Web portal za OPG

Smoljo, Ivan

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:039559

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-08-11**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

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

Preddiplomski sveučilišni studij

WEB PORTAL ZA OPG

Završni rad

Ivan Smoljo

Osijek, 2021.

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

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

Osijek, 20.09.2021.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Ivan Smoljo
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R3980, 26.07.2016.
OIB Pristupnika:	10802227609
Mentor:	Prof. dr. sc. Krešimir Nenadić
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Web portal za OPG
Znanstvena grana rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak završnog rad:	Kratko objasniti način obavljanja posla obiteljskih poljoprivrednih gospodarstava. Izraditi web portal na kojem će djelatnici OPG-a ponuditi informacije o svom OPG-u i proizvode koje prodaju. Omogućiti registriranim korisnicima, koji nisu zaposlenici OPG-a, naručivanje ponuđenih proizvoda putem web obrasca. Tema rezervirana: Ivan Smoljo
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	20.09.2021.
Datum potvrde ocjene od strane Odbora:	22.09.2021.

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

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 01.09.2022.

Ime i prezime studenta:	Ivan Smoljo
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3980, 26.07.2016.
Turnitin podudaranje [%]:	9

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

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

i sumentora ,

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

Potpis studenta:

SADRŽAJ

1. UVOD 1
1.2. Zadatak završnog rada 1
2. PREGLED SLIČNIH RJEŠENJA 1
2.1. „Eko Veselić“ 1
2.2. „OPG Lili“ 2
2.3. „OPG Toni Guberac“ 3
2.4. „OPG Ružić“ 3
2.5. „OPG Marušić“ 4
3. KORIŠTENE TEHNOLOGIJE 5
3.1 HTML 5
3.2. CSS 6
3.3. JavaScript 6
3.4. Firebase 7
3.5. Visual Studio Code 7
4. DIJELOVI WEB STRANICE 8
4.1. Zaglavlje 8
4.2. Početna stranica 10
4.3. Stranica proizvodima 13
4.4. Kontakt stranica 15
4.5. Prijava 18
4.6. Košarica 21

5. Zaključak 24

Literatura 25

Sažetak 26

Abstract 27

Životopis 27

Prilozi 28

1. UVOD

Ako se uzme da su prva seljačka gospodarstva nastala nakon ukidanja feudalizma, gospodarstva su stara više od 150 godina. Tijekom povijesti su se koristila razna pomagala. Kako je tehnologija napredovala, tako se poboljšavala kvaliteta proizvodnje i distribucije proizvoda. Kako bi bili u korak s vremenom, OPG-ovi su također prisvojili tehnologije koje će im pomoći u prodaji svojih proizvoda.

Kako ih se više okreće digitalnom načinu oglašavanja, u ovom radu će biti prikazana izrada web portala za OPG. Bit će opisane funkcionalnosti koje web portal pruža kako bi se posjetioci što bolje upoznali sa uslugama koje nudi OPG.

Nakon ovog uvoda, u drugom poglavlju su opisane tehnologije koje su korištene pri izradi web portala. Svaka tehnologija je ukratko opisana i opisane su osnovne i najvažnije funkcionalnosti. Zatim, u trećem poglavlju opisana je izrada pojedinih dijelova, neke specifičnosti za svaki dio, te kako se sve povezalo u jednu cjelinu.

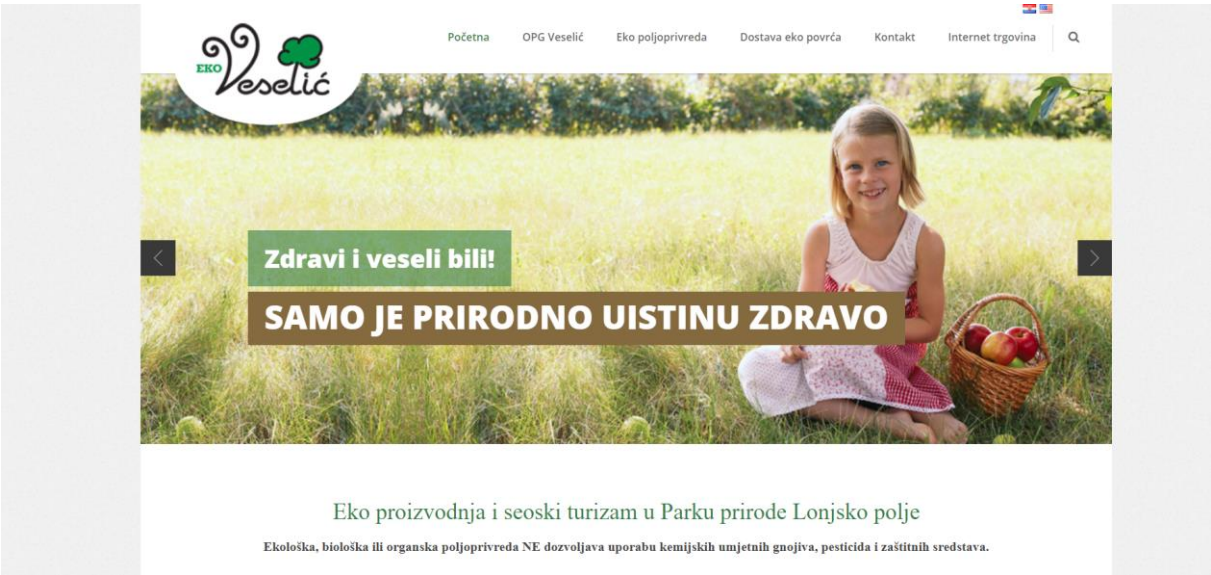
1.2. Zadatak završnog rada

Kratko objasniti način obavljanja posla obiteljskih poljoprivrednih gospodarstava. Izraditi web portal na kojem će djelatnici OPG-a ponuditi informacije o svom OPG-u i proizvode koje prodaju. Omogućiti registriranim korisnicima, koji nisu zaposlenici OPG-a, naručivanje ponuđenih proizvoda putem web obrasca.

