrane kategorije*

## ANIMACIJA PRILIKOM PRITISKA NA GUMB „ + „

Animacija ' +1 ' koja se pojavljuje pritiskom na gumb plus napravljena je putem JavaScripta, gdje je najprije postavljen broj pluseva na nulu, zatim zadan stil animacije i putem id-a određeno da kada se pritisne na određeni gumb da se pojavi animacija ' +1 '. Kako bi se animacija primjenjivala na svaki gumb posebno, a ne na sve odjednom, potrebno je svakom gumbu dodijeliti drugačiji id. Animacija je postavljena da se kreće po vertikali i to nešto malo iznad samog gumba. Također ima određeno trajanje i postepeno nestajanje.



*Slika 23. Animacija koja prikazuje +1 pritiskom na gumb*

```

$(document).ready(function () {
  // Store number of pluses
  let noOfPlusics = 0;

  // Plus animation #####
  $(".button-plus").click(function () {
    noOfPlusics++;
    var name = $(this).attr("id");

    $("#" + name).append(
      "<p id='plusic" +
        noOfPlusics +
        "' class='plus-sign text-primary-gold font-gilroyheavy absolute -z-
10'> +1 </p>"
    );

    // Plus sign animation
    var plusic =
      "#" +
      $("#" + name)
        .find("#plusic" + noOfPlusics)
        .attr("id");

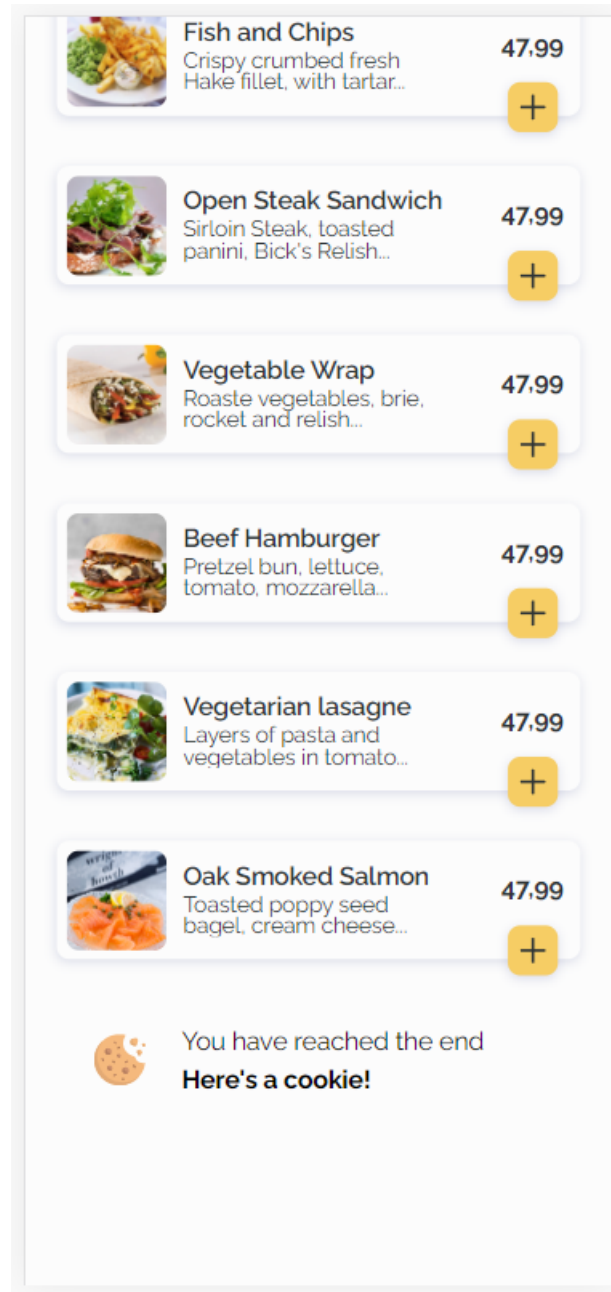
    // Plus sign animation
    var plusicAnimation = anime({
      targets: plusic,
      translateY: [
        {
          value: -20,
          duration: 1,
        },
        {
          value: -45,
          duration: 1000,
          easing: "cubicBezier(0.000, 0.560, 0.175, 0.925)",
        },
        {
          values: -20,
          duration: 1,
          delay: 20,
        },
      ],
      opacity: [
        {
          value: 1,
          duration: 1,
          delay: 20,
        },
      ],
    });
  });
}