2. PREGLED SLIČNIH RJEŠENJA

2.1. „Eko Veselić“

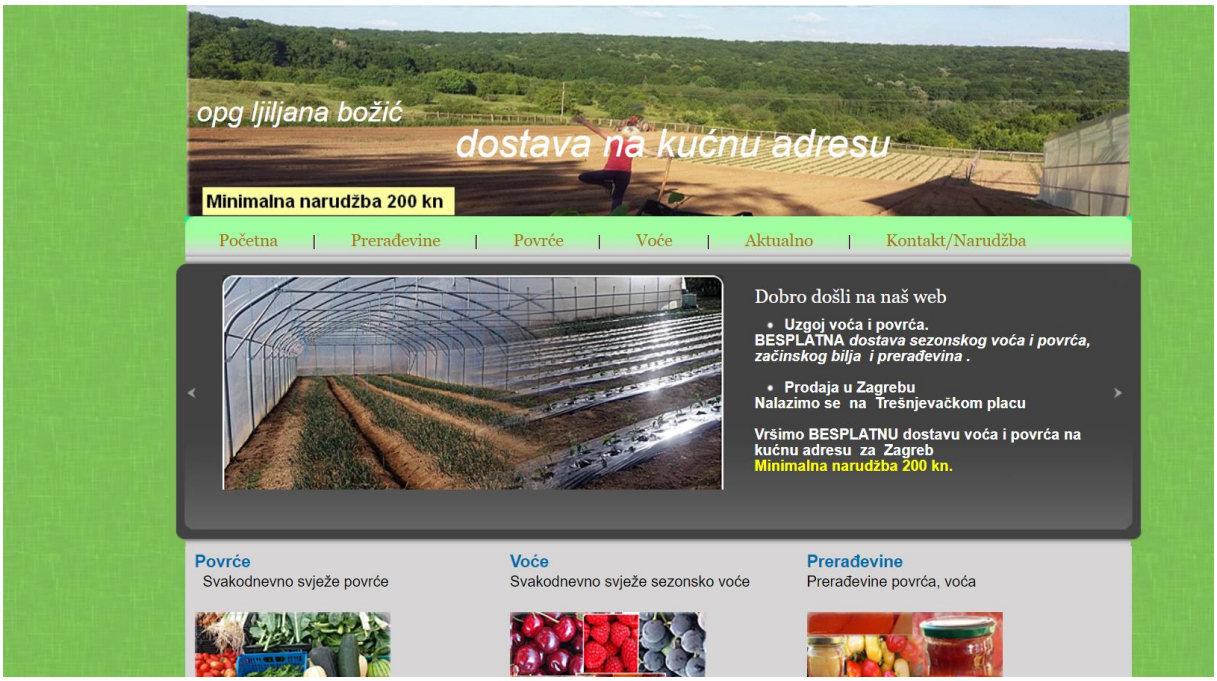
Web stranica „Eko Veselić“(https://eko-veselic.com/?page_id=242) prikazana slikom 2.1., primjer je kako je većina stranica ovakvog tipa rađena na sličan način. Mogu se vidjeti svi važni dijelovi stranice kao što su stranica koja opsuje čime se gospodarstvo bavi, kontakt stranica, te stranica za kupovinu proizvoda.



Slika 2.1. *Prikaz stranice „Eko Veselić“*

2.2. „OPG Lili“

Još jedan primjer stranice koja potvrđuje da su stranice ovakvog sadržaja slične je „OPG Lili“ (http://opg-bozic-lili.hr/Kontakt.html). Na slici 2.2. se vide slični elementi kao i u prethodnom primjeru.



Slika 2.2. *Prikaz stranice „OPG Lili“*

2.3. „OPG Toni Guberac“

„OPG Toni Guberac“(https://www.opg-toniguberac.hr/) stranica prikazuje ponudu svojih proizvoda za kupovinu što je prikazano slikom 2.3..

Slika 2.3. *Prikaz stranice „OPG Toni Guberac“*

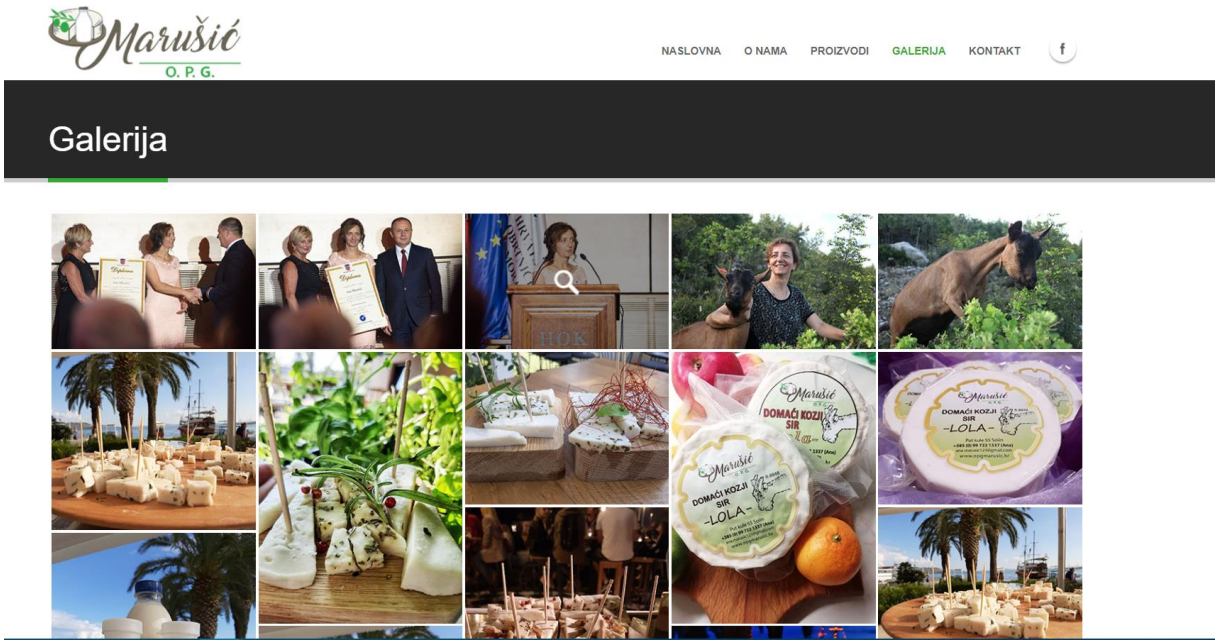
2.4. „OPG Ružić“

Na stranici „OPG Ružić“(https://opg-ruzic.com/) se može vidjeti kako gospodarstva upoznaju potencijalne kupce svojim opisom nastanka, napredovanja i planova za budućnost, što je vidljivo na slici 2.4..

Slika 2.4. *Prikaz stranice „OPG Ružić“*

2.5. „OPG Marušić“

Primjer gdje gospodarstvenik postavlja galeriju fotografija na kojima se vide proizvodi, članovi gospodarstva, nagrade i još mnogo toga je „OPG Marušić“(http://www.opgmarusic.hr/), što se vidi iz slike 2.5..



Slika 2.5. *Prikaz galerije*

3. KORIŠTENE TEHNOLOGIJE

U ovom poglavlju su navedene i ukratko objašnjene tehnologije koje su korištene za izradu web portala.

3.1 HTML

Hyper Text Markup Language ili skraćeno HTML opisni je jezik koji služi za izradu web stranica. Koristi se za opis strukture web stranice tako da preglednik zna kako će prikazati sadržaj korisniku. HTML se prvi put spominje kao dokument „HTML tags“ 1991. godine kojeg je opisao Tim Berners-Lee[3]. HTML se najčešće kombinira s nekim drugim jezicima kako bi stranica izgledala bolje i kako bi se uključile ostale funkcionalnosti. Osnovni elementi HTML-a su HTML oznake koje imaju svoju oznaku za otvaranje i zatvaranje. Dodavanje paragrafa u kod moramo prvo otvoriti oznakom . Zatim dolazi sadržaj koji treba biti u tom paragrafu i na kraju paragraf zatvorimo s oznakom . Pomoću ovih, i još mnogih drugih oznaka, kreira se kompleksnija stranica. Elementi tih oznaka imaju svojstvo ugnježđivanja, ali se pri tome mora paziti da se unutarnja oznaka zatvori prije vanjske.

Slika 3.1. *Izgled osnovne strukture HTML koda*

Slika 3.1. prikazuje osnovne elemente od kojih se sastoji HTML kod. U prvoj liniji koda se nalazi element <!DOCTYPE html> koja govori da se radi o HTML5 dokumentu. Nakon toga, u <html> elementu se nalazi sav sadržaj stranice i to je korijenski element. Unutar <html> elementa se nalaze oznake <head> i </head> između kojih se nalaze meta podatci i oznake

<title> i </title > između kojih je sadržan naslov dokumenta. Na kraju dolazi tijelo, odnosno <body> i </body>, gdje se nalazi sve što je vidljivo na web stranici stranici.

3.2. CSS

CSS (engl. *Cascading Style Sheets*) je jezik kojim se opisuje kakav će stil imati HTML dokument. S razvitkom weba, uočena je potreba za stilskim jezikom koji će oblikovati HTML dokument. CSS se može povezati na 3 načina s HTML dokumentom: vanjskom listom stilova, unutarnjom listom stilova te stilom unutar linije. Najpoželjnija je prva opcija zbog svojstva da se može povezati s više HTML dokumenata. Razdvajanje HTML-a i CSS-a omogućuje lakše i preglednije upravljanje sadržajem i izgledom stranice. CSS koristi selektore pomoću kojih „zna“ na koji element se primjenjuje zadani stil. Dvije najpoznatije vrste selektora su *id* selektor i klasni selektor. Na slici 3.2. prikazane su neke funkcionalnosti CSS-a.

```css
footer {
  margin-top: 1em;
  background-color: rgb(160, 60, 29);
  height: max-content;
  padding: 0.5em;
}
```

Slika 3.2. *Primjer stiliziranja u CSS-u*

3.3. JavaScript

JavaScript je skriptni programski jezik visoke razine. Dok se HTML i CSS koriste za sadržaj i izgled stranice, JavaScript dodaje još jednu funkcionalnost stranici, a to je interaktivnost. Prije dolaska JavaScripta, web stranice su bile slične i statične. Kako bi dodali funkcionalnosti JavaScripta u postojeću web stranicu, potrebno ga je dodati u <head> ili <body> dio HTML dokumenta ili u oba dijela. Dodavanje JavaScript koda prikazano je slikom 3.3..

```
<script>
    document.getElementById("ID").style.display ="JavaScript"
</script>
```

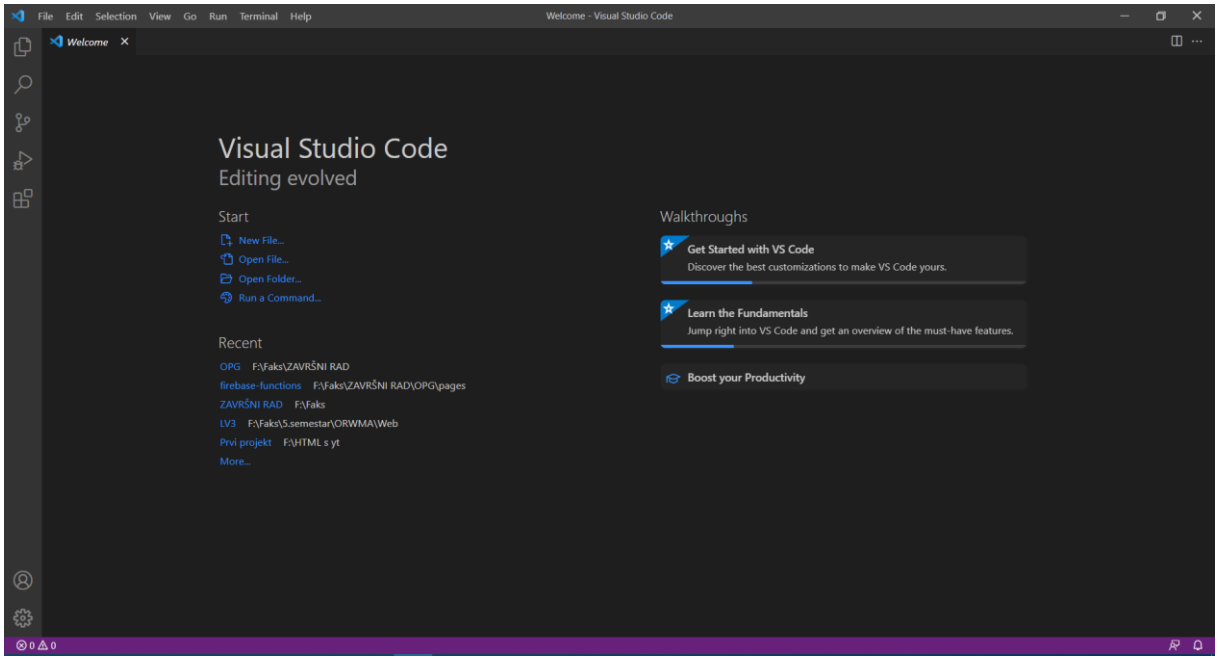
Slika 3.3. *Korištenje JavaScript jezika unutar HTML dokumenta*

3.4. Firebase

Firebase je platforma za razvoj web i mobilnih aplikacija. Tvrtka Firebase inc. 2012. godine ponudila je javnosti svoj prvi proizvod koji se zvao *Firebase Realtime Database*. Taj proizvod je omogućio korisnicima sinkronizaciju i pohranu podataka u stvarnom vremenu. Nakon toga izdali su još dva proizvoda, *Firebase Hosting* i *Firebase Authentication*, čime su se probili na tržište backenda za mobilne uređaje. *Firebase Authentication* je usluga koja pruža provjeru autentičnosti korisnika. Neki od načina provjere koje ova usluga podržava su korištenje zaporki, putem broja mobilnog telefona, pomoću Google, Facebook ili Twitter računa i mnogi drugi. Za korištenje svih navedenih usluga potrebno je prijaviti se na Firebase stranicu putem Google računa.

3.5. Visual Studio Code

Visual Studio Code je integrirano razvojno okruženje koje je razvio Microsoft za Windows, Linux i macOS. Većina izvornog koda dostupna je na GitHub-u. Korisničko sučelje koje je prikazano slikom 3.4. jednostavno je koristiti i pruža mogućnost dodavanja proširenja koja olakšavaju rad. Neke od korištenih proširenja koja su korištena u ovom radu su *Live Server*(Prikaz promjena u kodu u realnom vremenu) i *Prettier*(Poravnanje linija koda za bolju preglednost).



Slika 3.4. *Prikaz korisničkog sučelja*

4. DIJELOVI WEB STRANICE

Većina web stranica obiteljskih gospodarstava i web stranice koje nude slične sadržaje su slične u svom izgledu za pružanje informacija i usluga. Općenito se drže pravila gdje se opisuje kratka povijest OPG-a(Obiteljsko poljoprivredno gospodarstvo), kako je OPG započeo sa svojim radom, te iz kojih razloga. Navodi se kojim se sve djelatnostima bavi gospodarstvo i koje se sve usluge nude. Ovaj dio je najčešće prikazan na početnoj stranici ili u sekciji *O nama*. Proizvodi i usluge se često odvoje na zasebnu stranicu kako se na početnoj stranici ne bi nalazilo previše sadržaja. Česta praksa je pobliže opisati proizvode i usluge. Možda i najvažniji dio za svaki OPG jest onaj gdje se posjetitelji stranice mogu informirati o tome kako kontaktirati OPG, bilo putem telefona, adrese e-pošte ili društvenih mreža. Ovdje se također navodi i adresa kako bi zainteresirani kupci mogli i osobno posjetiti OPG. Neki vlasnici OPG-a se odluče i na uslugu prodaje svojih proizvoda na web stranici. Ovo je posebno važno za OPG-ove koji imaju specifične proizvode za određeno podneblje, a htjeli bi svoje proizvode prodavati i u dijelovima zemlje/svijeta u kojima takvih proizvoda nema. Ovom dijelu za kupovinu proizvoda moguće je pristupiti jedino kad se korisnik registrira ili prijavi(ako je već registriran).