```

Slika 24. Kôd za animaciju prilikom pristiska na gumb

## KRAJ STRANICE

Predzadnji dio je dio koji govori korisniku da je stigao do kraja stranice. Taj se dio sastoji od sličice keksa i teksta da je dosegnut kraj stranice.



Slika 25. Kraj stranice

Kôd za krajnji dio je kratak, s obzirom na to da se sastoji samo od dva div-a i paragrafa. Najprije su tekst i slike umotani u poseban div kojem je maksimalna širina prilagođena širini zaslona mobilnog uređaja, horizontalne margine su automatske, horizontalni razmak između elemenata je 14px i unutarnji odmak (padding) i margine su prilagođene kako bi sve skupa izgledalo uredno i konzistentno.

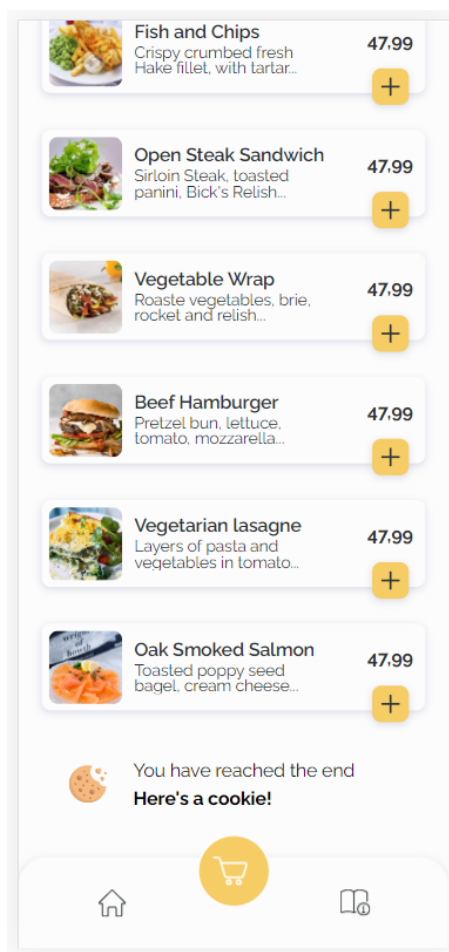
Sljedeća je stavljena ikona kolačića i zatim u još jedan div kojemu je negativna gornja margina 10px umotani su paragrafi. U prvom paragrafu je tekst „You have reached the end“, kojemu je poravnanje lijevo, font *raleway* i veličina teksta bazna. Drugi paragraf ima iste klase kao i prvi, samo je tekst podebljan i tekst je „Here's a cookie!“.

```
<!--END-->
  <div class="max-w-md mx-auto space-x-14 px-11 pt-12 pb-10 mb-20">
    
    <div class="-mt-10">
      <p class="text-left font-raleway text-
base">You have reached the end</p>
      <p class="text-left font-raleway font-bold">Here's a cookie!</p>
    </div>
  </div>
```

Slika 26. Kôd za prikaz kraja stranice

## FOOTER

Posljednji korak je fiksirano podnožje (footer) koje nudi mogućnost odabira na kućicu koja vodi na početnu stranicu (ovu), košaricu za kupovinu koja vodi na dio za plaćanje i knjižicu koja vodi na „O nama“ stranicu.