4.1. Zaglavlje

Svaka stranica u ovom radu ima zaglavlje koje se sastoji od tri dijela. Prvi dio nalazi se na krajnjem lijevom dijelu. To je logo koji je stiliziran da bude kružnog oblika. Za ovo jekorištena CSS naredba *border-radius,* te joj je vrijednost postavljena na 50% kako bi se slika kvadratnog oblika prikazala kao kružna. U sredini se nalazi ime OPG-a. U krajnjem desnom dijelu zaglavalja nalazi se navigacijski izbornik pomoću kojeg se pristupa ostalim stranicama. Cijelo zaglavlje se može vidjeti na slici 4.1..

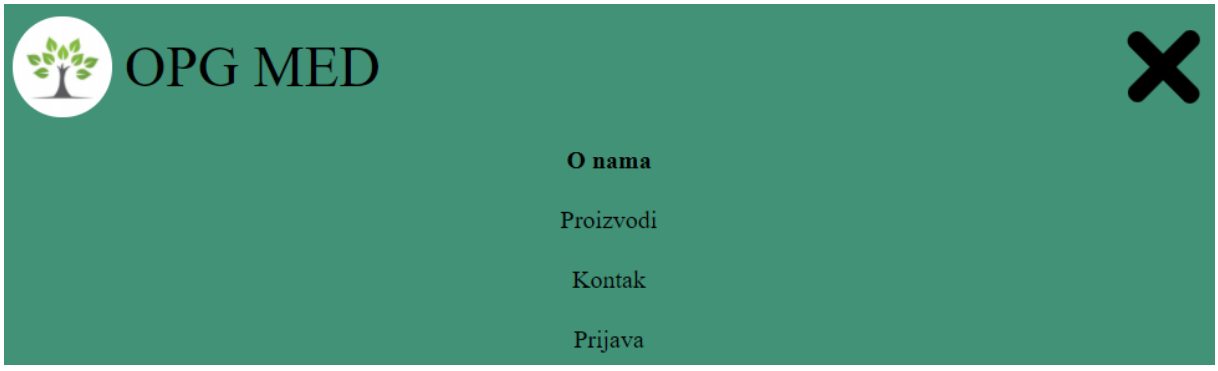
Navigacijski izbornik se sastoji od neporedane liste elemenata. Ona se sastoji od četiri elemenata pomoću kojih se pristupa ostali stranicama. Prva stranica je *O nama,* nakon nje dolazi dio s proizvodima, pa *Kontakt* i naposljetku *Prijava*. Kada se pokazivačem pređe preko jednih od tih elemenata, oni promijene boju. Ovo se postiglo naredbom *hover* i zatim naredbom *backround-color*. Kako bi korisnik znao na kojoj se stranici nalazi, tekst stranice bude podebljan ovisno o tome na kojoj se stranici korisnik nalazi. Cijelo zaglavlje se u svakom trenutku nalazi na vrhu stranice što je omogućeno u CSS-u pomoću *position:fixed.*

Slika 4.1. *Prikaz zaglavlja s hover funkcionalnosti*

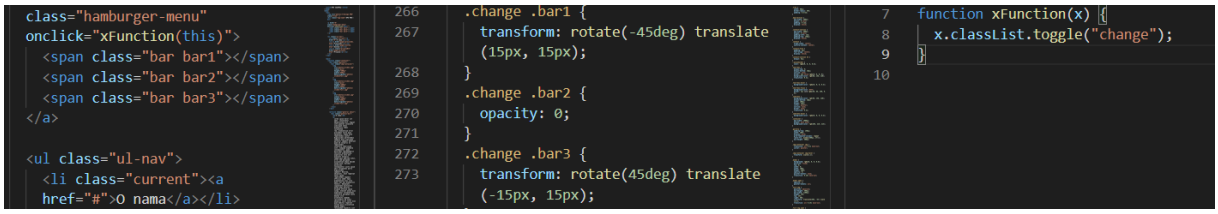
Budući da sve više korisnika pristupa web stranicama pomoću svojih mobilnih uređaja, zaglavlje je optimizirano tako da kad se promijeni širina prozora, odnosno kad se stranica ne otvara na osobnom računalu, promijeni se i izgled zaglavlja. Taj način prikaza je namijenjen manjim zaslonima i sastoji se od tzv. *hamburger* gumb. Ovaj način prikaza vidi se na slici 4.2.

Slika 4.2. *Prikaz zaglavlja pomoću hamburger gumba*

Pritiskom na hamburger gumb otvara se padajući izbornik sa istim opcijama za pristup ostalim stranicama što se može vidjeti na slici 4.3.. Kad se gumb pritisne, hamburger gumb se pretvara u znak X što je omogućeno pozivom funkcije koja mijenja izgled na osnovu CSS naredbi, a ta funkcija i CSS kod se može vidjeti na slici 4.4.



Slika 4.3. *Prikaz zaglavalja nakon pritisnutog gumba*



Slika 4.4. *Prikaz koda za izmjenu hamburger gumba*

Podnožje web stranice se također nalazi na svakoj stranici i napravljena je slično kao i zaglavlje. Isto se sastoji od tri dijela. Logo se nalazi na lijevoj strani, isto kao i kod zaglavlja. U sredini se nalazi oznaka <a> s tekstom *Na vrh* koja vraća korisnika na početak stranice na kojoj se nalazi. Na desnoj strani se nalaze tri ikone za pristup društvenim mrežama. Kao i navigacijski izbornik, ikone su smještene u nepovezanoj listi, te služe kao poveznice na određenu društvenu mrežu. Na slici 4.5. prikazano je podnožje web stranice.

Slika 4.5. *Prikaz podnožja web stranice*

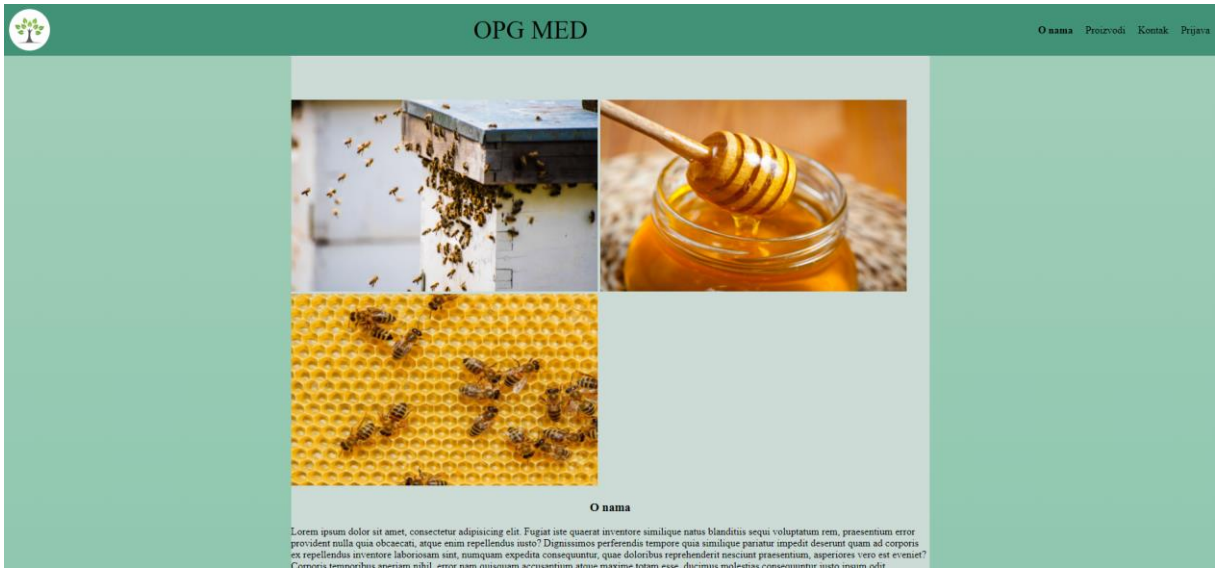
4.2. Početna stranica

Stranica koja se korisniku prva učita kad joj pristupi je stranica *O nama* koja se nalazi na navigacijskom izborniku i koja se može vidjeti na slici 4.7. Sastoji se od dva dijela. Prvi je galerija sa slikama na koje je moguće kliknuti i na taj način ju povećati. Ispod galerije nalazi se dio koji pruža više informacija o nastanku OPG-a, te čime se on bavi. Glavni dio stranice je stavljen pod oznaku <section> s imenom klase *container*, te je on podijeljen na dio, u kojem se nalazi galerija sa slikama, i <article> dio u kojem se nalazi tekst. U ovom slučaju je u dijelu *O nama* generiran nasumičan tekst, ali tu se može upisati bilo koja informacija o OPG-u.

Širina ovog sadržaja je postavljena na 900 piksela jer je za isti iznos ograničena uporaba za računala.Kad se širina uređaja smanji na razinu ispod 900 piksela, elementi unutar stranice će se, ovisno o zadanoj širini, pomaknuti kako bi se sadržaj bio vidljiv.Responzivnost je omogućena naredbom koja se može vidjeti na slici 4.6.. Unutar vitičastih zagrada se nalazi sav sadržaj stranice na koji želimo djelovati ovom naredbom.

```
6
7   @media only screen and (min-width: 900px) {
8
```

Slika 4.6. *CSS naredba za responzivnost*



Slika 4.7. *Prikaz početne/O nama stranice*

U galeriji se nalaze tri slike koje su umanjene od originalnih kako bi stale u zadane okvire. Kad se strelicom miša pređe preko pojedine slike, ona se malo uveća i pokazivač se promijeni u oblik koji korisniku daje do znanja da se na sliku može kliknuti. Ovo je omogućeno CSS naredbama na *img-container* u kojem se nalaze slike i koje su vidljive na slici 4.8..

```
.img-container img {
  transition: all 0.3s ease-out;
  cursor: pointer;
}

.img-container img:hover {
  transform: scale(1.1);
}
```

Slika 4.8. *CSS naredbe za promjenu pokazivača i povećanja slike prelaskom pokazivača*

Kad korisnik klikne na sliku, ona se otvara u većem obliku i pozadina se zatamni. Za zatvaranje slike iz cijelog prikaza potrebno je kliknuti bilo gdje izvan slike kako bi se vratilo na početnu

stranicu. Ovo je omogućeno dodavanjem elementa unutar <body> dijela stranice u HTML-u. Dodaje se na ovo mjesto jer jednom kad se slika otvori, potrebno ju je prikazati preko cijelog zaslona. Kako bi se to omogućilo,potrebno je u CSS kodu za ovaj element dodati svojstvo *opacity* koji se postavi na vrijednost 0 kako se ne bi vidio, te kasnije na 1 kako bi se pokazao. Ovo se može vidjeti na slici 4.9.. Za prikaz ovog dijela bilo je potrebno dodati JavaScript kod koji će dohvatiti elemente iz HTML koda i koji je vidljiv na slici 4.10..

```css
.modal {
  background: rgba(0, 0, 0, 0.6);
  position: fixed;
  top: 0;
  width: 0;
  width: 100%;
  height: 100%;
  opacity: 0;
  pointer-events: none;
  transition: 0.25s ease-out;
}

.modal.open {
  opacity: 1;
  pointer-events: all;
}
```

Slika 4.9. *CSS kod za prikaz cijele slike*

```
const modal = document.querySelector(".modal");
const previews = document.querySelectorAll(".gallery img");
const original = document.querySelector(".full-img");

previews.forEach((preview) => {
  preview.addEventListener("click", () => {
    modal.classList.add("open");
    original.classList.add("open");

    const originalSrc = preview.getAttribute("data-original");
    console.log(originalSrc);
    original.src = originalSrc;
  });
});

modal.addEventListener("click", (e) => {
  if (e.target.classList.contains("modal")) {
    modal.classList.remove("open");
    original.classList.remove("open");
  }
});
```

Slika 4.10. *JavaScript kod za ulazak u cijeli prikaz slike u galeriji*

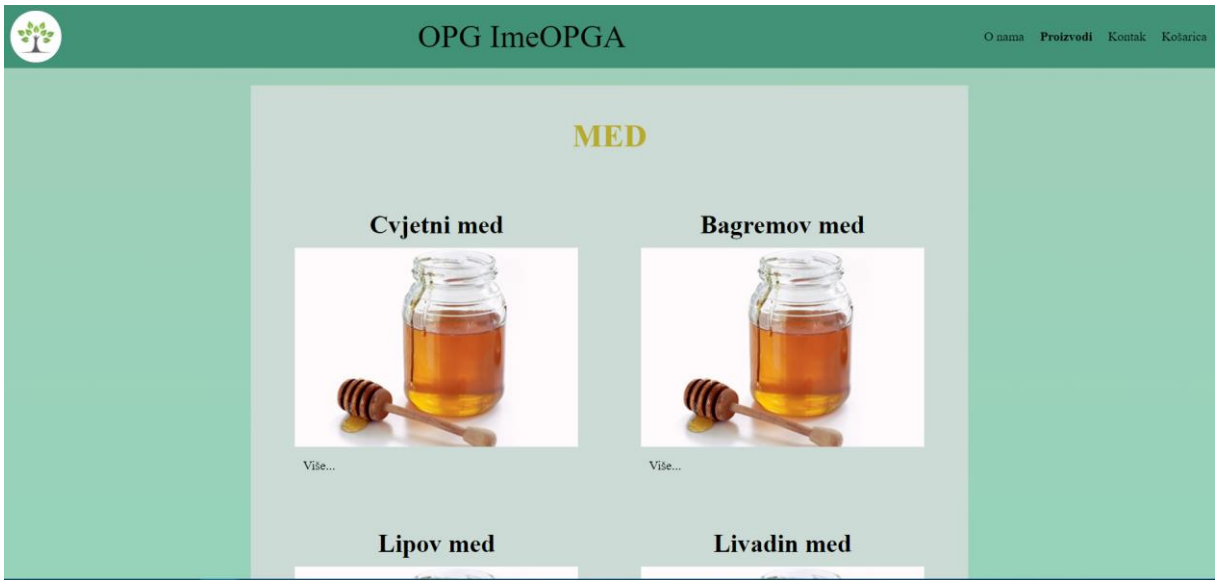
4.3. Stranica proizvodima

Kako bi posjetitelji stranica mogli vidjeti koje sve proizvode neki OPG nudi, većina vlasnika se odlučuje imati stranicu koja informira posjetitelje upravo o tome. Prikažu se slike proizvoda, njihov nazivi, te kratak opis proizvoda. Na sličan način je napravljena stranica proizvoda i u ovom radu koja je prikazana slikom 4.12..