Slika 27. Footer (podnožje stranice)

Kao i u svakom prijašnjem koraku, najprije je sve umotano u glavni div. Ovaj dio napravljen je pomoću grid-ova, stoga je div-u zadana klasa `grid grid-cols-3` i time napravljena rešetka sa 3 stupca od kojih će u svakome biti smješten po jedan element. Klasom `justify-items-center` određeno je da svi elementi budu horizontalno centrirani, `fixed` klasa govori da je cijeli taj dio fiksiran, što znači dok se bude listalo da će on cijelo vrijeme biti vidljiv na dnu. Klasa `bottom-0` korištena je kako bi se taj cijeli dio smjestio uz donji rub. Gornji kutovi pravokutnika imaju zaobljenje '3xl', te je iza cijelog elementa sjena.

U prvom stupcu nalazi se ikona kućice koja klikom na nju vodi na početnu stranicu, a to je ova trenutna.

U sljedećem stupcu nalazi se ikona košarice koja je istaknuta u odnosu na druge dvije ikone. Zaobljenje kutova postavljeno je na 'full' kako bi se formirao krug unutar kojeg će biti ikona. Kako bi ikona bila iznad druge dvije, a ne u ravnini s njima, mora prelaziti gornju granicu stupca, a to je postignuto tako što je div-u gornja negativna margina 12px kako bi se element „povukao“ gore. Pozadina div-a je predefinirane zlatne boje.

Zadnji stupac, kao i prvi, sadrži samo ikonu knjižice i nema u sebi nikakve postavljene klase.

Prilikom pritiska na gumb sa košaricom, prikazuje se kratka skalirajuća animacija. Krug se najprije poveća, zatim vrati na početno stanje. Trajanje animacije postavljeno je na 0,2 sekunde dok se povećava i isto toliko dok se smanjuje. Postavljena je lagana tranzicija prema van i unutra kako ne bi bila gruba animacija prilikom pritiska na gumb.

```
// Cart button animation
$(".button-cart").click(function () {
  var name = $(this).attr("id");
  var buttonname = "#" + name;

  var animation = anime({
    targets: buttonname,
    scale: [
      {
        value: 1,
        duration: 1,
        easing: "easeInOutSine",
      },
      {
        value: 1.1,
        duration: 200,
        easing: "easeInOutSine",
      },
      {
        value: 1,
        duration: 200,
        easing: "easeInOutSine",
      },
    ],
  });
});
```

Slika 28. Prikaz kôda za podnožje (footer) stranice

## Zaključak

S obzirom na današnju situaciju s pandemijom, sve više do izražaja dolazi tehnologija i njena upotreba kako bi se ograničila interakcija među ljudima. Potreba za tehnološkim rješenjima, u svim aspektima, je sve veća. Tako se uz sve ostalo problem pojavio i u kafićima i restoranima jer se interakcija osoblja i gostiju morala svesti na minimalnu, također su se privremeno ukinuli cjenici ne bi li svatko dirao po njima i zato je bilo potrebno brzo i jednostavno rješenje za sve to. Kao optimalno rješenje bila je aplikacija koja nudi mogućnost skeniranja QR kôda a na stolu i preko koje bi se mogla pregledavati ponuda objekta, naručivati i plaćati. Aplikacija je zasad namijenjena isključivo za mobilne uređaje s obzirom na to da danas veliki broj ljudi, gotovo svi, koriste pametne mobilne uređaje i imaju ih pri sebi, dok je manja vjerojatnost da će netko doći u kafić ili restoran sa laptopom ili tabletom.