Sadržaj prikaza proizvoda se sastoji od naziva skupine proizvoda, a ispod toga su prikazani svi proizvodi iz te skupine. Svaki proizvod u skupini ima svoje ime, sliku na kojoj se vidi izgled proizvoda, te opciju za prikaz više detalja o proizvodu. Svaka skupina proizvoda je pod zasebnom <section> oznakom uz ime klase *container*. Unutar toga se nalazi određeni broj oznaka (ovisno o broju proizvoda) gdje se nalaze gore spomenuti elementi svakog proizvoda što se vidi na slici 4.11..

```
<section class="container">
  <h2 class="naslov-proizvoda">MED</h2>
  <div class="vrste-proizvoda">
    <div class="konkretan-proizvod">
      <span class="ime-proizvoda">Cvjetni med</span>
      <img class="slika-proizvoda" src="../photos/slikaMeda.jpg" />
      <div class="detalji-proizvoda">
        <span class="vise-o-proizvodu">Više...</span>
      </div>
    </div>
    <div class="konkretan-proizvod">
      <span class="ime-proizvoda">Bagremov med</span>
      <img class="slika-proizvoda" src="../photos/slikaMeda.jpg" />
      <div class="detalji-proizvoda">
        <span class="vise-o-proizvodu">Više...</span>
      </div>
    </div>
    <div class="konkretan-proizvod">
      <span class="ime-proizvoda">Lipov med</span>
      <img class="slika-proizvoda" src="../photos/slikaMeda.jpg" />
      <div class="detalji-proizvoda">
        <span class="vise-o-proizvodu">Više...</span>
      </div>
    </div>
    <div class="konkretan-proizvod">
      <span class="ime-proizvoda">Livadin med</span>
      <img class="slika-proizvoda" src="../photos/slikaMeda.jpg" />
      <div class="detalji-proizvoda">
        <span class="vise-o-proizvodu">Više...</span>
      </div>
    </div>
  </div>
</section>
```

Slika 4.11. *HTML kod sadržaja stranice s proizvodima*



Slika 4.12. *Prikaz izgleda stranice s proizvodima*

4.4. Kontakt stranica

Ova stranica daje mogućnost posjetitelju web stranica da ostavi svoje ime, adresu e-pošte i upit vlasniku. Također je napisana fizička adresa OPG-a, adresa e-pošte na koji se korisnici također mogu javiti i broj mobitela i telefona.

Dio pomoću kojeg je korisniku omogućena opcija postavljanja upita vlasniku je napravljena pomoću <form> oznake. Ona se sastoji od dvije <input> oznake. Jedna za unos imena i prezimena, koja ima tip *text*, i druga za unos adrese e-pošte korisnika tipa *email*. Ispod toga se nalazi oznaka <textarea> koja omogućuje korisniku javljanje ili postavljanje pitanja i kojoj je definiran broj stupaca i redaka. Na kraju forme se nalazi gumb za slanje.

Kad korisnik želi postaviti upit, potrebno je dati ime, adresu e-pošte i sadržaj poruke. Ako korisnik pokuša poslati bez upisivanja podataka, izbacit će se upozorenje da neki dio nije popunjen. Ovo je vidljivo na slici 4.13.. Kad se popune svi dijelovi, ime, adresa e-pošte i sadržaj poruke se šalju u Firebase bazu podataka. Implementacija Firebaseove usluge *Firestore* prikazana je slikom 4.14.

Slika 4.13. *Izgled upozorenja nakon pritiska gumba „Pošalji“*

```
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.0.2/firebase-app.js";
  import {
    getFirestore,
    collection,
    addDoc,
  } from "https://www.gstatic.com/firebasejs/9.0.2/firebase-firestore.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  const firebaseConfig = {
    apiKey: "AIzaSyC5U4e-9GH3PAvsTfLJJCYdnNmGUPkxToE",
    authDomain: "zavrsni-e173c.firebaseapp.com",
    projectId: "zavrsni-e173c",
    storageBucket: "zavrsni-e173c.appspot.com",
    messagingSenderId: "988563151983",
    appId: "1:988563151983:web:33b3ad8b5015d451e0419a",
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
  const db = getFirestore(app);
```

Slika 4.14. *Implementacija usluge Firestore*

Kako bi bilo moguće poslati podatke unesene u formi za kontaktiranje u bazu podataka bilo je potrebno dodati funkcije vidljive na slici 4.15.. Ako dođe do neke greške prilikom slanja podataka, korisniku će se pojaviti poruka koja ga o tome obavještava.

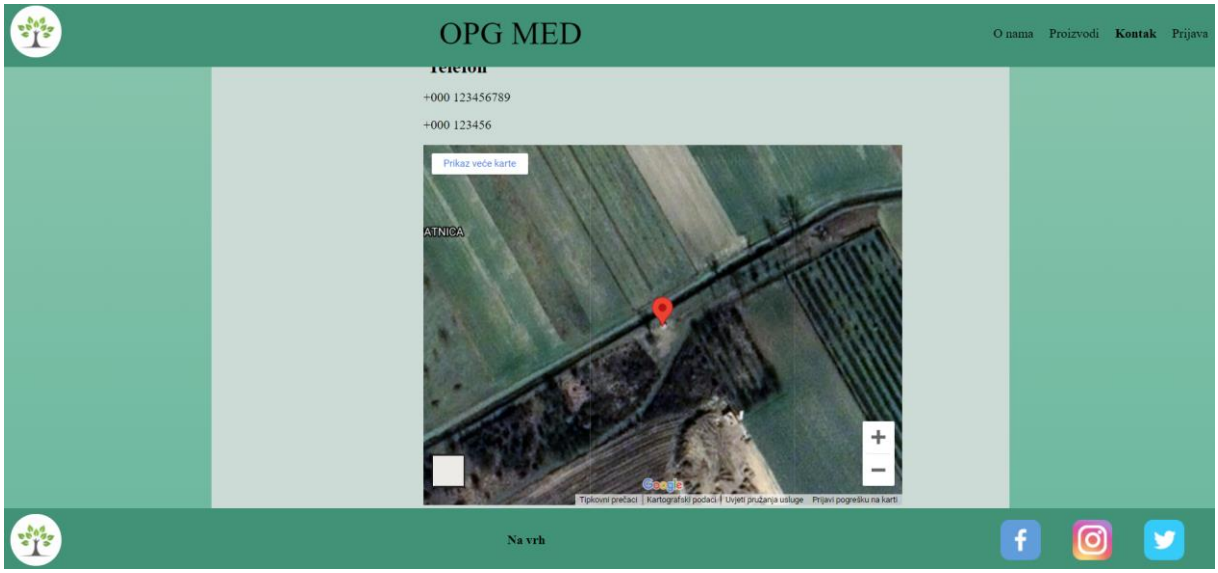
```
document
  .getElementById("contactForm")
  .addEventListener("submit", submitForm);

function submitForm(e) {
  e.preventDefault();

  var name1 = getInputVal("name");
  var email1 = getInputVal("email");
  var message1 = getInputVal("message");
  try {
    const docRef = addDoc(collection(db, "users"), {
      name: name1,
      email: email1,
      message: message1,
    }).then((docRef) => {
      console.log("Document written with ID: ", docRef.id);
    });
    alert("Uspješno ste poslali upit. Hvala!");
  } catch (e) {
    console.error("Error adding document: ", e);
    alert("Greška pri slanju upita!");
  }
}

function getInputVal(id) {
  return document.getElementById(id).value;
}
```

Slika 4.15. *Funkcije za slanje podataka u bazu podataka*

Nakon toga slijede oznake <h2> i <p> u kojima se nalaze informacije za kontaktiranje vlasnika putem adrese e-pošte ili telefona i mobitela. Piše i adresa OPG-a koja je ispod ovog dijela prikazana pomoću Google karte. Karta se u HTML dokument dodaje tako da se na Google karti označi lokacija koju želimo vidjeti gdje se nalazi, te se odabere opcija *Dijeli* gdje je ponuđen kod koji se kopira u HTML dokument. Na slici 4.16. prikazan je taj kod, a na slici 4.17. izgled karte umetnut u stranicu.

```
<iframe
  src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d659.1451108311375!2d18.392435829234813!3d45.
  33746599869372!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.
  1!3m3!1m2!1s0x0%3A0x0!2zNDXCsDIwJzE0LjkiTiAxOMKwMjMnMzQuNyJF!5e1!3m2!1shr!2shr!4v1626008128399!5m2!1shr!2shr"
  width="600"
  height="450"
  style="border: 0"
  allowfullscreen=""
  loading="lazy"
></iframe>
```

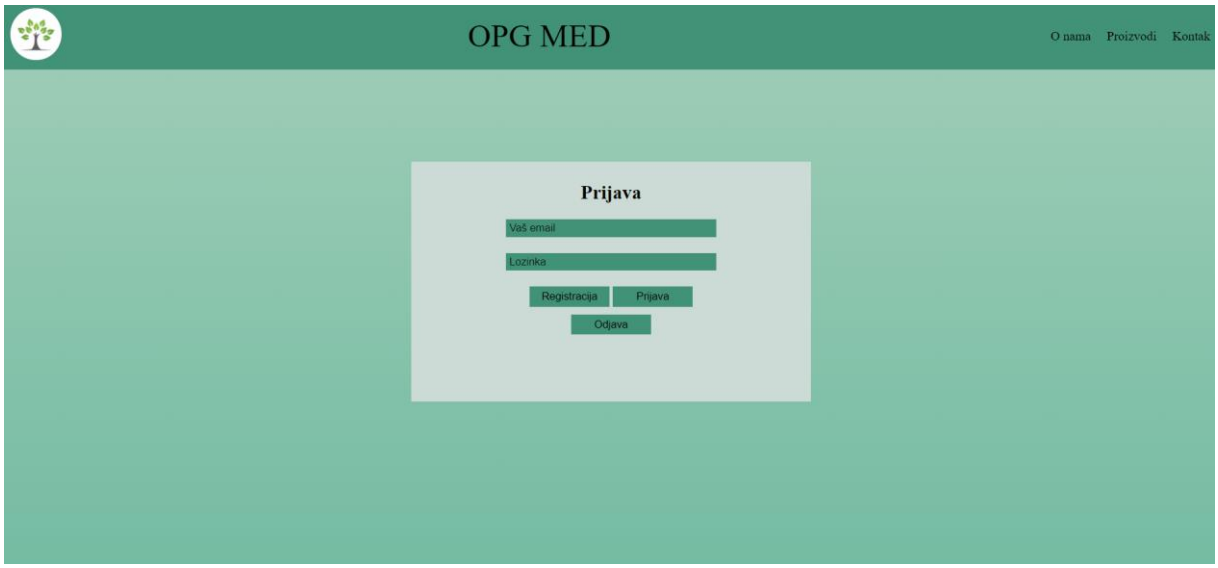
Slika 4.16. *Kod za prikaz karte na stranici*



Slika 4.17. *Prikaz karte s lokacijom na stranici*

4.5. Prijava

Za pristup košarici i kupovini proizvoda korisnik si mora registrirati ili prijaviti ako se već ranije registrirao. Za registraciju i prijavu napravljena je forma u kojoj se nalaze dva <input> elementa. Jedan za adresu e-pošte i drugi za lozinku. Izgled forme se može vidjeti na slici 4.18.. Ispod toga se nalaze gumbi za registraciju, prijavu i odjavu korisnika. Ako korisnik ne unese podatke i pokuša kliknuti na jedan od gumbova, izbacit će se greška. Korisnik mora unijeti podatke koji se provjeravaju pomoću Firebase usluge *Authentication.* Ako se korisnik pokuša registrirati s podacima koji već postoje pokazat će se skočni prozor koji obavijesti korisnika o tome. Isto će se dogoditi ako korisnik unese krivu lozinku.



Slika 4.18. *Izgled forme za registraciju/prijavu*

Kako bi se korisnik mogao registrirati/prijaviti, podatke iz forme bilo je potrebno predati *Firebase Authentication* funkcijama. Prikaz koda za implementaciju usluge je prikazan slikom 4.19., dok su funkcije za slanje podataka i otvaranje košarice prikazane slikama 4.20. i 4.21..

```
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.0.2/firebase-app.js";
  import { getAuth, onAuthStateChanged,createUserWithEmailAndPassword,signInWithEmailAndPassword,signOut } from "https://www.gstatic.com/
  firebasejs/9.0.2/firebase-auth.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
 const firebaseConfig = {
  apiKey: "AIzaSyC5U4e-9GH3PAvsTfLJJCYdnNmGUPkxToE",
  authDomain: "zavrsni-e173c.firebaseapp.com",
  projectId: "zavrsni-e173c",
  storageBucket: "zavrsni-e173c.appspot.com",
  messagingSenderId: "988563151983",
  appId: "1:988563151983:web:33b3ad8b5015d451e0419a",
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
```

Slika 4.19. *Kod implementacije usluge Authentication*

```
    onAuthStateChanged(auth, (user) => {
  if (user != null) {
    alert("Prijavljeni ste");
  }else{
    alert("Odjavljeni ste")
  }
});

const signUpBtn=document.getElementById("signUpButton");
signUpBtn.addEventListener("click", (e)=>{
  const email1=document.getElementById("email").value;
const pass1=document.getElementById("password").value;
createUserWithEmailAndPassword(auth, email1, pass1)
  .then((userCredential) => {
    // Signed in
    const user = userCredential.user;

            window.location="košarica.html";


    // ...
  })
  .catch((error) => {


    const errorCode = error.code;
    const errorMessage = error.message;
    alert(errorMessage);
    // ..
  });
})
```

Slika 4.20. *Funkcije za registraciju/prijavu*

```
const signInBtn=document.getElementById("signInButton");
signInBtn.addEventListener("click", (e)=>{
const email1=document.getElementById("email").value;
const pass1=document.getElementById("password").value;
signInWithEmailAndPassword(auth, email1, pass1)
  .then((userCredential) => {
    // Signed in
    const user = userCredential.user;

            window.location="košarica.html";
    // ...
  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
    alert(errorMessage);
    // ..
  });
})
const signOutBtn=document.getElementById("signOutButton");
signOutBtn.addEventListener("click", (e)=>{

signOut(auth).then(() => {
  // Sign-out successful.
}).catch((error) => {
  alert("Greska pri odjavi")
  // An error happened.
});
})
```

Slika 4.21. *Funkcije za registraciju/prijavu*

4.6. Košarica

Stranica za kupovinu proizvoda napravljena je na sličan način kao i stranica s proizvodima. Sastoji se od naslova košarice, ispod kojeg se nalazi ime skupine proizvoda za prodaju. Svaki proizvod se sastoji od svog imena, slike i detalja. Za razliku od stranice s proizvoda, ovdje se u dijelu sa detaljima nalazi oznaka sa imenom klase *cijena-proizvoda,* gdje je navedena cijena za određeni proizvod i oznaka <button> s tekstom *Dodaj u košaricu* pomoću koje se dodaje proizvod u košaricu koja se nalazi ispod svih ponuđenih proizvoda za prodaju.