## Popis slika

<i>Slika 1. Osnovni HTML kôd i njegov prikaz u pregledniku.....</i>	<i>4</i>
<i>Slika 2. Prikaz sadržaja unutar &lt;body&gt;. &lt;head&gt; i &lt;title&gt; elemenata.....</i>	<i>4</i>
<i>Slika 3. HTML komentari .....</i>	<i>5</i>
<i>Slika 4. Struktura HTML-a 5 .....</i>	<i>6</i>
<i>Slika 5. Struktura HTML-a .....</i>	<i>6</i>
<i>Slika 6. Prva web stranica, bez CSS-a.....</i>	<i>7</i>
<i>Slika 7. Ilustracija ideje.....</i>	<i>11</i>
<i>Slika 8. Dizajn aplikacije.....</i>	<i>13</i>
<i>Slika 9. Prikaz kôda unutar &lt;head&gt; tagova .....</i>	<i>22</i>
<i>Slika 10. Paleta boja.....</i>	<i>23</i>
<i>Slika 11. tailwind.config.js datoteka sa bojama, fontovima i sjenama.....</i>	<i>24</i>
<i>Slika 12. style.css datoteka sa fontovima.....</i>	<i>25</i>
<i>Slika 13. main.js datoteka sa Tippy elementima za okvir s porukom .....</i>	<i>26</i>
<i>Slika 14. Komponente za jezike, zvono, naslovi, okvir za pretraživanje .....</i>	<i>27</i>
<i>Slika 15. Prikaz kôda za naslov, zvono, jezike i okvir za pretraživanje .....</i>	<i>28</i>
<i>Slika 16. Kartice s ponudom jela.....</i>	<i>30</i>
<i>Slika 17. Prikaz kôda za kartice s ponudom jela .....</i>	<i>31</i>
<i>Slika 18. Kategorije .....</i>	<i>32</i>
<i>Slika 19. Kôd za kartice kategorija .....</i>	<i>34</i>
<i>Slika 20. Popis jela iz odabrane kategorije .....</i>	<i>35</i>
<i>Slika 21. Kompletni popis jela iz odabrane kategorije.....</i>	<i>37</i>
<i>Slika 22. Prikaz kôda za popis jela iz odabrane kategorije.....</i>	<i>38</i>
<i>Slika 23. Animacija koja prikazuje +1 pritiskom na gumb .....</i>	<i>39</i>
<i>Slika 24. Kôd za animaciju prilikom pristiska na gumb.....</i>	<i>40</i>

<i>Slika 25. Kraj stranice</i> .....	41
<i>Slika 26. Kôd za prikaz kraja stranice</i> .....	42
<i>Slika 27. Footer (podnožje stranice)</i> .....	43
<i>Slika 28. Prikaz kôda za podnožje (footer) stranice</i> .....	44

## Literatura

- [1] Dorman, M. (2020). *Introduction to Web Mapping* (1st ed.). CRC Press. doi: <https://doi.org/10.1201/9780429352874>, 2021.
- [2] Prelec, Andrija (2015) *Izrada responzivne internetske stranice upotrebom HTML5 i CSS3 tehnologija*. Završni rad - preddiplomski studij. Grafički fakultet. URI: <https://eprints.grf.unizg.hr/2293>, 2021.
- [3] J. Duckett, *HTML and CSS: Design and Build Websites*. Hoboken, NJ: Wiley, 2021.
- [4] Castro, Elizabeth, and Bruce Hyslop. *HTML5 a CSS3.*, 2021. <https://ptgmedia.pearsoncmg.com/images/9780321719614/samplepages/0321719611.pdf>, 2021.
- [5] M. Knapp, *HTML and CSS: Learn The Fundamentals In 7 days*, 2021.
- [6] Haverbeke, Marijn. *Eloquent JavaScript: a modern introduction to programming*. No Starch Press, 2021. <https://eloquentjavascript.net/index.html>

## Prilozi

Poveznica za prikaz gotovog dizajna aplikacije:

<https://drive.google.com/drive/folders/1z1Asi186uR3U5jz7Ehy3qfT-QyOA0WT9?usp=sharing>



Sveučilište  
Sjever



SVEUČILIŠTE  
SJEVER

IZJAVA O AUTORSTVU  
I  
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, Kristina Jurković (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Aplikacija za mobilne uređaje od ideje do realizacije (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica: KRISTINA  
(upisati ime i prezime) JURKOVIĆ

Kristina Jurković  
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, Kristina Jurković (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom Aplikacija za mobilne uređaje od ideje do realizacije (upisati naslov) čiji sam autor/ica.

Student/ica: KRISTINA  
(upisati ime i prezime) JURKOVIĆ

Kristina Jurković  
(vlastoručni potpis)