Košarica je smještena između oznaka <section> i te se sastoji od redaka koji predstavljaju proizvod koji je dodan, njegova cijena i količina koja se može mijenjati. Kada se proizvod doda u košaricu, u stupcu *Proizvod* se doda umanjeni prikaz slike odabranog proizvoda i njegovo ime. U stupcu *Cijena* se nalazi vrijednost proizvoda u kunama. U trećem stupcu se nalazi količina koja se može mijenjati za određeni proizvod i gumb za uklanjanje proizvoda iz košarice. U desnom kutu se nalazi ukupan iznos koji je dodan u košaricu, a ispod svega toga se nalazi gumb za kupovinu. Izgled košarice i proizvoda koji se mogu kupiti prikazani su slikom 4.22..

Slika 4.22. *Izgled proizvoda za prodaju i košarice*

Kad se proizvod doda u košaricu korisnik ima opciju da obriše proizvod iz košarice gumbom *Obriši.* Ova funkcionalnost je omogućena putem JavaScript koda koji dobavlja gumb za brisanje preko imena klase u HTML dokumentu. Dodaje se Event Listener pomoću kojeg se obavlja radnja brisanja proizvoda iz košarice. Funkcija za brisanje proizvoda iz košarice prikazana je slikom 4.23..

```
function removeCartItem(event) {
  var buttonClicked = event.target;
  buttonClicked.parentElement.parentElement.remove();
  updateCartTotal();
}
```

Slika 4.23. *Funkcija za uklanjanje proizvoda iz košarice*

Kao što se može vidjeti iz slike 4.23. funkcija također sadrži *updateCartTotal()* funkciju koja je odgovorna za ažuriranje ukupnog iznosa u košarici. Ova funkcija je dodana u sve ostale funkcije koje nakon svog izvršenja mijenjaju ukupan iznos u košarici. To su: funkcija koja dodaje proizvod u košaricu, funkcija koja je zadužena za mijenjanje količine proizvoda i funkcije koja obavlja kupovinu(kad se proizvod(i) kupe, ukupan iznos se postavlja na nulu). Dohvaćanje elemenata i izračun ukupnog iznosa prikazan je slikom 4.24..

```
function updateCartTotal() {
  var cartItemContainer = document.getElementsByClassName(
    "proizvodi-u-kosarici"
  )[0];
  var cartRows = cartItemContainer.getElementsByClassName("kosarica-redak");
  var total = 0;
  for (var i = 0; i < cartRows.length; i++) {
    var cartRow = cartRows[i];
    var priceElement = cartRow.getElementsByClassName("kosarica-cijena")[0];
    var quantityElement = cartRow.getElementsByClassName("unos-kolicine")[0];
    console.log(priceElement, quantityElement);
    var price = parseFloat(priceElement.innerText.replace("HRK", ""));
    var quantity = quantityElement.value;
    total = total + price * quantity;
  }
  total = Math.round(total * 100) / 100;
  document.getElementsByClassName("ukupno-cijena")[0].innerText = total + "HRK";
}
```

Slika 4.24. *Kod za izračun ukupnog iznosa košarice*

Kad korisnik pritisne gumb *Kupi*, prvo se provjerava ima li uopće dodanih proizvoda u košaricu. Ako ih nema korisnik dobije skočni prozor koji ga na to upozorava. Ako pak u košarici ima proizvoda, također se pojavi skočni prozor ali sa drugačijim sadržajem. Isto tako se brišu svi dodani proizvodi i vrijednost se postavlja na nulu ranije spomenutom funkcijom *updateCartTotal()*.

5. Zaključak

Kako bi obiteljska gospodarstva opstala, moraju se prilagođavati vremenima i novim tehnologijama. Kako se sve više djelatnosti odvija na internetu, tako se sve više koristi i za oglašavanje i kupovinu. To je jedan od glavnih načina oglašavanja za obiteljska gospodarstva u novijem dobu, pa je potražnja za izradu web stranica sve veća. Kako bi što bolje predstavili svoje gospodarstvo i proizvode koje nude, OPG-ovi se često odluče za taj korak. U ovom radu su prikazane tehnologije i postupak izrade web stranice za jedan takav OPG.

Korištene tehnologije pomogle su u izradi jedne takve stranice koja omogućava korisniku pregled svih važnih informacija, proizvoda i usluga koje OPG nudi. Korisniku je omogućeno slanje upita o proizvodima i uslugama. Isto tako je omogućena registracija korisnika kako bi mogao kupovati proizvode koje su u ponudi.

Prednost ovakvih web stranica je što se korisnici mogu na lagan način informirati o samom OPG-u, proizvodima i uslugama i na lagan način mogu kupiti proizvode. Još jedna prednost je to što se mogu steći određene vještine i znanja o tehnologijama koje su korištene.

Literatura

[1] HTML Tutorial, https://www.w3schools.com/html/default.asp, pristupljeno 11. srpnja 2021.

[2] Duckett J. HTML & CSS: design and build websites

[3] Berners-Lee T.,1991. HTML Tags

[4] CSS Tutorial, https://www.w3schools.com/css/default.asp, pristupljeno 11. srpnja 2021.

[5] JavaScript, https://hr.wikipedia.org/wiki/JavaScript, pristupljeno 11. srpnja 2021.

[6] Firebase, https://en.wikipedia.org/wiki/Firebase, pristupljeno 19. kolovoza 2021.

[7] Firebase, https://firebase.google.com/docs/build, pristupljeno 19. kolovoza 2021.

Sažetak

U ovom završnom radu prikazana je izrada i korištenje web stranica za OPG koji služi za informiranje korisnika i kupovinu proizvoda. Za izradu je korišten Microsoft Visual Studio Code. Tehnologije koje su korištene su HTML, CSS, JavaSript i Firebase. Ove tehnologije su ukratko opisane i prikazane kako su se koristile u ovom radu.

Ključne riječi: CSS, Firebase, HTML, JavaScript, web portal, web stranica

Abstract

Family farm web portal

This paper shows web page for family farms and it is used for giving information to the customers and for purchasing products. Some of the features shown in this paper are giving insight how family farms are organised, what products and services are available and giving opportunity to sign up. Microsoft Visual Code was used for making this paper. Technologies used: HTML, CSS, JavaScrpt and Firebase. Theese technologies are briefly described and shown how are they use din this paper.

Key words: CSS, Firebase, HTML, JavaScript, web portal, web stranica

Životopis

Ivan Smoljo je rođen 20. veljače 1997. u Đakovu. Osnovnoškolsko i srednjoškolsko obrazovanje završava u Đakovu. Srednju školu „Gimnazija Antuna Gustava Matoša“. Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek upisuje prvu godinu 2016. godine, smjer preddiplomski studij računarstva.

Prilozi

CD: Izvoran HTML, CSS, JavaScript, Firebase kod